Counting queries in ontology-based data access

Quentin Maniere

THÈSE PRÉSENTÉE

POUR OBTENIR LE GRADE DE

DOCTEUR DE L'UNIVERSITÉ DE BORDEAUX

ECOLE DOCTORALE MATHÉMATIQUES ET INFORMATIQUE

Par **Quentin Manière**

Counting queries in ontology-based data access

Dirigée par **Meghyn Bienvenu** et **Michaël Thomazo**

Soutenue publiquement le 30 septembre 2022

Composition du jury, présidé par Carsten Lutz :

Meghyn Bienvenu	Directrice de recherche	CNRS	Directrice
Bernardo Cuenca Grau	Professor	Université d'Oxford	Rapporteur
Carsten Lutz	Professor	Université de Leipzig	Rapporteur
Marie-Laure Mugnier	Professeure	Université de Montpellier	Examinatrice
Marie-Christine Rousset	Professeure	Université de Grenoble	Examinatrice
Mantas Šimkus	Associate professor	Université d'Umeå	Rapporteur
Michaël Thomazo	Chargé de recherche	INRIA	Invité

Remerciements

J'aimerais tout d'abord remercier mes deux merveilleux encadrants pour ces trois années de thèse. Meghyn, source intarissable de références et de connexions avec d'autres approches, et Michaël, joyeux destructeur de conjectures pleines d'espoir, j'ai adoré travailler avec vous et tant appris à vos côtés ! Pour cet enseignement, votre disponibilité, vos conseils et votre confiance, je vous suis infiniment reconnaissant. Merci aussi, Michaël, pour les petits-déjeuners dans la cour de l'ENS, au tout début. Merci aussi, Meghyn, pour les soirées jeux et le logis à la toute fin (merci d'ailleurs à Laurent et à Quentin pour leur accueil !). J'espère avoir la chance de retravailler avec vous deux à l'avenir.

Je tiens à remercier aussi Bernardo Cuenca Grau, Carsten Lutz et Mantas Šimkus d'avoir accepté de rapporter cette thèse. Vos retours m'ont permis d'ancrer plus clairement ce travail dans le paysage de notre communauté ; le présent manuscrit en ressort grandi. Je suis également reconnaissant à Marie-Laure Mugnier et Marie-Christine Rousset d'avoir pris part au jury. Si une grève des transports, coutume nationale que le monde entier nous envie, vous a empêchées d'assister en personne à la soutenance, nul doute que nous aurons tout de même des occasions de nous recroiser autrement qu'en ligne.

Merci bien sûr aux camarades du 123, simplement de passage ou "permanents" du bureau, vous avez été de fantastiques compagnons de route durant ces trois années ! Je pense à Govind, notre doyen qui a soutenu il y a déjà plus d'un an, à Noureddine et ses protocoles horriblement non-transitifs, à Shih-Shun et sa compagne désormais parents, à Jonathan et ses ~~coloriages~~ colorations qui se reconfigurent en quelques "tchak-tchak", à Jana et sa virtuosité en musique, à Pierre et son Tchoupi, à Angélique et son rire si reconnaissable, à Benoît qui a retrouvé ses lunettes, à Natacha et ses méandres au tableau, et à tous ceux que j'ai forcément oubliés dans cette tentative d'énumération.

Les camarades thésards ne se limitent évidemment pas au 123 : je remercie Zoé pour son engagement dans les Hanabis du mardi, Timothée pour l'ensemble de ses blagues, Clément pour ses cours de tricot, Claire pour ses origamis, Jojo(séphine) pour l'association pommes-Beethoven dont je n'arrive plus à me défaire, Corto, Antonio et Thibault pour leurs passages au bureau à l'heure du déjeuner, réglés comme du papier à musique, Aïda pour les précisions sur le statut juridique du patio, Rémi pour ses combines de magicien (spécialité "Escamotage"), Maxime pour son expertise inattendue en cucurbitacées, Sanja pour ses astuces en allemand, Sarah pour sa courageuse représentation des Méthodes Formelles-Formelles dans un bureau de CombAlgo, Aline pour ses slides sur lesquelles j'ai dû improviser, Arthur pour ses talents en improvisation justement, et Théo, Rohan, Alex, P-E, Elsa et bien d'autres.

Merci également aux différentes équipes de recherches dans lesquelles je me suis glissé. À toute l'équipe du DI, sous les toits de l'ENS, pour son accueil pendant mon stage de fin de Master qui a préfiguré cette thèse : Camille, Tatiana, Pierre, Luc, Chien-Chung, Julien, Garance et Michaël. À l'équipe RATIO et ses pique-niques dans l'herbe : Diego, Joanna, Anca, Igor, Marc, Vincent, Gianluca, Sanja et Meghyn. À l'équipe autour du projet INTENDED et de son université d'été très réussie. À l'équipe autour du projet CQFD, qui a financé ce travail et dont les rendez-vous annuels ont toujours été enrichissants.

Plus généralement, bien des membres du LaBRI ont embelli mes trois années passées à Bordeaux. Je pense notamment à Marthe et Pascal, qui ont formé un comité de suivi plein de précieux conseils, à Katel et Élia, qui ont coordonné les missions auxquelles j'ai pris part, à Corinne, chargée de l'entretien des bureaux et qui a eu bien d'autres casquettes à l'université, à François, Arnaud, Frédéric, David, Sébastien, Vincent et une nouvelle fois à Marthe "pour les nombreux fruits, gâteaux et thés" [Thomazo, 2013].

Une thèse au LaBRI constituerait une toute autre expérience sans l'AFoDIB, l'association des doctorants en informatique (épargnons-nous la signification exacte de l'acronyme), dont les événements rythment l'aventure doctorale. Ces occasions contribuent à rassembler les thésards, mais aussi les stagiaires et les permanents, pour parler science autour d'une tisane ou mettre à l'épreuve la théorie des jeux (de société) en salle de séminaire. Merci à l'AFoDIB donc, et à tous ses membres qui la font vivre ! Mention spéciale à Sarah et Corto pour avoir pris ma suite à la trésorerie et couvert le trou dans la caisse qui a financé mes dernières vacances.

Merci à mes amis de longue date qui m'ont accompagné avant et pendant cette aventure : Mathilde et Matthieu pour le gîte au pied du Vercors et l'atelier potager, Morgan l'indéfectible nantais, Martin, Thomas et Olivier que je ne croise pas aussi souvent que je le voudrais, sans oublier la bande orcéenne, Pablo, notre Hanabi-sensei, Paul, Fabien, Florian, Alexia et Thomas.

Merci à Odile Vallée, excellente professeure de mathématiques au lycée, et dont l'enseignement m'a laissé ce goût prononcé pour la logique qui ne m'a jamais quitté.

Je tiens également à remercier ma famille, Maman et Papa, Vincent et Pierre, Mamie Paule, Mamie Thé, les Manière et les Juranville, pour leur soutien de toujours.

Enfin, à Christèle, merci pour ta présence durant ces trois années, les horaires des trains Paris-Bordeaux n'ayant plus de secrets pour nous. Merci d'avoir supporté l'absurdité de l'Éduc' Nat' qui ne t'a pas laissée venir à Bordeaux, mais qui nous a gracieusement affectés au même endroit une fois ta dispo et ma démission posées (*soupir*). Je suis heureux d'avoir pu partager les bons moments de cette thèse avec toi, et je suis sûr que de nombreux camarades ont apprécié le partage de tes succulentes pâtisseries avec eux !

Contents

List of Figures

List of Tables

Résumé étendu en français

La réponse à des requêtes en présence d'une ontologie (OMQA, pour "ontology-mediated query answering") est une approche facilitant l'accès à des données par l'intermédiaire d'ontologies. Ces dernières sont des spécifications formelles de la terminologie et des connaissances conceptuelles d'un domaine d'intérêt. Les ontologies peuvent servir à fournir un vocabulaire adapté à la formulation de requêtes, ce qui est particulièrement adéquat lors de l'intégration de diverses sources de données. Les ontologies capturent également des connaissances sur le domaine étudié, qui peuvent être utilisées pour gérer des données incomplètes en inférant des informations implicites, ce qui permet d'enrichir les réponses aux requêtes posées. A partir d'environ 2005 et des premiers travaux de Poggi et al. [2008], OMQA est devenu un champ de recherche prolifique dans les communautés IA et bases de données. Les articles de synthèse [Bienvenu and Ortiz, 2015; Xiao et al., 2018] fournissent une introduction à ce domaine ainsi que de nombreuses références vers la littérature existante.

Cette thèse étudie la question de la réponse à des requêtes de comptage dans ce cadre OMQA, et plus particulièrement la complexité de ce problème. A ce jour, ce sujet n'a été étudié que dans des cas très restreints, et sans définition commune de ce que sont les requêtes de comptage. De plus, les résultats existants sont largement insatisfaisants puisque la plupart échouent à déterminer avec exactitude la complexité du problème, et ce malgré l'expressivité limitée des langages d'ontologies considérés.

Une nouvelle étude de ces requêtes de comptage en présence d'une ontologie est donc nécessaire. Nous définissons tout d'abord une notion simple et élégante des requêtes de comptage qui généralise plusieurs définitions précédentes. Nous étendons ensuite notre cadre à des langages d'ontologies plus expressifs, qui étendent ceux explorés jusqu'alors mais couvrent aussi d'autres logiques très populaires, notamment utilisées dans la pratique. Dans ce contexte élargi, nous caractérisons la complexité exacte du problème de réponse aux requêtes de comptage en présence de ces ontologies expressives, et déterminons ensuite comment celle-ci varie si l'on restreint la structure des requêtes et/ou l'expressivité du langage des ontologies. Notre travail ne clôt pas seulement les questions laissées ouvertes dans de précédents travaux, mais étend aussi notre compréhension des requêtes de comptage à des panels bien plus larges de situations OMQA.

Logiques de description

La plupart des travaux sur OMQA considère que la connaissance est représentée par des logiques de description, une famille de langages introduite dans les années 80 [Brachman and Schmolze, 1985], et qui a suscité beaucoup d'attention depuis [Baader et al., 2003, 2017]. Dans les logiques de descriptions, les notions élémentaires du domaine d'intérêt sont décrites par un vocabulaire consistant de concepts et de rôles, qui sont respectivement des prédicats unaires et binaires, et à partir desquels des concepts et rôles plus complexes peuvent être obtenus par divers constructeurs (par exemple la conjonction $\sqcap$). La diversité de ces constructeurs est dictée par la logique de description.

Une base de connaissance en logique de description se décompose en deux parties: une ontologie et des données. L'ontologie contient la connaissance terminologique du domaine, et consiste en un ensemble d'axiomes (tels que des inclusions $\sqsubseteq$) qui décrivent les relations entre les différents concepts et rôles. Les données représentent des connaissances factuelles, et précisent quels sont les concepts satisfaits par tel ou tel individu et quels rôles les connectent. Cela prend la forme d'un ensemble de faits que l'on peut assimiler à une base de données usuelle (mais restreinte à des faits unaires et binaires).

L'intérêt des logiques de description pour représenter des connaissances est désormais largement reconnu et celles-ci sont le fondement logique du langage d'ontologie web OWL, un standard W3C pour le web sémantique [Horrocks et al., 2003, 2006; Hitzler et al., 2009]. Une attention toute particulière a été portée aux familles DL-Lite et $\mathcal{EL}$ de logiques de description [Calvanese et al., 2007b; Baader et al., 2005], du fait de leurs bonnes propriétés en terme de complexité. DL-Lite est adapté pour des applications impliquant un grand volume de données et a donné naissance au profil OWL 2 QL, tandis que les logiques de la famille $\mathcal{EL}$ sous-tendent le profil OWL 2 EL[1] et sont utilisées pour exprimer des ontologies médicales à grande échelle telles que SNOMED CT[2] [Spackman, 2000].

[1] `https://www.w3.org/TR/owl2-profiles/`
[2] `http://www.ihtsdo.org/snomed-ct`

Là où les travaux existants sur les requêtes de comptage se cantonnent à des fragments de la famille DL-Lite, cette thèse étend le champ d'étude à la logique plus expressive $\mathcal{ALCHI}$, qui contient à la fois $\mathcal{EL}$ et des fragments très usités de la famille DL-Lite.

Opérations de déduction

Une couche ontologique sur des données conduit à de nouvelles opérations de déduction et accroît la complexité de celles usuellement considérées dans le contexte des bases de données. Les opérations communes incluent par exemple la question de la satisfiabilité, qui sert à détecter si une base de connaissance contient des informations contradictoires, et la question de la subsomption, qui teste si un concept donné est plus spécifique qu'un autre.

Comme son nom le laisse deviner, le cadre OMQA est particulièrement intéressé par les réponses à des requêtes, une opération très étudiée dans le cadre des bases de données relationnelles usuelles et qui correspond, quand des ontologies sont introduites, à se demander si une requête est une conséquence logique de la base de connaissances considérée.

La complexité de ces opérations de déductions augmentent naturellement avec l'expressivité de la logique de description considérée et du langage de requête. Un compromis est donc nécessaire entre la capacité des logiques de description à représenter des connaissances de façon satisfaisantes, et l'efficacité à raisonner avec des bases de connaissances exprimées dans ces logiques. La compréhension de la complexité des opérations de déduction est donc un enjeu majeur dans le paradigme OMQA: elle guide le choix du langage de requêtes et de la logique de description selon le cas applicatif étudié. De telles considérations pratiques ont menés au développement des logiques de description dites "légères", telles que les sus-mentionnées DL-Lite et $\mathcal{EL}$, qui permettent de bonnes performances [Calvanese et al., 2007b; Baader et al., 2005].

Les travaux existants sur les réponses à des requêtes de comptage [Kostylev and Reutter, 2015; Calvanese et al., 2020a] échouent à caractériser pleinement la complexité de ce problème de déduction, laissant souvent des trous béants entre les bornes de complexité supérieure et inférieure. Dans cette thèse, nous déterminons exactement la complexité du problème de réponse à des requêtes de comptage dans toutes les situations étudiées, clôturant ainsi les cas restés ouverts de la littérature mais fournissant également une compréhension fine de ce problème dans des contextes bien plus vastes.

Requêtes

La grande majorité des travaux dans le cadre OMQA suppose que l'utilisateur formule ses requêtes sous la forme de requêtes conjonctives. De telles requêtes demandent si une condition conjonctive donnée, la requête, est logiquement induite par la base de connaissance formée de l'ontologie et des données.

Cependant, il existe de nombreux autres types de requêtes, au delà des requêtes conjonctives, qui sont utiles en pratique. Cela a motivé des recherches sur l'adoption d'autres langages de requêtes pour OMQA. Tandis qu'enrichir les requêtes conjonctives par des conditions négatives ou des inégalités mène à des problèmes indécidables, même sous des hypothèses très restreintes [Gutiérrez-Basulto et al., 2015], la situation est plus favorable pour les requêtes navigationnelles (comme celles fondées sur des chemins réguliers), qui peuvent être utilisées sans perdre le caractère décidable du problème associé, et qui a même parfois une complexité tout à fait raisonnable vis-à-vis des données [Bienvenu et al., 2015].

Les requêtes d'agrégation, qui utilisent des opérateurs numériques (comme du comptage, des sommes ou des moyennes) pour résumer certaines parties du jeu de données, constituent une autre classe majeure des requêtes en bases de données. Bien que de telles requêtes soient largement utilisées pour l'analyse des données, elles demeurent peu explorées dans le contexte OMQA. Cela est peut-être dû au fait qu'il n'est pas évident de définir la sémantique de ces requêtes.

Plusieurs sémantiques ont ainsi été proposées ces dernières années [Calvanese et al., 2008; Kostylev and Reutter, 2015] pour répondre à ce problème, mais sans atteindre une définition unifiée et satisfaisante. Cette thèse définie une sémantique qui généralise celles explorées dans [Kostylev and Reutter, 2015] et permet des requêtes relativement expressives.

Contributions

Cette thèse présente un panorama exhaustif de la complexité du problème de la réponse à des requêtes de comptage en présence d'ontologies selon trois dimensions. La première est l'expressivité des ontologies: nous explorons systématiquement $\mathcal{ALCHI}$ et ses sous-logiques, $\mathcal{ALCHI}$ étant une logique de description très expressive qui capture notamment $\mathcal{EL}$ et les principales sous-logiques de DL-Lite. La deuxième dimension est le langage de requête: nous considérons une notion générale de requête conjonctive de comptage (CCQ, pour "counting conjunctive query"), et explorons ensuite deux sous-classes naturelles de CCQ, basées respectivement sur l'enracinement et l'atomicité, afin de déterminer si de telles restrictions syntaxiques réduisent la complexité de la réponse à ces CCQs. La troisième dimension est la mesure de la complexité utilisée. Nous considérons à la fois la complexité combinée, usuelle, et la complexité de données, la première explicitant la complexité totale du problème tandis que la seconde se concentre sur comment cette complexité évolue selon la taille des données.

Notre première contribution est la sémantique même des requêtes de comptage. Rappelons qu'un modèle est une façon de compléter les données afin de satisfaire à tous les axiomes de l'ontologie. Dans un modèle donné, nous nous intéressons aux façons de satisfaire une requête conjonctive, que l'on appelle des matches ; leur nombre constitue la réponse à la requête de comptage correspondante, et varie de modèle en modèle. La sémantique que nous définissons pour une réponse à une CCQ sur une base de connaissance consiste en des bornes sur ce nombre de matches, qui doivent être valides pour tout modèle de la base de connaissance. Ces réponses sont appellées des réponses certaines, étendent les sémantiques présentées dans [Kostylev and Reutter, 2015] et généralisent le problème usuel de réponse à des requêtes conjonctives.

Dans le cas général des CCQs, nous prouvons que le problème de réponse à ces requêtes est 2EXP-complet pour la plupart des sous-logiques d'$\mathcal{ALCHI}$, mais devient coNEXP-complet pour DL-Lite$_{\mathsf{core}}$. En terme de complexité de données, nous montrons que le problème est coNP-complet pour toutes les sous-logiques considérées. Les techniques développées s'appuient sur des manipulations précautionneuses des modèles, qui préservent à la fois le nombre de matches de la requête et déplient les régularités inhérentes au modèle. Nos constructions s'avérent robustes dans la mesure où elles nous permettent de clore une question voisine dans le domaine des prédicats clos, pour lequels certains prédicats ne peuvent s'interpréter au-delà de leur description dans les données. Nous montrons ainsi que le problème de satisfiabilité d'une base de connaissance exprimée dans DL-Lite$_{\mathsf{core}}$ et avec des prédicats clos est coNEXP, rejoignant ainsi une borne inférieure existante.

Dans la perspective d'identifier des cas profitant d'une meilleure complexité, nous considérons d'abord l'impact de la restriction aux CCQs enracinées. L'enracinement est en effet une restriction syntaxique bien connue pour réduire la complexité dans des cadres OMQA proches. Il s'avère cependant que l'adaptation la plus directe de cette restriction à nos CCQs ne conduit pas à de meilleures propriétés que dans le cas général. Cela nous conduit à nous concentrer sur une classe plus restreinte: les CCQs enracinées et exhaustives. Pour cette dernière classe, nous utilisons des variations des constructions développées précédemmemt afin d'obtenir quatre améliorations différentes, selon la logique de description considérée, allant de la PP-complétude à la coNEXP-complétude. Cette dernière repose notamment sur la présence de rôles inverses dans l'ontologie, une fonctionnalité déjà connue pour augmenter la complexité des requêtes enracinées. En terme de complexité de données, nous exhibons des cas raisonnables en pratique pour les ontologies exprimées dans DL-Lite$_{\mathsf{core}}$. Ce résultat positif s'appuie sur le fait que le modèle canonique minimise le nombre de matches.

Nous continuons notre quête de cas plus simples, en terme de complexité, que le cas général, par une autre restriction sur le langage de requêtes, sans lien avec l'enracinement: l'atomicité. La classe des CCQs consistant en un seul atome, que nous appelons des requêtes de cardinalité, sont disponibles en deux saveurs selon que le prédicat d'intérêt est unaire ou binaire. De nombreuses connexions naturelles avec la sémantique des prédicats clos sont exploitées afin de déterminer la complexité combinée du problème de réponse à ces requêtes de cardinalité. Nous prouvons que ce problème est coNP-complet pour les langages de la famille DL-Lite, tandis qu'il demeure EXP-complet pour $\mathcal{EL}$ et plusieurs de ses extensions. Quand les ontologies sont suffisamment expressives pour contraindre les modèles à être de taille exponentiellement grande, cette complexité augmente en coNEXP-complétude, ce qui est surprenamment élevé pour un cas particulier d'apparence si simple. Cependant, la situation est plus favorable en complexité de données, pour laquelle nous identifions des cas raisonnables pour des ontologies formulées dans la famille DL-Lite. De façon remarquable, ces derniers résultats ne reposent pas sur l'existence d'un modèle canonique optimal mais plutôt sur l'existence d'une famille de modèles dans laquelle un modèle optimal peut toujours être trouvé. Finalement, nous éclaircissons la complexité des requêtes de cardinalité dans la famille DL-Lite par une analyse de cette complexité pour chaque paire requête-ontologie. En particulier, nous caractérisons complètement la complexité de données de ces paires, pour des ontologies exprimées dans DL-Lite$^{\mathcal{H}}_{\mathsf{pos}}$, et montrons une trichotomie (TC^0, coNP, ou logspace-équivalent à Perfect Matching).

1 Introduction

Ontology-mediated query answering (OMQA) facilitates access to data through the use of ontologies, which are formal specifications of the terminology and conceptual knowledge of a given application domain. Ontologies can serve to provide a convenient vocabulary for query formulation, which is especially relevant when integrating data from different sources, and they also provide domain knowledge that can be exploited at query time to infer implicit information and obtain more complete query results, thus helping to tackle data incompleteness. Starting from around 2005 and the seminal work of Poggi et al. [2008], OMQA has grown into an active topic of research in the AI and database communities. The survey articles [Bienvenu and Ortiz, 2015; Xiao et al., 2018] provide introductions to the area and pointers to the literature.

This thesis investigates the issue of answering counting queries in the OMQA framework and focuses in particular on pinpointing the precise computational complexity of this problem. So far, this topic has only been explored for very restricted settings, without even a unified notion of what is a counting query. Furthermore, existing complexity results remain unsatisfactory as many of them fail to pinpoint the precise complexity of the problem, despite the limited expressiveness of the considered ontology languages.

This motivates us to take a fresh look at counting queries in OMQA. We begin by defining a simple yet elegant notion of counting query, which is a natural generalization of some existing notions. We further extend the scope of our study to more expressive ontology languages, which properly extend those explored so far but also cover other popular logics that are used in practical applications. In this broader context, we characterize the precise complexity of answering counting queries over these expressive ontologies, and further determine how the complexity varies if we restrict the structure of the counting queries and/or the expressiveness of the ontology language. Our work not only closes the complexity gaps that had been left open in the literature, but it also extends our understanding of counting queries to a much wider range of OMQA settings.

Description Logics

Much of the work on OMQA considers ontologies formulated in fragments of first-order logics such as description logics (DLs) or existential rules (also known as Datalog$^{\pm}$). In this work, we focus on description logics ontologies, a family of knowledge representation languages introduced in the 80's [Brachman and Schmolze, 1985] and which has drawn a lot of attention since then [Baader et al., 2003, 2017]. In DLs, the basic notions of the domain of interest are described using a vocabulary consisting of *concept and roles names*, that are respectively unary and binary predicates, from which complex concepts and roles can be further built using various constructors (e.g. conjunction $\sqcap$ or existential restriction $\exists$). The set of available constructors is dictated by the considered DL.

A DL knowledge base consists of two components: a TBox and an ABox. The TBox (or ontology) contains the terminological knowledge about a domain and consists of a set of axioms (such as inclusions, $\sqsubseteq$) that describe the relationship between different concepts and roles. The ABox captures the assertional knowledge by specifying which concepts, resp. roles, hold on which individuals, resp. connect which pairs of individuals, where individuals are constants. It takes the form of a set of ground facts and can be thought of as a classic database instance (but restricted to unary and binary facts).

Let us give a toy example to illustrate these latter definitions. It is common knowledge that a *mule* is an animal that is the offspring of a male *donkey* and a female *horse*, these two latter being distinct species of animals. Assume we know a mule molly, but no horse nor donkey. Using DL notations, our toy example could be captured with the following TBox consisting of 4 axioms:

$$\text{Mule} \sqsubseteq \text{Animal} \sqcap \exists \text{MaleParent.Horse} \sqcap \exists \text{FemaleParent.Donkey}$$

$$\text{Horse} \sqcap \text{Donkey} \sqsubseteq \bot \qquad \text{Horse} \sqsubseteq \text{Animal} \qquad \text{Donkey} \sqsubseteq \text{Animal}$$

and by the ABox containing a single fact:

$$\text{Mule}(\mathsf{molly}).$$

The interest of description logics to represent knowledge is now widely recognized, and DLs notably provide the logical foundations of the OWL web ontology language, a W3C standardized language for the Semantic Web [Horrocks et al., 2003, 2006; Hitzler et al., 2009]. Particular attention has been paid to the DL-Lite [Calvanese et al., 2005, 2007b; Artale et al., 2009] and $\mathcal{EL}$ families [Baader et al., 1999, 2005, 2008], due to their favorable computational properties. DL-Lite is well suited for data-intensive applications and gave rise to the OWL 2 QL profile, while DLs of the $\mathcal{EL}$ family underly the OWL 2 EL profile[1] and are used to specify large-scale medical ontologies such as SNOMED CT[2] [Spackman, 2000].

While existing work concerning counting queries in OMQA has remained limited to fragments of the DL-Lite family, this thesis extends the scope to the expressive description logic $\mathcal{ALCHI}$, with subsumes both $\mathcal{EL}$ and popular dialects of the DL-Lite family.

Reasoning tasks

Adding an ontological layer on top of data motivates looking at new reasoning tasks, and it typically also increases the complexity of the usual computational tasks considered in the database domain. Common reasoning tasks include for example satisfiability, that serves to detect whether a knowledge base contains contradictory information (e.g. if there exists an animal that is both an horse and a donkey, in our toy example) and subsumption, that tests if a concept is more specific than another (e.g. it can be deduced that $\mathsf{Mule} \sqsubseteq \mathsf{Animal}$ even though this axiom is not explicitly given in our toy TBox).

As its name suggests, OMQA is additionally concerned with query answering, a task that is well studied for classical relational databases, and which corresponds, when ontologies are introduced, to testing if a query is logically entailed from the knowledge base of interest.

The complexity of these reasoning tasks generally increases with the expressiveness of the considered DL and query language. A trade-off hence arises between the capacity of DLs to provide a satisfactory representation of the domain knowledge, and the desired efficiency to reason over DL KBs. Understanding the complexity of the reasoning tasks of interest is a major issue in OMQA: it guides the choice of which DL and which query language should be used for a given application. Practical considerations led to the development of so-called 'lightweight' DLs, such as the previously mentioned DL-Lite and $\mathcal{EL}$ families, which enjoy favorable computational properties [Calvanese et al., 2007b; Baader et al., 2005].

Existing work on the task of answering counting queries [Kostylev and Reutter, 2015; Calvanese et al., 2020a] fails to fully characterize the complexity of the query answering task, with many open gaps between the obtained upper and lower complexity bounds. In this thesis, we pinpoint the exact complexity of answering counting queries in all of the considered situations, thereby closing the open cases

[1] `https://www.w3.org/TR/owl2-profiles/`
[2] `http://www.ihtsdo.org/snomed-ct`

from the literature as well as providing a precise understanding of the problem for a much wider range of settings.

Queries

The question of how to query DL knowledge bases (KBs), composed of a TBox (ontology) and ABox (data), has been explored since the early days of DL research. Initially, the focus was on instance queries [Baader et al., 2003], where the task is to determine all members of a given concept or role and which basically corresponds to testing entailment of atomic facts from the KB. However, starting from the works of [Calvanese et al., 1998; Levy and Rousset, 1998; Horrocks and Tessaris, 2000; Calvanese et al., 2005], and motivated by the interest of using DL ontologies to improve data access, attention shifted to the more expressive conjunctive queries (CQs), and the vast majority of work on OMQA takes CQs as the query language. Such queries consist of a conjunction of atoms and have been widely studied in the database community, as they correspond to the Select-Project-Join fragment of the SQL query language. In the OMQA setting, the problem of answering CQs essentially corresponds to asking whether the conjunctive condition given by the query and candidate answer tuple is entailed from the knowledge base.

In our toy example, one can ask "Who has a parent who is a horse?". Using a classical database management system (DBMS), this query would not admit any answer as no horse occurs in the data. In the OMQA setting, however, the domain knowledge can be used to infer that the mule molly must have a (male) parent who is a horse, hence, we are able to deliver the intended answer: molly.

However, there are many other kinds of database queries, beyond plain CQs, that are relevant in practice. This motivated research into the feasibility of adopting other database query languages for OMQA. While enriching CQs with either negated atoms or inequalities has been shown to lead to undecidability even in very restricted settings [Gutiérrez-Basulto et al., 2012, 2015], the situation is more positive for navigational queries (like regular path queries), which can be adopted without losing decidability, sometimes even retaining tractable data complexity [Calvanese et al., 2007a; Ortiz et al., 2011; Stefanoni et al., 2014; Bienvenu et al., 2015].

Aggregate queries, which use numeric operators (e.g. count, sum, max) to summarize selected parts of a dataset, constitute another prominent class of database queries. While they have been studied for a broad range of related settings[3], from relational databases to extensions of rule-based languages such as Datalog or Answer Set Programming (ASP), and are widely used for data analysis,

[3]See Section 3.1.1 for a presentation of the state-of-the-art regarding aggregate functions.

these queries have been little explored in the framework of OMQA. This may be partly due to the fact that it is not at all obvious how to define the semantics of such queries in the OMQA setting. In our toy example, one can ask "How many animals are there?". A classic DBMS will return the answer 0 (it only knows molly as a mule, not an animal). The expected answer is less clear in the OMQA setting: should it be 1 because of molly? or rather 3 if also counting its two parents? or "at least 3"? or maybe "at least 2" as our toy ontology does not prevent molly from being its own parent?

Several semantics of counting queries over OMQA have hence been proposed in the past years [Calvanese et al., 2008; Kostylev and Reutter, 2015] to address this question, without reaching a satisfactory unique definition. This thesis defines a semantics unifying both those explored in [Kostylev and Reutter, 2015] and allowing for rather expressive queries.

Structure of the thesis

We present a complete picture of the complexity landscape of answering counting queries, along three main dimensions. The first dimension is the expressive power of the ontology: we systematically explore a variety of sublogics of $\mathcal{ALCHI}$, an expressive description logic that notably captures both $\mathcal{EL}$ and central dialects of the DL-Lite family. The second dimension is the query language. We consider a general notion of counting conjunctive query (CCQ) and further explore two natural subclasses of CCQs, based upon rootedness and atomicity, to determine whether such syntactic restrictions lower the complexity of CCQ answering. The third dimension is the complexity measure. In this work, we consider both the standard combined complexity as well as data complexity, with the former elucidating the overall complexity of the problem, and the latter focusing on how the complexity scales w.r.t. the size of the data.

We now present the global structure of this dissertation, organized according to the second dimension.

Chapter 2. This chapter introduces the necessary notions for later chapters: it defines the investigated description logics, the associated semantics, and recalls the standard reasoning tasks and their complexities.

Chapter 3. This chapter formally defines the consider query language of CCQs and pinpoints the precise complexity of answering these queries. Based on the construction of interlacings, which are models enjoying good properties with respect to the query of interest, we prove that the combined complexity ranges from coNEXP-completeness to 2EXP-completeness, depending on the considered DL, while we obtain coNP-complete data complexity for all considered logics. Interestingly, our approach also allows to answer an open question from the related problem of OMQA with closed predicates.

Chapter 4. This chapter explores rootedness, a structural restriction on queries that is known to lower the complexity of reasoning in related OMQA settings. We show that the most straightforward adaptation of rootedness to CCQs does not lead to improved complexity, which motivates us to focus on a natural subclass of exhaustive rooted CCQs. For this latter class, we use variations of the constructions developed for the general case to obtain four different improvements depending on the considered DL, ranging from PP-completeness to coNEXP-completeness, for the combined complexity measure. For data complexity, we prove that exhaustive rooted CCQ answering over DL-Lite$_{\mathsf{core}}$ ontologies is tractable and enjoys the lowest possible complexity (TC^0).

Chapter 5. This chapter explores cardinality queries, which are CCQs consisting of a single atom. Several connections with OMQA with closed predicates are exhibited, which we use to determine the combined complexity of cardinality query answering in all of our considered DLs. In particular, we prove that the problem is coNP-complete for the DL-Lite family and is EXP-complete for $\mathcal{EL}$ and several of its extensions. This complexity even rises to coNEXP-completeness for the most expressive investigated DLs. The situation is more favorable in data complexity, as we obtain tractable cases (TC^0) in the DL-Lite family. Finally, we gain further insights into the complexity of cardinality query answering in the DL-Lite family by performing a non-uniform complexity analysis that aims to determine the data complexity associated with each particular ontology-mediated query (OMQ). In particular, we are able to fully characterize the data complexity of OMQs consisting of a cardinality query and DL-Lite$^{\mathcal{H}}_{\mathsf{pos}}$ ontology, exhibiting a complexity trichotomy (TC^0, coNP, or logspace-equivalent to Perfect Matching).

Chapter 6. This chapter summarizes the results of the thesis and suggests several further directions of research.

Annex A. Additional proof material that has not been included within the thesis is available in this annex.

Annex B. This annex aims to facilitate the understanding of the four investigated flavors of interlacings by centralizing useful definitions and figures. We encourage the reader to keep a printed version of this annex close at hand.

Related publications

Some of the results presented in this thesis have already been published:

- The semantics of CCQs introduced in Chapter 3, the associated coNP and DP procedures for DL-Lite ontologies with respect to data complexity, and some results from Chapter 4 regarding rooted CCQs, again over DL-Lite, can be found in [Bienvenu et al., 2020].

- In Chapter 5, all of the data complexity results for cardinality query answering over DL-Lite ontologies have been presented in [Bienvenu et al., 2021a], including the complexity classification in Section 5.5.

- The generalization of these DL-Lite approaches to sublogics of $\mathcal{ALCHI}$ between $\mathcal{EL}$ and $\mathcal{ELHI}_\bot$ appeared in a workshop paper [Bienvenu et al., 2021b], establishing in particular data complexity results for CCQ answering over these ontologies that appear in Chapter 3, while the exact combined complexity remained open.

- In Chapter 3, optimal bounds for CCQ answering over these sublogics of $\mathcal{ELHI}_\bot$ with respect to combined complexity have been further presented in [Bienvenu et al., 2022]. This latter publication also provides all the results from Chapter 5 concerning cardinality queries answering over these DLs.

By contrast, all of the results in this thesis that concern $\mathcal{ALC}$ or its extensions have not been published yet, and the study of rooted CCQs beyond DL-Lite, notably in extensions of $\mathcal{EL}$, is also a novelty.

2 Preliminaries

This chapter introduces the background notions required for the later chapters. We begin with the description logics (DLs) investigated in this thesis: we define $\mathcal{ALCHI}$ knowledge bases, their interpretations and models. Closed predicates are also briefly introduced as several connections to this setting will be made and exploited in later chapters. We further give an overview of the usual reasoning tasks and recall their associated complexities.

2.1 Description Logics

$\mathcal{ALCHI}$ is an extension of the central $\mathcal{ALC}$ description logic, which serves as a base for many more expressive DLs. We hereby recall its syntax and its relevant sublogics. $\mathcal{ALCHI}$ notably extends $\mathcal{EL}$ and, by allowing for inverse roles, several dialects of the DL-Lite family such as DL-Lite$_{\mathsf{core}}$. Importantly, and following Bienvenu et al. [2014a], we allow negative role inclusions in $\mathcal{ALCHI}$ so that it admits DL-Lite$^{\mathcal{H}}_{\mathsf{core}}$ as a sublogic. We further recall the usual set semantics for these DLs, through the notions of interpretations and models, present a normal form for $\mathcal{ALCHI}$ ontologies, and briefly introduce the semantics of closed predicates.

2.1.1 $\mathcal{ALCHI}$ and its sublogics

Description logics usually split the representation of knowledge into two parts: a terminological one, the ontology, and an assertional one, the data. The second accounts for ground facts that are typically stored and processed by database management systems, while the first represents the domain knowledge, given by a logical theory. The building blocks of these two parts are concept and role names (unary and binary predicates) that can further be combined, using the constructors provided by the considered DL, to obtain complex concepts and roles.

		DL-Lite				Suffixes			Examples	
		pos	core	$\mathcal{EL}$	$\mathcal{ALC}$	$\mathcal{H}$	$\mathcal{I}$	$\bot$	$\mathcal{ALCI}$	$\mathcal{ELH}_\bot$
Concepts	$\mathrm{A} \mid \exists \mathrm{R}.\top$	✓	✓	✓	✓				✓	✓
	$\top \mid \mathrm{B}_1 \sqcap \mathrm{B}_2 \mid \exists \mathrm{R}.\mathrm{B}$			✓	✓				✓	✓
	$\bot$				✓			✓	✓	✓
	$\neg \mathrm{B} \mid \mathrm{B}_1 \sqcup \mathrm{B}_2 \mid \forall \mathrm{R}.\mathrm{B}$				✓				✓	
Roles	$\mathrm{R} \in \mathsf{N_R}$	✓	✓	✓	✓		✓		✓	✓
	$\mathrm{R} \in \mathsf{N_R^\pm}$	✓	✓				✓		✓	
Axioms	Positive concept incl.	✓	✓	✓	✓				✓	✓
	Positive role incl.					✓				✓
	Corresp. neg. incl.		✓		✓			✓	✓	✓

Table 2.1: Allowed features depending on the considered DL.

Definition 1. *We assume mutually disjoint sets* $\mathsf{N_C}$*,* $\mathsf{N_R}$*, and* $\mathsf{N_I}$ *of* concept, role, and individual names*, their union being the basic vocabulary used to represent knowledge. A* knowledge base (KB) $\mathcal{K} := (\mathcal{T}, \mathcal{A})$ *consists of a terminological part* $\mathcal{T}$ *called a TBox, and an assertional part* $\mathcal{A}$ *called an ABox. An* ABox *is a finite set of* concept assertions $\mathrm{A}(\mathsf{b})$ *(with* $\mathrm{A} \in \mathsf{N_C}$*,* $\mathsf{b} \in \mathsf{N_I}$*) and* role assertions $\mathrm{P}(\mathsf{a}, \mathsf{b})$ *(with* $\mathrm{P} \in \mathsf{N_R}$*,* $\mathsf{a}, \mathsf{b} \in \mathsf{N_I}$*). We denote by* $\mathsf{Ind}(\mathcal{A})$ *the set of individuals occurring in an ABox* $\mathcal{A}$*. A* TBox *is a finite set of axioms, whose forms are dictated by the DL in question.*

We shall use $\mathsf{sig}(\mathcal{T})$ *(resp.* $\mathsf{sig}(\mathcal{A})$ *and* $\mathsf{sig}(\mathcal{K})$*) to denote the* signature *of a TBox* $\mathcal{T}$ *(resp. ABox* $\mathcal{A}$ *and KB* $\mathcal{K}$*), i.e. the set of concept and role names appearing in* $\mathcal{T}$ *(resp.* $\mathcal{A}$ *and* $\mathcal{K}$*).*

The types of axioms that can appear in a TBox depends on the chosen DL, but the most common form of TBox axiom are *inclusions* that can represent hierarchies between concepts or roles, but also enforce disjointness of such predicates. To define the syntax of the DLs considered in this thesis, it will be helpful to distinguish four possible shapes of inclusions in the TBox.

Definition 2. *We distinguish four kinds of possible axioms in a TBox:* positive concept inclusions $\mathrm{B}_1 \sqsubseteq \mathrm{B}_2$, negative concept inclusions $\mathrm{B}_1 \sqcap \mathrm{B}_2 \sqsubseteq \bot$ *(alternatively denoted* $\mathrm{B}_1 \sqsubseteq \neg \mathrm{B}_2$*),* positive role inclusions $\mathrm{R}_1 \sqsubseteq \mathrm{R}_2$*, and* negative role inclusions $\mathrm{R}_1 \sqcap \mathrm{R}_2 \sqsubseteq \bot$ *(alternatively denoted* $\mathrm{R}_1 \sqsubseteq \neg \mathrm{R}_2$*), where* $\mathrm{B}_1, \mathrm{B}_2$*, resp.* $\mathrm{R}_1, \mathrm{R}_2$*, are concepts, resp. roles, whose forms are dictated by the DL of interest.*

We begin with the most expressive DL considered in this thesis, namely $\mathcal{ALCHI}$, which allows all four shapes of inclusions in the TBox, with roles R drawn from the set $\mathsf{N}_\mathsf{R}^{\pm} := \{\mathrm{P}, \mathrm{P}^- \mid \mathrm{P} \in \mathsf{N}_\mathsf{R}\}$, consisting of all role names P and their *inverse role* P^-, and with concepts B constructed according to the following grammar:

$$\mathrm{B} := \top \mid \bot \mid \mathrm{A} \mid \neg \mathrm{B} \mid \mathrm{B}_1 \sqcap \mathrm{B}_2 \mid \mathrm{B}_1 \sqcup \mathrm{B}_2 \mid \exists \mathrm{R}.\mathrm{B} \mid \forall \mathrm{R}.\mathrm{B} \quad \text{with } \mathrm{A} \in \mathsf{N}_\mathsf{C}, \mathrm{R} \in \mathsf{N}_\mathsf{R}^{\pm}.$$

Remark 1. *We follow e.g. Bienvenu et al. [2014a] by including negative role inclusions in $\mathcal{ALCHI}$, so that it has DL-Lite$^{\mathcal{H}}_{\mathsf{core}}$ (defined later) as a sublogic. We also remark that, in the case of $\mathcal{ALCHI}$, allowing negative concept inclusions is redundant as positive concept inclusions already allow us to express such negative concept inclusions, due to the expressive syntax of concepts in $\mathcal{ALCHI}$*

Various sublogics of $\mathcal{ALCHI}$ can be obtained by disallowing some forms of inclusions, inverse roles, and/or several concept constructors. For example, the well-known $\mathcal{EL}$ [Baader et al., 1999, 2005, 2008] is obtained by removing negative concept inclusions, both shapes of roles inclusions, inverse roles and restricting to the concepts B obtained from the following grammar:

$$\mathrm{B} := \top \mid \mathrm{A} \mid \mathrm{B}_1 \sqcap \mathrm{B}_2 \mid \exists \mathrm{R}.\mathrm{B} \quad \text{with } \mathrm{A} \in \mathsf{N}_\mathsf{C}, \mathrm{R} \in \mathsf{N}_\mathsf{R}.$$

We shall also consider some DL-Lite dialects that are fragments of $\mathcal{ALCHI}$. The most expressive dialect is DL-Lite$^{\mathcal{H}}_{\mathsf{core}}$ (alternatively known as DL-Lite$_{\mathcal{R}}$) which allows the four kind of inclusions, inverse roles, and the following restricted forms of concepts:

$$\mathrm{D_i} := \mathrm{A} \mid \exists \mathrm{R}.\top \quad \text{with } \mathrm{A} \in \mathsf{N}_\mathsf{C}, \mathrm{R} \in \mathsf{N}_\mathsf{R}^{\pm}.$$

The logics DL-Lite$^{\mathcal{H}}_{\mathsf{pos}}$, DL-Lite$_{\mathsf{core}}$, and DL-Lite$_{\mathsf{pos}}$ are obtained respectively by dropping negative inclusions, role inclusions, or both features [Calvanese et al., 2005, 2007b; Artale et al., 2009].

Table 2.1 summarizes the naming conventions underlying each explored combination: the row entries are the possible features and column entries are (parts of) names of DLs. A feature is allowed in a DL if the symbol ✓ is present in a column corresponding to (a part of) the name of the DL. Generally speaking, one starts from either DL-Lite$_{\mathsf{pos}}$, DL-Lite$_{\mathsf{core}}$, $\mathcal{EL}$ or $\mathcal{ALC}$ and adds combinations of the 3 available suffixes $\mathcal{H}$, $\mathcal{I}$ and / or $_\bot$, that are not intended to be considered alone. The use of $\mathcal{H}$ indicates that role inclusions are allowed, use of $\mathcal{I}$ that inverse roles are allowed (both for building concepts and in eventual role inclusions), and $_\bot$ that the concept $\bot$ is allowed along with the negative inclusions corresponding to the positive permitted inclusions. Several combinations are of course irrelevant as redundant, e.g. DL-Lite$^{\mathcal{I}}_{\mathsf{pos}}$ or $\mathcal{ALC}_\bot$.

In addition to the already introduced $\mathcal{EL}$ and DL-Lite logics, two examples are detailed in the right most columns, namely $\mathcal{ALCI}$ and $\mathcal{ELH}_\bot$. Notice that $\mathcal{ELH}_\bot$

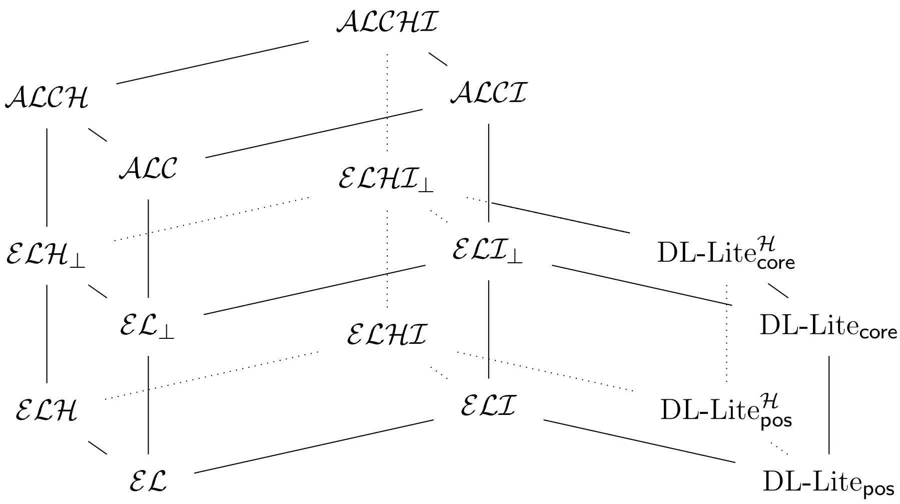


Figure 2.1: The 16 investigated sublogics of $\mathcal{ALCHI}$.

allows for both positive concept and role inclusions, and for *the corresponding negative inclusions*, that is all four shapes of inclusions, while $\mathcal{ALCI}$ only permits positive and negative concept inclusions, but neither positive nor negative role inclusions. The hierarchy, w.r.t. relative expressiveness, of the 16 sublogics of $\mathcal{ALCHI}$ explored in this thesis and obtained from combinations of the presented restrictions, is depicted in Figure 2.1

Example 1. *We reuse the example from the introduction: the knowledge base $\mathcal{K}_{ex}$ is the pair $(\mathcal{T}_{ex}, \mathcal{A}_{ex})$ where $\mathcal{A}_{ex} := \{\text{Mule}(\mathsf{molly})\}$ is the ABox and $\mathcal{T}_{ex}$ is the TBox consisting of the 4 axioms:*

$$\text{Mule} \sqsubseteq \text{Animal} \sqcap \exists\text{MaleParent.Horse} \sqcap \exists\text{FemaleParent.Donkey}$$

$$\text{Horse} \sqcap \text{Donkey} \sqsubseteq \bot \qquad \text{Horse} \sqsubseteq \text{Animal} \qquad \text{Donkey} \sqsubseteq \text{Animal}.$$

There are 4 concept names and 2 role names in the signature of $\mathcal{T}_{ex}$, which contains 3 positive concept inclusions, 1 negative concept inclusion and no role inclusions. It is easily verified that $\mathcal{T}_{ex}$ is an $\mathcal{EL}_\bot$ TBox.

2.1.2 Set semantics

$\mathcal{ALCHI}$ knowledge bases are well known to correspond to decidable fragments of first-order logic. More precisely, $\mathcal{ALCHI}$ KBs translate into the two-variable fragment of first-order logic (see e.g. [Baader et al., 2017]), whose entailment problem is known to be in NEXP [Grädel et al., 1997]. As fragments of first-order logic, $\mathcal{ALCHI}$ knowledge bases are equipped with the standard set semantics, based upon interpretations, recalled next.

Constructor	Syntax	Interpretation
Inverse role	P^-	$\{(y,x) \mid (x,y) \in \mathrm{P}^{\mathcal{I}}\}$
Bottom	$\bot$	$\emptyset$
Top	$\top$	$\Delta^{\mathcal{I}}$
Negation	$\neg\mathrm{B}$	$\Delta^{\mathcal{I}} \setminus \mathrm{B}^{\mathcal{I}}$
Conjunction	$\mathrm{B}_1 \sqcap \mathrm{B}_2$	$\mathrm{B}_1^{\mathcal{I}} \cap \mathrm{B}_2^{\mathcal{I}}$
Disjunction	$\mathrm{B}_1 \sqcup \mathrm{B}_2$	$\mathrm{B}_1^{\mathcal{I}} \cup \mathrm{B}_2^{\mathcal{I}}$
Existential restriction	$\exists\mathrm{R}.\mathrm{B}$	$\{d \mid \exists e \in \Delta^{\mathcal{I}}, (d,e) \in \mathrm{R}^{\mathcal{I}} \wedge e \in \mathrm{B}^{\mathcal{I}}\}$
Universal restriction	$\forall\mathrm{R}.\mathrm{B}$	$\{d \mid \forall e \in \Delta^{\mathcal{I}}, (d,e) \in \mathrm{R}^{\mathcal{I}} \rightarrow e \in \mathrm{B}^{\mathcal{I}}\}$

Table 2.2: Semantics of concept and role constructors.

Definition 3. *An interpretation takes the form $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where $\Delta^{\mathcal{I}}$ is a non-empty set (called the domain) and $\cdot^{\mathcal{I}}$ is the interpretation function that maps each $\mathrm{A} \in \mathsf{N_C}$ to $\mathrm{A}^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, each $\mathrm{P} \in \mathsf{N_R}$ to $\mathrm{P}^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, and each $\mathsf{a} \in \mathsf{N_I}$ to $\mathsf{a}^{\mathcal{I}}$. In this thesis, we will make the* Standard Names Assumption *by setting $\mathsf{a}^{\mathcal{I}} = \mathsf{a}$. Note however that our results only rely upon the weaker* Unique Names Assumption *(UNA), which stipulates that $\mathsf{a}^{\mathcal{I}} \neq \mathsf{b}^{\mathcal{I}}$ whenever $\mathsf{a} \neq \mathsf{b}$. The function $\cdot^{\mathcal{I}}$ extends to roles and complex concepts as summarized in Table 2.2.*

Remark 2. *Notice the set $\Delta^{\mathcal{I}}$ is required not to be empty and to contain at least $\mathsf{N_I}$ (due to the SNA), but is otherwise unrestricted. This is sometimes referred to as the open domain assumption, as opposed to the closed domain assumption underlying usual databases in which no elements outside of those mentioned in the data (the individuals of the ABox, from the OMQA perspective) are considered.*

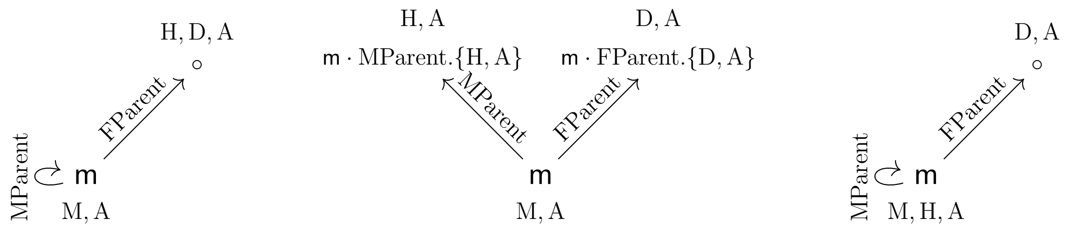
As DL KBs only use unary and binary predicates, an interpretation $\mathcal{I}$ is easily represented as a labeled directed graph according to the two following rules: (i) each $e \in \Delta^{\mathcal{I}}$ is represented by a vertex e, labeled with all the concept names $\mathrm{A} \in \mathsf{N_C}$ such that $e \in \mathrm{A}^{\mathcal{I}}$; (ii) there is a directed edge (e_1, e_2) in the graph representation of $\mathcal{I}$ iff there exists a role name $\mathrm{P} \in \mathsf{N_R}$, such that $(e_1, e_2) \in \mathrm{P}^{\mathcal{I}}$, in which case the edge (e_1, e_2) is labeled with all such role names P. For readability, we often replace each vertex representing an element e from $\Delta^{\mathcal{I}} \setminus \mathsf{Ind}(\mathcal{A})$ by a placeholder $\circ$ to avoid specifying the exact definition of $\Delta^{\mathcal{I}}$.

We now move to the notion of models of a knowledge base.

Definition 4. *An inclusion $\mathrm{G} \sqsubseteq \mathrm{H}$ is satisfied in $\mathcal{I}$ if $\mathrm{G}^{\mathcal{I}} \subseteq \mathrm{H}^{\mathcal{I}}$; an assertion $\mathrm{A}(\mathsf{b})$ (resp. $\mathrm{P}(\mathsf{a}, \mathsf{b})$) is satisfied in $\mathcal{I}$ if $\mathsf{b} \in \mathrm{A}^{\mathcal{I}}$ (resp. $(\mathsf{a}, \mathsf{b}) \in \mathrm{P}^{\mathcal{I}}$). An interpretation is* *a* model *of a TBox $\mathcal{T}$ (resp. KB $\mathcal{K}$) if it satisfies all axioms in $\mathcal{T}$ (resp. axioms and assertions in $\mathcal{K}$). A KB is* satisfiable *if it has at least one model. An inclusion (resp. assertion) Φ is* entailed *from $\mathcal{T}$ (resp. $\mathcal{K}$), written $\mathcal{T} \models \Phi$ (resp. $\mathcal{K} \models \Phi$), if Φ is satisfied in every model of $\mathcal{T}$ (resp. $\mathcal{K}$).*

Example 2. *Continuing Example 1, three interpretations of $\mathcal{K}_{ex}$ are depicted in Figure 2.2, in which individual and concept names have been abbreviated to their first letter for readability. The interpretation $\mathcal{I}_1$ is not a model as the anonymous element (depicted by $\circ$) satisfies both* Horse *and* Donkey*, hence violating axiom* $\text{Horse} \sqcap \text{Donkey} \sqsubseteq \bot$*. It also violates the first axiom of $\mathcal{T}_{ex}$ since* molly *is a* Mule *that does not have a* MaleParent *being a* Horse*.*

The two other interpretations are indeed models of $\mathcal{K}_{ex}$. Notice in particular our ontology is somehow ill-formed as it permits molly *to be its own male parent being a horse. One could exclude this kind of model, e.g. by adding a negative concept inclusion* $\text{Mule} \sqcap \text{Horse} \sqsubseteq \bot$ *to the ontology.*



(a) Interp. $\mathcal{I}_1$ of $\mathcal{K}_{ex}$ (b) Model $\mathcal{I}_2$ of $\mathcal{K}_{ex}$ (c) Model $\mathcal{I}_3$ of $\mathcal{K}_{ex}$

Figure 2.2: Interpretations of $\mathcal{K}_{ex}$ for Example 2.

Remark 3. *Notice that a model can interpret some individual names as satisfying more concept names than required by the KB: in our example,* molly *is a horse in model $\mathcal{I}_3$ but not in model $\mathcal{I}_2$, hence the fact* Horse(molly) *is not entailed by $\mathcal{K}_{ex}$, but it doesn't contradict $\mathcal{I}_{ex}$ being a model. This is referred to as the open world assumption, as opposed to the closed world assumption underlying usual databases in which unsure facts are assumed to be false.*

It will often be useful to manipulate interpretations similarly to ABoxes, which motivates the following definition.

Definition 5. *We can view an interpretation $\mathcal{I}$ as a (possibly infinite) set of assertions $\mathcal{A}_\mathcal{I} = \{\text{A}(e) \mid e \in \text{A}^\mathcal{I}, \text{A} \in \mathsf{N_C}\} \cup \{\text{P}(e, e') \mid (e, e') \in \text{P}^\mathcal{I}, \text{P} \in \mathsf{N_R}\}$. We say that $\mathcal{I}$ is $\mathcal{T}$-*satisfiable *if $\mathcal{T} \cup \mathcal{A}_\mathcal{I}$ has a model.*

2.1.3 Normal forms

It is standard to assume that the TBoxes are in a given normal form, which is often tailored to the setting of interest. Such a normal form simplifies the description of algorithms and proofs of modelhood by restricting the shapes of axioms that need to be considered. So long as the transformation to normal form preserves the central properties of the considered problem (*e.g.* satisfiability of the TBox), it can be assumed without loss of generality that all TBoxes are in normal form. In our own study, we shall work with $\mathcal{ALCHI}$ TBoxes that in the following normal form, which extends the normal form presented in Baader et al. [2017].

The normalization proceeds in three steps. The first step aims at removing *nested* occurrences of concepts. We say a concept B has a nested occurrence in a concept D if B is not a concept name and D has one of the following shapes: $\neg B \mid B \sqcap C \mid B \sqcup C \mid C \sqcap B \mid C \sqcup B \mid \exists R.B \mid \forall R.B$, where C is any concept (nested or not). As is standard (see e.g. Baader et al. [2017]), we can assume w.l.o.g. that there are no such nested occurrences by introducing linearly many fresh concept names and axioms in the TBox.

The second step replaces each axiom $B_1 \sqsubseteq B_2$ by the two axioms $B_1 \sqsubseteq A_{B_1,B_2}$ and $A_{B_1,B_2} \sqsubseteq B_2$, where A_{B_1,B_2} is a dedicated fresh concept name. This only doubles the size of the TBox and ensures each concept inclusion has now a single concept name on its left-hand side or on its right-hand side. We are therefore left with the following 15 shapes of concept inclusions:

$$
\begin{array}{rl}
\multicolumn{2}{c}{A \sqsubseteq B} \\
\top \sqsubseteq A & A \sqsubseteq \top \\
\bot \sqsubseteq A & A \sqsubseteq \bot \\
A_1 \sqcap A_2 \sqsubseteq A & A \sqsubseteq A_1 \sqcap A_2 \\
A_1 \sqcup A_2 \sqsubseteq A & A \sqsubseteq A_1 \sqcup A_2 \\
\exists R.B \sqsubseteq A & A \sqsubseteq \exists R.B \\
\forall R.B \sqsubseteq A & A \sqsubseteq \forall R.B \\
\neg B \sqsubseteq A & A \sqsubseteq \neg B
\end{array}
\qquad \text{with } A, A_1, A_2, B \in \mathsf{N_C} \text{ and } R \in \mathsf{N_R^{\pm}}
$$

The third step applies the 9 kinds of substitutions from Table 2.3 to reduce to the following 6 shapes of concept axioms:

$$\top \sqsubseteq A \qquad A_1 \sqcap A_2 \sqsubseteq A \qquad \exists R.B \sqsubseteq A \qquad A \sqsubseteq \exists R.B \qquad \neg B \sqsubseteq A \qquad A \sqsubseteq \neg B,$$

with $A, A_1, A_2, B \in \mathsf{N_C}$ and $R \in \mathsf{N_R^{\pm}}$. Each substitution from Table 2.3 directly gives axioms with the desired shapes, so a single iteration of these rules is required.

This latter step gives us the desired normal form.

$$\begin{array}{llll}
\mathrm{A} \sqsubseteq \top & \rightsquigarrow & \mathrm{A} \sqsubseteq \mathrm{B} & \rightsquigarrow \mathrm{A} \sqcap \mathrm{C}_\top \sqsubseteq \mathrm{B} \\
\mathrm{A} \sqsubseteq \bot & \rightsquigarrow \begin{cases} \mathrm{A} \sqsubseteq \neg \mathrm{C}_\top \\ \top \sqsubseteq \mathrm{C}_\top \end{cases} & \bot \sqsubseteq \mathrm{A} & \rightsquigarrow \\
\mathrm{A} \sqsubseteq \mathrm{A}_1 \sqcap \mathrm{A}_2 & \rightsquigarrow \begin{cases} \mathrm{A} \sqcap \mathrm{C}_\top \sqsubseteq \mathrm{A}_1 \\ \mathrm{A} \sqcap \mathrm{C}_\top \sqsubseteq \mathrm{A}_2 \\ \top \sqsubseteq \mathrm{C}_\top \end{cases} & & \\
\mathrm{A} \sqsubseteq \mathrm{A}_1 \sqcup \mathrm{A}_2 & \rightsquigarrow \begin{cases} \mathrm{A} \sqsubseteq \neg \mathrm{C}_{\neg \mathrm{A}_1 \sqcap \neg \mathrm{A}_2} \\ \mathrm{C}_{\neg \mathrm{A}_1} \sqcap \mathrm{C}_{\neg \mathrm{A}_2} \sqsubseteq \mathrm{C}_{\neg \mathrm{A}_1 \sqcap \neg \mathrm{A}_2} \\ \neg \mathrm{A}_1 \sqsubseteq \mathrm{C}_{\neg \mathrm{A}_1} \\ \neg \mathrm{A}_2 \sqsubseteq \mathrm{C}_{\neg \mathrm{A}_2} \end{cases} & \mathrm{A}_1 \sqcup \mathrm{A}_2 \sqsubseteq \mathrm{A} & \rightsquigarrow \begin{cases} \mathrm{A}_1 \sqcap \mathrm{C}_\top \sqsubseteq \mathrm{A} \\ \mathrm{A}_2 \sqcap \mathrm{C}_\top \sqsubseteq \mathrm{A} \\ \top \sqsubseteq \mathrm{C}_\top \end{cases} \\
\mathrm{A} \sqsubseteq \forall \mathrm{R}.\mathrm{B} & \rightsquigarrow \begin{cases} \mathrm{A} \sqsubseteq \neg \mathrm{C}_{\exists \mathrm{R}.\neg \mathrm{B}} \\ \exists \mathrm{R}.\mathrm{C}_{\neg \mathrm{B}} \sqsubseteq \mathrm{C}_{\exists \mathrm{R}.\neg \mathrm{B}} \\ \neg \mathrm{B} \sqsubseteq \mathrm{C}_{\neg \mathrm{B}} \end{cases} & \forall \mathrm{R}.\mathrm{B} \sqsubseteq \mathrm{A} & \rightsquigarrow \begin{cases} \neg \mathrm{C}_{\exists \mathrm{R}.\neg \mathrm{B}} \sqsubseteq \mathrm{A} \\ \exists \mathrm{R}.\mathrm{C}_{\neg \mathrm{B}} \sqsubseteq \mathrm{C}_{\exists \mathrm{R}.\neg \mathrm{B}} \\ \neg \mathrm{B} \sqsubseteq \mathrm{C}_{\neg \mathrm{B}} \end{cases}
\end{array}$$

where concepts C_X are fresh concept names representing (complex) concepts X.

Table 2.3: Normalization of $\mathcal{ALCHI}$ ontologies.

Definition 6. *An $\mathcal{ALCHI}$ TBox $\mathcal{T}$ is said to be in* normal form *if every concept inclusion in $\mathcal{T}$ has one of the following shapes:*

$$\top \sqsubseteq \mathrm{A} \qquad \mathrm{A}_1 \sqcap \mathrm{A}_2 \sqsubseteq \mathrm{A} \qquad \exists \mathrm{R}.\mathrm{B} \sqsubseteq \mathrm{A} \qquad \mathrm{A} \sqsubseteq \exists \mathrm{R}.\mathrm{B} \qquad \neg \mathrm{B} \sqsubseteq \mathrm{A} \qquad \mathrm{A} \sqsubseteq \neg \mathrm{B},$$

with $\mathrm{A}, \mathrm{A}_1, \mathrm{A}_2, \mathrm{B} \in \mathsf{N_C}$ *and* $\mathrm{R} \in \mathsf{N_R^\pm}$.

To ensure the normalization procedure does not affect the outcome of the reasoning tasks we consider in this thesis, it is sufficient to ensure that the normalized TBox is a conservative extension of the initial TBox.

Definition 7. *A TBox $\mathcal{T}'$ is a* conservative extension *of a TBox $\mathcal{T}$ if the three following conditions are satisfied:*

- $\mathsf{sig}(\mathcal{T}) \subseteq \mathsf{sig}(\mathcal{T}')$;
- *Every model of $\mathcal{T}'$ is a model of $\mathcal{T}$;*
- *For every model $\mathcal{I}$ of $\mathcal{T}$, there exists a model $\mathcal{I}'$ of $\mathcal{T}'$ such that the restriction of $\cdot^{\mathcal{I}'}$ to $\mathsf{sig}(\mathcal{T})$ is $\cdot^{\mathcal{I}}$.*

The desired properties of the normalization procedure can now be summarized as follows.

Theorem 1. *Every $\mathcal{ALCHI}$ TBox $\mathcal{T}$ can be transformed in linear time into a conservative extension $\mathcal{T}'$ of $\mathcal{T}$ such that $\mathcal{T}'$ is in normal form and has linear size w.r.t. the size of $\mathcal{T}$.*

In particular, this transformation does not affect the outcome of upcoming reasoning tasks, nor the associated complexity results.

2.1.4 Canonical models for $\mathcal{ELHI}_\bot$ KBs

As previously noted, many DLs of the DL-Lite and $\mathcal{EL}$ families [Calvanese et al., 2005; Baader et al., 1999] allow for efficient reasoning due to their carefully restricted syntax. Such logics belong to the broader class of Horn DLs, which are those that cannot express (implicitly or explicitly) any form of disjunction, and thus do not require reasoning by cases. More expressive Horn DLs can be defined by selecting a (highly) expressive DL, like $\mathcal{SHIQ}$, and suitably restricting its syntax to exclude the need for disjunctive reasoning, yielding e.g. Horn-$\mathcal{SHIQ}$ [Hustadt et al., 2005; Krötzsch et al., 2013]. The key property of Horn DLs is that every satisfiable KB admits a canonical (or universal) model that embeds homomorphically into each of its models. Such a canonical model plays a central role in designing reasoning procedures as it often suffices to restrict the attention to this single model.

In our setting, it is well known that every satisfiable $\mathcal{ELHI}_\bot$ KB admits a canonical model ($\mathcal{ELHI}_\bot$ being essentially Horn-$\mathcal{ALCHI}$, up to some syntactic reformulations relying on inverse roles). We recall how such a model $\mathcal{C}_\mathcal{K}$ can be constructed (see [Bienvenu and Ortiz, 2015]).

Definition 8. *The domain $\Delta^{\mathcal{C}_\mathcal{K}}$ consists of all sequences $\mathsf{a} \cdot \mathrm{R_1.M_1} \cdots \mathrm{R_n.M_n}$ $(\mathsf{n} \geq 0)$ such that $\mathsf{a} \in \mathsf{Ind}(\mathcal{A})$, each $\mathrm{R_i}$ belongs to $\mathsf{N}_\mathsf{R}^\pm$, each $\mathrm{M_i}$ is a conjunction of concepts from $\mathsf{N_C} \cup \{\top\}$ (which is treated as a set when convenient), and the following conditions hold:*

1. *If $\mathsf{n} \geq 1$, then $\mathcal{T} \models \mathrm{M_0} \sqsubseteq \exists \mathrm{R_1.M_1}$ where $\mathrm{M_0} = \{\mathrm{A} \in \mathsf{N_C} \cup \{\top\} \mid \mathcal{K} \models \mathrm{A}(\mathsf{a})\}$ and $\mathrm{M_1}$ is maximal, as a set of concept names, for this property.*

2. *If $\mathsf{n} \geq 1$, then there is no $\mathsf{b} \in \mathsf{Ind}(\mathcal{A})$ such that $\mathcal{K} \models \mathrm{M_1}(\mathsf{b})$ and $\mathcal{K} \models \mathrm{R_1}(\mathsf{a}, \mathsf{b})$.*

3. *For every $1 \leq \mathsf{i} < \mathsf{n}$, $\mathcal{T} \models \mathrm{M_i} \sqsubseteq \exists \mathrm{R_{i+1}.M_{i+1}}$ and $\mathrm{M_{i+1}}$ is maximal, as a set of concept names, for this property.*

4. *For every $1 \leq \mathsf{i} \leq \mathsf{n} - 2$, $\mathcal{T} \not\models \mathrm{M_i} \sqsubseteq \mathrm{M_{i+2}}$ or $\mathcal{T} \not\models \mathrm{R_{i+1}} \sqsubseteq \mathrm{R^-_{i+2}}$.*

Individual names are interpreted as themselves ($\mathsf{a}^{\mathcal{C}_\mathcal{K}} = \mathsf{a}$), and concept and role *names are interpreted as follows:*

$$\begin{array}{rcll} \mathrm{A}^{\mathcal{C}_{\mathcal{K}}} & := & \{\mathsf{a} \mid \mathcal{K} \models \mathrm{A}(\mathsf{a})\} & (1) \\ & & \cup \{e \cdot \mathrm{R.M} \mid \mathrm{A} \in \mathrm{M}\} & (2) \end{array}$$

$$\begin{array}{rcll} \mathrm{P}^{\mathcal{C}_{\mathcal{K}}} & := & \{(\mathsf{a}, \mathsf{b}) \mid \mathcal{K} \models \mathrm{P}(\mathsf{a}, \mathsf{b})\} & (1) \\ & & \cup \{(e, e \cdot \mathrm{P}_0.\mathrm{M}) \mid \mathcal{T} \models \mathrm{P}_0 \sqsubseteq \mathrm{P}\} & (2^+) \\ & & \cup \{(e \cdot \mathrm{P}_0.\mathrm{M}, e) \mid \mathcal{T} \models \mathrm{P}_0 \sqsubseteq \mathrm{P}^-\} & (2^-) \end{array}$$

The canonical model $\mathcal{C}_{\mathcal{K}}$ plays a central role in conjunctive query (CQ) answering, as, by virtue of embedding in every model of the KB $\mathcal{K}$ of interest, it provides either the assurance that the query further embeds in every model $\mathcal{I}$ of $\mathcal{K}$ (if a embedding of the CQ in $\mathcal{C}_{\mathcal{K}}$ exists), or a countermodel for the query (that is, an example of a model in which there are no embedding of the CQ). This issue will be recalled in more detail later in this chapter. To formalize this central property, it is necessary to properly recall the definition of a homomorphism of interpretations.

Definition 9. *Given two interpretations $\mathcal{I}_1$ and $\mathcal{I}_2$, a function $f : \Delta^{\mathcal{I}_1} \to \Delta^{\mathcal{I}_2}$ is a homomorphism of $\mathcal{I}_1$ into $\mathcal{I}_2$, denoted $f : \mathcal{I}_1 \to \mathcal{I}_2$, if the three following conditions hold: (i) $f(\mathsf{a}^{\mathcal{I}_1}) = \mathsf{a}^{\mathcal{I}_2}$ for all $\mathsf{a} \in \mathsf{N_I}$; (ii) $f(\mathrm{A}^{\mathcal{I}_1}) \subseteq \mathrm{A}^{\mathcal{I}_2}$ for all concept name $\mathrm{A} \in \mathsf{N_C}$; (iii) $f(\mathrm{A}^{\mathcal{I}_1}) \subseteq \mathrm{A}^{\mathcal{I}_2}$ for all role name $\mathrm{P} \in \mathsf{N_R}$.*

We can now formally recall the central property of the canonical model, which motivates its name.

Theorem 2. *Let $\mathcal{K}$ be a satisfiable $\mathcal{ELHI}_\bot$ KB. The canonical model $\mathcal{C}_{\mathcal{K}}$ is a model of $\mathcal{K}$ and for every model $\mathcal{I}$ of $\mathcal{K}$, there exists a homomorphism $f : \mathcal{C}_{\mathcal{K}} \to \mathcal{I}$.*

Example 3. *The canonical model of $\mathcal{K}_{ex}$ from Example 1 is depicted as model $\mathcal{I}_2$ in Figure 2.2. It embeds in model $\mathcal{I}_3$ by mapping* $\mathsf{molly} \cdot \mathrm{MaleParent}.\{\mathrm{Horse}, \mathrm{Animal}\}$ *to* molly *and* $\mathsf{molly} \cdot \mathrm{FemaleParent}.\{\mathrm{Donkey}, \mathrm{Animal}\}$ *to itself.*

2.1.5 Closed predicates

The open-world and open-domain assumptions are natural in settings where the data is incomplete, and there may be missing facts and a need to reason about unnamed objects. Many scenarios, however, may involve some parts of the data which are incomplete, and other parts which are known to be complete (*e.g.* when considering the list of countries). The combination of DL reasoning with (partially) complete data was first explored in Franconi et al. [2011] and led to a line of work on DL KBs with closed predicates [Lutz et al., 2013; Ngo et al., 2016], which allow for a trade-off between the closed- and open-world assumptions. Formally, one adapts the notions of KBs and models as follows, where the interpretations of some predicates are stated to be fully known, and hence should not extend beyond the instances explicitly given in the ABox.

Definition 10. *A* KB with closed predicates *consists of a KB* $(\mathcal{T}, \mathcal{A})$ *and a set* $\Sigma \subseteq \mathsf{N_C} \cup \mathsf{N_R}$ *of* closed predicates. *An interpretation* $\mathcal{I}$ *is a* model of $(\mathcal{T}, \mathcal{A}, \Sigma)$ *if it is a model of* $(\mathcal{T}, \mathcal{A})$ *which interprets the closed predicates according to* $\mathcal{A}$*, i.e.* $\mathrm{A}^\mathcal{I} = \{\mathsf{a} \mid \mathrm{A}(\mathsf{a}) \in \mathcal{A}\}$ *for every* $\mathrm{A} \in \Sigma \cap \mathsf{N_C}$ *and* $\mathrm{P}^\mathcal{I} = \{(\mathsf{a}, \mathsf{b}) \mid \mathrm{P}(a, b) \in \mathcal{A}\}$ *for every* $\mathrm{P} \in \Sigma \cap \mathsf{N_C}$*.*

Example 4. *In our running example,* $\mathcal{K}_{ex}$ *admits models as presented before, but the corresponding KB with a single closed predicate* Animal*, that is* $\mathcal{K}'_{ex} := (\mathcal{T}_{ex}, \{\mathrm{Animal}\}, \mathcal{A}_{ex}\}$ *becomes unsatisfiable.*

Closed predicates have been explored for a range of DLs and have been shown to increase significantly the complexity of the most common reasoning tasks compared to the classical setting without closed predicates [Franconi et al., 2011; Lutz et al., 2013; Ngo et al., 2016].

2.2 Reasoning tasks

We now recall the usual reasoning tasks associated with knowledge bases and summarize the known complexity of answering these problems. For each task, we distinguish between *combined complexity* in which everything is part of the input, and *data complexity* in which only the data, that is, the ABox, is considered as input and the other parameters are treated as fixed.

Both combined and data complexity measures consider the worst-case complexity of the problem. It can also be interesting to pinpoint the complexity of a particular ontology or ontology-mediated query (OMQ), i.e. an ontology-query pair. This more refined approach has yielded several dichotomy results and complexity classifications, which identify what are the possible complexities and pinpoint the tractable and intractable cases [Lutz and Wolter, 2012; Lutz et al., 2012; Bienvenu et al., 2014b; Lutz and Sabellek, 2017].

2.2.1 Satisfiability, subsumption and instance checking

The most basic reasoning task associated with a TBox or a KB is arguably to ask whether it is consistent or not. This is known as the satisfiability problem.

Definition 11. *Given a TBox* $\mathcal{T}$*, resp. a KB* $\mathcal{K}$*, the satisfiability problem is to decide whether* $\mathcal{T}$*, resp.* $\mathcal{K}$*, admits a model.*

	Satisfiability		Instance checking		CQ answering	
	Data	*Combined*	*Data*	*Combined*	*Data*	*Combined*
DL-Lite$^{(\mathcal{H})}_{\mathsf{core}}$	in AC^0	NL	in AC^0	NL	in AC^0	NP
$\mathcal{EL}$, $\mathcal{ELH}_\bot$	P	P	P	P	P	NP
$\mathcal{ELI}$, $\mathcal{ELHI}_\bot$	P	EXP	P	EXP	P	EXP
$\mathcal{ALC}$	NP	EXP	coNP	EXP	coNP	EXP
$\mathcal{ALCI}$	NP	EXP	coNP	EXP	coNP	2EXP

Table 2.4: Complexity of common reasoning tasks in standard DLs. Lower bounds for satisfiability do not apply for $\mathcal{EL}$ and $\mathcal{ELI}$ KBs which always admit a model.

The second task concerns TBoxes, and asks whether a new inclusion can be inferred from the given ones, which is known as the subsumption problem.

Definition 12. *Given a TBox $\mathcal{T}$ and two concepts* $\mathrm{C}_1, \mathrm{C}_2$*, the subsumption problem is to decide whether $\mathcal{T}$ entails* $\mathrm{C}_1 \sqsubseteq \mathrm{C}_2$.

The third problem is the assertional counterpart of the subsumption problem, asking whether a given assertion can be inferred from a given KB, which is known as instance checking.

Definition 13. *Given a KB $\mathcal{K}$ and a concept* C *and an individual name* $\mathsf{a} \in \mathsf{N_I}$*, the instance checking problem is to decide whether $\mathcal{K}$ entails the assertion* $\mathrm{C}(\mathsf{a})$.

These three reasoning tasks are known to be reducible to each other as soon as disjointness is expressible in the TBox (e.g. with DL-Lite$_{\mathsf{core}}$ or $\mathcal{EL}_\bot$), and both their data and combined complexities are well understood for sublogics of $\mathcal{ALCI}$ and of $\mathcal{ELHI}_\bot$. These results are recalled in Table 2.4, borrowed from [Bienvenu and Ortiz, 2015], and have been obtained from a variety of techniques.

One prominent approach for Horn DLs is query rewriting, in which reasoning tasks are reduced to the more well-known problems of evaluating first-order (FO) or Datalog queries over databases. More precisely, query rewriting takes a TBox and query as input and produces an FO-query (resp. Datalog-query) that incorporates the relevant knowledge from the TBox and is such that evaluating this query over the ABox yields the required result for the initial reasoning task. It is known that FO query evaluation is PSPACE-complete w.r.t. combined complexity [Vardi, 1982] and in AC^0 w.r.t. data complexity [Vardi, 1995], while Datalog query evaluation is EXP-complete w.r.t. combined complexity [Vardi, 1982] and P-complete w.r.t. data complexity [Immerman, 1986]. First-order query rewriting can be used to obtain an AC^0 procedure (w.r.t. data complexity) for instance checking for a range of DL-Lite dialects, including DL-Lite$^{\mathcal{H}}_{\mathsf{core}}$, and can also be used to show an NL upper bound in combined complexity [Artale et al., 2009]. By rephrasing satisfiability as a query answering task, we can obtain the same upper bounds for satisfiability of DL-Lite KBs. For $\mathcal{EL}$ and its extensions, it is not always possible to reduce to FO query evaluation, but Datalog rewriting can be used to establish tight upper bounds for instance checking and satisfiability in $\mathcal{ELHI}_\bot$ [Hustadt et al., 2005]. Note that for $\mathcal{EL}$ and $\mathcal{ELI}$, only the instance checking problem is of interest, as the satisfiability task is trivial, due to the absence of disjointness, negation, or other constraints.

Another prominent reasoning technique for Horn DLs is saturation (or materialization), which consists in iteratively adding (some of) the facts that can be entailed from the KB, then checking whether the target query has been produced. The P upper bounds in combined complexity for $\mathcal{EL}$ and its extension $\mathcal{ELH}_\bot$ were originally established using such saturation techniques [Baader et al., 2005] (see Calvanese et al. [2006] for the matching lower bounds).

For expressive DLs, reasoning tasks are often rephrased as satisfiability checks and addressed with tableaux techniques. In a nutshell, tableaux algorithms test the satisfiability of the input KB by trying to construct a (representation of a) model. They can be seen as extending saturation procedures by exploring the different ways of adding facts to account for the disjunctive features allowed in the KBs. A tableaux procedure for the DL $\mathcal{ALCI}$ can be found in Donini and Massacci [2000], and the even more expressive DL $\mathcal{SHIQ}$ was addressed in Tobies [2001].

Throughout the later chapters, we will often need to perform some satisfiability (resp. subsumption and instance checking) checks, relying upon these complexity results. For satisfiability tests for our slightly non-standard versions of $\mathcal{ALCH}$ and $\mathcal{ALCHI}$ allowing negative role inclusions, we prove that it remains EXP-complete with respect to combined complexity (satisfiability of $\mathcal{ALCHI}$ KBs without negative role inclusions being EXP-complete as proven in [Tobies, 2001] and [Schild, 1991]). Although we will not need the corresponding statement for data complexity, it also remains NP-complete, as follows from a later result (Theorem 8).

Theorem 3. *The satisfiability of a $\mathcal{ALCHI}$ KB with role disjointness is* EXP-*complete w.r.t. combined complexity.*

Proof. EXP-hardness is immediate as $\mathcal{ALCHI}$ extends $\mathcal{ALC}$, for which the satisfiability task is already EXP-complete [Schild, 1991]. For the upper bound, we reduce our problem to the satisfiability problem of $\mathcal{ALCI}b$ KBs, also known to be EXP-complete (see Theorem 4.42 in Tobies [2001]). An $\mathcal{ALCI}b$ KB extends the $\mathcal{ALCI}$ KBs presented in this chapter by allowing more expressive combinations of roles in the construction of concepts (see Definition 4.17 in Tobies [2001]): "An $\mathcal{ALCI}b$-role expression ω is built from $\mathcal{ALCI}b$-roles [*i.e.* roles from $\mathsf{N}_\mathsf{R}^\pm$] using the operators $\sqcap$ (role intersection), $\sqcup$ (role union), and $\neg$ (role complement), with the restriction that, when transformed into disjunctive normal form, every disjunct contains at least one non-negated conjunct. A role expression that satisfies this constraint is called safe."

Consider a $\mathcal{ALCHI}$ KB $\mathcal{K} := (\mathcal{T}, \mathcal{A})$ in normal form. We first construct an extension $\mathcal{A}'$ of the ABox $\mathcal{A}$. For each assertion $\mathrm{R}(\mathsf{a}, \mathsf{b}) \in \mathcal{A}$ and each (eventually inverse) role S such that $\mathcal{T} \models \mathrm{R} \sqsubseteq \mathrm{S}$, we add the assertion $\mathrm{S}(\mathsf{a}, \mathsf{b})$ (or $\mathrm{S}(\mathsf{b}, \mathsf{a})$ in the inverse case) to $\mathcal{A}'$. We now turn to a modified version $\mathcal{T}'$ of the TBox $\mathcal{T}$, in which each role inclusion is dropped, each axiom $\exists \mathrm{R}.\mathrm{B} \sqsubseteq \mathrm{A} \in \mathcal{T}$ is replaced by the axiom $\exists \omega.\mathrm{B} \sqsubseteq \mathrm{A}$, where ω is the following safe role expression:

$$\omega := \bigcup_{\mathcal{T} \models \mathrm{S} \sqsubseteq \mathrm{R}} \mathrm{S},$$

and each axiom $\mathrm{A} \sqsubseteq \exists \mathrm{R}.\mathrm{B} \in \mathcal{T}$ is replaced by the axiom $\mathrm{A} \sqsubseteq \exists \omega.\mathrm{B}$ where ω is the following safe role expression:

$$\omega := \left(\bigcap_{\mathcal{T} \models \mathrm{R} \sqsubseteq \mathrm{S}} \mathrm{S} \right) \cap \left(\bigcap_{\mathcal{T} \models \mathrm{R} \sqcap \mathrm{T} \sqsubseteq \bot} \neg \mathrm{T} \right).$$

Note that all the role inclusion checks in this construction can be polynomially decided due to the very limited role constructors and inclusions in $\mathcal{ALCHI}$. It remains to establish the following claim:

$$\mathcal{K}' := (\mathcal{T}', \mathcal{A}') \text{ is satisfiable iff } \mathcal{K} \text{ is satisfiable.}$$

($\Leftarrow$). It is easily checked that every model $\mathcal{I}$ of $\mathcal{K}$ is also a model of $\mathcal{K}'$.

($\Rightarrow$). Consider a model $\mathcal{I}'$ of $\mathcal{K}'$. Axioms $\exists \mathrm{R}.\mathrm{B} \sqsubseteq \mathrm{A} \in \mathcal{T}$ are clearly satisfied in $\mathcal{I}'$. However, $\mathcal{I}'$ may violate some role inclusions and role disjointness axioms from $\mathcal{T}$. For each $\mathrm{A} \sqsubseteq \exists \mathrm{R}.\mathrm{B} \in \mathcal{T}$, and each element $e \in \mathrm{A}^{\mathcal{I}'}$, our construction ensures that there exists at least one successor to e for the corresponding $\mathcal{T}'$ axiom that respects both the positive and negative role inclusions from $\mathcal{T}$. Dropping all role facts in $\mathcal{I}'$ that are neither involved in such a successor relationship nor entailed on individuals by $\mathcal{K}$, we obtain a model $\mathcal{I}$ of $\mathcal{K}$. □

2.2.2 Query answering

As the counting conjunctive queries we study in this thesis correspond to an extension of classical conjunctive queries, we briefly recall the definition of such queries, which constitute a simple, yet practically relevant and much studied, query language. A *conjunctive query (CQ)* takes the form $q(\mathbf{x}) = \exists \mathbf{y}\ \psi(\mathbf{x}, \mathbf{y})$, where $\mathbf{x}, \mathbf{y}$ are tuples of *answer and existential variables*, respectively, and ψ is a conjunction of concept and role atoms with terms from $\mathsf{N_I} \cup \mathbf{x} \cup \mathbf{y}$. We use $\mathsf{terms}(q)$ for the set of all terms occurring in q. A CQ q is said to be *Boolean* if $\mathbf{x} = \emptyset$.

A *match* for a CQ q in an interpretation $\mathcal{I}$ is a homomorphism from q into $\mathcal{I}$, i.e. a function π that maps each term in q to an element of $\Delta^{\mathcal{I}}$ such that $\pi(t) = t$ when $t \in \mathsf{N_I}$, $\pi(t) \in \mathrm{A}^{\mathcal{I}}$ for every $\mathrm{A}(t) \in q$, and $(\pi(t), \pi(t')) \in \mathrm{P}^{\mathcal{I}}$ for every $\mathrm{P}(t, t') \in q$. The set of *answers* to q in $\mathcal{I}$, denoted $q^{\mathcal{I}}$, contains all tuples $\mathbf{a}$ of individuals from $\mathsf{N_I}$ such that there exists a match of $q(\mathbf{a})$ in $\mathcal{I}$. A *certain answer* to a CQ q w.r.t. $\mathcal{K}$ is an answer in every model of $\mathcal{K}$, that is, a tuple from $q^{\mathcal{K}} := \bigcap_{\mathcal{I} \models \mathcal{K}} q^{\mathcal{I}}$.

Definition 14. *Given a KB $\mathcal{K}$, a CQ q and a tuple* $\mathbf{a}$, *the problem of CQ answering is to decide whether $\mathbf{a} \in q^{\mathcal{K}}$.*

A summary of the complexity results for CQ answering over the considered DLs is provided in Table 2.4 and, as for the previous reasoning tasks, the upper bounds often rely on rewriting techniques. Query rewriting notably underlies the data complexity results for the DL-Lite family, and there have been many rewriting algorithms developed since the original PerfectRef algorithm [Calvanese et al., 2005, 2007b]. Rewriting techniques have been employed to answers CQs in the presence of KBs formulated in the $\mathcal{EL}$ family [Rosati, 2007; Krisnadhi and Lutz, 2007; Krötzsch and Rudolph, 2007] and have been extended to handle Horn versions of expressive DLs [Eiter et al., 2012a].

For expressive DLs, rewriting procedures to Disjunctive-Datalog are possible [Motik, 2006; Hustadt et al., 2007; Rudolph et al., 2012] (see notably Lutz [2008] for some lower bounds). Saturation techniques or variations of tableaux-based procedure can also prove useful for CQ answering in such DLs [Ortiz et al., 2008]. Other algorithms used to handle expressive DLs mostly rely on two main steps [Glimm et al., 2008; Eiter et al., 2008, 2012b; Kikot et al., 2012]. The first step is to split the query into a part mapping on individuals from the ABox while other parts are to be mapped on tree-shaped interpretations completing the ABox. Whether such mappings in tree-shaped structures exists in all models of the KB of interest form the second step of the algorithm. This step can notably reuse existing results on instance checking since the selected tree-like parts of the query can be expressed as a single concept. However, the first step often creates an exponential number of instances for the second step, based on the possible decompositions of the query, that may result in an exponential increase in combined complexity between instance checking and CQ answering (see *e.g.* the situation for $\mathcal{ALCI}$ KBs in Table 2.4).

Interestingly, several works have explored the possibility to mixing rewriting and saturation procedures in order to keep the best of both techniques, resulting in the so-called combined approach. This provides alternative ways to tackle $\mathcal{EL}$ and several of its extensions [Lutz et al., 2009], some dialects of the DL-Lite family [Kontchakov et al., 2011] and even the Horn version of $\mathcal{ALCHOIQ}$ [Carral et al., 2018].

3 Counting Conjunctive Queries

In this chapter, we introduce the semantics of counting conjunctive queries (CCQs) and the corresponding ontology-mediated query answering problem (OMQA). We further study the computational complexity of this problem for knowledge bases (KBs) expressed in $\mathcal{ALCHI}$ and its sublogics. Our results are summarized in Table 3.1.

	Combined complexity	*Data complexity*
DL-Lite$^{\mathcal{H}}_{\mathsf{pos}}$, $\mathcal{EL}$, $\mathcal{ALCHI}$	2EXP-complete	coNP-complete$^{\natural}$
DL-Lite$_{\mathsf{pos}}$, DL-Lite$_{\mathsf{core}}$	coNEXP-complete	coNP-complete$^{\natural}$

Table 3.1: Complexity of CCQ answering. $^{\natural}$: previously known lower bound.

Section 3.1 presents the semantics of CCQs, its connection with existing work, and the associated decision problem in term of combined and data complexities. Section 3.2 investigates a family of models, namely *interlacings*, built from an initial model of interest, from which they retain desirable properties with respect to CCQs while enjoying a more tree-shaped structure. Based on those interlacings, Section 3.3 establishes a 2EXP procedure, with respect to combined complexity, to answer CCQs over $\mathcal{ALCHI}$ KBs. Afterwards, in Section 3.4, it is shown how to construct optimal models of bounded size, yielding a coNP procedure for CCQ answering over $\mathcal{ALCHI}$ KBs with respect to data complexity, and allowing us to refine the 2EXP algorithm in combined complexity into a coNEXP procedure for DL-Lite$_{\mathsf{core}}$ KBs. Section 3.5 concludes the chapter by providing matching lower bounds and draws a first connection to closed predicates.

Organization of Chapter 3

3.1 Preliminaries

Aggregate queries, which use numeric operators (*e.g.* count, sum, max) to summarize selected parts of a dataset, constitute a prominent class of database queries. Although such queries are widely used for data analysis, they have been little explored in context of OMQA. This may be partly due to the fact that it is not at all obvious how to define the semantics of such queries in the OMQA setting.

3.1.1 Related work

Aggregate queries have been first studied for relational databases before being integrated in other knowledge representation frameworks. In Klug [1982], these queries are formulated with the standard relational query language SQL[1] (see *e.g.* Ullman [1988] for a presentation of SQL) and allow to aggregate the values from selected entries of a relational table. The expressive power of SQL has notably drawn attention due to the support of these aggregate operators [Libkin,

[1] `https://www.iso.org/standard/63555.html`

2003]. Similar aggregate features have also been investigated in the RDF query language SPARQL[2] [Kaminski et al., 2016], and are now supported by modern implementations such as RDFox[3] [Nenov et al., 2015]. The upcoming standard for querying graph databases GQL[4], inspired, among others, by both standards SQL and SPARQL, plans to integrate aggregate features too [Deutsch et al., 2021].

Answering aggregate queries over inconsistent databases has also received attention: in Arenas et al. [2003], a range semantics is proposed to bound the answers of an aggregate query across the repairs of a database violating some functional dependencies (with the notion of answer in a repair defined as in the relational setting).

Aggregate query answering over incomplete data is also addressed. In presence of conditional tables, that allow to manipulate unknown or missing information in relational databases by specifying various conditions on entries whose exact values are unknown, data can still be aggregated, resulting in an answer being itself a conditional table [Lechtenbörger et al., 2002]. In presence of source-to-target tuple-generating dependencies (s-t tgds), various semantics have been proposed in Afrati and Kolaitis [2008] to account for the possible nulls that may arise. It is worth mentioning that their count operator, denoted $count(*)$ in the reference, is allowed to count null elements in the considered models (while other aggregate operators simply drop these nulls), and that the complexity of deciding model-independent bounds on these count numbers is in P due to the restricted retained notion of models (endomorphisms of the canonical model, for the interested reader).

The rule-based language Datalog [Ceri et al., 1990; Ullman, 1988] has also been extended with aggregate operators. They have indeed been studied to enrich the expressive power of rules expressed in Datalog [Consens and Mendelzon, 1993], or in disjunctive Datalog with a notable implementation in the DLV system [Dell'Armi et al., 2003]. More recently, restrictions of Datalog$_{\mathbb{Z}}$, an extension of Datalog which captures many data aggregation tasks by allowing arithmetic functions over integers at the cost of undecidability, have been studied to regain decidability, resulting in the fragment Limit Datalog$_{\mathbb{Z}}$ [Cuenca Grau et al., 2020] whose expressive power has been further studied [Kaminski et al., 2021].

Integration of aggregate functions in another prominent rule-based declarative language, namely Answer Set Programming (ASP), has also drawn particular attention as it notably extends ASP with the possibility to express functional dependencies (see [Gelfond and Lifschitz, 1991] for the original semantics underlying ASP and [Brewka et al., 2011] for a more recent presentation). Several semantics have been proposed to handle more and more forms of aggregates: monotone and

[2] `https://www.w3.org/TR/sparql11-query/`
[3] `https://docs.oxfordsemantic.tech/`
[4] `https://www.iso.org/standard/76120.html`

convex aggregates [Liu and Truszczynski, 2006], non-negated aggregates [Faber et al., 2011; Ferraris, 2011], or aggregate over conditional expressions [Cabalar et al., 2020].

Apart from the mentioned exception of Afrati and Kolaitis [2008], all the above works do not involve elements that are unknown in the original data, while such anonymous elements are one of the main features the OMQA framework aims to take into account. We recall that, in this thesis, our attention focuses on this latter setting and notably differs from the presented works so far as we adopt the open domain and open world assumptions, in particular with expressive DLs that often rely on elements outside of the original data to be satisfiable.

Closer to the realm of description logics, some attempts have been made to enrich the ontology language with aggregate operators (e.g. by allowing concepts that already perform aggregate operation). Equipping the well-known DL $\mathcal{ALC}$ with such features quickly leads to undecidable basic reasoning tasks such as satisfiability and subsumption [Baader and Sattler, 2003], while the situation is more favorable when extending the less expressive DL-Lite family [Artale et al., 2012; Savkovic and Calvanese, 2012; Hernich et al., 2017]. By contrast, in this thesis, we investigate the impact of counting features on the query language rather than on the ontology language.

In the OMQA framework, a first exploration of aggregate queries was conducted by Calvanese et al. [2008]. They argued that the most straightforward adaptation of classical certain answer semantics to aggregate queries was unsatisfactory, as often values would differ from model to model, leading to no certain answers. For this reason, an epistemic semantics was proposed, in which variables involved in the aggregates are required to match to data constants. However, as discussed in Kostylev and Reutter [2015], this semantics can also give unintuitive results by ignoring ways of mapping aggregate variables to anonymous elements inferred due the ontology axioms. For instance, if no children of alex are listed in the data, then a query that asks to return the number of children will yield 0 under epistemic semantics, even if it can be inferred (e.g. due to a family tax benefit) that there must be at least 3 children. This led Kostylev and Reutter to define an alternative semantics for two kinds of counting queries (inspired by the COUNT and COUNT DISTINCT in SQL) which adopts a form of certain answer semantics but considers lower and upper bounds on the count value across different models. This latter semantics relates to those explored for aggregate queries over inconsistent databases, in Arenas et al. [2003], and for data exchange, in Afrati and Kolaitis [2008].

The semantics by Kostylev and Reutter was adopted in later work by Calvanese et al. [2020a], in which DL-Lite ontologies coupled with various restrictions on the counting query shape have been explored. In this latter reference and in Calvanese et al. [2020c], a rewriting procedure is also provided for connected and rooted counting queries, based upon the canonical model being sufficient to decide the problem in this particular setting.

Interestingly, techniques to decide the multiplicity of an answer for a rooted CQ with respect to bag semantics, notably investigated in Nikolaou et al. [2019], are similar to those investigated for rooted counting CQs with respect to set semantics. However, no immediate reduction from one setting to the other seems possible, as discussed in Calvanese et al. [2020a] (see Example 1 in the reference).

Another recent study by Feier et al. [2021] classifies the complexity of counting the number of certain answers (rather than the number of ways a certain answer is obtained) for guarded existential rules. This notably gives lower bounds on the number of answers that might be relevant when allowed to also count outside individual elements, but the converse is false in general (see the discussion following Example 6, later in this chapter).

Instead of counting the (certain) answers, a closely related approach consists of enumerating them, a topic that has been extensively studied in the database setting (see for example the survey [Berkholz et al., 2020]). In enumeration, a preprocessing phase is allowed after which answers must be returned with a permitted delay between two successive answers, the efficiency of the enumeration being measured according to the lengths of both the preprocessing phase and the delay. A recent study by Lutz and Przybylko [2022] studied the enumeration of certain answers to CQs over ontologies expressed in $\mathcal{ELI}$ or as a set of guarded-TGDs.

3.1.2 Semantics of counting conjunctive queries

We propose a new notion of counting CQ that generalizes the two forms of queries from Kostylev and Reutter [2015], hence also those considered in Calvanese et al. [2020a].

Definition 15. *A* counting conjunctive query (CCQ) *takes the form*

$$q(\mathbf{x}) = \exists \mathbf{y}\ \exists \mathbf{z}\ \psi(\mathbf{x}, \mathbf{y}, \mathbf{z}),$$

where $\mathbf{x}, \mathbf{y}, \mathbf{z}$ *are tuples of* answer, existential*, and* counting variables*, respectively, and* ψ *is a conjunction of concept and role atoms with terms from* $\mathsf{N_I} \cup \mathbf{x} \cup \mathbf{y} \cup \mathbf{z}$*. We use* $\mathsf{terms}(q)$ *for the set of all terms occurring in* q*, and we treat queries as sets of atoms when convenient. A CCQ* q *is* Boolean *if* $\mathbf{x} = \emptyset$*.*

The usual notion of conjunctive queries (CQ) is captured by CCQs without counting variables, i.e. $\mathbf{z} = \emptyset$. The counting queries studied in Kostylev and Reutter [2015] were CCQs restricted by $|\mathbf{z}| = 1$, denoted $q(\mathbf{x}, \mathit{Cntd}(z))$ in the reference, and CCQs restricted by $\mathbf{y} = \emptyset$, denoted $q(\mathbf{x}, \mathit{Count}())$ in the reference. Calvanese et al. [2020a] continued the study of the latter subclass of CCQs.

The CQ obtained by replacing each counting variable of a CCQ by a fresh existential variable is referred to as the *underlying CQ of the CCQ*. For readability, it is convenient to represent a CCQ q as a graph: each term t is represented by a vertex v_t labeled by t and by concept names A such that $\mathrm{A}(t) \in q$, and an oriented edge (v_{t_1}, v_{t_2}) labeled with P is added for each atom $\mathrm{P}(t_1, t_2) \in q$. To easily distinguish the status of each term (and often to omit the name of the term), the node v_t is depicted as $\bullet$ if $t \in \mathsf{Ind} \cup \mathbf{x}$, as $\circ$ if $t \in \mathbf{y}$, and as ⬤ if $t \in \mathbf{z}$.

Example 5. *Let us illustrate the notion of CCQ with a toy example, inspired by Bienvenu and Ortiz [2015]. A logician enters a vegetarian-friendly and kid-friendly restaurant* r *in which the menu is partially ripped off, so that only the following facts are readable, here encoded as an ABox* $\mathcal{A}_e$*:*

$\mathrm{VegFriendly}(\mathsf{r})$	$\mathrm{GivesChoice}(\mathsf{m}_1, \mathsf{carb})$	$\mathrm{WithMeat}(\mathsf{carb})$
$\mathrm{KidFriendly}(\mathsf{r})$	$\mathrm{GivesChoice}(\mathsf{m}_2, \mathsf{carb})$	
$\mathrm{Offers}(\mathsf{r}, \mathsf{m}_1)$	$\mathrm{GivesChoice}(\mathsf{m}_2, \mathsf{regi})$	$\mathrm{WithMeat}(\mathsf{regi})$
$\mathrm{Offers}(\mathsf{r}, \mathsf{m}_2)$	$\mathrm{GivesChoice}(\mathsf{m}_2, \mathsf{tira})$	$\mathrm{Dessert}(\mathsf{tira})$
$\mathrm{Menu}(\mathsf{m}_1)$	$\mathrm{GivesChoice}(\mathsf{m}_2, \mathsf{baba})$	$\mathrm{Dessert}(\mathsf{baba})$
$\mathrm{Menu}(\mathsf{m}_2)$		$\mathrm{WithAlcohol}(\mathsf{baba})$

The ABox $\mathcal{A}_e$ *is depicted in Figure 3.1.*

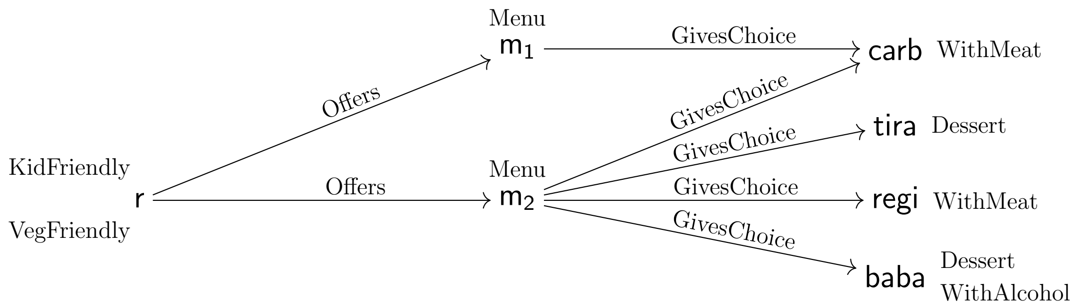


Figure 3.1: The ABox $\mathcal{A}_e$ from Example 5.

Furthermore, clients can expect some general principles to hold, encoded in the following $\mathcal{ALC}$ *TBox* $\mathcal{T}_e$*:*

$$
\begin{aligned}
\mathrm{Menu} &\sqsubseteq \exists \mathrm{GivesChoice}.\mathrm{MainDish} \sqcap \exists \mathrm{GivesChoice}.\mathrm{Dessert} \\
\mathrm{VegFriendly} &\sqsubseteq \exists \mathrm{Offers}.(\mathrm{Menu} \sqcap \forall \mathrm{GivesChoice}.\neg \mathrm{WithMeat}) \\
\mathrm{KidFriendly} &\sqsubseteq \exists \mathrm{Offers}.(\mathrm{Menu} \sqcap \forall \mathrm{GivesChoice}.\neg \mathrm{WithAlcohol}) \\
\mathrm{WithMeat} &\sqsubseteq \mathrm{MainDish} \\
\mathrm{MainDish} \sqcap \mathrm{Dessert} &\sqsubseteq \bot
\end{aligned}
$$

Our logician wonders how many combinations of dish z_1 *and dessert* z_2 *can be ordered in this restaurant* r *as long as each such combination is permitted within* *some menu y. This can be seen as evaluating the following CCQ q_e, also depicted in Figure 3.2, over the KB $\mathcal{K}_e := (\mathcal{T}_e, \mathcal{A}_e)$:*

$$q_e := \exists y\ \exists z_1\ \exists z_2\ \mathrm{Offers}(\mathsf{r}, y) \wedge \mathrm{Menu}(y) \wedge \mathrm{GivesChoice}(y, z_1) \wedge \mathrm{MainDish}(z_1) \\ \wedge \mathrm{GivesChoice}(y, z_2) \wedge \mathrm{Dessert}(z_2)$$

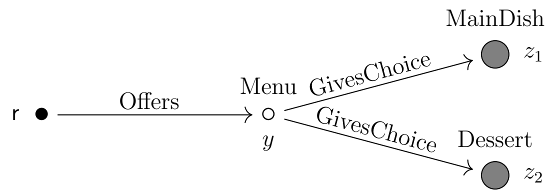


Figure 3.2: The query q_e from Example 5.

The query q_e is Boolean and y being an existential variable means that a value for the pair (z_1, z_2) obtained from two different menus should only be counted once.

The answers to a CCQ in a fixed model $\mathcal{I}$ are defined using *counting matches*, which are defined similarly to the classical notion of *matches* for a (plain) CQ, but are then restricted to the counting variables from $\mathbf{z}$.

Definition 16. *A* match *for a CCQ q in an interpretation $\mathcal{I}$ is a homomorphism from q into $\mathcal{I}$, i.e. a function π that maps each term in q to an element of $\Delta^{\mathcal{I}}$ such that $\pi(t) = t$ when $t \in \mathsf{N_I}$, $\pi(t) \in \mathrm{A}^{\mathcal{I}}$ for every $\mathrm{A}(t) \in q$, and $(\pi(t), \pi(t')) \in \mathrm{P}^{\mathcal{I}}$ for every $\mathrm{P}(t, t') \in q$. If a match π maps $\mathbf{x}$ to $\mathbf{a}$, then the restriction of π to $\mathbf{z}$ is called a* counting match (c-match) *of $q(\mathbf{a})$ in $\mathcal{I}$.*

The usual problem of CQ answering is to decide whether there exists a match in every model of the KB of interest. With counting conjunctive queries, we are interested in how many counting matches exist in such models. However, the exact number from a model to another might vary, especially since ontologies expressed with $\mathcal{ALCHI}$ cannot constrain the size of the models:

Proposition 1. *If a CCQ q is satisfied in a model of an $\mathcal{ALCHI}$ KB $\mathcal{K}$ and $\mathbf{z} \neq \emptyset$, then there exists a model of $\mathcal{K}$ with an infinite number of counting matches for q.*

Proof. Let $\mathcal{I}$ be a model of $\mathcal{K} := (\mathcal{T}, \mathcal{A})$ in which q is satisfied. Let ρ_k be the mapping renaming an element e into e_k. Let $\mathcal{I}_\infty$ be the interpretation with domain $\Delta^{\mathcal{I}_\infty} := \bigcup_{k=0}^{+\infty} \rho_k(\Delta^{\mathcal{I}})$, which interprets each individual a as a_0 (slightly abusing the SNA), and each concept name A and role name P as follows:

$$\mathrm{A}^{\mathcal{I}_\infty} := \bigcup_{k=0}^{+\infty} \rho_k(\mathrm{A}^{\mathcal{I}}) \qquad \mathrm{P}^{\mathcal{I}_\infty} := \bigcup_{i=0}^{+\infty} \bigcup_{j=0}^{+\infty} (\rho_i \times \rho_j)(\mathrm{P}^{\mathcal{I}})$$

Since $\mathcal{I}_\infty$ embeds in $\mathcal{I}$ by dropping all indexes and that $\mathcal{I}$ embeds in each layer $\mathcal{I}_k := \rho_k(\mathcal{I})$ of $\mathcal{I}_\infty$, it is easily verified that $\mathcal{I}_\infty$ is a model of the $\mathcal{ALCHI}$ KB $\mathcal{K}$ and that the counting match $\pi : \mathbf{z} \to \Delta^\mathcal{I}$ yields an infinite number of distinct counting matches $\pi_k := \rho_k \circ \pi$ in $\mathcal{I}_\infty$ (recall $\mathbf{z} \neq \emptyset$). □

Therefore, a notion of certain answer requiring that there exist *exactly* n counting matches for q in every model $\mathcal{K}$ will likely return false for every integer n. To address this issue, we follow Kostylev and Reutter [2015] and consider bounds on the exact number of counting matches. More precisely, answers to a CCQ in a model are all intervals bounding the exact number of counting matches.

Definition 17. *The set of* answers *to q in $\mathcal{I}$, denoted $q^\mathcal{I}$, contains all pairs $(\mathbf{a}, [m, M])$, with $m, M \in \mathbb{N} \cup \{+\infty\}$, such that the number of distinct counting matches of $q(\mathbf{a})$ in $\mathcal{I}$ belongs to the interval $[m, M]$.*

Importantly, these bounds are taking into consideration counting matches that are mapping counting variables $\mathbf{z}$ outside of individual elements of $\mathcal{I}$. Hence these elements may not be shared across models, as opposed to values of the answer variables $\mathbf{x}$. It does not cause any issue to define certain answers as we are only interested in (bounds on) the number of such counting matches from a model to another. Furthermore, let us emphasize those bounds hold on the number of counting matches, not on the number of matches, treating equally a counting match obtained from a single match and a counting match obtained from an eventually infinite number of matches. Notice the pair $(\mathbf{a}, [0, +\infty])$ is always an answer, for any suitable $\mathbf{a}$, over any interpretation as $[0, +\infty]$ is a trivial bound on the number of counting matches. The notion of certain answer is then defined as usual certain answers for CQs, that is as the intersection of answers across all models:

Definition 18. *A* certain answer *to q w.r.t. $\mathcal{K}$ is an answer in every model of $\mathcal{K}$, that is a pair from $\bigcap_{\mathcal{I} \models \mathcal{K}} q^\mathcal{I}$. In particular, if $\mathcal{K}$ is unsatisfiable, then all couples $(\mathsf{a}, [m, M])$, with $\mathsf{a} \in \mathsf{Ind}(\mathcal{A})$ and $m, M \in \mathbb{N} \cup \{+\infty\}$, are certain answers.*

Let us illustrate the notions of matches, counting matches, answers and certain answers with the following example, which is a continuation of Example 5.

Example 6 (Example 5 continued)**.** *Two models $\mathcal{I}_e^1$ and $\mathcal{I}_e^2$ of the KB $\mathcal{K}_e$ are depicted in Figures 3.3 and 3.4. Matches and counting matches of q_e in each model are presented in Table 3.2. The number of counting matches in $\mathcal{I}_e^1$ is 5, while it is 6 in $\mathcal{I}_e^2$.*

In the model $\mathcal{I}_e^2$, there equal numbers of matches and counting matches, but this doesn't hold in general as illustrated by model $\mathcal{I}_e^1$. In the latter, we indeed retain a single occurrence of the pair $(\mathsf{carb}, \mathsf{tira})$ *even though it can be obtained in two different ways by mapping the existential variable y to either* m_1 *or to* m_2*.*

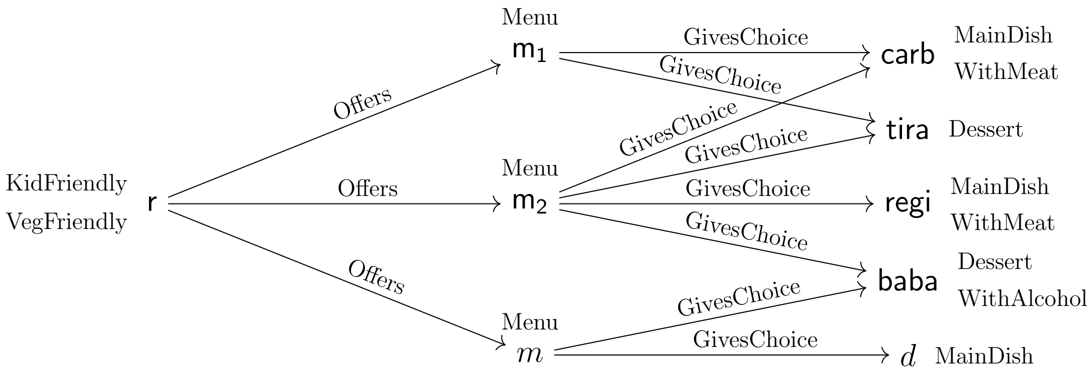


Figure 3.3: Model $\mathcal{I}_e^1$ from Example 6.

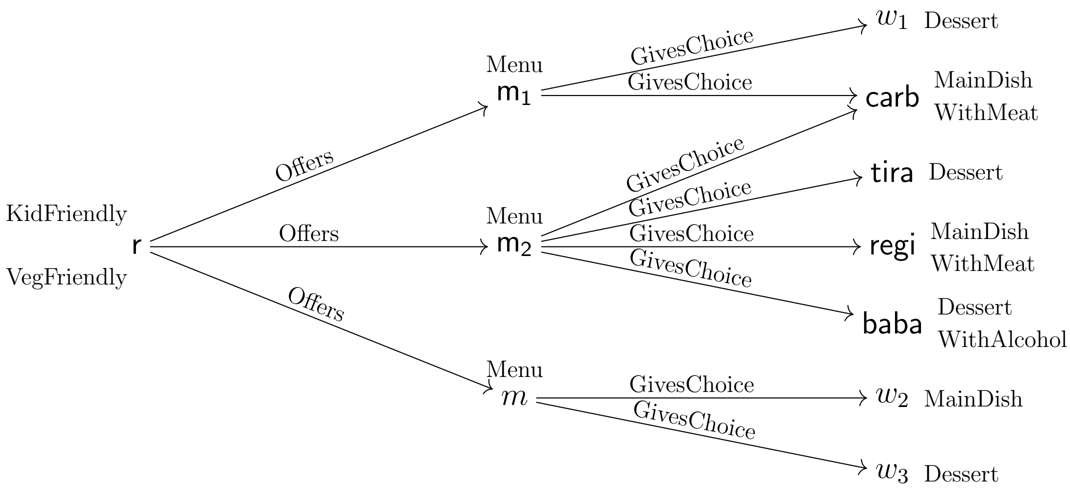


Figure 3.4: Model $\mathcal{I}_e^2$ from Example 6

The answers to q_e in $\mathcal{I}_e^1$ are precisely those intervals containing 5*, hence the pair* $(\emptyset, [6, +\infty])$ *is* not *an answer in $\mathcal{I}_e^1$, while it is an answer in $\mathcal{I}_e^2$. It follows that* $(\emptyset, [6, +\infty])$ *is not a certain answer. The pair* $(\emptyset, [5, 7])$ *is an answer in both models $\mathcal{I}_e^1$ and $\mathcal{I}_e^2$, but one can easily come up with another model containing, say,* 8 *matches, proving that* $(\emptyset, [5, 7])$ *is not a certain answer.*

It is not hard to see that $(\emptyset, [4, +\infty])$ *is a certain answer as the* 4 *common matches of $\mathcal{I}_e^1$ and $\mathcal{I}_e^2$ (those involving menu* m_2*) are actually entailed by the KB and hence yield* 4 *distinct counting matches in every model (recall that models are required to comply with the unique name assumption). Interestingly, this lower bound of* 4 *can be obtained by counting the certain answers of the usual CQ $q_e'(x_1, x_2)$ obtained by considering our CCQ q_e in which we replace the two counting variables z_1 and z_2 by answer variables x_1 and x_2.*

y	z_1	z_2
m_1	carb	tira
m_2	carb	tira
m_2	carb	baba
m_2	regi	tira
m_2	regi	baba
m	d	baba

(a) Matches in $\mathcal{I}_e^1$

z_1	z_2
carb	tira
carb	baba
regi	tira
regi	baba
d	baba

(b) Counting matches in $\mathcal{I}_e^1$

y	z_1	z_2
m_1	carb	w_1
m_2	carb	tira
m_2	carb	baba
m_2	regi	tira
m_2	regi	baba
m	w_2	w_3

(c) Matches in $\mathcal{I}_e^2$

z_1	z_2
carb	w_1
carb	tira
carb	baba
regi	tira
regi	baba
w_2	w_3

(d) Counting matches in $\mathcal{I}_e^2$

Table 3.2: Matches and counting matches of q_e in $\mathcal{I}_e^1$ and $\mathcal{I}_e^2$

Note that a tighter certain answer exists, as $(\emptyset, [5, +\infty])$ *is also a certain answer, and that* $[5, +\infty]$ *is included in* $[4, +\infty]$. *This is because the vegetarian menu that each model must contain always yields an extra counting match as the* 4 *entailed matches all involve non-vegetarian main dishes.*

To conclude this example, notice that $\mathcal{K}_e$ *admits universal models in all of which model* $\mathcal{I}_e^2$ *embeds. From the above discussion, it follows* $(\emptyset, [6, +\infty])$ *not being a certain answer for* q_e *cannot be determined by considering universal models of* $\mathcal{K}_e$. *This is in contrast with CQ answering, for which we know that the certain answers are precisely the answers in any universal model, whenever such a model exists.*

The connection mentioned in Example 6 between counting the certain answers of a usual CQ, a reasoning task notably explored in Feier et al. [2021], and the proposed notion of certain answers for CCQ actually holds in general: if m is the number of certain answers of the CQ $q(\mathbf{x}, \mathbf{z}) = \exists \mathbf{y}\ \phi(\mathbf{x}, \mathbf{y}, \mathbf{z})$ mapping $\mathbf{x}$ to $\mathbf{a}$, then $(\mathbf{a}, [m, +\infty])$ is a certain answer to the CCQ $q(\mathbf{x}) = \exists \mathbf{y}\ \exists \mathbf{z}\ \phi(\mathbf{x}, \mathbf{y}, \mathbf{z})$. The converse is not true in general.

Example 6 may have convinced the reader that the notation of answers and certain answers as pairs is cumbersome. It should hence be a relief that, as usual, it is sufficient to consider the Boolean case: $(\mathbf{a}, [m, M])$ is a certain answer to a CCQ $q(\mathbf{x})$ iff $(\emptyset, [m, M])$ is a certain answer to the Boolean CCQ $q(\mathbf{a})$ obtained by replacing $\mathbf{x}$ with $\mathbf{a}$. Thus, from now on, *we focus on Boolean CCQs*, and work with answers and certain answers $[m, M]$ in place of $(\emptyset, [m, M])$.

Furthermore, and as already mentioned in Remark 1, $\mathcal{ALCHI}$ cannot restrict the size of models, hence *the least* upper bound M in a certain answer $[m, M]$ is:

- 0 if the underlying CQ is unsatisfiable w.r.t. $\mathcal{T}$;
- 1 if q has a match in every model but $\mathbf{z} = \emptyset$;

- $+\infty$ otherwise.

As the first two cases can be readily handled using existing techniques, *we focus on identifying certain answers of the form* $[m, +\infty]$.

Remark 4. *The question of upper bounds* M*, that we have so quickly dismissed in the* $\mathcal{ALCHI}$ *setting, arises naturally in closely related contexts, for example when considering functionality axioms in the ontology or dealing with closed predicates. It is for this reason that we chose to present our semantics with intervals of the form* $[m, M]$ *rather than directly focusing on intervals* $[m, +\infty]$*.*

3.1.3 Decision problems

CCQ answering Given a $\mathcal{ALCHI}$ knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, a Boolean CCQ q, and an integer $m \geq 0$ (in binary), we are interested in the complexity of deciding whether $[m, +\infty]$ is a certain answer to q w.r.t. $\mathcal{K}$. We refer to this decision problem as *CCQ answering* and consider the two usual complexity measures: *combined complexity* which is in terms of the size of the whole input, and *data complexity* which is only in terms of the size of $\mathcal{A}$ and m ($\mathcal{T}$ and q are treated as fixed).

Recall that if O is a TBox, ABox, KB, or CCQ, then the *size* of O, denoted $|O|$, is the number of occurrences of concept and role names in O and that m is written in *binary*. This latter point will not appear crucial in the present chapter as reductions involved in the proofs of lower complexity bounds happen to only construct polynomially large such integers m w.r.t. the size of the instance of the reduced decision problem, if not constant (e.g. for data complexity). In the two following chapters, however, several reductions strongly require a binary encoding of these integers m.

When deciding whether a given $[m, +\infty]$ is a certain answer for some CCQ over some KB, we use the term of *countermodel* to refer to a model with less than m counting matches. Similarly, an *optimal model* is a model minimizing the number of counting matches.

Tightest variant The definition of certain answers implies that if $[m, +\infty]$ is a certain answer, then so is $[m', +\infty]$, for every $m' \leq m$. It is naturally of interest to focus on certain answers providing the greatest m, i.e., the *tightest certain answer* $[m_{opt}, +\infty]$, being the intersection of all certain answers. Given the same input as CCQ answering, we refer to the problem of deciding if $[m, +\infty]$ is the tightest certain answer as *tight CCQ answering*.

This optimization variant has already been formulated as an open question in Kostylev and Reutter [2015], in presence of coNP-complete situations for CCQ answering over DL-Lite$^{\mathcal{H}}_{\mathsf{core}}$ KBs w.r.t. data complexity. We close these questions in Subsection 3.5.3 and prove tight CCQ answering is DP-complete not only in the settings considered in Kostylev and Reutter [2015] but also for $\mathcal{EL}$ ontologies.

Remark 5. *Notice $[m, M]$ being the tightest certain answer doesn't imply that for all $n \in [m, M]$ there exists a model containing exactly n matches. Consider the Boolean query $q_{\mathrm{A}\times\mathrm{A}} := \exists z_1\ \exists z_2\ \mathrm{A}(z_1) \wedge \mathrm{A}(z_2)$ for any concept name* A*, evaluated over the empty KB. The number of counting matches of $q_{\mathrm{A}\times\mathrm{A}}$ in an interpretation $\mathcal{I}$ is $\left|\mathrm{A}^{\mathcal{I}}\right|^2$ and therefore only perfect squares can be reached despite $[0, +\infty]$ being the tightest certain answer. Interestingly, this could motivate a more general setting in which answers and certain answers allow more refined subsets of integers instead of intervals $[m, M]$.*

3.2 Interlacings

Looking to existing DL-Lite approaches [Kostylev and Reutter, 2015], we observe that the high-level idea to answer CCQs is to start from an arbitrary optimal model $\mathcal{I}$ and merge its elements so as to reduce its size, while at the same time *not introducing any new query matches.* This ensures that if a countermodel exists for the candidate integer m, then there exists one with size at most the size of the model obtained when merging the initial optimal model. This technique allowed Kostylev and Reutter [2015] to obtain, in combined complexity, a coN2EXP algorithm for answering the two subclasses of CCQs they considered over DL-Lite$^{\mathcal{H}}_{\mathsf{core}}$, refined into a coNEXP algorithm over DL-Lite$_{\mathsf{core}}$, and also yielding a coNP upper bound in data complexity.

But how can we decide which elements of the starting model $\mathcal{I}$ can be safely merged? We observe that they proceed in two steps. First, they define an intermediate model $\mathcal{I}'$ (called *interleaving*) that, informally, retains the useful parts of $\mathcal{I}$ (i.e., those involved in query matches or needed to satisfy the ABox) and replaces the rest with tree-shaped structures taken from the corresponding parts of the canonical model. With this more structured countermodel $\mathcal{I}'$, it is easier to identify, via a well-chosen equivalence relation, the elements that behave similarly and thus can be safely merged. In a second step, elements of $\mathcal{I}'$ from the same equivalence class are merged to obtain the desired bounded-size countermodel.

A naïve adaptation of the DL-Lite approach to $\mathcal{ALCHI}$ fails at the very first step as the existence of a canonical model is not guaranteed. Furthermore, even when a canonical models exists, due to conjunction in the LHS of concept inclusions, for example in $\mathcal{EL}$ TBoxes, the interleaving need not be a model as the next example illustrates. Generally speaking, the issue is that the canonical model may not contain elements witnessing conjunctions of concepts that occur in the initial countermodel, so it is not enough to copy over parts of the canonical model.

Example 7. *Consider the $\mathcal{EL}$ KB $\mathcal{K}_0$ whose ABox only contains the assertion* $\mathrm{A}(\mathsf{a})$ *and whose TBox contains the four following axioms:*

$$\mathrm{A} \sqsubseteq \exists \mathrm{R.B} \qquad \mathrm{A} \sqsubseteq \exists \mathrm{R.C} \qquad \mathrm{B} \sqsubseteq \mathrm{D} \qquad \mathrm{C} \sqsubseteq \mathrm{D} \qquad \mathrm{B} \sqcap \mathrm{C} \sqsubseteq \exists \mathrm{R.A}$$

A countermodel $\mathcal{I}_0$ for integer 2 and CCQ $q_0 := \exists z\ \mathrm{D}(z)$ over $\mathcal{K}_0$ is depicted on Figure 3.5a. It contains a single counting match: $z \mapsto \delta$. The canonical model $\mathcal{C}_{\mathcal{K}_0}$ of $\mathcal{K}_0$ is depicted on Figure 3.5b and embeds in the countermodel through the homomorphism $f_0 : \mathsf{a} \mapsto \mathsf{a}, \mathsf{a} \cdot \mathrm{R}.\{\mathrm{B}, \mathrm{D}\} \mapsto \delta, \mathsf{a} \cdot \mathrm{R}.\{\mathrm{C}, \mathrm{D}\} \mapsto \delta$. The interleaving as defined in Kostylev and Reutter [2015] considers the interpretation obtained from $\mathcal{C}_{\mathcal{K}_0}$ by merging together elements u, v from $\Delta^{\mathcal{C}_{\mathcal{K}_0}}$ iff $f_0(u) = f_0(v)$ and this element of $\mathcal{I}_0$ is reached by a counting match of $\mathcal{I}_0$. In our case, it holds that $f_0(\mathsf{a} \cdot \mathrm{R}.\{\mathrm{B}, \mathrm{D}\}) = f_0(\mathsf{a} \cdot \mathrm{R}.\{\mathrm{C}, \mathrm{D}\}) = \delta$ and that δ is reached by a counting match, hence the interleaving depicted on Figure 3.5c. This latter interpretation violates axiom $\mathrm{B} \sqcap \mathrm{C} \sqsubseteq \exists \mathrm{R.A}$, *hence fails to be a model of $\mathcal{K}_0$.*

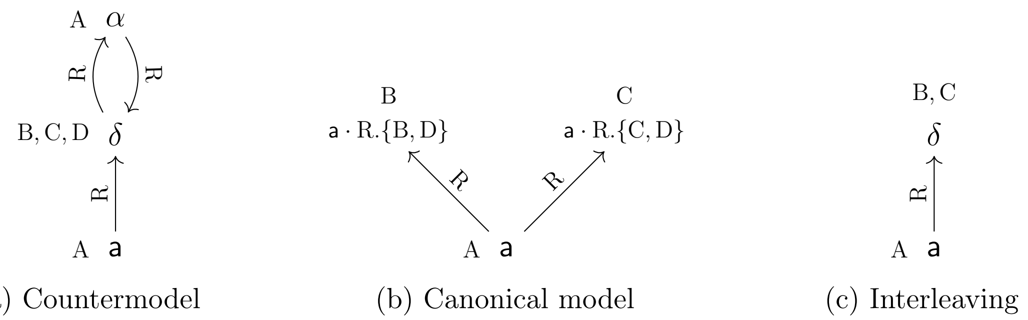


Figure 3.5: DL-Lite interleaving applied on the $\mathcal{EL}$ KB $\mathcal{K}_0$

In this section, we present a family of models of a KB $\mathcal{K}$ that is built from a starting model of interest $\mathcal{I}$. The construction proceeds in two steps. First, it unfolds $\mathcal{I}$ into a tree-shaped domain called the existential extraction which keeps track of the RHS existential concepts satisfied in $\mathcal{I}$. This existential extraction embeds in the initial model $\mathcal{I}$ through a mapping f. Second, it folds back parts of the existential extraction according to a parameter f' being a function allowed to merge together elements u and v of the existential extraction, i.e. $f'(u) = f'(v)$, only if $f(u) = f(v)$. This condition is sufficient to ensure that the resulting interpretation, called the f'-interlacing of $\mathcal{I}$, is a model of $\mathcal{K}$.

Depending on the chosen function f', the f'-interlacing may retain desirable features of the initial model $\mathcal{I}$. By choosing $f' := \mathsf{Id}$, we show that the resulting interlacing can be collapsed into a finite model with at most exponential size, which provides a countermodel for large values of the candidate integer m when evaluating a CCQ q over $\mathcal{K}$.

To handle the remaining values of m, we further explore another function $f' := f^*$ whose corresponding interlacing has at most as many counting matches as the initial model $\mathcal{I}$, but also partially inherits from the tree-shaped structure of the existential extraction. This latter f^*-interlacing motivates the 2EXP procedure presented in the next section and allows us to build finite models that minimize the number of matches.

Other interlacings, obtained *via* refined functions f', will further prove useful to answer rooted CCQs (see Chapter 4).

To illustrate the various constructions presented in this section and the next one, we rely on the following KB $\mathcal{K}_e$ and CCQ q_e as a running example.

Example 8. *Let* $\mathcal{K}_e$ *be the KB whose ABox* $\mathcal{A}$ *only contains the assertion* $\mathrm{A}(\mathsf{a})$ *and whose ontology* $\mathcal{T}$ *contains the following axioms:*

$$\begin{array}{cccc} \mathrm{A} \sqsubseteq \exists \mathrm{R}.\mathrm{A}' & \mathrm{B} \sqsubseteq \mathrm{B}' \sqcup \mathrm{D} & \mathrm{D} \sqsubseteq \exists \mathrm{S}.\mathrm{D} & \mathrm{A} \sqcap \mathrm{B} \sqsubseteq \bot \\ \mathrm{A}' \sqsubseteq \exists \mathrm{R}.\mathrm{A} & \mathrm{B}' \sqsubseteq \exists \mathrm{R}.\mathrm{C} & \mathrm{C} \sqsubseteq \exists \mathrm{S}.\mathrm{A} & \mathrm{R} \sqcap \mathrm{R}^- \sqsubseteq \bot \\ \mathrm{A}' \sqsubseteq \exists \mathrm{R}.\mathrm{B} & \mathrm{B}' \sqsubseteq \exists \mathrm{T}.\mathrm{D} & \mathrm{C} \sqsubseteq \exists \mathrm{S}.\mathrm{B} & \mathrm{D} \sqcap \exists \mathrm{R}^-.\mathrm{A}' \sqsubseteq \bot \end{array}$$

A model $\mathcal{I}_e$ *of* $\mathcal{K}_e$ *is depicted on Figure 3.6. Consider the Boolean CCQ* $q_e := \exists y_1 \, \exists y_2 \, \exists z \; \mathrm{R}(y_1, y_2) \wedge \mathrm{S}(y_2, z)$*, which admits* 2 *counting matches in* $\mathcal{I}_e$*, mapping respectively* z *to* a *or to* ε*.*

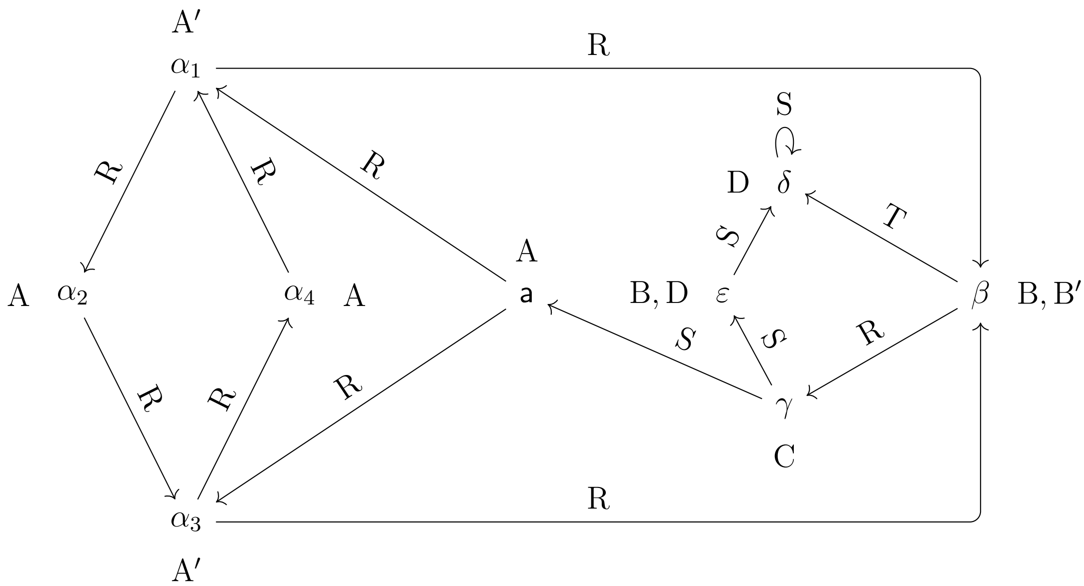


Figure 3.6: Model $\mathcal{I}_e$ of $\mathcal{K}_e$

3.2.1 Existential extraction

We fix a satisfiable $\mathcal{ALCHI}$ KB $\mathcal{K} := (\mathcal{T}, \mathcal{A})$ and a model $\mathcal{I}$ of $\mathcal{K}$. The definition of existential extraction uses the alphabet Ω consisting of all R.A such that $\exists$R.A is the RHS of an axiom in $\mathcal{T}$. Furthermore, it assumes that, for every $\mathrm{R.A} \in \Omega$, we have chosen a function $\mathsf{succ}^{\mathcal{I}}_{\mathrm{R.A}}$ that maps every element $e \in (\exists \mathrm{R.A})^{\mathcal{I}}$ to an element $e' \in \Delta^{\mathcal{I}}$ such that $(e, e') \in \mathrm{R}^{\mathcal{I}}$ and $e' \in \mathrm{A}^{\mathcal{I}}$.

Definition 19. *Over the set* $\mathsf{Ind}(\mathcal{A}) \cdot \Omega^*$*, inductively build the following mapping:*

$$\begin{array}{rcll} f : \mathsf{Ind}(\mathcal{A}) \cdot \Omega^* & \to & \Delta^{\mathcal{I}} \cup \{\uparrow\} & \\ \mathsf{a} & \mapsto & \mathsf{a} & \\ w \cdot \mathrm{R.A} & \mapsto & \begin{cases} \uparrow & \text{if } f(w) = \uparrow \text{ or } f(w) \notin (\exists \mathrm{R.A})^{\mathcal{I}} \\ \mathsf{succ}^{\mathcal{I}}_{\mathrm{R.A}}(f(w)) & \text{otherwise} \end{cases} \end{array}$$

where $\uparrow$ *is a fresh symbol witnessing the absence of a proper image for an element of* $\mathsf{Ind}(\mathcal{A}) \cdot \Omega^*$*. The* existential extraction[5] *of* $\mathcal{I}$ *is* $\Delta^\circ := \{w \mid w \in \mathsf{Ind}(\mathcal{A}) \cdot \Omega^*, f(w) \neq \uparrow\}$*. Slightly abusing the notation, the mapping* $f_{|\Delta^\circ} : \Delta^\circ \to \Delta^{\mathcal{I}}$ *is also denoted* f *for readability.*

Remark 6. Δ° *can be seen as the domain of a form of unravelling of* $\mathcal{I}$ *starting from* $\mathsf{Ind}(\mathcal{A})$*, in which we only follow the selected successors for the RHS existential concepts.*

Example 9. *For* $\mathcal{K}_e$*, we have* $\Omega := \{\mathrm{R.A}, \mathrm{R.A'}, \mathrm{R.B}, \mathrm{R.C}, \mathrm{S.A}, \mathrm{S.B}, \mathrm{S.D}, \mathrm{T.D}\}$*. We chose* $\mathsf{succ}^{\mathcal{I}_e}_{\mathrm{R.A'}}(\mathsf{a}) := \alpha_1$*, one could have also chosen* α_3*. All the other choices of successors in* $\mathcal{I}_e$ *are unique. The existential extraction* Δ°_e *of* $\mathcal{I}_e$ *is depicted on Figure 3.7 as a directed graph: an element* w *belongs to* Δ°_e *iff there exists a path* p *from the node* a *to a node* n_w *that produces* w *when concatenating* a *with the encountered labels along* p*. For example, element* $\mathsf{a} \cdot \mathrm{R.A'} \cdot \mathrm{R.B}$ *belongs to* Δ°_e *while* $\mathsf{a} \cdot \mathrm{R.A'} \cdot \mathrm{R.A'}$ *doesn't. Notice* Δ°_e *is infinite. The image of an element* w *by* f *is indicated as* $(f(w))$ *on the node* n_w*. Hence* $f(\mathsf{a} \cdot \mathrm{R.A'} \cdot \mathrm{R.B}) = \beta$ *and* $f(\mathsf{a} \cdot \mathrm{R.A'} \cdot \mathrm{R.B} \cdot \mathrm{R.C} \cdot \mathrm{S.A}) = \mathsf{a}$.

As illustrated by the above example, existential extractions contain many regularities. In particular, the branches issuing from two elements of an existential extraction that map on the same element in the starting model are similar. This is formalized by the following lemma.

Lemma 1. *Let* $u, v \in \mathsf{Ind}(\mathcal{A}) \cdot \Omega^*$ *such that* $f(u) = f(v)$*. For all* $w \in \Omega^*$*, we have* $f(u \cdot w) = f(v \cdot w)$*; hence in particular* $u \cdot w \in \Delta^\circ$ *iff* $v \cdot w \in \Delta^\circ$*.*

[5] While the definitions of f, Δ°, and later constructions depend on the choice of successor functions, all choices lead to the desired result.

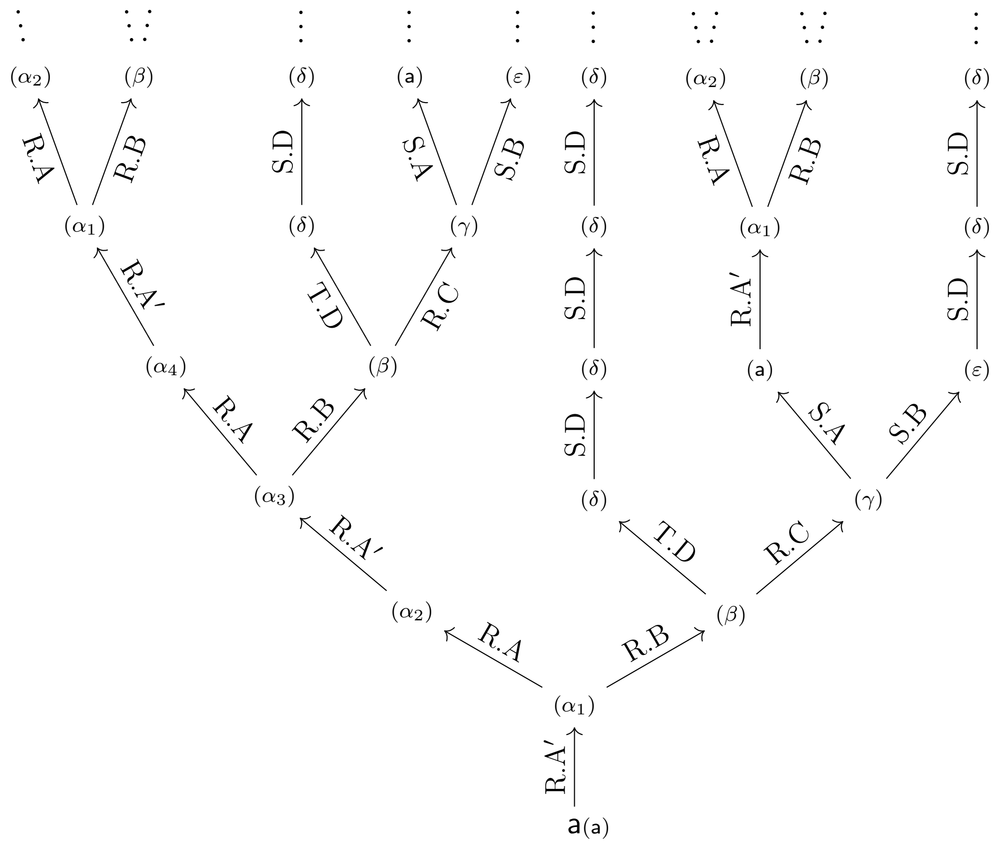


Figure 3.7: A representation of f and of the existential extraction of $\mathcal{I}_e$

Proof. We proceed by induction on $w \in \Omega^*$. For w being the empty word, it is trivial. Assume now it holds for w and consider $w \cdot \text{R.A}$. Observe that the definition of $f(u \cdot w \cdot \text{R.A})$ only depends on $f(u \cdot w)$, which is equal to $f(v \cdot w)$ by induction hypothesis. Therefore $f(u \cdot w \cdot \text{R.A})$ yields the same value as $f(v \cdot w \cdot \text{R.A})$. □

3.2.2 A family of models: interlacings

Consider a satisfiable $\mathcal{ALCHI}$ KB $\mathcal{K} := (\mathcal{T}, \mathcal{A})$, a model $\mathcal{I}$ of $\mathcal{K}$ and Δ° an existential extraction Δ° of $\mathcal{I}$ for some choice of successors in $\mathcal{I}$. We recall that $f : \Delta^\circ \to \Delta^\mathcal{I}$ denotes the mapping built along the domain Δ°. We proceed to define f'-interlacings, parametrized by a function of interest f'.

Definition 20. *The* f'-interlacing $\mathcal{I}'$ *of* $\mathcal{I}$ *is the interpretation whose domain is*

$\Delta^{\mathcal{I}'} := f'(\Delta^\circ)$ *and which interprets concept and role names as follows:*

$$\begin{aligned}
\mathrm{A}^{\mathcal{I}'} &:= \{f'(u) \mid u \in \Delta^\circ, f(u) \in \mathrm{A}^{\mathcal{I}}\} \\
\mathrm{P}^{\mathcal{I}'} &:= \{(\mathsf{a},\mathsf{b}) \mid \mathsf{a},\mathsf{b} \in \mathsf{Ind}(\mathcal{A}) \wedge \mathcal{K} \models \mathrm{P}(\mathsf{a},\mathsf{b})\} && (\triangledown_0) \\
&\cup \ \{(f'(u), f'(u \cdot \mathrm{R.B})) \mid u, u \cdot \mathrm{R.B} \in \Delta^\circ \wedge \mathcal{T} \models \mathrm{R} \sqsubseteq \mathrm{P}\} && (\triangledown_+) \\
&\cup \ \{(f'(u \cdot \mathrm{R.B}), f'(u)) \mid u, u \cdot \mathrm{R.B} \in \Delta^\circ \wedge \mathcal{T} \models \mathrm{R}^- \sqsubseteq \mathrm{P}\} && (\triangledown_-)
\end{aligned}$$

Intuitively, the Id-interlacing is the interpretation with domain Δ° with concepts imported from $\mathcal{I}$ and roles interpreted in a tree-shaped manner (apart from the ABox part) issuing from the occurring successors in $\mathcal{I}$. The f'-interlacing is then the image of the Id-interlacing by f'. In particular, by setting $f' := f$, we obtain the f-interlacing being a sub-interpretation of the original model $\mathcal{I}$.

Notice the interpretation of roles is mainly defined from the existential extraction, which is similar in spirit to the interpretation of roles in the canonical model when it exists, and relates to the original model as follows:

Lemma 2. *For all $u, v \in \Delta^\circ$ and all role $\mathrm{R} \in \mathsf{N}_\mathsf{R}^\pm$, if $(f'(u), f'(v)) \in \mathrm{R}^{\mathcal{I}'}$, then $(f(u), f(v)) \in \mathrm{R}^{\mathcal{I}}$.*

Proof. Let $u, v \in \Delta^\circ$ and $\mathrm{R} \in \mathsf{N}_\mathsf{R}^\pm$ such that $(f'(u), f'(v)) \in \mathrm{R}^{\mathcal{I}'}$. We distinguish the three cases from definition of $\mathrm{R}^{\mathcal{I}'}$:

$\triangledown_0$. We have $u, v \in \mathsf{Ind}(\mathcal{A})$ and $\mathcal{K} \models \mathrm{P}(u, v)$. In particular $f(u) = u$ and $f(v) = v$, and since $\mathcal{I}$ is a model, it immediately gives $(f(u), f(v)) \in \mathrm{R}^{\mathcal{I}}$.

$\triangledown_+$. There exists $\mathrm{P.B} \in \Omega$ such that $\mathcal{T} \models \mathrm{P} \sqsubseteq \mathrm{R}$ and $v = u \cdot \mathrm{P.B}$. Since $u{\cdot}\mathrm{P.B} \in \Delta^\circ$, we have that $\mathsf{succ}^{\mathcal{I}}_{\mathrm{P.B}}(f(u))$ is defined and the definition of f yields $f(v) = \mathsf{succ}^{\mathcal{I}}_{\mathrm{P.B}}(f(u))$. By definition of $\mathsf{succ}^{\mathcal{I}}_{\mathrm{P.B}}(f(u))$, we have in particular $(f(u), f(v)) \in \mathrm{P}^{\mathcal{I}}$. Since $\mathcal{I}$ is a model, it now ensures $(f(u), f(v)) \in \mathrm{R}^{\mathcal{I}}$.

$\triangledown_-$. There exists $\mathrm{P.B} \in \Omega$ such that $\mathcal{T} \models \mathrm{P}^- \sqsubseteq \mathrm{R}$ and $u = v \cdot \mathrm{P.B}$. Since $v{\cdot}\mathrm{P.B} \in \Delta^\circ$, we have that $\mathsf{succ}^{\mathcal{I}}_{\mathrm{P.B}}(f(v))$ is defined and the definition of f yields $f(u) = \mathsf{succ}^{\mathcal{I}}_{\mathrm{P.B}}(f(v))$. By definition of $\mathsf{succ}^{\mathcal{I}}_{\mathrm{P.B}}(f(v))$, we have in particular $(f(v), f(u)) \in \mathrm{P}^{\mathcal{I}}$. Since $\mathcal{I}$ is a model, it now ensures $(f(u), f(v)) \in \mathrm{R}^{\mathcal{I}}$. □

In general, the f'-interlacing may not be a model of $\mathcal{K}$. For example if the function f' maps two elements u and v on a same element $e := f'(u) = f'(v)$, and that $u \in \mathrm{A}^{\mathcal{I}}$ and $v \in \mathrm{B}^{\mathcal{I}}$ for some concepts A and B such that $\mathcal{T} \models \mathrm{A} \sqcap \mathrm{B} \sqsubseteq \bot$, then the element e satisfies both A and B in $f'(\mathcal{I})$, proving the latter is not a model of $\mathcal{K}$. We hence explore a sufficient condition ensuring modelhood, namely pseudo-injectivity of f', which intuitively requires the function f' not to merge together elements that are not already merged by the function f.

Definition 21. *A function* $f' : \Delta^\circ \to E$ *is* pseudo-injective *if: for all* $u, v \in \Delta^\circ$*, if* $f'(u) = f'(v)$*, then* $f(u) = f(v)$*.*

Under this condition, we obtain modelhood but also prove that such a f'-interlacing embeds in $\mathcal{I}$.

Theorem 4. *If* $f' : \Delta^\circ \to E$ *is pseudo-injective, then* $\mathcal{I}'$ *is a model of* $\mathcal{K}$ *and the following mapping is a homomorphism from* $\mathcal{I}'$ *to* $\mathcal{I}$*:*

$$\sigma : \begin{array}{ccc} \Delta^{\mathcal{I}'} & \to & \Delta^{\mathcal{I}} \\ f'(u) & \mapsto & f(u) \end{array}$$

Notice that f' *being pseudo-injective ensures* σ *is indeed well-defined.*

Proof. We start by showing that $\mathcal{I}'$ is a model, by considering each possible shape of assertions and axioms (recall that $\mathcal{T}$ is in normal form):

- $\mathrm{A}(\mathsf{a})$. Since $\mathcal{I}$ is a model, we have $f(\mathsf{a}) = \mathsf{a} \in \mathrm{A}^{\mathcal{I}}$. Therefore, the definition of $\mathrm{A}^{\mathcal{I}'}$ gives $f'(\mathsf{a}) = \mathsf{a} \in \mathrm{A}^{\mathcal{I}'}$.

- $\mathrm{P}(\mathsf{a}, \mathsf{b})$. Setting $\mathrm{P}_0 := \mathrm{P}$ in Case ∇_0 of the definition of $\mathrm{P}^{\mathcal{I}'}$ yields $(f'(\mathsf{a}), f'(\mathsf{b})) = (\mathsf{a}, \mathsf{b}) \in \mathrm{P}^{\mathcal{I}'}$.

- $\top \sqsubseteq \mathrm{A}$. Let $u \in \top^{\mathcal{I}'} = \Delta^{\mathcal{I}'}$. By definition of $\Delta^{\mathcal{I}'}$, there exists $u_0 \in \Delta^\circ$ such that $f'(u_0) = u$. Since $f(u_0) \in \top^{\mathcal{I}}$ and $\mathcal{I}$ is a model, it ensures $f(u_0) \in \mathrm{A}^{\mathcal{I}}$. Therefore $u = f'(u_0) \in \mathrm{A}^{\mathcal{I}'}$.

- $\mathrm{A}_1 \sqcap \mathrm{A}_2 \sqsubseteq \mathrm{A}$. Let $u \in (\mathrm{A}_1 \sqcap \mathrm{A}_2)^{\mathcal{I}'}$. By definition of $\mathrm{A}_1^{\mathcal{I}'}$ and $\mathrm{A}_2^{\mathcal{I}'}$, there exists $u_1, u_2 \in \Delta^\circ$ with $f'(u_1) = f'(u_2) = u$ and such that $f(u_1) \in \mathrm{A}_1^{\mathcal{I}}$ and $f(u_2) \in \mathrm{A}_2^{\mathcal{I}}$. Since f' is pseudo-injective, it yields $f(u_1) = f(u_2)$, hence $f(u_1) \in (\mathrm{A}_1 \sqcap \mathrm{A}_2)^{\mathcal{I}}$. Since $\mathcal{I}$ is a model, it ensures $f(u_1) \in \mathrm{A}^{\mathcal{I}}$. Recalling $f'(u_1) = u$, we obtain $u \in \mathrm{A}^{\mathcal{I}'}$.

- $\mathrm{A} \sqsubseteq \exists \mathrm{R}.\mathrm{B}$. Let $u \in \mathrm{A}^{\mathcal{I}'}$. By definition there exists $v \in \Delta^\circ$ with $f'(v) = u$ and such that $f(v) \in \mathrm{A}^{\mathcal{I}}$. Since $\mathcal{I}$ is a model, it ensures $\mathsf{succ}^{\mathcal{I}}_{\mathrm{R}.\mathrm{B}}(f(v))$ is defined. Therefore $v \cdot \mathrm{R}.\mathrm{B} \in \Delta^\circ$ and element $w := f'(v \cdot \mathrm{R}.\mathrm{B})$ satisfies:

  - $(u, w) \in \mathrm{R}^{\mathcal{I}'}$ since $(u, w) = (f'(v), f'(v \cdot \mathrm{R}.\mathrm{B}))$;
  - $w \in \mathrm{B}^{\mathcal{I}'}$ since $f(v \cdot \mathrm{R}.\mathrm{B}) = \mathsf{succ}^{\mathcal{I}}_{\mathrm{R}.\mathrm{B}}(f(v)) \in \mathrm{B}^{\mathcal{I}}$ and $f'(v \cdot \mathrm{R}.\mathrm{B}) = w$.

  Hence $u \in (\exists \mathrm{R}.\mathrm{B})^{\mathcal{I}'}$.

$\exists$R.B $\sqsubseteq$ A. Let $u \in (\exists \mathrm{R.B})^{\mathcal{I}'}$, that is there exists $v \in \mathrm{B}^{\mathcal{I}'}$ with $(u, v) \in \mathrm{R}^{\mathcal{I}'}$. By definition of $\Delta^{\mathcal{I}'}$, there exists $u_0 \in \Delta^\circ$ such that $f'(u_0) = u$, and by definition of $\mathrm{B}^{\mathcal{I}'}$ there also exists $v_0 \in \Delta^\circ$ such that $f'(v_0) = v$ and $f(v_0) \in \mathrm{B}^{\mathcal{I}}$. Notice $(f'(u_0), f'(v_0)) \in \mathrm{R}^{\mathcal{I}'}$ hence Lemma 2 gives $(f(u_0), f(v_0)) \in \mathrm{R}^{\mathcal{I}}$. Therefore $f(u_1) \in (\exists \mathrm{R.B})^{\mathcal{I}}$. Since $\mathcal{I}$ is a model, it ensures $f(u_1) \in \mathrm{A}^{\mathcal{I}}$, yielding $u = f'(u_1) \in \mathrm{A}^{\mathcal{I}'}$.

A $\sqsubseteq$ $\neg$B. By contradiction, assume $u \in \mathrm{A}^{\mathcal{I}'} \cap \mathrm{B}^{\mathcal{I}'}$. By definition there exists $v, w \in \Delta^\circ$ with $f'(v) = f'(w) = u$ and such that $f(v) \in \mathrm{A}^{\mathcal{I}}$ and $f(w) \in \mathrm{B}^{\mathcal{I}}$. Since f' is pseudo-injective, we obtain $f(v) = f(w)$. Hence $f(v) \in \mathrm{A}^{\mathcal{I}} \cap \mathrm{B}^{\mathcal{I}}$, contradicting $\mathcal{I}$ being a model.

$\neg$B $\sqsubseteq$ A. Let $u \in \neg \mathrm{B}^{\mathcal{I}'}$. By definition of $\Delta^{\mathcal{I}'}$, there exists $v \in \Delta^\circ$ such that $f'(v) = u$. Since $u \notin \mathrm{B}^{\mathcal{I}'}$, we have $f(v) \notin \mathrm{B}^{\mathcal{I}}$. Hence $\mathcal{I}$ being a model gives $f(v) \in \mathrm{A}^{\mathcal{I}}$, yielding by definition $u = f'(v) \in \mathrm{A}^{\mathcal{I}'}$.

P $\sqsubseteq$ R. Let $(u, v) \in \mathrm{P}^{\mathcal{I}'}$. By definition of $\Delta^{\mathcal{I}'}$, there exists $u_0, v_0 \in \Delta^\circ$ such that $f'(u_0) = u$ and $f'(v_0) = v$. In case $\triangledown_0$ from the definition of $\mathrm{P}^{\mathcal{I}'}$ we have $\mathcal{K} \models \mathrm{P}(u, v)$, hence $\mathcal{K} \models \mathrm{R}(u, v)$ and $(u, v) \in \mathrm{R}^{\mathcal{I}'}$ by definition of $\mathrm{R}^{\mathcal{I}'}$. Otherwise both cases $\triangledown_+$ and $\triangledown_-$ provides a subrole $\mathrm{P}_0 \in \mathsf{N}_{\mathsf{R}}^{\pm}$ of P, hence also of R, which triggers the corresponding case for $\mathrm{R}^{\mathcal{I}'}$ and ensures $(u, v) \in \mathrm{R}^{\mathcal{I}'}$.

$\mathrm{R}_1 \sqcap \mathrm{R}_2 \sqsubseteq$ A. By contradiction, assume one can find $(u, v) \in (\mathrm{R}_1 \sqcap \mathrm{R}_2)^{\mathcal{I}'}$. By definition of $\Delta^{\mathcal{I}'}$, there exists $u_0, v_0 \in \Delta^\circ$ such that $f'(u_0) = u$ and $f'(v_0) = v$. Notice $(f'(u_0), f'(v_0)) \in \mathrm{R}_1^{\mathcal{I}'}$ and $(f'(u_0), f'(v_0)) \in \mathrm{R}_2^{\mathcal{I}'}$, hence Lemma 2 gives $(f(u_0), f(v_0)) \in \mathrm{R}_1^{\mathcal{I}}$ and $(f(u_0), f(v_0)) \in \mathrm{R}_2^{\mathcal{I}}$, that is $(f(u_0), f(v_0)) \in (\mathrm{R}_1 \sqcap \mathrm{R}_2)^{\mathcal{I}}$. However, $\mathcal{I}$ being a model ensures $(\mathrm{R}_1 \sqcap \mathrm{R}_2)^{\mathcal{I}} = \emptyset$, hence a contradiction.

We now prove that σ is a homomorphism:

- Let $u \in \mathrm{A}^{\mathcal{I}'}$. By definition of $\mathrm{A}^{\mathcal{I}'}$, we have $u_0 \in \mathrm{A}^{\Delta^\circ}$ such that $f'(u_0) = u$ and $f(u_0) \in \mathrm{A}^{\mathcal{I}}$. Remark 7 provides $\sigma(f'(u_0)) = f(u_0)$, hence $\sigma(u) \in \mathrm{A}^{\mathcal{I}}$.
- Let $(u, v) \in \mathrm{R}^{\mathcal{I}'}$. By definition of $\Delta^{\mathcal{I}'}$, we have $u_0, v_0 \in \Delta^\circ$ such that $f'(u_0) = u$ and $f'(v_0) = v$. Notice $(f'(u_0), f'(v_0)) \in \mathrm{R}^{\mathcal{I}'}$, hence Lemma 2 gives $(f(u_0), f(v_0)) \in \mathrm{R}^{\mathcal{I}}$. Using Remark 7, we obtain $(\sigma(u), \sigma(v)) \in \mathrm{R}^{\mathcal{I}}$. □

Remark 7. *It is immediate that $\sigma \circ f' = f$.*

Interestingly, using a very simple pseudo-injective function, this construction allows to equip the existential extraction with an interpretation being a model.

Remark 8. $\mathsf{Id} : \Delta^\circ \to \Delta^\circ$ *is pseudo-injective, hence the* Id*-interlacing is a model.*

3.2.3 Finite models

We now exhibit a finite model $\mathcal{I}_\mathcal{K}$ for each satisfiable $\mathcal{ALCHI}$ KB $\mathcal{K}$. This will allow us to answer a CCQ q over $\mathcal{K}$ with candidate integer greater than $|\mathcal{I}_\mathcal{K}|^{|q|}$ simply by checking satisfiability. The existence of such finite model is obtained by merging elements from the Id-interlacing of a model of $\mathcal{K}$, primarily according to the atomic concepts they satisfy but also with some additional conditions due to role disjointness axioms.

Theorem 5. *If $\mathcal{K} := (\mathcal{T}, \mathcal{A})$ is a satisfiable $\mathcal{ALCHI}$ KB, then it admits a model with size at most $|\mathsf{Ind}(\mathcal{A})| + 3\,|\mathcal{T}|\,2^{|\mathcal{T}|}$.*

Proof. Assume $\mathcal{K}$ admits a model $\mathcal{I}$. Consider its Id-interlacing $\mathcal{I}'$. For each element of $\Delta^{\mathcal{I}'}$, we define its size: the size $|\mathsf{a}|$ of an individual a is 1, the size $|w \cdot \mathrm{R.B}|$ of a non-individual element $w \cdot \mathrm{R.B}$ is $|w| + 1$. We now equip $\Delta^{\mathcal{I}'}$ with the following equivalence relation $\sim$: each individual is only equivalent to itself, while two non-individual elements $w_1 \cdot \mathrm{R}_1.\mathrm{B}_1$ and $w_2 \cdot \mathrm{R}_2.\mathrm{B}_2$ are equivalent iff they satisfy the same concept names, that $\mathrm{R}_1.\mathrm{B}_1 = \mathrm{R}_2.\mathrm{B}_2$ and $|w_1| = |w_2| \mod 3$. Let $\widetilde{u}$ denote the equivalence class of the element u w.r.t $\sim$ and $\nu : d \mapsto \widetilde{d}$ the canonical projection.

We claim that the interpretation $\mathcal{M} := \mathcal{I}' / \sim$ with domain $\Delta^{\mathcal{I}'} / \sim$ and interpretation function $\cdot^{\mathcal{M}} := \nu \circ \cdot^{\mathcal{I}'}$ is a model. Notice it has the desired number of elements as each equivalence class is either a single individual, or fully characterized by an integer modulo 3, a role from $\mathsf{sig}(\mathcal{T})$ and a set of concepts from $\mathsf{sig}(\mathcal{T})$.

We consider in turn each of the possible forms of assertions and axioms:

$\mathrm{A}(\mathsf{a})$. Since $\mathcal{I}'$ is a model, we have $\mathsf{a} \in \mathrm{A}^{\mathcal{I}'}$. Therefore, the definition of $\mathrm{A}^{\mathcal{M}}$ gives $\widetilde{\mathsf{a}} = \mathsf{a} \in \mathrm{A}^{\mathcal{M}}$.

$\mathrm{P}(\mathsf{a}, \mathsf{b})$. Since $\mathcal{I}'$ is a model, we have $(\mathsf{a}, \mathsf{b}) \in \mathrm{P}^{\mathcal{I}'}$. Therefore, the definition of $\mathrm{P}^{\mathcal{M}}$ gives $(\widetilde{\mathsf{a}}, \widetilde{\mathsf{b}}) = (\mathsf{a}, \mathsf{b}) \in \mathrm{P}^{\mathcal{M}}$.

$\top \sqsubseteq \mathrm{A}$. Let $u \in \top^{\mathcal{M}} = \Delta^{\mathcal{M}}$. By definition of $\Delta^{\mathcal{M}}$, there exists $u_0 \in \Delta^{\mathcal{I}'}$ such that $\widetilde{u_0} = u$. Since $u_0 \in \top^{\mathcal{I}'}$ and $\mathcal{I}'$ is a model, it ensures $u_0 \in \mathrm{A}^{\mathcal{I}'}$. Therefore the definition of $\mathrm{A}^{\mathcal{M}}$ gives $u = \widetilde{u_0} \in \mathrm{A}^{\mathcal{M}}$.

$\mathrm{A}_1 \sqcap \mathrm{A}_2 \sqsubseteq \mathrm{A}$. Let $u \in (\mathrm{A}_1 \sqcap \mathrm{A}_2)^{\mathcal{M}}$. By definition of $\mathrm{A}_1^{\mathcal{M}}$ and $\mathrm{A}_2^{\mathcal{M}}$, there exists $u_1 \in \mathrm{A}_1^{\mathcal{I}'}$ and $u_2 \in \mathrm{A}_2^{\mathcal{I}'}$ with $\widetilde{u_1} = \widetilde{u_2} = u$. Since $\widetilde{u_1} = \widetilde{u_2}$, elements u_1 and u_2 satisfy the same concepts. In particular $u_1 \in (\mathrm{A}_1 \sqcap \mathrm{A}_2)^{\mathcal{I}'}$. Since $\mathcal{I}'$ is a model, it ensures $u_1 \in \mathrm{A}^{\mathcal{I}'}$, yielding by definition of $\mathrm{A}^{\mathcal{M}}$ that $u = \widetilde{u_1} \in \mathrm{A}^{\mathcal{M}}$.

$A \sqsubseteq \exists R.B$. Let $u \in A^{\mathcal{M}}$. By definition of $A^{\mathcal{M}}$ there exists $u_0 \in A^{\mathcal{I}'}$ with $\widetilde{u_0} = u$. Since $\mathcal{I}'$ is a model, it ensures there exists $v_0 \in B^{\mathcal{I}'}$ with $(u_0, v_0) \in R^{\mathcal{I}'}$. By definition of $B^{\mathcal{M}}$ and $R^{\mathcal{M}}$, the element $v := \widetilde{v_0}$ satisfies both $v \in B^{\mathcal{M}}$ and $(u, v) \in R^{\mathcal{M}}$, that is $u \in (\exists R.B)^{\mathcal{M}}$.

$\exists R.B \sqsubseteq A$. Let $u \in (\exists R.B)^{\mathcal{M}}$, that is there exists $v \in B^{\mathcal{M}}$ with $(u, v) \in R^{\mathcal{M}}$. By definition of $B^{\mathcal{M}}$ and $R^{\mathcal{M}}$, there exist $(u_0, v_0) \in R^{\mathcal{I}'}$ and $v_1 \in B^{\mathcal{I}'}$ such that $\widetilde{u_0} = u$ and $\widetilde{v_0} = \widetilde{v_1} = v$. Again, since $\widetilde{v_0} = \widetilde{v_1}$ both v_0 and v_1 satisfy the same concepts, that is in particular $u_0 \in (\exists R.B)^{\mathcal{I}'}$. Since $\mathcal{I}'$ is a model, it ensures $u_0 \in A^{\mathcal{I}'}$, yielding by definition of $A^{\mathcal{M}}$ that $u = \widetilde{u_0} \in A^{\mathcal{M}}$.

$A \sqsubseteq \neg B$. By contradiction, assume $u \in A^{\mathcal{M}} \cap B^{\mathcal{M}}$. By definition there exists $v \in A^{\mathcal{I}'}$ and $w \in B^{\mathcal{I}'}$ with $\widetilde{v} = \widetilde{w} = u$. Since $\widetilde{v} = \widetilde{w}$, both v and w satisfy the same concepts, contradicting $\mathcal{I}'$ being a model.

$\neg B \sqsubseteq A$. Let $u \in \neg B^{\mathcal{M}}$. By definition of $\Delta^{\mathcal{M}}$, there exists $v \in \mathcal{I}'$ such that $\widetilde{v} = u$. Since $u \notin B^{\mathcal{M}}$, we have $v \notin B^{\mathcal{I}'}$. Hence $\mathcal{I}'$ being a model gives $v \in A^{\mathcal{I}'}$, yielding by definition $u = \widetilde{v} \in A^{\mathcal{M}}$.

$P \sqsubseteq R$. Let $(u, v) \in P^{\mathcal{M}}$. By definition of $P^{\mathcal{M}}$, there exists $(u_0, v_0) \in P^{\mathcal{I}'}$ such that $\widetilde{u_0} = u$ and $\widetilde{v_0} = v$. Since $\mathcal{I}'$ is a model, it ensures $(u_0, v_0) \in R^{\mathcal{I}'}$, hence $(\widetilde{u_0}, \widetilde{v_0}) = (u, v) \in R^{\mathcal{M}}$ by definition of $R^{\mathcal{M}}$.

$R_1 \sqcap R_2 \sqsubseteq \bot$. By contradiction, assume one can find $(u, v) \in (R_1 \sqcap R_2)^{\mathcal{M}}$. By definition of $R_1^{\mathcal{M}}$ and $R_2^{\mathcal{M}}$, there exists $(u_1, v_1) \in R_1^{\mathcal{I}'}$ and $(u_2, v_2) \in R_2^{\mathcal{I}'}$ such that $\widetilde{u_1} = \widetilde{u_2} = u$ and $\widetilde{v_1} = \widetilde{v_2} = v$.
If either $\mathcal{K} \models R_1(u_1, v_1)$ or $\mathcal{K} \models R_2(u_2, v_2)$, then, each individual being alone in its equivalence class, we have $u_1 = u_2$ and $v_1 = v_2$. In particular it gives $(u_1, v_1) \in (R_1 \sqcap R_2)^{\mathcal{I}'}$, contradicting $\mathcal{I}'$ being a model.
Otherwise we distinguish the four possible cases:

- $v_1 = u_1 \cdot P_1.B_1$ and $\mathcal{T} \models P_1 \sqsubseteq R_1$.
  - $v_2 = u_2 \cdot P_2.B_2$ and $\mathcal{T} \models P_2 \sqsubseteq R_2$. Since $\widetilde{v_1} = \widetilde{v_2}$ we have $P_1.B_1 = P_2.B_2$. In particular $(u_1, v_1) \in {R_2}^{\mathcal{I}'}$, which contradicts $\mathcal{I}'$ being a model.
  - $u_2 = v_2 \cdot P_2.B_2$ and $\mathcal{T} \models P_2 \sqsubseteq R_2^-$. In particular $|v_1| = |u_1| + 1 \mod 3$ and $|u_2| = |v_2| + 1 \mod 3$. Recall that $\widetilde{u_1} = \widetilde{u_2}$ and $\widetilde{v_1} = \widetilde{v_2}$, hence $|u_1| = |u_2| \mod 3$ and $|v_1| = |v_2| \mod 3$. It yields $0 = 2 \mod 3$, contradiction.
- $u_1 = v_1 \cdot P_1.B_1$ and $\mathcal{T} \models P_1 \sqsubseteq R_1^-$.

- $v_2 = u_2 \cdot \mathrm{P}_2.\mathrm{B}_2$ and $\mathcal{T} \models \mathrm{P}_2 \sqsubseteq \mathrm{R}_2$. Symmetric to the previous case, leading to a contradiction.
- $u_2 = v_2 \cdot \mathrm{P}_2.\mathrm{B}_2$ and $\mathcal{T} \models \mathrm{P}_2 \sqsubseteq \mathrm{R}_2^-$. Since $\widetilde{u_1} = \widetilde{u_2}$ we have $\mathrm{P}_1.\mathrm{B}_1 = \mathrm{P}_2.\mathrm{B}_2$. In particular $(u_1, v_1) \in \mathrm{R}_2{}^{\mathcal{I}'}$, which contradicts $\mathcal{I}'$ being a model. □

Lemma 3. *For every satisfiable $\mathcal{ALCHI}$ KB $\mathcal{K} := (\mathcal{T}, \mathcal{A})$ and CCQ q, there exists a model with at most $(|\mathsf{Ind}(\mathcal{A})| + 3\,|\mathcal{T}|\,2^{|\mathcal{T}|})^{|q|}$ counting matches.*

Proof. The model exhibited in Theorem 5 is such a model. □

3.2.4 Countermodels via interlacings

We now consider a CCQ q and investigate a more specific function f^*, whose f^*-interlacing has at most as many counting-matches as the original model $\mathcal{I}$. This latter property along with the locally tree-shaped structures inherited from the existential extraction will play key roles in the 2EXP procedure developed further in this chapter and will be the starting point for the constructions of models with polynomial size w.r.t. data complexity. To define this new interlacing function f^*, we first need to capture which elements in the starting model $\mathcal{I}$ are involved in counting matches.

Definition 22. *Given an interpretation $\mathcal{I}$, we let Δ^* be the set of individuals from $\mathcal{A}$ plus all the elements from $\Delta^{\mathcal{I}}$ reached by counting matches. More formally:*

$$\Delta^* := \mathsf{Ind}(\mathcal{A}) \cup \bigcup_{\substack{\pi : q \to \mathcal{I} \\ \text{match}}} \pi(\mathbf{z})$$

We can now define f^* as a function which mimics f when it reaches Δ^*, and preserves Δ° otherwise. This definition is a direct adaptation of the function used in Kostylev and Reutter [2015], but using existential extraction and interlacings in place of canonical models and interleavings.

Definition 23. *The f^* mapping of $\mathcal{I}$ is:*

$$\begin{array}{rccl} f^*: & \Delta^\circ & \to & \Delta^* \uplus (\Delta^\circ \setminus \Delta^*) \\ & w & \mapsto & \begin{cases} f(w) & \text{if } f(w) \in \Delta^* \\ w & \text{otherwise} \end{cases} \end{array}$$

Example 10. *Figure 3.8 depicts the f^*-interlacing $\mathcal{I}_e{}'$ of $\mathcal{I}_e$. It has an infinite domain inherited from the existential extraction, but only 2 counting matches for q_e like the initial model $\mathcal{I}_e$. On the other hand, the number of matches for q_e in $\mathcal{I}_e{}'$ is infinite while it was 2 in $\mathcal{I}_e$.*

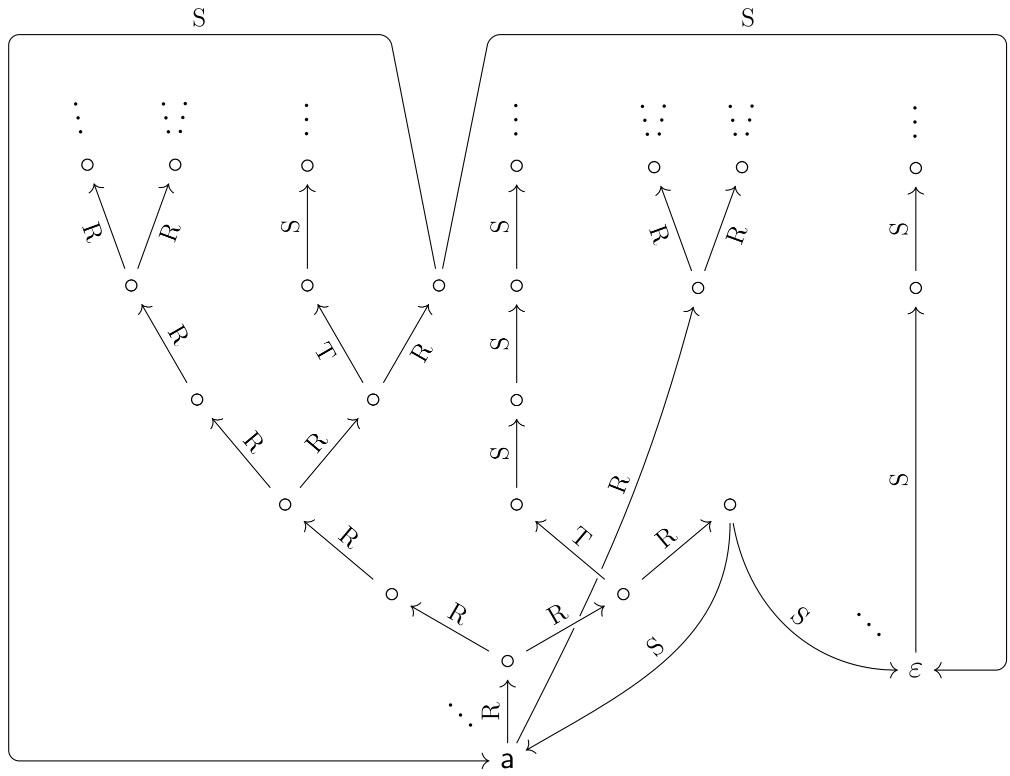


Figure 3.8: Initial portion of the f^*-interlacing of $\mathcal{I}_e$

To ensure modelhood of the f^*-interlacing, we rely on Theorem 4 and thus concentrate on proving f^* is pseudo-injective.

Lemma 4. *f^* is pseudo-injective.*

Proof. Let $u, v \in \Delta^\circ$ such that $f^*(u) = f^*(v)$. We distinguish two cases based on $f(u)$ belonging, or not, to Δ^*.

$f(u) \notin \Delta^*$. Therefore $f^*(u) = u$ and in particular $f^*(u) \notin \Delta^*$. Recall $f^*(v) = f^*(u)$, hence $f^*(v) \notin \Delta^*$. Therefore $f^*(v) = v$, yielding $u = v$, hence $f(u) = f(v)$.

$f(u) \in \Delta^*$. Therefore $f^*(u) = f(u) \in \Delta^*$. Recall $f^*(v) = f^*(u)$, hence $f^*(v) \in \Delta^*$. Therefore $f^*(v) = f(v)$, yielding $f(u) = f(v)$. □

We now turn to the counting matches in the f^*-interlacing $\mathcal{I}'$, which are exactly those of the original model $\mathcal{I}$. Notice however that the number of *matches* extending each counting match may have increased (eventually becoming infinite as in Example 10). The general idea is to use the homomorphism σ also provided by

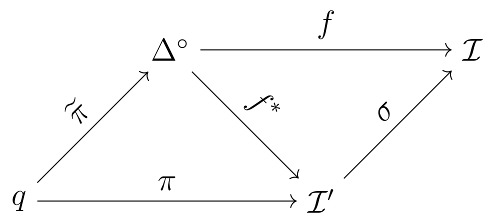


Figure 3.9: Mappings involved in the proof of Lemma 5.

Theorem 4 to injectively transform counting matches of $\mathcal{I}'$ into counting matches of $\mathcal{I}$, from which it follows by definition of Δ^* and f^* that these counting matches coincide.

Lemma 5. *The f^*-interlacing of a model $\mathcal{I}$ has at most as many counting matches for q as $\mathcal{I}$.*

Proof. Let $\mathcal{I}'$ be the f^*-interlacing of $\mathcal{I}$. We prove that if $\pi : q \to \mathcal{I}'$ is a match, then $(\sigma \circ \pi)_{|\mathbf{z}} = \pi_{|\mathbf{z}}$. Since $\sigma \circ \pi$ is a match of q in $\mathcal{I}$, it proves in particular that σ injects the c-matches of $\mathcal{I}'$ in the c-matches of $\mathcal{I}$, hence the claim. Let us thus consider a match $\pi : q \to \mathcal{I}'$. By definition of $\Delta^{\mathcal{I}'}$, we can pick some $\widetilde{\pi} : q \to \Delta^\circ$ such that $\pi = f^* \circ \widetilde{\pi}$ (note that $\widetilde{\pi}$ is not a match since we do not define an interpretation on Δ°). By Theorem 4, $\sigma \circ \pi$ is a match of q in $\mathcal{I}$. Therefore we have $(\sigma \circ \pi)(\mathbf{z}) \subseteq \Delta^*$, that is $(\sigma \circ f^* \circ \widetilde{\pi})(\mathbf{z}) \subseteq \Delta^*$. Remark 7 ensures $\sigma \circ f^* = f$, hence it is also $(f \circ \widetilde{\pi})(\mathbf{z}) \subseteq \Delta^*$. Along with the definition of f^*, it gives $f^* \circ \widetilde{\pi}_{|\mathbf{z}} = f \circ \widetilde{\pi}_{|\mathbf{z}}$. Recall $\pi = f^* \circ \widetilde{\pi}$, by definition of $\widetilde{\pi}$, hence in particular $\pi_{|\mathbf{z}} = f^* \circ \widetilde{\pi}_{|\mathbf{z}}$. Therefore $\sigma \circ \pi_{|\mathbf{z}} = \sigma \circ f^* \circ \widetilde{\pi}_{|\mathbf{z}} = f \circ \widetilde{\pi}_{|\mathbf{z}} = f^* \circ \widetilde{\pi}_{|\mathbf{z}} = \pi_{|\mathbf{z}}$. □

3.3 Answering CCQs over $\mathcal{ALCHI}$ ontologies

In this section, we devise a procedure that computes in double-exponential time the minimum number of counting matches, which immediately yields the following upper bound:

Theorem 6. *CCQ answering in $\mathcal{ALCHI}$ is in* 2EXP *w.r.t. combined complexity.*

Our approach is based upon the f^*-interlacings from Section 3.2.4, witnessing that there exists a model minimizing the count value that consists of an arbitrary structure $\mathcal{I}^*$ containing all assignments for the counting variables, augmented with structures that are tree-shaped, provided we ignore edges to and from $\mathcal{I}^*$. Importantly, we can bound the size of the central component $\mathcal{I}^*$, which enables us to explore all possible options for $\mathcal{I}^*$. Checking whether a given $\mathcal{I}^*$ can be extended to a model preserving the minimum count value can be done by specifying a set of *patterns* (intuitively representing a pair of adjacent elements), and testing via local consistency conditions whether they can be coherently assembled. This latter step takes inspiration from a CQ answering technique for existential rules found in Thomazo et al. [2012], and is also similar in spirit to type-elimination style procedures, which have been employed for reasoning with expressive DLs, see e.g. Rudolph et al. [2012]; Eiter et al. [2009].

3.3.1 Patterns

We fix an $\mathcal{ALCHI}$ KB $\mathcal{K} := (\mathcal{T}, \mathcal{A})$, a CCQ q. If $\mathcal{K}$ is not satisfiable, which can be tested in EXP (see Theorem 3), then the minimum number of counting matches is $+\infty$ (as every $[m, +\infty]$ is a certain answer). We henceforth focus on the case of $\mathcal{K}$ being satisfiable. It follows from Lemma 3 that the minimum is at most $M := (|\mathsf{Ind}(\mathcal{A})| + 3\,|\mathcal{T}|\,2^{|\mathcal{T}|})^{|q|}$. Hence, in any model $\mathcal{I}$ having a minimum number of counting matches, the set $\Delta^* \subseteq \Delta^{\mathcal{I}}$ (see Definition 22) of elements appearing in the image of a c-match has size at most $M \cdot |q|$. We can thus iterate over all such Δ^*, and even over all induced interpretations $\mathcal{I}^* = \mathcal{I}_{|\Delta^*}$, in double-exponential time w.r.t. combined complexity. The core task will then be to determine, given such a candidate $\mathcal{I}^*$, whether we can extend $\mathcal{I}^*$ into a model of $\mathcal{K}$ *without introducing new c-matches.*

Let us fix our candidate $\mathcal{I}^*$ and see how to check for a suitable extension. The challenging axioms to handle are those of the form $\mathrm{A} \sqsubseteq \exists \mathrm{R}.\mathrm{B}$, as they might require us to introduce new elements. We recall the set $\Omega := \{\mathrm{R}.\mathrm{B} \mid \mathrm{A} \sqsubseteq \exists \mathrm{R}.\mathrm{B} \in \mathcal{T}\}$ and shall refer to its members as *(existential) heads*. Importantly, as the f^*-interlacings from Section 3.2.4 witness, it is sufficient to consider extensions of $\mathcal{I}^*$ which are obtained by adding tree-shaped structures of new elements, plus some edges between the new elements and $\Delta^{\mathcal{I}^*}$ (we may need to use elements from $\Delta^{\mathcal{I}^*}$ as witnesses for existential heads to avoid new query matches). This property enables us to build such an extension by piecing together local interpretations corresponding to the addition of a single edge, using two distinguished symbols $\circledcirc$ and $\circledast$ as placeholders for fresh elements. We shall call these building blocks *patterns*, as they are inspired by a notion of the same name introduced for CQ answering with existential rules [Thomazo et al., 2012]. To be easily connected, these local interpretations are required to be *saturated*, in the following sense.

Definition 24. *An interpretation $\mathcal{I}$ is $\mathcal{T}$-saturated if it can be extended to a model $\mathcal{J}$ such that $\mathcal{J}_{|\Delta^{\mathcal{I}}} = \mathcal{I}$. In particular, $\mathcal{I}$ must contain all certain facts that can be inferred from $\mathcal{I}$ and $\mathcal{T}$.*

Remark 9. *Testing the $\mathcal{T}$-saturation of an interpretation $\mathcal{I}$ can be done by testing the $\mathcal{T}'$-satisfiability of $\mathcal{I}$ enriched by: all facts $\overline{\mathrm{A}}(e)$ with $e \in \Delta^{\mathcal{I}} \setminus \mathrm{A}^{\mathcal{I}}$, where $\overline{\mathrm{A}}$ is* *a fresh concept name, and* $\mathcal{T}'$ *contains* $\mathcal{T}$ *and axioms* $\mathrm{A} \sqcap \overline{\mathrm{A}} \sqsubseteq \bot$*; and by facts* $\overline{\mathrm{R}}(e_1, e_2)$ *with* $(e_1, e_2) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \setminus \mathrm{R}^{\mathcal{I}}$*, where* $\overline{\mathrm{R}}$ *is fresh role name, and* $\mathcal{T}'$ *also contains axioms* $\mathrm{R} \sqcap \overline{\mathrm{R}} \sqsubseteq \bot$*.*

Patterns not only consist of a local interpretation, but also other information needed to ensure that assembled patterns do not violate any TBox axioms or introduce any new matches. In particular, we shall keep track of (partial) query matches involving the local elements using the notion of a coherent specification. Intuitively, such a specification tells us which matches should be realized in the constructed extension, and naturally contains at least the matches of subqueries of q already realized in the local interpretation.

Definition 25. *Let* $\mathcal{I}$ *be an interpretation.*

- *The* specification $\mathfrak{M}^{\mathcal{I}}$ induced by $\mathcal{I}$ *is the set of pairs* (r, π) *such that* $r \subseteq q$ *and* $\pi : r \to \mathcal{I}$ *is a (full) match.*
- *A* coherent specification $\mathfrak{M}$ *over* $\mathcal{I}$ *is a set of pairs* (r, π) *where* $r \subseteq q$ *and* π *is a partial mapping from* $\mathsf{terms}(r)$ *to* $\Delta^{\mathcal{I}}$ *such that:*
  - $\mathfrak{M}^{\mathcal{I}} \subseteq \mathfrak{M}$*;*
  - *If* $(r_1, \pi_1), (r_2, \pi_2) \in \mathfrak{M}$ *with* π_1 *and* π_2 *defined and equal on* $\mathsf{var}(r_1) \cap \mathsf{var}(r_2)$*, then* $(r_1 \cup r_2, \pi_1 \cup \pi_2) \in \mathfrak{M}$*.*

To check the compatibility of different specifications, we will need to be able to restrict them to a subdomain:

Definition 26. *The* restriction of a specification $\mathfrak{M}$ *over an interpretation* $\mathcal{I}$ *to a domain* $\Delta \subseteq \Delta^{\mathcal{I}}$*, denoted* $\mathfrak{M}_{|\Delta}$*, is the set of pairs* (r, π') *such that* π' *is the restriction of* π *to* $\pi^{-1}(\Delta)$ *for some* $(r, \pi) \in \mathfrak{M}$*.*

Remark 10. *Induced specifications and restrictions of coherent specifications are both coherent specifications.*

Patterns will contain a further kind of information called a prediction, defined next. The purpose will be explained in more detail once we introduce links between patterns, but roughly it serves to coordinate the successor patterns of a pattern to avoid violating negative role inclusions.

Definition 27. *A prediction is a function* $\mathsf{next} : \Omega \to \Delta^{\mathcal{I}^*} \cup \Omega$ *verifying that: for all* $\mathrm{R}_1.\mathrm{B}_1, \mathrm{R}_2.\mathrm{B}_2 \in \Omega$*, if* $\mathcal{T} \models \mathrm{R}_1 \sqcap \mathrm{R}_2 \sqsubseteq \bot$*, then* $\mathsf{next}(\mathrm{R}_1.\mathrm{B}_1) \neq \mathsf{next}(\mathrm{R}_2.\mathrm{B}_2)$*.*

We now formally define the central notion of pattern, relative to $\mathcal{I}^*$ and a candidate specification $\mathfrak{M}^*$ over $\mathcal{I}^*$.

Definition 28. *A pattern* $\mathbb{P}$ *(w.r.t.* $\mathcal{I}^*$ *and* $\mathfrak{M}^*$*) is a tuple* $(\mathfrak{fr}^{\mathbb{P}}, \mathfrak{gen}^{\mathbb{P}}, \mathfrak{I}^{\mathbb{P}}, \mathfrak{M}^{\mathbb{P}}, \mathsf{next}_{\mathbb{P}})$ *where:*

- *The frontier and generated domains* $\mathfrak{fr}^{\mathbb{P}}$ *and* $\mathfrak{gen}^{\mathbb{P}}$ *are* disjoint *sets of elements from* $\Delta^{\mathcal{I}^*} \cup \{\circledcirc, \circledast\}$*;*
- $\mathfrak{I}^{\mathbb{P}}$ *is a* $\mathcal{T}$*-saturated and* $\mathcal{T}$*-satisfiable interpretation with* $\Delta^{\mathfrak{I}^{\mathbb{P}}} = \Delta^{\mathcal{I}^*} \cup \mathfrak{fr}^{\mathbb{P}} \cup \mathfrak{gen}^{\mathbb{P}}$ *and such that* $\mathfrak{I}^{\mathbb{P}}_{|\Delta^{\mathcal{I}^*}} = \mathcal{I}^*$*;*
- $\mathfrak{M}^{\mathbb{P}}$ *is a coherent specification of* q *over* $\mathfrak{I}^{\mathbb{P}}$ *that preserves* $\mathfrak{M}^*$*, that is* $(\mathfrak{M}^{\mathbb{P}})_{|\Delta^{\mathcal{I}^*}} = \mathfrak{M}^*$*;*
- $\mathsf{next}_{\mathbb{P}}$ *is a prediction.*

We shall be interested in two types of patterns. The (unique) *initial pattern* $\mathbb{P}^* := (\emptyset, \Delta^{\mathcal{I}^*}, \mathcal{I}^*, \mathfrak{M}^*, \mathsf{Id})$ simply represents $\mathcal{I}^*$ and $\mathfrak{M}^*$. All other patterns of interest represent additions of a pair of adjacent elements, and $\mathfrak{fr}^{\mathbb{P}}$ and $\mathfrak{gen}^{\mathbb{P}}$ will be singletons (representing these two elements).

Example 11. *In our running example,* $\Delta^*_e := \{\mathsf{a}, \varepsilon\}$ *(z maps to only these elements). The initial pattern* $\mathbb{P}^*_e$ *has frontier* $\emptyset$*, generated terms* Δ^*_e*, interpretation* $\mathcal{I}^*_e := (\mathcal{I}_e)_{|\Delta^*_e}$ *depicted in Figure 3.10a, and specification given in Table 3.3. Non-initial patterns will be illustrated later.*

We now define how to combine patterns together, and first, *when* it is necessary to combine them.

Definition 29. *We say that* $\mathrm{R.B} \in \Omega$ *is applicable to* e *in* a pattern $\mathbb{P}$ *if* $e \in \mathfrak{gen}^{\mathbb{P}}$ *and there exists* $\mathrm{A} \sqsubseteq \exists \mathrm{R.B} \in \mathcal{T}$ *with* $e \in \mathrm{A}^{\mathfrak{I}^{\mathbb{P}}}$ *but* $e \notin (\exists \mathrm{R.B})^{\mathfrak{I}^{\mathbb{P}}}$.

When a head is applicable to a pattern, we need to find another pattern that can realize the head. This is formalized by the following notion of link between patterns, which requires that the two patterns are compatible (Conditions 1, 2, 3), the second pattern realizes the head (Condition 4), and certain consistency conditions hold (Conditions 5, 6).

Definition 30. *Let* $\mathrm{R.B}$ *be an applicable head on* e_1 *in a pattern* $\mathbb{P}_1$*. There is a* $(\mathrm{R.B}, e_1)$*-link from* $\mathbb{P}_1$ *to* $\mathbb{P}_2$ *if:*

1. $\mathfrak{fr}^{\mathbb{P}_2} = \{e_1\}$ *and* $\mathfrak{gen}^{\mathbb{P}_2}$ *is a singleton, say* $\{e_2\}$*;*
2. *For all concept names* A*, we have* $e_1 \in \mathrm{A}^{\mathfrak{I}^{\mathbb{P}_1}}$ *iff* $e_1 \in \mathrm{A}^{\mathfrak{I}^{\mathbb{P}_2}}$*;*
3. $\mathfrak{M}^{\mathbb{P}_1}_{|\Delta^{\mathcal{I}^*} \cup \{e_1\}} = \mathfrak{M}^{\mathbb{P}_2}_{|\Delta^{\mathcal{I}^*} \cup \{e_1\}}$*;*

p	y_1	y_2	z	$\mathfrak{M}^*$	$\mathfrak{M}_1$	$\mathfrak{M}_2$	$\mathfrak{M}_3$	$\mathfrak{M}_4$	$\mathfrak{M}_5$	$\mathfrak{M}_6$	$\mathfrak{M}_7$	$\mathfrak{M}_8$	$\mathfrak{M}_9$	$\mathfrak{M}_{10}$	$\mathfrak{M}_{11}$
$\emptyset$				✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
p_{R}	a			✓					✓	✓	✓	✓	✓	✓	✓
p_{R}				✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
p_{R}		$\circledcirc$			✓	✓			✓	✓					
p_{R}	a	$\circledcirc$			✓	✓	✓	✓							
p_{R}	$\circledcirc$				✓	✓	✓	✓							
p_{R}	$\circledcirc$	$\circledast$				✓		✓							
p_{R}	$\circledast$					✓		✓				✓			
p_{R}		$\circledast$					✓	✓			✓	✓			
p_{R}	$\circledast$	$\circledcirc$					✓				✓				
p_{S}		$\circledcirc$	a						✓	✓	✓				
p_{S}		$\circledcirc$	ε						✓	✓	✓				
p_{S}				✓	✓	✓	✓	✓	✓	✓	✓				
p_{S}		ε		✓	✓	✓	✓	✓	✓	✓	✓				
p_{S}			a	✓	✓	✓	✓	✓				✓	✓	✓	✓
p_{S}			ε	✓	✓	✓	✓	✓				✓	✓	✓	✓
p_{S}		ε	$\circledcirc$									✓	✓	✓	✓
p_{S}		$\circledcirc$	$\circledcirc$									✓	✓	✓	✓
p_{S}		ε	$\circledast$											✓	✓
p_{S}		$\circledast$	$\circledast$											✓	✓
p_{S}		$\circledcirc$	$\circledast$											✓	✓
p_{S}		$\circledast$	$\circledcirc$											✓	✓
q_e			a	✓	✓	✓	✓						✓	✓	✓
q_e			ε	✓	✓	✓	✓						✓	✓	✓
q_e	$\circledast$		a					✓				✓			
q_e	$\circledast$		ε					✓				✓			
q_e		$\circledcirc$	a						✓	✓					
q_e		$\circledcirc$	ε						✓	✓					
q_e	$\circledast$	$\circledcirc$	a								✓				
q_e	$\circledast$	$\circledcirc$	ε								✓				

Table 3.3: Specifications from Example 12. ✓ in a $\mathfrak{M}$-column indicates that the pair (p, π) given by the first 4 columns, with π a partial match of p, belongs to $\mathfrak{M}$.

4. $e_2 \in \mathrm{B}^{\mathfrak{I}^{\mathbb{P}_2}}$ *and for all* $\mathrm{P} \in \mathsf{N_R}$*:*

$$\mathrm{P}^{\mathfrak{I}^{\mathbb{P}_2}} = \mathrm{P}^{\mathcal{I}^*} \cup \{(e_1, e_2) \mid \mathcal{T} \models \mathrm{R} \sqsubseteq \mathrm{P}\} \cup \{(e_2, e_1) \mid \mathcal{T} \models \mathrm{R}^- \sqsubseteq \mathrm{P}\}$$

5. If ever $e_2 \in \Delta^{\mathcal{I}^*} \cap \mathfrak{fr}^{\mathbb{P}_1}$*, then* $\mathfrak{I}^{\mathbb{P}_1} \cup \mathfrak{I}^{\mathbb{P}_2}$ *is* $\mathcal{T}$*-satisfiable.*

6. If $e_2 \in \Delta^{\mathcal{I}^*}$*, then* $e_2 = \mathsf{next}_{\mathbb{P}_1}(\mathrm{R.B})$*.*

We denote by $\mathbb{L}^{\mathrm{R.B}}_{\mathbb{P}_1,e_1}$ *the set of patterns* $\mathbb{P}_2$ *such that there is a* $(\mathrm{R.B}, e_1)$*-link from* $\mathbb{P}_1$ *to* $\mathbb{P}_2$*.*

Remark 11. *Predictions are used in Condition 6 to avoid problematic situations where two successor patterns merge back to the same element of* $\Delta^{\mathcal{I}^*}$*. Specifically, if we have a* $\mathrm{R}_1.\mathrm{B}_1$*-link from* $\mathbb{P}_0$ *to* $\mathbb{P}_1$ *and a* $\mathrm{R}_2.\mathrm{B}_2$*-link from* $\mathbb{P}_0$ *to* $\mathbb{P}_2$*, with* $\mathcal{T} \models \mathrm{R}_1 \sqcap \mathrm{R}_2 \sqsubseteq \bot$*, then* $\mathsf{next}_{\mathbb{P}_0}(\mathrm{R}_1.\mathrm{B}_1) \neq \mathsf{next}_{\mathbb{P}_0}(\mathrm{R}_2.\mathrm{B}_2)$*, preventing* $\mathbb{P}_1$ *and* $\mathbb{P}_2$ *from using the same element of* $\Delta^{\mathcal{I}^*}$ *as generated term (which would violate* $\mathcal{T}$*). Condition 5 is similar in spirit, handling the case of the pattern* $\mathbb{P}_1$ *using the frontier element of* $\mathbb{P}_0$ *as a generated term.*

Example 12. *We consider patterns* $\mathbb{P}^e_1, \ldots, \mathbb{P}^e_{11}$ *whose interpretations are depicted in Figure 3.10. Frontier terms are indicated using square-purple and generated terms by circle-green. Predictions are always* Id *except for* $\mathsf{next}_{\mathbb{P}^e_7}$*, which maps* $\mathrm{S.A}$ *to* a *and* $\mathrm{S.B}$ *to* ε*. The specifications* $\mathfrak{M}_i$ *are given in Table 3.3, with* p_R *being the* R*-atom of* q_e *and* p_S *its* S*-atom. Links between our patterns are depicted in Figure 3.11.*

Let us illustrate the underlying mechanisms of specifications and Condition 3 with the link $\mathbb{P}^e_6 \in \mathbb{L}^{\mathrm{S.B}}_{\mathbb{P}^e_7,\circledcirc}$*. Notice that despite* $\mathfrak{I}^{\mathbb{P}^e_7}$ *interprets* S *as empty, its specification* $\mathfrak{M}_7$ *contains the pair* $(q_e, (y_1, y_2, z) \mapsto (\circledast, \circledcirc, \varepsilon))$ *witnessing for a complete match it doesn't have full knowledge of. This is made possible by: (i) the fact* $\mathrm{R}(\circledast, \circledcirc)$ *satisfied in the interpretation of* $\mathbb{P}^e_7$ *ensures* $(p_\mathrm{R}, (y_1, y_2) \mapsto (\circledast, \circledcirc)) \in \mathfrak{M}_7$ *as* $\mathfrak{M}_7$ *must be coherent; (ii) similarly* $(p_\mathrm{S}, (y_2, z) \mapsto (\circledcirc, \varepsilon)) \in \mathfrak{M}_6$ *from the coherence of* $\mathfrak{M}_6$*; (iii) Condition 3 requires the restrictions of their specifications to* $\{\mathsf{a}, \varepsilon, \circledcirc\}$ *to coincide, hence* $(p_\mathrm{S}, (y_2, z) \mapsto (\circledcirc, \varepsilon)) \in \mathfrak{M}_7$*; (iv) coherence of* $\mathfrak{M}_7$ *requires that the combination of* $(p_\mathrm{S}, (y_2, z) \mapsto (\circledcirc, \varepsilon))$ *and* $(p_\mathrm{R}, (y_1, y_2) \mapsto (\circledast, \circledcirc))$ *belongs to* $\mathfrak{M}_7$*, which is the desired pair. This highlights how specifications, if well assembled, suffice to capture complete matches of the CCQ of interest despite the local interpretations they are attached to.*

We now characterize patterns that cannot be used to satisfy a head without introducing a new c-match.

Definition 31. *A pattern* $\mathbb{P}$ *is* rejecting *if one of the two following conditions holds:*

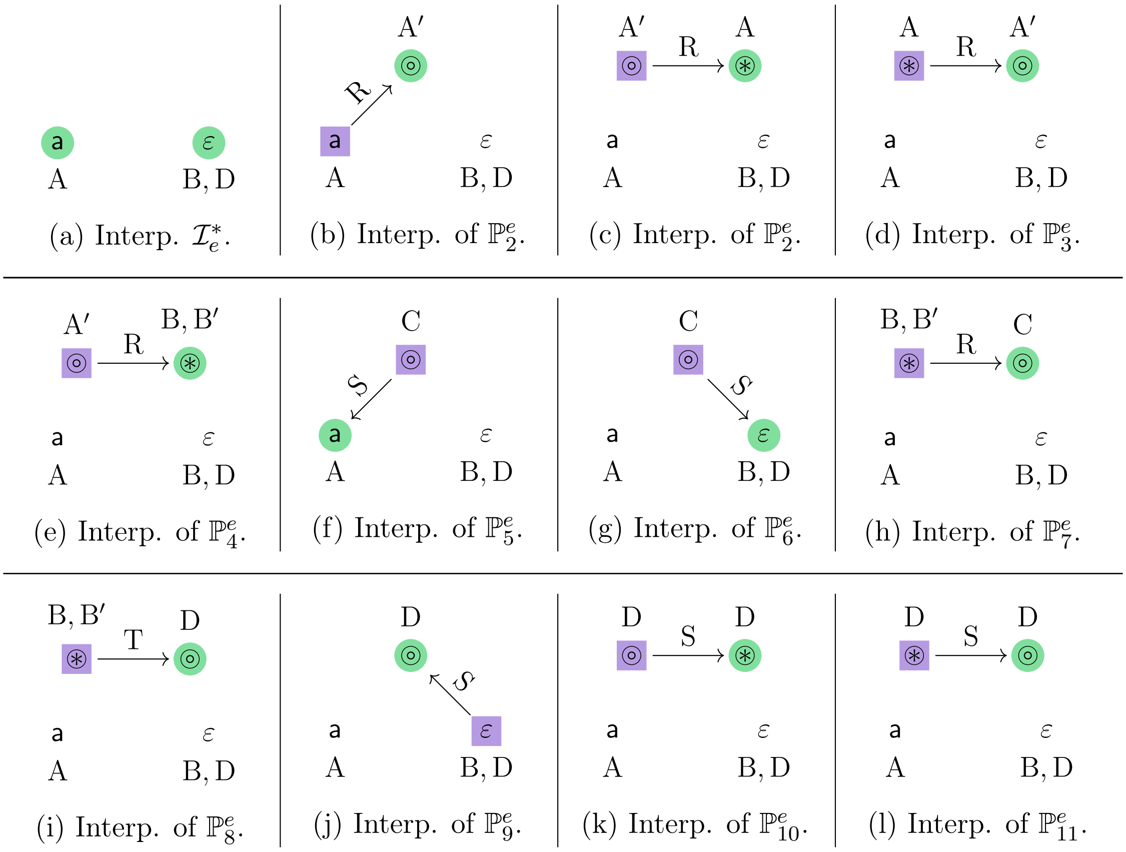


Figure 3.10: Interpretations of the patterns from Example 12

- *There exists* $(q, \pi) \in \mathfrak{M}^{\mathbb{P}}$ *with* $\pi(\mathbf{z}) \cap \{\circledcirc, \circledast\} \neq \emptyset$*;*
- *There exists an existential head* R.B *that applies on* e *in* $\mathbb{P}$ *such that all patterns* $\mathbb{P}' \in \mathbb{L}^{\mathrm{R.B}}_{\mathbb{P},e}$ *are rejecting.*

A pattern is accepting if it is not rejecting.

The acceptance of the initial pattern $\mathbb{P}^*$ is a sufficient condition ensuring $\mathcal{I}^*$ extends to a model having no more counting matches than encoded in $\mathfrak{M}^*$, i.e. the pairs $(q, \pi) \in \mathfrak{M}^*$ such that π is defined for all counting variables.

Lemma 6. *If* $\mathbb{P}^* := (\emptyset, \Delta^*, \mathcal{I}^*, \mathfrak{M}^*, \mathsf{Id})$ *is accepting, then there exists a model* $\mathcal{I}^\diamondsuit$ *such that* $\mathcal{I}^* \subseteq \mathcal{I}^\diamondsuit$ *and if* $\pi : q \to \mathcal{I}^\diamondsuit$ *is a c-match, then* $(q, \pi) \in \mathfrak{M}^*$*. In particular,* $\mathcal{I}^\diamondsuit$ *has at most as many c-matches as those encoded in* $\mathfrak{M}^*$*.*

Furthermore, the minimum number of counting matches is reached among initial patterns due to the following result:

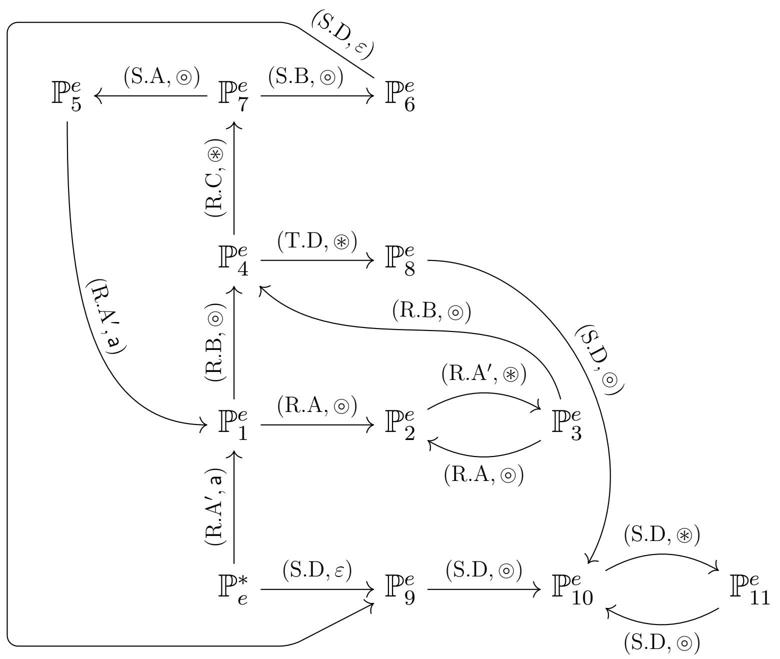


Figure 3.11: Links between the 12 patterns from Example 12. Label (h, e) on an edge (u, v) indicates a (h, e)-link between patterns u and v.

Lemma 7. *If $\mathcal{I}$ is a model of $\mathcal{K}$ with m counting matches, then there exists an accepting initial pattern whose specification encodes* exactly *m c-matches.*

Before proving Lemmas 6 and 7, let us recap the overall double-exponential procedure underlying Theorem 6:

Proof of Theorem 6. We consider all possible initial patterns $\mathbb{P}^*$ with an interpretation domain Δ^* such that $\mathsf{Ind}(\mathcal{A}) \subseteq \Delta^*$ and $|\Delta^*| \leq M|q|$ (recall Lemma 3). Every such $\mathbb{P}^*$ is of single-exponential size w.r.t. combined complexity (observe that its specification $\mathfrak{M}^*$ corresponds to a subset of $2^q \times (\Delta^* \cup \{\uparrow\})^q$), and thus are double-exponential in number (up to isomorphism) and can be enumerated in double-exponential time. For each such $\mathbb{P}^*$, we construct in double-exponential time the set of all possible descendant patterns of $\mathbb{P}^*$ (which are of single-exponential size, having at most $|\Delta^*| + 2$ elements). We then check whether each possible pattern ($\mathbb{P}^*$ or candidate descendant) is in fact a well-defined pattern, in particular, its interpretation is $\mathcal{T}$-satisfiable and $\mathcal{T}$-saturated. These verifications can be done in double-exponential time, recalling that KB satisfiability and instance checking are in EXP for $\mathcal{ALCHI}$ (even this variant with negative role inclusions, see 3). Acceptance of $\mathbb{P}^*$ is tested (again in deterministic exponential time) by repeatedly iterating over the set of patterns and removing those that are rejecting either due to their specification, or due to the removal of all patterns that could provide a link for an applicable head. If $\mathbb{P}^*$ is found to be accepting and $\mathfrak{M}^*$ encodes m c-matches, then Lemma 6 ensures the existence of a model with at most m c-matches. Conversely, Lemma 7 ensures that we can find the smallest such m among the accepting initial patterns. □

3.3.2 Soundness: from patterns to models

To prove Lemma 6, assume we are given an initial pattern $\mathbb{P}^* := (\emptyset, \Delta^*, \mathcal{I}^*, \mathfrak{M}^*, \mathsf{Id})$ that is accepting. Our aim is to construct a model $\mathcal{I}$ that extends $\mathcal{I}^*$ and is such that $(q, \pi) \in \mathfrak{M}^*$ for every c-match $\pi : q \to \mathcal{I}$.

We proceed as follows. For each accepting descendant pattern $\mathbb{P}$ (w.r.t. $\mathcal{I}^*$ and $\mathfrak{M}^*$) and each head R.B applicable to e in $\mathbb{P}$, we choose an accepting pattern $\mathsf{ch}^{\mathrm{R.B}}_{\mathbb{P},e}$ from $\mathbb{L}^{\mathrm{R.B}}_{\mathbb{P},e}$. Then, starting from $\mathbb{P}^*$, we build a tree-shaped set of words, whose letters consist of an accepting pattern and existential head, and which witnesses the acceptance of $\mathbb{P}^*$.

Definition 32. *The pattern tree $\mathcal{P}$ is the smallest set of words such that:*

- $(\mathbb{P}^*, \emptyset) \in \mathcal{P}$*;*

- *If* $w \cdot (\mathbb{P}, h) \in \mathcal{P}$ *and* R.B *is applicable to* e *in* $\mathbb{P}$*, then* $w \cdot (\mathbb{P}, h) \cdot (\mathsf{ch}^{\mathrm{R.B}}_{\mathbb{P},e}, \mathrm{R.B}) \in \mathcal{P}$.

Remark 12. *Each element from $\mathcal{P}$ has shape $w \cdot (\mathbb{P}, h)$, where w is eventually the empty word. In particular, in what follows, when we let $w \cdot (\mathbb{P}, h) \in \mathcal{P}$ it includes the case of the initial pair $w \cdot (\mathbb{P}, h) = (\mathbb{P}^*, \emptyset)$.*

It remains to 'glue' together the interpretations $\mathfrak{I}^{\mathbb{P}}$ according to the structure of $\mathcal{P}$. Since a pattern $\mathbb{P}$ may occur more than once, we create a copy of $\mathfrak{I}^{\mathbb{P}}$ for each node in $\mathcal{P}$ of the form $w \cdot (\mathbb{P}, h)$. We do not duplicate however elements from $\mathcal{I}^*$ as they precisely are those we want to reuse. Hence only the frontier term and the generated term may be duplicated (provided they do not belong to Δ^*). When a node $w \cdot (\mathbb{P}_1, h_1) \cdot (\mathbb{P}_2, h_2)$ is encountered, we merge the frontier term of $\mathbb{P}_2$ with the already-introduced copy of the generated element from $\mathbb{P}_1$ on which h_2 is applied (which is the only element in $\mathfrak{fr}^{\mathbb{P}_2}$). Therefore, when considering such a node $w \cdot (\mathbb{P}_1, h_1) \cdot (\mathbb{P}_2, h_2)$, the only element we might have to introduce is a copy of the generated term e of $\mathbb{P}_2$ (unless $e \in \Delta^*$), which we shall simply name $w \cdot (\mathbb{P}_1, h_1) \cdot (\mathbb{P}_2, h_2)$. Formally, the copying and merging of elements is achieved by the following family of duplicating functions, defined inductively for each $w \cdot (\mathbb{P}, h) \in \mathcal{P}$.

$$\lambda_{w\cdot(\mathbb{P},h)} : \Delta^{\mathfrak{I}^{\mathbb{P}}} \to \Delta^{\mathcal{I}^*} \cup \{w, w \cdot (\mathbb{P}, h)\}$$
$$e \mapsto \begin{cases} e & \text{if } e \in \Delta^{\mathcal{I}^*} \\ w & \text{if } e \in \mathfrak{fr}^{\mathbb{P}} \setminus \Delta^{\mathcal{I}^*} \\ w \cdot (\mathbb{P}, h) & \text{if } e \in \mathfrak{gen}^{\mathbb{P}} \setminus \Delta^{\mathcal{I}^*} \end{cases}$$

Note that if $e \in \mathfrak{fr}^{\mathbb{P}_2} \setminus \Delta^{\mathcal{I}^*}$, then $e \in \mathfrak{gen}^{\mathbb{P}_1} \setminus \Delta^{\mathcal{I}^*}$, hence

$$\lambda_{w\cdot(\mathbb{P}_1,h_1)\cdot(\mathbb{P}_2,h_2)}(e) = \lambda_{w\cdot(\mathbb{P}_1,h_1)}(e) = w \cdot (\mathbb{P}_1, h_1).$$

The desired model $\mathcal{I}$ can then be defined as follows:

$$\mathcal{I} := \bigcup_{w\cdot(\mathbb{P},h)\in\mathcal{P}} \lambda_{w\cdot(\mathbb{P},h)}(\mathfrak{I}^{\mathbb{P}}),$$

that is the domain (resp. the interpretation of each concept name and each role name) of $\mathcal{I}$ is the union across all $w \cdot (\mathbb{P}, h) \in \mathcal{P}$ of the image by $\lambda_{w\cdot(\mathbb{P},h)}$ of the domain (resp. the interpretation of each concept name and each role name) of $\mathfrak{I}^{\mathbb{P}}$.

Example 13. *The patterns introduced in Example 12 are sufficient to witness that $\mathbb{P}_e^*$ is accepting. The corresponding pattern tree $\mathcal{P}_e$ can be obtained by "unfolding" the links between patterns depicted in Figure 3.11, starting from the pattern $\mathbb{P}_e^*$. The resulting $\mathcal{I}_e$ is depicted in Figure 3.8, which coincides with the f^*-interlacing of the original model $\mathcal{I}_e$. Notice how it inherits the tree-shaped structure of $\mathcal{P}_e$ up to roles collapsing back in $\mathcal{I}_e^*$.*

By definition, each $\lambda_{w\cdot(\mathbb{P},h)}$ is a homomorphism from $\mathfrak{I}^{\mathbb{P}}$ to $\mathcal{I}$. Due to Condition 2 of Definition 30, the shared element of linked patterns must belong to the same concepts, so concept membership in $\mathcal{I}$ transfers back to $\mathfrak{I}^{\mathbb{P}}$:

Lemma 8. *For all $w \cdot (\mathbb{P}, h) \in \mathcal{P}$, for all $e \in \Delta^{\mathfrak{I}^{\mathbb{P}}}$ and for all $\mathrm{A} \in \mathsf{N_C}$, if $\lambda_{w\cdot(\mathbb{P},h)}(e) \in \mathrm{A}^{\mathcal{I}}$, then $e \in \mathrm{A}^{\mathfrak{I}^{\mathbb{P}}}$.*

Proof. Let $w_1 \cdot (\mathbb{P}_1, h_1) \in \mathcal{P}$ be a node from the pattern tree, e_1 an element from $\Delta^{\mathfrak{I}^{\mathbb{P}_1}}$ and A a concept name. Assume $\lambda_{w_1\cdot\mathbb{P}_1}(e_1) \in \mathrm{A}^{\mathcal{I}}$. By definition of $\mathrm{A}^{\mathcal{I}}$ there exists a node $w_2 \cdot (\mathbb{P}_2, h_2)$ from the pattern tree, and an element $e_2 \in \mathrm{A}^{\mathfrak{I}^{\mathbb{P}_2}}$ such that $\lambda_{w_1\cdot(\mathbb{P}_1,h_1)}(e_1) = \lambda_{w_2\cdot(\mathbb{P}_2,h_2)}(e_2)$. We further refer to this equality as $(*)$. We distinguish 5 cases.

1. $e_1 \in \Delta^{\mathcal{I}^*}$ or $e_2 \in \Delta^{\mathcal{I}^*}$.
   $(*)$ yields $e_1 = e_2$. Interpretation $\mathfrak{I}^{\mathbb{P}_2}$ preserves $\mathcal{I}^*$, hence $e_2 \in \mathrm{A}^{\mathcal{I}^*}$. Interpretation $\mathfrak{I}^{\mathbb{P}_1}$ preserves $\mathcal{I}^*$, hence $e_1 \in \mathrm{A}^{\mathfrak{I}^{\mathbb{P}_1}}$.

   In the remaining cases, we assume $e_1, e_2 \notin \Delta^{\mathcal{I}^*}$, which ensures $\mathbb{P}_1 \neq \mathbb{P}^*$ and $\mathbb{P}_2 \neq \mathbb{P}^*$. In particular, $\mathfrak{fr}^{\mathbb{P}_1}$, $\mathfrak{gen}^{\mathbb{P}_1}$, $\mathfrak{fr}^{\mathbb{P}_2}$ and $\mathfrak{gen}^{\mathbb{P}_2}$ are singletons.

2. $e_1 \in \mathfrak{gen}^{\mathbb{P}_1} \setminus \Delta^{\mathcal{I}^*}$ and $e_2 \in \mathfrak{gen}^{\mathbb{P}_2} \setminus \Delta^{\mathcal{I}^*}$.
   $(*)$ yields $\mathbb{P}_1 = \mathbb{P}_2$. Recall $\mathfrak{gen}^{\mathbb{P}_1}$ is a singleton hence $e_1 = e_2$, which concludes.

3. $e_1 \in \mathfrak{fr}^{\mathbb{P}_1} \setminus \Delta^{\mathcal{I}^*}$ and $e_2 \in \mathfrak{gen}^{\mathbb{P}_2} \setminus \Delta^{\mathcal{I}^*}$.
   $(*)$ yields $w_1 = w_2 \cdot (\mathbb{P}_2, h_2)$. In particular $w_2 \cdot (\mathbb{P}_2, h_2) \cdot (\mathbb{P}_1, h_1) \in \mathcal{P}$, hence $\mathbb{P}_1 = \mathsf{ch}^{h_1}_{\mathbb{P}_2, e_2}$. From the definition of a link, $e_1 = e_2$ (Condition 1) and e_1 satisfies the same concepts in both interpretations (Condition 2) hence $e_1 \in \mathrm{A}^{\mathfrak{I}^{\mathbb{P}_1}}$.

4. $e_1 \in \mathfrak{gen}^{\mathbb{P}_1} \setminus \Delta^{\mathcal{I}^*}$ and $e_2 \in \mathfrak{fr}^{\mathbb{P}_2} \setminus \Delta^{\mathcal{I}^*}$.
   Same arguments as for Case 3 but this time with $\mathbb{P}_2 = \mathsf{ch}^{h_2}_{\mathbb{P}_1, e_1}$.

5. $e_1 \in \mathfrak{fr}^{\mathbb{P}_1} \setminus \Delta^{\mathcal{I}^*}$ and $e_2 \in \mathfrak{fr}^{\mathbb{P}_2} \setminus \Delta^{\mathcal{I}^*}$.
   $(*)$ yields the existence of $w \cdot (\mathbb{Q}, h)$ such that $w_1 = w_2 = w \cdot (\mathbb{Q}, h)$. In particular $w \cdot (\mathbb{Q}, h) \cdot (\mathbb{P}_2, h_2) \in \mathcal{P}$, hence $\mathbb{P}_2 = \mathsf{ch}^{h_2}_{\mathbb{Q}, e_2}$. By definition of a link, e_2 satisfies the same concepts in both interpretations (Condition 2) hence $e_2 \in \mathrm{A}^{\mathfrak{I}^{\mathbb{Q}}}$. Similarly, $w \cdot (\mathbb{Q}, h) \cdot (\mathbb{P}_1, h_1) \in \mathcal{P}$, hence $\mathbb{P}_1 = \mathsf{ch}^{h_1}_{\mathbb{Q}, e_2}$. By definition of a link $e_1 = e_2$ (Condition 1) and e_1 satisfies the same concepts in both interpretations (Condition 2) hence $e_1 \in \mathrm{A}^{\mathfrak{I}^{\mathbb{P}_1}}$. □

An analogous property fails however for roles, as two patterns $\mathbb{P}_1 = \mathsf{ch}^{\mathrm{R}_1.\mathrm{B}_1}_{\mathbb{P}, e}$ and $\mathbb{P}_2 = \mathsf{ch}^{\mathrm{R}_2.\mathrm{B}_2}_{\mathbb{P}, e}$ may reuse the same element from Δ^*, that is, $\mathfrak{gen}^{\mathbb{P}_1} = \mathfrak{gen}^{\mathbb{P}_2} \subseteq \Delta^*$. In that case, we have $\lambda_{w \cdot (\mathbb{P}, h) \cdot (\mathbb{P}_1, \mathrm{R}_1.\mathrm{B}_1)}(\Delta^{\mathfrak{I}^{\mathbb{P}_1}}) = \lambda_{w \cdot (\mathbb{P}, h) \cdot (\mathbb{P}_2, \mathrm{R}_2.\mathrm{B}_2)}(\Delta^{\mathfrak{I}^{\mathbb{P}_2}})$ hence $\mathfrak{I}^{\mathbb{P}_1}$ maps somewhere in $\mathcal{I}$ satisfying the role R_2, but there is no reason for R_2 to be satisfied in $\mathfrak{I}^{\mathbb{P}_1}$. Such a situation may also arise for $\mathbb{P}$ and $\mathbb{P}_1$ as above if $\mathfrak{gen}^{\mathbb{P}_1} \subseteq \mathfrak{fr}^{\mathbb{P}} \subseteq \Delta^*$. Conditions 5 and 6 from the definition of a link, respectively handling the second and the first of the two cases described above, allow us to show the following weaker property, sufficient for our purposes:

Lemma 9. *For all* $w \cdot (\mathbb{P}, h) \in \mathcal{P}$*,* $d, e \in \Delta^{\mathfrak{I}^{\mathbb{P}}}$*, and* $\mathrm{P} \in \mathsf{N_R}$*: if* $(\lambda_{w \cdot (\mathbb{P}, h)}(d), \lambda_{w \cdot (\mathbb{P}, h)}(e)) \in \mathrm{P}^{\mathcal{I}}$*, then* $\mathfrak{I}^{\mathbb{P}}$ *remains* $\mathcal{T}$*-satisfiable if we add* (d, e) *to* $\mathrm{P}^{\mathfrak{I}^{\mathbb{P}}}$*.*

Proof sketch. The full proof can be found in the appendix and proceeds by a case analysis similar in spirit to the proof of Lemma 8 (except there are twice as many elements to consider). As mentioned, this is the main proof in which Conditions 5 and 6 from the definition of a link are required. We recall the purpose of these conditions has been discussed in Remark 11. □

With Lemmas 8 and 9 in hand, we are ready to show that $\mathcal{I}$ is a model of $\mathcal{K}$.

Lemma 10. $\mathcal{I}$ *is a model of* $\mathcal{K}$*.*

Proof. We consider each possible shape of assertion and axiom in $\mathcal{K}$:

$\mathrm{A}(\mathsf{a})$. Since $\mathcal{I}^*$ is a model of $\mathcal{A}$, we have $\mathsf{a} \in \mathrm{A}^{\mathcal{I}^*}$. Recall $\mathcal{I}^*$ is the interpretation of the initial pattern. Therefore the definition of $\mathrm{A}^{\mathcal{I}}$ gives $\mathsf{a} = \lambda_{\mathbb{P}^*,\emptyset}(\mathsf{a}) \in \mathrm{A}^{\mathcal{I}}$.

$\mathrm{P}(\mathsf{a},\mathsf{b})$. Since $\mathcal{I}^*$ is a model of $\mathcal{A}$, we have $(\mathsf{a},\mathsf{b}) \in \mathrm{P}^{\mathcal{I}^*}$. Recall $\mathcal{I}^*$ is the interpretation of the initial pattern. Therefore the definition of $\mathrm{P}^{\mathcal{I}}$ gives $(\mathsf{a},\mathsf{b}) = (\lambda_{\mathbb{P}^*,\emptyset}(\mathsf{a}), \lambda_{\mathbb{P}^*,\emptyset}(\mathsf{b})) \in \mathrm{P}^{\mathcal{I}}$.

$\top \sqsubseteq \mathrm{A}$. Let $u \in \top^{\mathcal{I}} = \Delta^{\mathcal{I}}$. By definition of $\Delta^{\mathcal{I}}$, we have $w \cdot (\mathbb{P}, h) \in \mathcal{P}$ and an element $e \in \Delta^{\mathfrak{I}^{\mathbb{P}}}$ such that $\lambda_{w\cdot\mathbb{P},h}(e) = u$. Since $e \in \top^{\mathfrak{I}^{\mathbb{P}}}$ and $\mathfrak{I}^{\mathbb{P}}$ is $\mathcal{T}$-saturated, it ensures $e \in \mathrm{A}^{\mathfrak{I}^{\mathbb{P}}}$. Therefore the definition of $\mathrm{A}^{\mathcal{I}}$ gives $u = \lambda_{w\cdot(\mathbb{P},h)}(e) \in \mathrm{A}^{\mathcal{I}}$.

$\mathrm{A}_1 \sqcap \mathrm{A}_2 \sqsubseteq \mathrm{A}$. Let $u \in \mathrm{A}_1 \sqcap \mathrm{A}_2{}^{\mathcal{I}}$. By definition of $\Delta^{\mathcal{I}}$, there exist $w \cdot (\mathbb{P}, h) \in \mathcal{P}$ and an element $e \in \Delta^{\mathfrak{I}^{\mathbb{P}}}$ such that $\lambda_{w\cdot(\mathbb{P},h)}(e) = u$. Lemma 8 applied on both concepts A_1 and A_2 ensures $e \in \mathrm{A}_1 \sqcap \mathrm{A}_2{}^{\mathfrak{I}^{\mathbb{P}}}$. Since $\mathfrak{I}^{\mathbb{P}}$ is $\mathcal{T}$-saturated, it ensures $e \in \mathrm{A}^{\mathfrak{I}^{\mathbb{P}}}$. Therefore the definition of $\mathrm{A}^{\mathcal{I}}$ gives $u = \lambda_{w\cdot(\mathbb{P},h)}(e) \in \mathrm{A}^{\mathcal{I}}$.

$\mathrm{A}_1 \sqsubseteq \exists\mathrm{R}.\mathrm{A}_2$. Let $u \in \mathrm{A}_1^{\mathcal{I}}$. By definition of $\mathrm{A}_1{}^{\mathcal{I}}$, there exist $w \cdot (\mathbb{P}, h) \in \mathcal{P}$ and an element $e \in \Delta^{\mathfrak{I}^{\mathbb{P}}}$ such that $e \in \mathrm{A}_1{}^{\mathfrak{I}^{\mathbb{P}}}$ and $\lambda_{w\cdot(\mathbb{P},h)}(e) = u$.
We first prove that w.l.o.g. we can assume that $e \in \mathfrak{gen}^{\mathbb{P}}$. Indeed, if $e \in \Delta^*$, then e is in the generated domain of the initial pattern $\mathbb{P}^*$ and Lemma 8 gives $e \in \mathrm{A}_1{}^{\mathcal{I}^*}$. Otherwise, if $e \in \mathfrak{fr}^{\mathbb{P}}$, then w cannot be empty (recall the initial pattern has an empty frontier!) and therefore we have $w = w' \cdot (\mathbb{P}_0, h_0)$ with $e \in \mathfrak{gen}^{\mathbb{P}_0}$ and $\lambda_{w'\cdot(\mathbb{P}_0,h_0)}(e) = u$. Again, Lemma 8 gives $e \in \mathrm{A}_1{}^{\mathfrak{I}^{\mathbb{P}_0}}$. Therefore, up to switching $\mathbb{P}$ to $\mathbb{P}^*$ or to $\mathbb{P}_0$, we can assume w.l.o.g. that $e \in \mathfrak{gen}^{\mathbb{P}}$.
If $\mathrm{R}.\mathrm{A}_2$ is not applicable to e in $\mathbb{P}$, then this is because there exists $e' \in \mathrm{A}_2{}^{\mathfrak{I}^{\mathbb{P}}}$ with $(e, e') \in \mathrm{R}^{\mathfrak{I}^{\mathbb{P}}}$. Set $v := \lambda_{w\cdot(\mathbb{P},h)}(e')$. By definition of $\mathrm{R}^{\mathcal{I}}$ and $\mathrm{A}_2{}^{\mathcal{I}}$, we obtain $v \in \mathrm{A}_2^{\mathcal{I}}$ and $(u, v) \in \mathrm{R}^{\mathcal{I}}$.
If $\mathrm{R}.\mathrm{A}_2$ is applicable to e in $\mathbb{P}$, then since $\mathbb{P}$ is accepting there must exist an accepting pattern $\mathbb{P}_1 \in \mathsf{ch}_{\mathbb{P},e}^{\mathrm{R}.\mathrm{A}_2}$. In particular $w \cdot (\mathbb{P}, h) \cdot (\mathbb{P}_1, \mathrm{R}.\mathrm{A}_2) \in \mathcal{P}$. Let e' be the generated term of $\mathbb{P}_1$. From the definition of a link between patterns, we have $(e, e') \in \mathrm{R}^{\mathfrak{I}^{\mathbb{P}_1}}$ and $e' \in \mathrm{A}_2^{\mathfrak{I}^{\mathbb{P}_1}}$. Set $v := \lambda_{w\cdot(\mathbb{P},h)\cdot(\mathbb{P}_1,\mathrm{R}.\mathrm{A}_2)}(e')$. Noticing $\lambda_{w\cdot(\mathbb{P},h)}(e) = \lambda_{w\cdot(\mathbb{P},h)\cdot(\mathbb{P}_1,\mathrm{R}.\mathrm{A}_2)}(e)$ and by definition of $\mathrm{R}^{\mathcal{I}}$ and $\mathrm{A}_2{}^{\mathcal{I}}$, we obtain $v \in \mathrm{A}_2^{\mathcal{I}}$ and $(u, v) \in \mathrm{R}^{\mathcal{I}}$.

$\exists\mathrm{R}.\mathrm{A}_1 \sqsubseteq \mathrm{A}_2$. Let $u \in (\exists\mathrm{R}.\mathrm{A}_1)^{\mathcal{I}}$, that is, there exists $v \in \mathrm{A}_1^{\mathcal{I}}$ with $(u, v) \in \mathrm{R}^{\mathcal{I}}$. By definition of $\mathrm{R}^{\mathcal{I}}$, there exist $w \cdot (\mathbb{P}, h) \in \mathcal{P}$ and elements $e, e' \in \Delta^{\mathfrak{I}^{\mathbb{P}}}$ such that $(e, e') \in \mathrm{R}^{\mathfrak{I}^{\mathbb{P}}}$, $\lambda_{w\cdot(\mathbb{P},h)}(e) = u$ and $\lambda_{w\cdot(\mathbb{P},h)}(e') = v$. By

Lemma 8 we obtain $e' \in \mathrm{A_1}^{\mathfrak{I}^{\mathbb{P}}}$. Since $\mathfrak{I}^{\mathbb{P}}$ is $\mathcal{T}$-saturated, we have $e \in \mathrm{A_2}^{\mathfrak{I}^{\mathbb{P}}}$. Therefore by definition of $\mathrm{A_2}^{\mathcal{I}}$ we obtain $u \in \mathrm{A_2}^{\mathcal{I}}$.

$\mathrm{A} \sqsubseteq \neg\mathrm{B}$. By contradiction, assume there exists $u \in \mathrm{A}^{\mathcal{I}} \cap \mathrm{B}^{\mathcal{I}}$. By definition of $\Delta^{\mathcal{I}}$, there exist $w \cdot (\mathbb{P}, h) \in \mathcal{P}$ and an element $e \in \Delta^{\mathfrak{I}^{\mathbb{P}}}$ such that $\lambda_{w\cdot(\mathbb{P},h)}(e) = u$. Lemma 8 applied on both concepts A and B ensures $e \in (\mathrm{A} \sqcap \mathrm{B})^{\mathfrak{I}^{\mathbb{P}}}$, contradicting $\mathfrak{I}^{\mathbb{P}}$ being $\mathcal{T}$-satisfiable.

$\neg\mathrm{B} \sqsubseteq \mathrm{A}$. Let $u \in \neg\mathrm{B}^{\mathcal{I}}$. By definition of $\Delta^{\mathcal{I}}$, there exist $w \cdot (\mathbb{P}, h) \in \mathcal{P}$ and an element $e \in \Delta^{\mathfrak{I}^{\mathbb{P}}}$ such that $\lambda_{w\cdot(\mathbb{P},h)}(e) = u$. Since $u \notin \mathrm{B}^{\mathcal{I}'}$, we have $e \notin \mathrm{B}^{\mathfrak{I}^{\mathbb{P}}}$. Since $\mathfrak{I}^{\mathbb{P}}$ is $\mathcal{T}$-saturated, it gives $e \in \mathrm{A}^{\mathfrak{I}^{\mathbb{P}}}$, yielding by definition $u \in \mathrm{A}^{\mathcal{I}}$.

$\mathrm{P} \sqsubseteq \mathrm{R}$. Let $(u, v) \in \mathrm{P}^{\mathcal{I}}$. By definition of $\mathrm{P}^{\mathcal{I}}$, there exist $w \cdot (\mathbb{P}, h) \in \mathcal{P}$ and elements $e, e' \in \Delta^{\mathfrak{I}^{\mathbb{P}}}$ such that $(e, e') \in \mathrm{P}^{\mathfrak{I}^{\mathbb{P}}}$, $\lambda_{w\cdot(\mathbb{P},h)}(e) = u$ and $\lambda_{w\cdot(\mathbb{P},h)}(e') = v$. Since $\mathfrak{I}^{\mathbb{P}}$ is $\mathcal{T}$-saturated, we have $(e, e') \in \mathrm{R}^{\mathfrak{I}^{\mathbb{P}}}$. Therefore by definition of $\mathrm{R}^{\mathcal{I}}$ we obtain $(u, v) \in \mathrm{R}^{\mathcal{I}}$.

$\mathrm{R_1} \sqcap \mathrm{R_2} \sqsubseteq \bot$. Let $(u, v) \in \mathrm{R_1} \sqcap \mathrm{R_2}^{\mathcal{I}}$. By definition of $\mathrm{R}_1^{\mathcal{I}}$, there exist $w_1 \cdot (\mathbb{P}_1, h_1) \in \mathcal{P}$ and elements $d_1, e_1 \in \Delta^{\mathfrak{I}^{\mathbb{P}_1}}$ such that $(d_1, e_1) \in \mathrm{R}_1^{\mathfrak{I}^{\mathbb{P}_1}}$, $\lambda_{w_1\cdot(\mathbb{P}_1,h_1)}(d_1) = u$ and $\lambda_{w_1\cdot(\mathbb{P}_1,h_1)}(e_1) = v$. Similarly, by definition of $\mathrm{R}_2^{\mathcal{I}}$, there exist a pattern $w_2 \cdot (\mathbb{P}_2, h_2)$ and elements $d_2, e_2 \in \Delta^{\mathfrak{I}^{\mathbb{P}_2}}$ such that $(d_2, e_2) \in \mathrm{R}_2^{\mathfrak{I}^{\mathbb{P}_2}}$, $\lambda_{w_2\cdot(\mathbb{P}_2,h_2)}(d_2) = u$ and $\lambda_{w_2\cdot(\mathbb{P}_2,h_2)}(e_2) = v$. In particular we have $\lambda_{w_1\cdot(\mathbb{P}_1,h_1)}(d_1) = \lambda_{w_2\cdot(\mathbb{P}_2,h_2)}(d_2)$ and $\lambda_{w_1\cdot(\mathbb{P}_1,h_1)}(e_1) = \lambda_{w_2\cdot(\mathbb{P}_2,h_2)}(e_2)$. By Lemma 9, we can add (d_1, e_1) to $\mathrm{R}_2^{\mathfrak{I}^{\mathbb{P}_1}}$ while retaining $\mathcal{T}$-satisfiability, contradicting the fact that $\mathcal{T}$ contains $\mathrm{R_1} \sqcap \mathrm{R_2} \sqsubseteq \bot$. □

It remains to verify that there are no additional c-matches for q in $\mathcal{I}$, that is, no more than encoded in $\mathfrak{M}^*$. The inherited tree-like structure of $\mathcal{I}$, along with the specifications having to be preserved between linked patterns, ensures that if a match $\pi : q \to \mathcal{I}$ exists, then it is actually already taken into account in the specification of the patterns from $\mathcal{P}$. Therefore, if a match maps a counting variable z onto an element of shape $w \cdot (\mathbb{P}, h)$ in $\mathcal{I}$, we shall ensure that $(q, z \mapsto s)$, with s either ⊚ or ⊛, belongs to $\mathfrak{M}^{\mathbb{P}}$. This would contradict $\mathbb{P}$ being accepting. The exact (stronger) statement is as follows.

Lemma 11. *If $\pi : r \to \mathcal{I}$ is a match of $r \subseteq q$, then for all $w \cdot (\mathbb{P}, h) \in \mathcal{P}$, we have $(r, \pi') \in \mathfrak{M}^{\mathbb{P}}$ where $\pi' := (\lambda_{w\cdot(\mathbb{P},h)})^{-1} \circ \pi_{|\Delta}$ with $\Delta := \pi^{-1}(\lambda_{w\cdot(\mathbb{P},h)}(\Delta^{\mathfrak{I}^{\mathbb{P}}}))$.*

Proof sketch. The full proof can be found in the appendix and proceeds by a breadth-first induction on the pattern tree $\mathcal{P}$ to verify the statement holds for matches that map in the intermediate interpretation $\mathcal{I}_W$ obtained by piecing patterns until W, that is restricting the union in the definition of $\mathcal{I}$ to those elements $w \cdot (\mathbb{P}, h) \in \mathcal{P}$

that are at smaller or equal to W w.r.t. a breadth-first ordering on $\mathcal{P}$. This is sufficient to prove our statement since any match mapping to the full model $\mathcal{I}$ only requires a finite set of facts $\phi_1, \ldots \phi_k$, coming from the images of interpretations $\mathfrak{I}^{\mathbb{P}_1}, \ldots \mathfrak{I}^{\mathbb{P}_k}$ by some duplicating functions $\lambda_{w_1 \cdot (\mathbb{P}_1, h_1)}, \ldots \lambda_{w_k \cdot (\mathbb{P}_k, h_k)}$. Picking W as the greatest such node $w_i \cdot (\mathbb{P}_i, h_i)$, with $i = 1, \ldots k$, w.r.t. the breadth-first ordering on $\mathcal{P}$, we obtain that our match of interest is already a match in $\mathcal{I}_W$.

At such a step W, we prove the desired property on the specification holds for all patterns $\mathbb{P}$ occurring in some node $w \cdot (\mathbb{P}, h)$ introduced prior to W. This is also achieved by induction, but on the distance, in the pattern tree $\mathcal{P}$, between W and the node $w \cdot (\mathbb{P}, h)$. □

The two latter lemmas yield the following, concluding the proof of Lemma 6.

Proposition 2. *$\mathcal{I}$ is a model of $\mathcal{K}$ whose c-matches are included in those encoded in $\mathfrak{M}^*$.*

Proof. Modelhood follows directly from Lemma 10, itself based on Lemmas 8 and 9, while the number of c-matches is handled by Lemma 11. □

3.3.3 Completeness: from models to patterns

We now turn to the proof of Lemma 7. We fix a model $\mathcal{I}$ of $\mathcal{K}$, and our task is to construct an accepting initial pattern having the same number of c-matches as $\mathcal{I}$.

Let Δ^* be the subset of $\Delta^{\mathcal{I}}$ consisting of all individuals in $\mathcal{A}$ and all elements e such that $e = \pi(z)$ for some $\pi : q \to \mathcal{I}$ and counting variable z. Set $\mathcal{I}^* := \mathcal{I}_{|\Delta^*}$ and $\mathfrak{M}^* := (\mathfrak{M}^{\mathcal{I}})_{|\Delta^*}$. Notice in particular that the number of c-matches for q encoded in $\mathfrak{M}^*$ is exactly the number of c-matches for q in $\mathcal{I}$. We claim that $\mathbb{P}^* := (\emptyset, \Delta^*, \mathcal{I}^*, \mathfrak{M}^*, \mathsf{Id})$ is accepting.

To prove this, we shall build a set of patterns, whose every pattern $\mathbb{P}$ is *not trivially rejecting*, i.e. $\mathbb{P}$ does not satisfy the base-case condition of a rejecting pattern, and which is *realized in* $\mathcal{I}$, meaning that $\mathfrak{I}^{\mathbb{P}}$ homomorphically embeds into $\mathcal{I}$. Observe that the initial pattern $\mathbb{P}^*$ satisfies both conditions. To pursue the construction, given any pattern $\mathbb{P}$ satisfying the two conditions and a head h applicable to $\mathbb{P}$, we show how to extract from $\mathcal{I}$ another $\mathbb{Q}$ which satisfies the conditions and which makes h hold for $\mathbb{P}$. Since the number of patterns is finite, every sequence of patterns constructed in such a manner either leads to a trivially accepting pattern (i.e. one with no applicable heads) or loops back to an already explored pattern satisfying the conditions. It follows that all patterns in the set are accepting (in particular, $\mathbb{P}^*$).

To formalize the construction, we shall introduce along with each pattern $\mathbb{P}$ a function τ being a homomorphism $\mathfrak{I}^{\mathbb{P}} \to \mathcal{I}$. Recall that for every R.A $\in \Omega$, we have chosen a function $\mathsf{succ}^{\mathcal{I}}_{\mathrm{R.A}}$ that maps every element $e \in (\exists \mathrm{R.A})^{\mathcal{I}}$ to an element $e' \in \Delta^{\mathcal{I}}$ such that $(e, e') \in \mathrm{R}^{\mathcal{I}}$ and $e' \in \mathrm{A}^{\mathcal{I}}$.

Definition 33. *Base case: the construction begins with the pair* $(\mathbb{P}^*, \mathsf{Id}_{\mathcal{I}^* \to \mathcal{I}})$, *where* $\mathsf{Id}_{\mathcal{I}^* \to \mathcal{I}}$ *denotes the identity function.*

Induction case: consider some already constructed pair $(\mathbb{P}_1, \tau_1)$, *and a head* R.B *that is applicable to* e_1 *in* $\mathbb{P}_1$. *Since* R.B *applies to* e, *there must exist* $\mathrm{A} \in \mathsf{N_C}$ *such that* $e \in \mathrm{A}^{\mathfrak{I}^{\mathbb{P}_1}}$ *and* $\mathcal{T} \models \mathrm{A} \sqsubseteq \exists \mathrm{R.B}$. *Set* $e'_1 := \tau_1(e_1)$. *Since* τ_1 *is a homomorphism and* $\mathcal{I}$ *is a model of* $\mathcal{T}$, *we obtain* $e'_1 \in (\exists \mathrm{R.B})^{\mathcal{I}}$ *and can set* $e'_2 := \mathsf{succ}^{\mathcal{I}}_{\mathrm{R.B}}(e'_1)$. *If* $e'_2 \in \Delta^*$, *then we set* $e_2 := e'_2$, *otherwise we set* e_2 *to either* $\circledcirc$ *or* $\circledast$ *such that* $e_1 \neq e_2$.

We now define a new pattern $\mathbb{P}_2$. *Its frontier is* e_1 *and its generated term is* e_2. *Its interpretation is given by:*

$$\begin{array}{rcl} \mathrm{C}^{\mathfrak{I}^{\mathbb{P}_2}} & := & \mathrm{C}^{\mathcal{I}^*} \cup \{e_k \mid e'_k \in \mathrm{C}^{\mathcal{I}}, k = 1, 2\} \\ \mathrm{P}^{\mathfrak{I}^{\mathbb{P}_2}} & := & \mathrm{P}^{\mathcal{I}^*} \cup \{(e_1, e_2) \mid \mathcal{T} \models \mathrm{R} \sqsubseteq \mathrm{P}\} \\ & & \cup \{(e_2, e_1) \mid \mathcal{T} \models \mathrm{R}^- \sqsubseteq \mathrm{P}\} \end{array}$$

Its specification is $(\mathfrak{M}^{\mathcal{I}})_{|\Delta^* \cup \{e'_1, e'_2\}}$ *in which* e'_1 *(resp.* e'_2*) has been replaced by* e_1 *(resp.* e_2*). Its prediction maps a head* h *to the value of* $\mathsf{succ}^{\mathcal{I}}_h(e'_2)$ *if it is defined, else to* h. *Finally, we let* τ_2 *be the function that maps elements of* Δ^* *to themselves,* e_1 *to* e'_1 *and* e_2 *to* e'_2. *We obtain a new pair* $(\mathbb{P}_2, \tau_2)$.

Example 14. *In the model* $\mathcal{I}_e$ *from Example 8, depicted in Figure 3.6, we can set* $\mathsf{succ}^{\mathcal{I}_e}_{\mathrm{R.A'}}(\mathsf{a}) := \alpha_1$ *(other choices of successors are unique), and then apply the preceding construction to obtain the accepting patterns from Example 12. Figure 3.12 illustrates where these patterns are realized in* $\mathcal{I}_e$. *A same pattern can be realized several times, e.g.* $\mathbb{P}^e_2$. *Patterns* $\mathbb{P}^e_{10}$ *and* $\mathbb{P}^e_{11}$ *illustrate how a loop in the original model unfolds as two patterns. In their specifications, notice the partial matches* $(y_2, z) \mapsto (\circledcirc, \circledcirc)$ *and* $(y_2, z) \mapsto (\circledast, \circledast)$ *witnessing this loop that can not be retrieved in their respective interpretations.*

Recalling that $\mathcal{I}$ is a model, of $\mathcal{K}$ it is then straightforward to verify that $\mathbb{P}_2$ is a well-defined not-trivially-rejecting pattern, satisfying $\mathbb{P}_2 \in \mathbb{L}^{\mathrm{R.B}}_{\mathbb{P}_1, e_1}$, and that τ_2 is indeed a homomorphism. These properties are verified by the next two lemmas.

Lemma 12. *Each pair* $(\mathbb{P}, \tau)$ *built according to Definition 33 yields consists of a well-defined and non-trivially rejecting pattern* $\mathbb{P}$ *and a homomorphism* $\tau : \mathfrak{I}^{\mathbb{P}} \to \mathcal{I}$.

Proof. The base case consisting of verifying that $\mathbb{P}^*$ is non-trivially rejecting is trivial, and $\mathsf{Id}_{\mathcal{I}^* \to \mathcal{I}}$ is indeed a homomorphism.

We move to the induction case: assume $(\mathbb{P}_1, \tau_1)$ is obtained by the described procedure with $\mathbb{P}_1$ a well-defined and non-trivially rejecting pattern and $\tau_1 : \mathfrak{I}^{\mathbb{P}_1} \to \mathcal{I}$

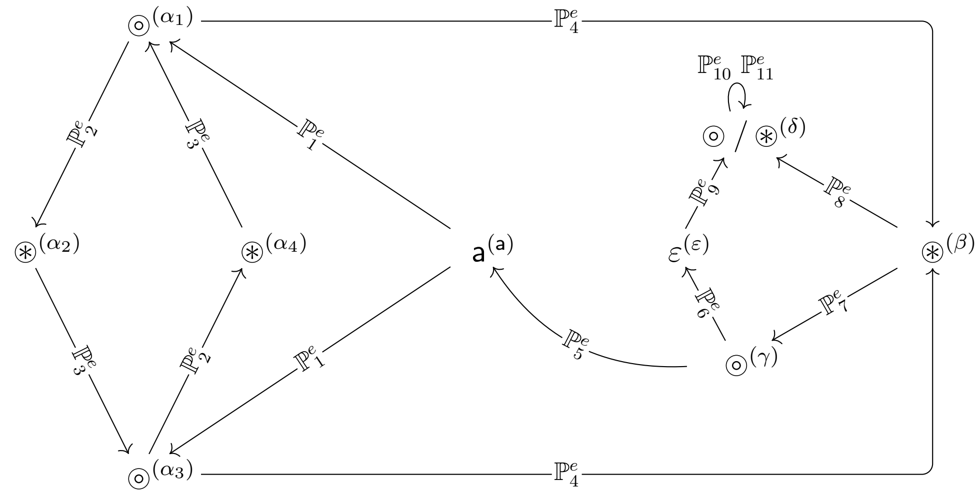


Figure 3.12: Patterns from Example 12 as realized in $\mathcal{I}_e$. An edge $(u^{(a)}, v^b)$ with label p indicates there exists a built pair (p, τ) such that $\tau(u) = a$ and $\tau(v) = b$.

a homomorphism. Let R.B an existential head that applies on e_1 in $\mathbb{P}_1$ and consider the pair $(\mathbb{P}_2, \tau_2)$ obtained by applying Definition 33 on $(\mathbb{P}_1, \tau_1)$ for this head.

We first verify that τ_2 is a homomorphism:

- Let $u \in \mathrm{A}^{\mathfrak{I}^{\mathbb{P}_2}}$. If $u \in \mathrm{A}^{\mathcal{I}^*}$, then in particular $u \in \Delta^*$ hence $\tau_2(u) = u \in \mathrm{A}^{\mathcal{I}^*} \subseteq \mathrm{A}^{\mathcal{I}}$. Otherwise, $u = e_k$ for $k = 1$ or $k = 2$ with $e'_k \in \mathrm{A}^{\mathcal{I}}$. In that case, notice $\tau_2(u) = e'_k$ which concludes.
- Let $(u, v) \in \mathrm{P}^{\mathfrak{I}^{\mathbb{P}_2}}$. If $(u, v) \in \mathrm{P}^{\mathcal{I}^*}$, then in particular $u, v \in \Delta^*$, hence $(\tau_2(u), \tau_2(v)) = (u, v) \in \mathrm{P}^{\mathcal{I}^*} \subseteq \mathrm{P}^{\mathcal{I}}$. Otherwise, if $(u, v) = (e_1, e_2)$ with $\mathcal{T} \models \mathrm{R} \sqsubseteq \mathrm{P}$, then notice that $(\tau_2(u), \tau_2(v)) = (e'_1, e'_2)$. Since e'_2 is the successor of e'_1 for R.B in $\mathcal{I}$, and $\mathcal{I}$ models $\mathcal{T}$, we obtain $(e'_1, e'_2) \in \mathrm{P}^{\mathcal{I}}$ as desired. Otherwise we have $(u, v) = (e_2, e_1)$ with $\mathcal{T} \models \mathrm{R}^- \sqsubseteq \mathrm{P}$, then notice that $(\tau_2(u), \tau_2(v)) = (e'_1, e'_1)$. Since e'_2 is the successor of e'_1 for R.B in $\mathcal{I}$, and $\mathcal{I}$ models $\mathcal{T}$, we have $(e'_2, e'_1) \in \mathrm{P}^{\mathcal{I}}$ as desired.

We now verify that $\mathbb{P}_2$ is a well-defined pattern.

- The frontier e_1 and the generated term e_2 of $\mathbb{P}_2$ are elements from $\Delta^* \cup \{\circledcirc, \circledast\}$.
- The interpretation $\mathfrak{I}^{\mathbb{P}_2}$ is $\mathcal{T}$-satisfiable as it embeds by τ_2 in $\mathcal{I}$ being a model of $\mathcal{T}$. It is $\mathcal{T}$-saturated since concepts and roles on $\mathcal{I}^*$ are fully preserved as they come from the model $\mathcal{I}$. The additional concepts on e_1 and e_2 are also all preserved from those on e'_1 and e'_2. The additional roles between e_1 and e_2 are all defined as induced by $\mathrm{R}(e_1, e_2)$ which ensures this edge is also

saturated. Finally, it indeed preserves $\mathcal{I}^*$: this is trivial for concepts, and for roles it suffices to verify that e'_1 and e'_2 cannot be both elements of Δ^* (hence no new role fact). Since $e'_1, e'_2 \in \Delta^*$ would contradict R.B being applicable on $e_1 = e'_1$ since $e_2 = e'_2$ is the R.B successor in $\mathcal{I}$, this case is indeed excluded.

- Restrictions of induced specifications are coherent, hence $\mathfrak{M}^{\mathbb{P}_2}$ is indeed coherent. It is a technicality to verify that $((\mathfrak{M}^{\mathcal{I}})_{|\Delta^* \cup (\tau_2)^{-1}(\{e'_1,e'_2\})})_{|\Delta^*} = (\mathfrak{M}^{\mathcal{I}})_{|\Delta^*}$, which proves $(\mathfrak{M}^{\mathbb{P}_2})_{|\Delta^*} = \mathfrak{M}^*$.
- Let $\mathrm{R}_1.\mathrm{B}_1$ and $\mathrm{R}_2.\mathrm{B}_2$ be two heads such that $\mathcal{T} \models \mathrm{R}_1 \sqcap \mathrm{R}_2 \sqsubseteq \bot$. By definition of next_2, if it maps $\mathrm{R}_1.\mathrm{B}_1$ and $\mathrm{R}_2.\mathrm{B}_2$ to the same element, then the successors of e'_1 for these two heads in $\mathcal{I}$ are equal, contradicting $\mathcal{I}$ being a model.

The fact that $\mathbb{P}_2$ is not trivially rejecting is immediate as its specification is a restriction of the induced specification of $\mathcal{I}$, which doesn't contain pairs (q, π) with π mapping outside Δ^* (that is precisely the definition of Δ^*). □

Lemma 13. *In the induction step of Definition 33, we have* $\mathbb{P}_2 \in \mathbb{L}^{\mathrm{R.B}}_{\mathbb{P}_1,e_1}$.

Proof. We verify $\mathbb{P}_2 \in \mathbb{L}^{\mathrm{R.B}}_{\mathbb{P}_1,e_1}$ by checking each condition from Definition 30.

1. $\mathfrak{fr}^{\mathbb{P}_2} = \{e_1\}$ and $\mathfrak{gen}^{\mathbb{P}_2} = \{e_2\}$ are indeed singletons.
2. $\mathbb{P}_1$ can either be the initial pattern or a non-initial one. In both cases, the concepts satisfied on e_1 in $\mathbb{P}_1$ are inherited from those on e'_1. Since it is also the case for $\mathbb{P}_2$, this condition holds.
3. $\mathbb{P}_1$ can either be the initial pattern or non-initial one. In both cases, the specification is the induced specification of $\mathcal{I}$ restricted to the domain of $\mathfrak{I}^{\mathbb{P}_1}$. We directly have the desired equality as both e_1 (seen in $\mathbb{P}_1$ and $\mathbb{P}_2$) comes from the *same* e'_1.
4. This condition matches the definition of $\mathfrak{I}^{\mathbb{P}_2}$.
5. A violation of this condition would imply that e'_1 is the successor of e'_2 for a head incompatible with h, which would contradict $\mathcal{I}$ being a model.
6. This follows from the *fixed* choice of successors in $\mathcal{I}$. □

3.4 Countermodels with bounded size

In this section, we prove that starting from a model with a minimum number of counting matches, we can construct such an optimal model whose size is polynomial w.r.t. data complexity, double-exponential w.r.t. combined complexity, and single-exponential if the size of the CCQ q is fixed (see further Theorem 8). Although this result doesn't allow us to improve upon our previous 2EXP algorithm for answering CCQ over $\mathcal{ALCHI}$ KBs w.r.t. combined complexity, it does immediately yield the following result for data complexity:

Theorem 7. *CCQ answering in $\mathcal{ALCHI}$ is in* coNP *w.r.t. data complexity.*

We further refine the construction in the case of DL-Lite$_{\mathsf{core}}$ to obtain an optimal model with exponential size, which yields a coNEXP procedure w.r.t. combined complexity.

3.4.1 Equivalence relation based on neighbourhoods

To obtain optimal models of bounded size, we start from the f^*-interlacing $\mathcal{I}'$ of an optimal model $\mathcal{I}$. It remains to merge elements of $\mathcal{I}'$ to obtain a model of the required size. To identify similar elements, we define a notion of neighbourhood.

Definition 34. *Consider an interpretation $\mathcal{M}$ and an element $c \in \Delta^{\mathcal{M}}$. Its n-neighbourhood $\mathcal{N}_n^{\mathcal{M},\Delta}(c)$ w.r.t. a subdomain $\Delta \subseteq \Delta^{\mathcal{M}}$ is defined inductively as:*

$$\begin{aligned} \mathcal{N}_0^{\mathcal{M},\Delta}(c) &:= \{c\} \\ \mathcal{N}_{n+1}^{\mathcal{M},\Delta}(c) &:= \mathcal{N}_n^{\mathcal{M},\Delta}(c) \cup \left\{ e \;\middle|\; \begin{array}{l} \exists d \in \mathcal{N}_n^{\mathcal{M},\Delta}(c) \setminus \Delta, \\ \exists \mathrm{R} \in \mathsf{N}_{\mathsf{R}}^{\pm},\ (d,e) \in \mathrm{R}^{\mathcal{M}} \end{array} \right\} \end{aligned}$$

Observe that we stop adding successors when we reach Δ. In particular, for $c \in \Delta$, we have $\mathcal{N}_n^{\mathcal{M},\Delta}(c) = \{c\}$ for every value of n. It follows that the statement '$c_1 \in \mathcal{N}_n^{\mathcal{M},\Delta}(c_2)$ iff $c_2 \in \mathcal{N}_n^{\mathcal{M},\Delta}(c_1)$' does not hold in general.

Recall that the definition of $\Delta^{\mathcal{I}'}$ ensures that any $c \in \Delta^{\mathcal{I}'} \setminus \Delta^*$ is actually an element of Δ° and therefore we have $c = \mathsf{a}w$ for some individual name a and word $w \in \Omega^*$. The tree-shaped structure of Δ° ensures that for all n, there exists a unique prefix $r_{n,c}$ of $\mathsf{a}w$ such that *(i)* $f^*(r_{n,c}) \in \mathcal{N}_n^{\mathcal{I}',\Delta^*}(c)$ and *(ii)* for any $d \in \mathcal{N}_n^{\mathcal{I}',\Delta^*}(c)$, there exists a unique word $w_{n,c}^d$ such that $d = f^*(r_{n,c} \cdot w_{n,c}^d)$.

This leads us to characterize the n-neighbourhood of an element $c \in \mathcal{I}'$ via the following function $\chi_{n,c}$, whose domain Ω_n is the set of words over Ω with length $\leq 2n$. Notice that, departing from Kostylev and Reutter [2015], we keep track of *sets* of satisfied concepts, in order to handle conjunctions of concepts in the left-hand sides of axioms.

$$\begin{aligned} &\chi_{n,c} : \Omega_n \to \Delta^* \cup 2^{\mathsf{sig}(\mathcal{T})} \cup \{\emptyset\} \\ &w \mapsto \begin{cases} \emptyset & \text{if } f^*(r_{n,c} \cdot w) \text{ undefined} \\ f^*(r_{n,c} \cdot w) & \text{if } f^*(r_{n,c} \cdot w) \in \Delta^* \\ \{\mathrm{A} \in \mathsf{sig}(\mathcal{T}) \mid f^*(r_{n,c} \cdot w) \in \mathrm{A}^{\mathcal{I}}\} & \text{otherwise} \end{cases} \end{aligned}$$

We can now introduce the equivalence relation we use to merge elements:

Definition 35. *The equivalence relation* $\sim_n$ *on* $\Delta^{\mathcal{I}'}$ *is defined as follows: an element* $e \in \Delta^*$ *is* $\sim_n$*-equivalent only to itself; elements* c_1, c_2 *from* $\Delta^{\mathcal{I}'} \setminus \Delta^*$ *are* $\sim_n$*-equivalent iff* $w^{c_1}_{n,c_1} = w^{c_2}_{n,c_2}$*,* $\chi_{n,c_1} = \chi_{n,c_2}$*, and* $|c_1| = |c_2| \bmod 2|q| + 3$.

We obtain a finite model of the required size by merging elements with respect to $\sim_{|q|+1}$.

Theorem 8. *The interpretation* $\mathcal{J} := \mathcal{I}'/{\sim_{|q|+1}}$ *is a model of* $\mathcal{K}$ *that has at most as many c-matches for* q *as* $\mathcal{I}$*. Its size is polynomial w.r.t. data complexity, double-exponential w.r.t. combined complexity, and single-exponential if the size of the CCQ* q *is fixed.*

Proof sketch. The key to proving that the number of c-matches does not increase as a result of the quotient operation is to exhibit suitable local homomorphisms. Indeed, a match of q in $\mathcal{J}$ maps each connected component C of q into a $|q|$-neighbourhood $\mathcal{N}^{\mathcal{J},\overline{\Delta^*}}_{|q|}(\overline{c})$, where $\overline{c}$ denotes the equivalence class of c w.r.t. $\sim_{|q|+1}$ and $\overline{\Delta^*}$ stands for the set $\{\overline{e} \mid e \in \Delta^*\}$. By exhibiting a homomorphism $\rho_c : \mathcal{N}^{\mathcal{J},\overline{\Delta^*}}_{|q|}(\overline{c}) \to \mathcal{N}^{\mathcal{I}',\Delta^*}_{|q|}(c)$ such that $\rho^{-1}_{n,c}(\overline{\Delta^*}) \subseteq \overline{\Delta^*}$, we can find a match of C in $\mathcal{I}'$. Such matches for q's connected components together form a match of the full q in $\mathcal{I}'$. It is mostly straightforward to show that $\mathcal{J}$ is a model, except for negative role inclusions, where the homomorphisms ρ_c are needed to move violations of $\mathrm{R}_1 \sqcap \mathrm{R}_2 \sqsubseteq \bot$ in $\mathcal{J}$ back into $\mathcal{I}'$. The claimed upper bounds are obtained by analyzing the size of $\mathcal{J}$ (i.e. counting the equivalence classes in $\Delta^{\mathcal{J}}$), keeping in mind that due to Lemma 3, we may assume that $|\Delta^*| \leq |\mathsf{Ind}(\mathcal{A})| + |q|\,(|\mathsf{Ind}(\mathcal{A})| + 3\,|\mathcal{T}|\,2^{|\mathcal{T}|})^{|q|}$. □

Example 15. *To illustrate this construction, consider the ABox* $\mathcal{A}_e = \{\mathrm{A}(\mathsf{a}), \mathrm{B}(\mathsf{b})\}$ *and the* $\mathcal{ELHI}_\bot$ *TBox* $\mathcal{T}_e$*:*

$$\begin{array}{ccccc} \mathrm{A} \sqsubseteq \exists \mathrm{P}.\mathrm{A}' & \mathrm{B} \sqsubseteq \exists \mathrm{Q}.\mathrm{B}' & \mathrm{A}' \sqcap \mathrm{B}' \sqsubseteq \mathrm{A}_0 & \mathrm{A}' \sqsubseteq \mathrm{D} & \mathrm{B}' \sqsubseteq \mathrm{D} \\ \mathrm{A}_0 \sqsubseteq \exists \mathrm{R}_1.\mathrm{A}_1 & \mathrm{A}_1 \sqsubseteq \exists \mathrm{R}_2.\mathrm{A}_2 & \mathrm{A}_2 \sqsubseteq \exists \mathrm{R}_3.\mathrm{A}_3 & \mathrm{A}_3 \sqsubseteq \exists \mathrm{S}.\mathrm{B}_0 & \mathrm{B}_0 \sqsubseteq \exists \mathrm{V}.\mathrm{B}'_0 \\ \mathrm{B}_0 \sqsubseteq \exists \mathrm{U}.\mathrm{C}_0 & \mathrm{U} \sqsubseteq \mathrm{V} & \mathrm{C}_0 \sqsubseteq \exists \mathrm{V}_1.\mathrm{C}_1 & \mathrm{C}_1 \sqsubseteq \exists \mathrm{V}_2.\mathrm{C}_2 & \mathrm{C}_2 \sqsubseteq \exists \mathrm{V}_3.\mathrm{D} \end{array}$$

Our example KB is $\mathcal{K}_e := (\mathcal{T}_e, \mathcal{A}_e)$*. Figure 3.13 depicts several model of* $\mathcal{K}_e$*. A countermodel* $\mathcal{I}_e$ *for the CCQ* $q_e := \exists z\, \mathrm{D}(z)$ *and integer* 3 *is depicted on Figure 3.13a.*

A part of the Id*-interlacing of* $\mathcal{I}_e$ *is depicted on Figure 3.13b (a tree-structure similar to the one following the* P*-edge issued from* a *also follows the* Q*-edge issued from* b*).*

The corresponding part of the f^**-interlacing of* $\mathcal{I}_e$ *is depicted on Figure 3.13c. Like the initial model* $\mathcal{I}_e$*, it is a countermodel for* q_e *and integer* 3*. Two neighbourhoods* $\mathcal{N}^{\mathcal{I}',\Delta^*}_2(\gamma)$ *and* $\mathcal{N}^{\mathcal{I}',\Delta^*}_2(\delta)$ *are depicted (in green, resp. blue). In particular, notice* $\mathsf{a} \notin \mathcal{N}^{\mathcal{I}',\Delta^*}_2(\delta)$ *since* $\alpha \in \Delta^*$*.*

Finally, the model $\mathcal{J}_e$ *obtained by merging elements of the* f^**-interlacing of* $\mathcal{I}_e$ *according to* $\sim_5$ *is depicted in Figure 3.13d, together with two* 2*-neighbourhoods*

$\mathcal{N}_2^{\mathcal{J}_e,\overline{\Delta^*}}(\overline{\gamma})$ and $\mathcal{N}_2^{\mathcal{J}_e,\overline{\Delta^*}}(\overline{\delta})$. Notice $\mathcal{J}_e$ remains a countermodel for q_e and candidate integer 3.

The remainder of this subsection is devoted to a proof of Theorem 8, that is, proving $\mathcal{J}$ is indeed a model and contains at most as many counting matches as $\mathcal{I}'$. Let us first formulate two remarks concerning the constructed interpretation $\mathcal{J}$.

Remark 13. *The set of concepts from $\mathsf{sig}(\mathcal{T})$ satisfied by $c \in \Delta^{\mathcal{I}'}$ is exactly $\chi_{n,c}(w^c_{n,c})$. Therefore, if $c \sim_n c'$, then c and c' satisfy the same concept names.*

Remark 14. *If $c \sim_n c'$, then $c \sim_m c'$ for any $m \leq n$.*

We now define homomorphisms ρ_c, mentioned in the proof sketch, inductively on $\mathcal{N}_k^{\mathcal{J},\overline{\Delta^*}}(\overline{c})$ with k increasing from 0 to $|q|$. Figure 3.14 summarizes the structures and mappings involved in the construction. Starting from the element $\overline{c} \in \mathcal{N}_0^{\mathcal{J},\overline{\Delta^*}}(\overline{c})$, we can naturally carry it back as $\rho_c(\overline{c}) = c \in \mathcal{N}_0^{\mathcal{I}',\overline{\Delta^*}}(c)$. Assume now that we have defined $\rho_c(\overline{d})$ for some $\overline{d} \in \mathcal{N}_n^{\mathcal{J},\overline{\Delta^*}}(\overline{c})$ and that we are moving further to an element $\overline{e} \in \mathcal{N}_{n+1}^{\mathcal{J},\overline{\Delta^*}}(\overline{c})$ along an edge $(\overline{d}, \overline{e})$ in $\mathcal{J}$. In the case of $\overline{e} \notin \overline{\Delta^*}$, the following lemma produces a candidate $\rho_c(\overline{e})$, namely e', which is to $\rho_c(\overline{d})$, namely d', what $\overline{e}$ is to $\overline{d}$.

Lemma 14. *Given two elements $\overline{d}, \overline{e} \in \Delta^{\mathcal{J}} \setminus \overline{\Delta^*}$, if there exists a role P from $\mathsf{N}_\mathsf{R}^\pm$ such that $(\overline{d}, \overline{e}) \in \mathrm{P}^{\mathcal{J}}$, then there exists a unique element $\mathrm{R.B} \in \Omega$ such that one of the two following conditions is satisfied:*

***edge*$^+$.** $|e| = |d| + 1 \mod 2|q| + 3$, $w^e_{|q|+1,e} = w^d_{|q|,d} \cdot \mathrm{R.B}$ *and* $\mathcal{T} \models \mathrm{R} \sqsubseteq \mathrm{P}$.

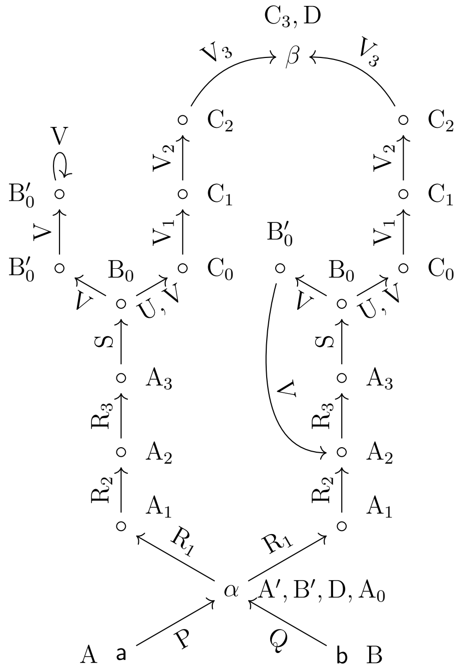
Furthermore, for all $d' \sim_k d$, the element $e' := d' \cdot \mathrm{R.B}$ belongs to $\Delta^{\mathcal{I}'}$ and satisfies $e' \sim_{k-1} e$.

***edge*$^-$.** $|d| = |e| + 1 \mod 2|q| + 3$, $w^d_{|q|+1,d} = w^e_{|q|,e} \cdot \mathrm{R.B}$ *and* $\mathcal{T} \models \mathrm{R}^- \sqsubseteq \mathrm{P}$.

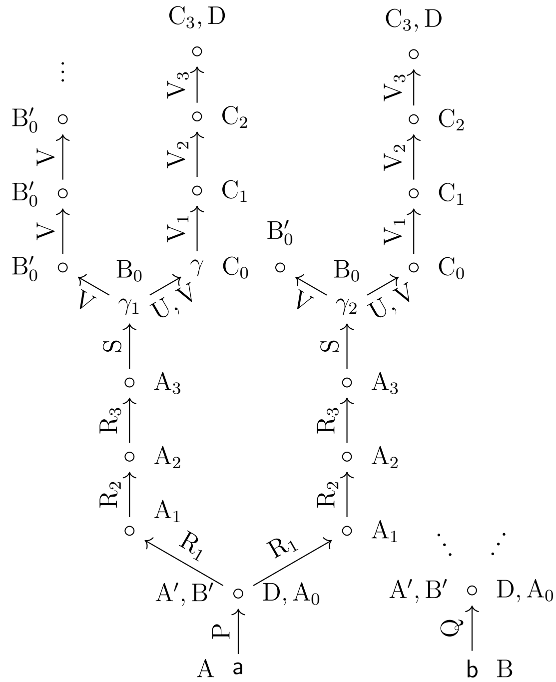
Furthermore, for all $d' \sim_k d$, we have e' such that $d' = e' \cdot \mathrm{R.B}$ and the prefix e' satisfies $e' \sim_{k-1} e$.

Proof. Notice the two conditions are mutually exclusive: $|e| = |d| + 1 \mod 2|q| + 3$ and $|d| = |e| + 1 \mod 2|q| + 3$ would imply $0 = 2 \mod 2|q| + 3$, which is impossible as $2|q| + 3 > 2$. Furthermore, in each case R.B is defined as the last letter of the word $w^e_{|q|+1,e}$ (resp $w^d_{|q|+1,d}$), which is unique and does not depends on the choice of e (resp d) nor on P. This proves the uniqueness.

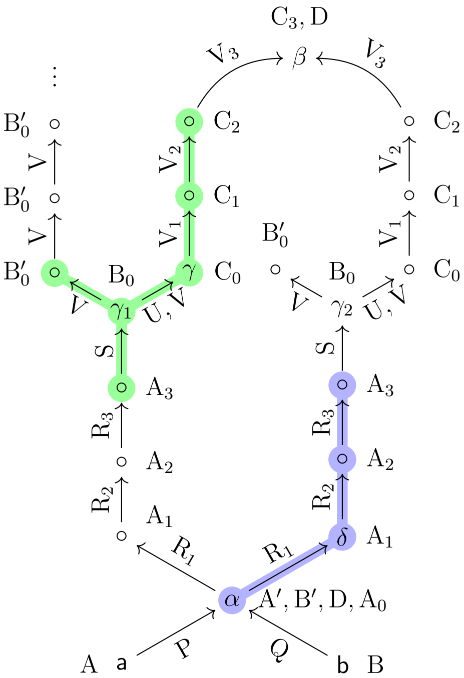
We now focus on the existence and the additional property. From the definition of $\mathrm{P}^{\mathcal{J}}$, there exist $(d_0, e_0) \in \mathrm{P}^{\mathcal{I}'}$ such that $\overline{d_0} = \overline{d}$ and $\overline{e_0} = \overline{e}$. Recall $\overline{d}, \overline{e} \notin \overline{\Delta^*}$, hence $d_0, e_0 \notin \Delta^*$. In that case the definition of f' ensures the only antecedent of d_0 (resp e_0) by f' is itself. Therefore the definition of $\mathrm{P}^{\mathcal{I}'}$, that is $\sigma(\mathrm{P}^{\mathcal{I}})$, yields two cases:



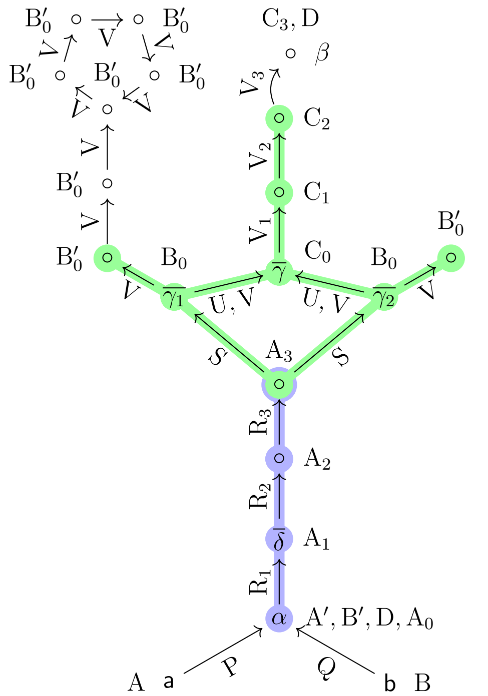
(a) Initial countermodel $\mathcal{I}_e$.



(b) Id-interlacing of $\mathcal{I}_e$.



(c) f^*-interlacing of $\mathcal{I}_e$.



(d) $\mathcal{J}_e$ from the f^*-interlacing of $\mathcal{I}_e$.

Figure 3.13: Models from Example 15.

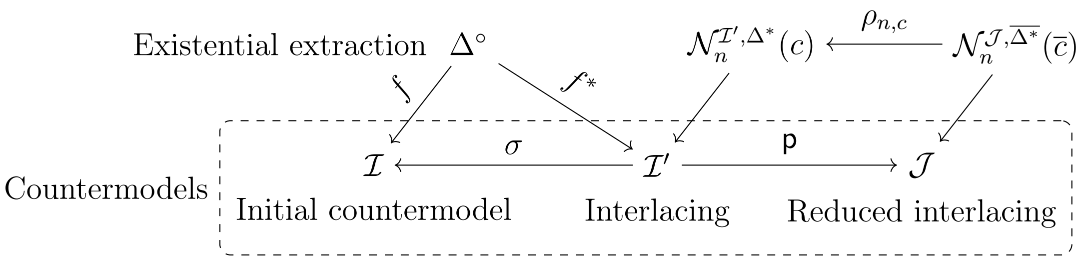


Figure 3.14: Models, domains, and mappings involved in Section 3.4.1.

- We have $e_0 = d_0 \cdot \text{R.B}$ with $\mathcal{T} \models \text{R} \sqsubseteq \text{P}$. It follows that $|e_0| = |d_0| + 1 \mod 2|q| + 3$ and $w^{e_0}_{|q|+1,e_0} = w^{d_0}_{|q|+1-1,d_0} \cdot \text{R.B}$, immediately yielding the same properties for d and e as $(\overline{d_0}, \overline{e_0}) = (\overline{d}, \overline{e})$.

  Let now $1 \leq k \leq |q| + 1$ be an integer and $d' \sim_k d$. Transitivity gives $d' \sim_k d_0$, and we have in particular $\chi_{k,d'} = \chi_{k,d_0}$ and $w^{d'}_{k,d'} = w^{d_0}_{k,d_0}$. Recall that $e_0 = d_0 \cdot \text{R.B}$, hence we have $\chi_{k,d_0}(w^{d_0}_{k,d_0} \cdot \text{R.B}) \neq \emptyset$, hence $\chi_{k,d'}(w^{d'}_{k,d'} \cdot \text{R.B}) \neq \emptyset$, that is $d' \cdot \text{R.B}$ is well-defined.

  Notice it is now sufficient to prove $d' \cdot \text{R.B} \sim_{k-1} e_0$: that is because $\overline{e} = \overline{e_0}$, hence transitivity will conclude the proof. It should be clear that $w^{d' \cdot \text{R.B}}_{k-1,d' \cdot \text{R.B}} = w^{e_0}_{k-1,e_0}$ and $|d' \cdot \text{R.B}| = |e_0| \mod 2|q| + 3$. Hence we are only left proving that $\chi_{k,e_0} = \chi_{k,d' \cdot \text{R.B}}$.

  First, $e_0 = d_0 \cdot \text{R.B}$ ensures that χ_{k,d_0} fully determines χ_{k-1,e_0}. Moreover, $\chi_{k,d'}$ fully determines $\chi_{k-1,d' \cdot \text{R.B}}$. But since $\chi_{k,d_0} = \chi_{k,d'}$ and $w^{e_0}_{k,d_0} = w^{d' \cdot \text{R.B}}_{k,d'}$, we obtain: $\chi_{k,e_0} = \chi_{k,d' \cdot \text{R.B}}$, concluding the proof.

- We have $d_0 = e_0 \cdot \text{R.B}$ with $\mathcal{T} \models \text{R}^- \sqsubseteq \text{P}$. It follows that $|d_0| = |e_0| + 1 \mod 2|q| + 3$ and $w^{d_0}_{|q|+1,d_0} = w^{e_0}_{|q|+1-1,e_0} \cdot \text{R.B}$, immediately yielding the same properties for d and e as $(\overline{d_0}, \overline{e_0}) = (\overline{d}, \overline{e})$.

  Let now $1 \leq k \leq |q| + 1 + 1$ be an integer and $d' \sim_k d$. Transitivity gives $d' \sim_k d_0$, and we have in particular $w^{d'}_{1,d'} = w^{d_0}_{1,d_0} = \text{R.B}$ (very important to have $k \geq 1$ here!). That is d' ends by R.B, and therefore we can indeed have prefix e' such that $d' = e' \cdot \text{R.B}$. □

Notice the "strength" of the equivalence relation $\sim_k$ between $\overline{e}$ and $\rho_c(\overline{e})$ decreases as we move further in the neighbourhood of $\overline{c}$. However, since we start from $\rho_c(\overline{c}) := c \sim_{|q|+1} c$ and explore a $|q|$-neighbourhood, the index remains at least 1. This is essential as $\sim_1$ encodes relations to elements of $\overline{\Delta^*}$ as the next lemma shows. It allows in particular to treat the case of $\overline{e} \in \overline{\Delta^*}$.

Lemma 15. *If $(\overline{d}, \overline{e}) \in \text{R}^{\mathcal{J}}$ for some $e \in \Delta^*$, and if $d' \sim_1 d$, then $(d', e) \in \text{R}^{\mathcal{I}'}$.*

Proof. Recall that since $e \in \Delta^*$ we have $\overline{e} = \{e\}$. The definition of $\mathrm{R}^{\mathcal{J}}$ and further of $\mathrm{R}^{\mathcal{I}'}$ provide $d_0, e_0 \in \Delta^\circ$ such that: $\overline{f^*(d_0)} = \overline{d}$, $f^*(e_0) = e$ and satisfying $(f^*(d_0), f^*(e_0)) \in \mathrm{R}^{\mathcal{I}'}$ from one of the following three cases:

- $(f^*(d_0), f^*(e_0)) \in \mathrm{R}^{\mathcal{I}^*}$. In particular $f^*(d_0) \in \Delta^*$, hence $f^*(d_0) = d = d'$. Therefore $(d', e) = (f^*(d_0), f^*(e_0)) \in \mathrm{R}^{\mathcal{I}'}$.
- $e_0 = d_0 \cdot \mathrm{P.B}$ with $\mathcal{T} \models \mathrm{P} \sqsubseteq \mathrm{R}$. If $f^*(d_0) \in \Delta^*$, then we again have $f^*(d_0) = d = d'$ immediately yielding $(d', e) \in \mathrm{R}^{\mathcal{I}'}$. Otherwise we have $\chi_{1,f^*(d_0)}(w_{1,f^*(d_0)}^{f^*(d_0))} \cdot \mathrm{P.B}) = f^*(e_0) = e$. But since $f^*(d_0) \sim_1 d \sim_1 d'$, we have $\chi_{1,d'} = \chi_{1,f^*(d_0)}$ and $w_{1,d'}^{d'} = w_{1,f^*(d_0)}^{f^*(d_0)}$. Therefore $e = \chi_{1,f^*(d_0)}(w_{1,f^*(d_0)}^{f^*(d_0)} \cdot \mathrm{P.B}) = \chi_{1,d'}(w_{1,d'}^{d'} \cdot \mathrm{P.B}) = f^*(r_{1,d'} \cdot w_{1,d'}^{d'} \cdot \mathrm{P.B})$. Recalling that $d' = f^*(r_{1,d'} \cdot w_{1,d'}^{d'})$, we hence obtain $(d', e) = (f^*(r_{1,d'} \cdot w_{1,d'}^{d'}), f^*(r_{1,d'} \cdot w_{1,d'}^{d'} \cdot \mathrm{P.B})) \in \mathrm{P}^{\mathcal{I}'} \subseteq \mathrm{R}^{\mathcal{I}'}$.
- $d_0 = e_0 \cdot \mathrm{P.B}$ with $\mathcal{T} \models \mathrm{P} \sqsubseteq \mathrm{R}^-$. If $f^*(d_0) \in \Delta^*$, then we again have $f^*(d_0) = d = d'$ immediately yielding $(d', e) \in \mathrm{R}^{\mathcal{I}'}$. Otherwise the 1-root of $f^*(d_0) = d_0$ is e_0 and $w_{1,d}^{d} = \mathrm{P.B}$. We thus have: $\chi_{1,f^*(d_0)}(\varepsilon) = f^*(e_0) = e$ (where ε denotes the empty word). But since $f^*(d_0) \sim_1 d \sim_1 d'$, we have $\chi_{1,d'} = \chi_{1,f^*(d_0)}$ and $w_{1,d'}^{d'} = w_{1,d}^{d}$. Combining the preceding facts, we obtain $(d', e) = (f^*(r_{1,d'} \cdot w_{1,d'}^{d'}), \chi_{1,d'}(\varepsilon)) = (f^*(r_{1,d'} \cdot \mathrm{P.B}), f^*(r_{1,d'})) \in (\mathrm{P}^-)^{\mathcal{I}'} \subseteq \mathrm{R}^{\mathcal{I}'}$. □

It remains to free ourselves from the particular choice of $\overline{d}$, which is likely not to be the only element of $\mathcal{N}_n^{\mathcal{J},\overline{\Delta^*}}(\overline{c})$ connected to $\overline{e}$. Taking a closer look at Lemma 14, we observe that $\rho_c(\overline{e})$, that is e', is obtained either by adding a letter to $\rho_c(\overline{d})$, that is d', or by removing the last letter of $\rho_c(\overline{d})$, and that these letters coincide with those in the suffixes of elements d and e. Therefore, when moving from $\overline{c}$ to $\overline{e}$ and ignoring self-cancelling steps, each added letter must appear in the suffix of e and, similarly, each removed letter must appear in the suffix of c.

The challenge is therefore to quantify the number of additions and removals to build $\rho_c(\overline{e})$ directly from c and $\overline{e}$. The next definition captures the relative difference of letters between $\overline{c}$ and $\overline{e}$, encoded in $|c|$ and $|e| \mod 2|q| + 3$.

Definition 36. *Let $\overline{c} \in \Delta^{\mathcal{J}}$ and $n \leq |q|$. The* relative depth *of $\overline{e} \in \mathcal{N}_n^{\mathcal{J},\overline{\Delta^*}}(\overline{c})$ from $\overline{c}$ is the integer $\delta_{\overline{c}}(\overline{e}) \in [-n, n]$ such that $|e| = |c| + \delta_{\overline{c}}(\overline{e}) \mod 2|q| + 3$.*

Remark 15. *By induction on $n \leq |q|$, it is straightforward to see that $\delta_{\overline{c}}(\overline{e})$ is well defined. Unicity is ensured by $\delta_{\overline{c}}(\overline{e}) \leq n \leq |q|$. A consequence of Lemma 14 is that for the smallest $n \leq |q|$ such that $\overline{e} \in \mathcal{N}_n^{\mathcal{J},\overline{\Delta^*}}(\overline{c})$ we have $\delta_{\overline{c}}(\overline{e}) = n \mod 2$.*

We can now identify how many additions and removals cancelled each other. Indeed, if it takes n steps to reach $\overline{e}$ from $\overline{c}$, with relative difference of $\delta := \delta_{\overline{c}}(\overline{e})$, then $n - |\delta|$ is the length of the self-cancelling path, hence: $\frac{n-|\delta|}{2}$ cancelled additions and $\frac{n-|\delta|}{2}$ cancelled removals. Therefore, the actual number of additions is $\frac{n-|\delta|}{2} + \delta$ if $\delta \geq 0$, or $\frac{n-|\delta|}{2}$ if $\delta \leq 0$, that is in both cases $\frac{n+\delta}{2}$. Similarly we obtain $\frac{n-\delta}{2}$ for the actual number of removals. The next theorem formalizes all these intuitions: $\rho_{n,c}(\overline{e})$ (in non-trivial cases) is obtained by removing the $\frac{n-\delta}{2}$ last letters of c and keeping the $\frac{n+\delta}{2}$ last letters from the suffix of e. It is then a technicality to verify these syntactical operations on words make sense in the domain of $\mathcal{I}'$.

Theorem 9. *For all $c \in \Delta^{\mathcal{I}'}$ and all $n \leq |q|$, the following mapping:*

$$\rho_{n,c}(\overline{e}) : \mathcal{N}_n^{\mathcal{J},\overline{\Delta^*}}(\overline{c}) \to \mathcal{N}_n^{\mathcal{I}',\Delta^*}(c) \qquad \overline{e} \mapsto \begin{cases} \rho_{n-1,c}(\overline{e}) & \textit{if } \overline{e} \in \mathcal{N}_{n-1}^{\mathcal{J},\overline{\Delta^*}}(\overline{c}) \\ e & \textit{if } \overline{e} \in \overline{\Delta^*} \\ r_{\frac{n-\delta_{\overline{c}}(\overline{e})}{2},c} \cdot w^e_{\frac{n+\delta_{\overline{c}}(\overline{e})}{2},e} & \textit{otherwise} \end{cases}$$

is a homomorphism satisfying $\rho_{n,c}(\overline{e}) \sim_{|q|+1-n} e$ and $\rho_{n,c}^{-1}(\overline{\Delta^*}) \subseteq \overline{\Delta^*}$.

Proof sketch. The full proof can be found in the appendix and proceeds by induction on the radius n of the considered neighbourhood. In the induction step, the two cases highlighted by Lemma 14 arise and allow us to verify each considered element in the definition of $\rho_{n,c}$ is indeed well defined as an element of $\mathcal{I}'$. □

Let us now complete the proof of Theorem 8 with Theorem 9 in hand.

Proof of Theorem 8.

Modelhood. We first prove that $\mathcal{J}$ is indeed a model by considering each possible shape of assertions and axioms:

- $\mathrm{A}(\mathsf{a})$. Since $\mathcal{I}'$ is a model, we have $\mathsf{a} \in \mathrm{A}^{\mathcal{I}'}$. Therefore, the definition of $\mathrm{A}^{\mathcal{J}}$ gives $\overline{\mathsf{a}} = \mathsf{a} \in \mathrm{A}^{\mathcal{J}}$.
- $\mathrm{P}(\mathsf{a}, \mathsf{b})$. Since $\mathcal{I}'$ is a model, we have $(\mathsf{a}, \mathsf{b}) \in \mathrm{P}^{\mathcal{I}'}$. Therefore, the definition of $\mathrm{P}^{\mathcal{J}}$ gives $(\overline{\mathsf{a}}, \overline{\mathsf{b}}) = (\mathsf{a}, \mathsf{b}) \in \mathrm{P}^{\mathcal{J}}$.
- $\top \sqsubseteq \mathrm{A}$. Let $u \in \top^{\mathcal{J}} = \Delta^{\mathcal{J}}$. By definition of $\Delta^{\mathcal{J}}$, there exists $u_0 \in \Delta^{\mathcal{I}'}$ such that $\overline{u_0} = u$. Since $u_0 \in \top^{\mathcal{I}'}$ and $\mathcal{I}'$ is a model, it ensures $u_0 \in \mathrm{A}^{\mathcal{I}'}$. Therefore the definition of $\mathrm{A}^{\mathcal{J}}$ gives $u = \overline{u_0} \in \mathrm{A}^{\mathcal{J}}$.
- $\mathrm{A}_1 \sqcap \mathrm{A}_2 \sqsubseteq \mathrm{A}$. Let $u \in (\mathrm{A}_1 \sqcap \mathrm{A}_2)^{\mathcal{J}}$. By definition of $\mathrm{A}_1^{\mathcal{J}}$ and $\mathrm{A}_2^{\mathcal{J}}$, there exists $u_1 \in \mathrm{A}_1^{\mathcal{I}'}$ and $u_2 \in \mathrm{A}_2^{\mathcal{I}'}$ with $\overline{u_1} = \overline{u_2} = u$. Remark 13 ensures u_1 and u_2 satisfy the same concepts, that is in particular $u_1 \in (\mathrm{A}_1 \sqcap \mathrm{A}_2)^{\mathcal{I}'}$. Since $\mathcal{I}'$ is a model, it ensures $u_1 \in \mathrm{A}^{\mathcal{I}'}$, yielding by definition of $\mathrm{A}^{\mathcal{J}}$ that $u = \overline{u_1} \in \mathrm{A}^{\mathcal{J}}$.

$\mathrm{A}_1 \sqsubseteq \exists\mathrm{R}.\mathrm{A}_2$. Let $u \in \mathrm{A}_1^{\mathcal{J}}$. By definition of ${\mathrm{A}_1}^{\mathcal{J}}$ there exists $u_0 \in {\mathrm{A}_1}^{\mathcal{I}'}$ with $\overline{u_0} = u$. Since $\mathcal{I}'$ is a model, it ensures there exists $v_0 \in {\mathrm{A}_2}^{\mathcal{I}'}$ with $(u_0, v_0) \in \mathrm{R}^{\mathcal{I}'}$. By definition of ${\mathrm{A}_2}^{\mathcal{J}}$ and $\mathrm{R}^{\mathcal{J}}$, the element $v := \overline{v_0}$ satisfies both $v \in {\mathrm{A}_2}^{\mathcal{J}}$ and $(u, v) \in \mathrm{R}^{\mathcal{J}}$, that is $u \in (\exists\mathrm{R}.\mathrm{A}_2)^{\mathcal{J}}$.

$\exists\mathrm{R}.\mathrm{A}_1 \sqsubseteq \mathrm{A}_2$. Let $u \in (\exists\mathrm{R}.\mathrm{A}_1)^{\mathcal{J}}$, that is, there exists $v \in \mathrm{A}_1^{\mathcal{J}}$ with $(u, v) \in \mathrm{R}^{\mathcal{J}}$. By definition of $\mathrm{A}_1^{\mathcal{J}}$ and $\mathrm{R}^{\mathcal{J}}$, there exist $(u_0, v_0) \in \mathrm{R}^{\mathcal{I}'}$ and $v_1 \in \mathrm{A}_1^{\mathcal{I}'}$ such that $\overline{u_0} = u$ and $\overline{v_0} = \overline{v_1} = v$. Remark 13 ensures v_0 and v_1 satisfy the same concepts, in particular $u_0 \in (\exists\mathrm{R}.\mathrm{A}_1)^{\mathcal{I}'}$. Since $\mathcal{I}'$ is a model, this ensures $u_0 \in \mathrm{A}_2^{\mathcal{I}'}$, yielding by definition of $\mathrm{A}_2^{\mathcal{J}}$ that $u = \overline{u_0} \in \mathrm{A}_2^{\mathcal{J}}$.

$\mathrm{A} \sqsubseteq \neg\mathrm{B}$. By contradiction, assume $u \in \mathrm{A}^{\mathcal{J}} \cap \mathrm{B}^{\mathcal{J}}$. By definition there exists $v \in \mathrm{A}^{\mathcal{I}'}$ and $w \in \mathrm{B}^{\mathcal{I}'}$ with $\overline{v} = \overline{w} = u$. Remark 13 ensures v and w satisfy the same concepts, contradicting $\mathcal{I}'$ being a model.

$\neg\mathrm{B} \sqsubseteq \mathrm{A}$. Let $u \in \neg\mathrm{B}^{\mathcal{J}}$. By definition of $\Delta^{\mathcal{J}}$, there exists $v \in \mathcal{I}'$ such that $\overline{v} = u$. Since $u \notin \mathrm{B}^{\mathcal{J}}$, we have $v \notin \mathrm{B}^{\mathcal{I}'}$. Hence $\mathcal{I}'$ being a model gives $v \in \mathrm{A}^{\mathcal{I}'}$, yielding by definition $u = \overline{v} \in \mathrm{A}^{\mathcal{J}}$.

$\mathrm{P} \sqsubseteq \mathrm{R}$. Let $(u, v) \in \mathrm{P}^{\mathcal{J}}$. By definition of $\mathrm{P}^{\mathcal{J}}$, there exists $(u_0, v_0) \in \mathrm{P}^{\mathcal{I}'}$ such that $\overline{u_0} = u$ and $\overline{v_0} = v$. Since $\mathcal{I}'$ is a model, it ensures $(u_0, v_0) \in \mathrm{R}^{\mathcal{I}'}$, hence $(\overline{u_0}, \overline{v_0}) = (u, v) \in \mathrm{R}^{\mathcal{J}}$ by definition of $\mathrm{R}^{\mathcal{J}}$.

$\mathrm{R}_1 \sqcap \mathrm{R}_2 \sqsubseteq \bot$. By contradiction, assume one can find $(u, v) \in (\mathrm{R}_1 \sqcap \mathrm{R}_2)^{\mathcal{J}}$. By definition of $\mathrm{R}_1^{\mathcal{J}}$ and $\mathrm{R}_2^{\mathcal{J}}$, there exists $(u_1, v_1) \in \mathrm{R}_1^{\mathcal{I}'}$ and $(u_2, v_2) \in \mathrm{R}_2^{\mathcal{I}'}$ such that $\overline{u_1} = \overline{u_2} = u$ and $\overline{v_1} = \overline{v_2} = v$.
If $u_1, v_1 \in \Delta^*$, then, each element from Δ^* being alone in its equivalence class, we have $u_1 = u_2$ and $v_1 = v_2$. In particular it gives $(u_1, v_1) \in (\mathrm{R}_1 \sqcap \mathrm{R}_2)^{\mathcal{I}'}$, contradicting $\mathcal{I}'$ being a model.
Otherwise say $u_1 \notin \Delta^*$ (the case of $v_1 \notin \Delta^*$ is symmetrical), hence $\overline{v_1} \in \mathcal{N}_1^{\mathcal{J},\overline{\Delta^*}}(\overline{u_1})$. Theorem 9 gives a homomorphism from $\mathcal{N}_1^{\mathcal{J},\overline{\Delta^*}}(\overline{u_1})$ to $\mathcal{N}_1^{\mathcal{I}',\Delta^*}(u_1)$. But since $(\overline{u_1}, \overline{v_1}) \in (\mathrm{R}_1 \sqcap \mathrm{R}_2)^{\mathcal{J}}$, we obtain a contradiction with $\mathcal{I}'$ being a model.

Number of c-matches. We now prove $\mathcal{J}$ contains at most as many matches as $\mathcal{I}'$ by building an injection from matches in $\mathcal{J}$ to matches in $\mathcal{I}'$. Assume we have a match $\pi : q \to \mathcal{J}$. Consider the set of variables $\mathbf{v}_\pi := \{v \mid v \in \mathbf{y} \cup \mathbf{z}, \pi(v) \notin \overline{\Delta^*}\}$. Let $\mathcal{C}$ denote the set of connected components of $\mathbf{v}_\pi$ in $q_{|\mathbf{v}_\pi}$ (that is the query obtained by keeping only those atoms containing variables from $\mathbf{v}_\pi$). For each connected component $C \in \mathcal{C}$, choose a reference variable $v_C \in C$. Since π is a homomorphism and $|C| \leq |q|$, every variable $v \in C$ satisfies $\pi(v) \in \mathcal{N}_{|q|}^{\mathcal{J},\overline{\Delta^*}}(\pi(v_C))$. Let $d_C \in \Delta^{\mathcal{I}'}$ denote your favourite representative for the class of $\pi(v_C)$ (that is $\overline{d_C} = \pi(v_C)$).

From Theorem 9, we have a homomorphism $\rho_C : \mathcal{N}_{|q|}^{\mathcal{J},\overline{\Delta^*}}(\pi(v_C)) \to \mathcal{N}_{|q|}^{\mathcal{I}',\Delta^*}(d_C)$. Using these ρ_C, one per $C \in \mathcal{C}$, we define:

$$\begin{array}{rcl} \pi' : \mathbf{x} \cup \mathbf{y} \cup \mathbf{z} & \to & \Delta^{\mathcal{I}'} \\ v & \mapsto & \begin{cases} \rho_C(\pi(v)) & \text{if } v \in C, C \in \mathcal{C} \\ e & \text{if } \pi(v) = \overline{e} \in \overline{\Delta^*} \end{cases} \end{array}$$

Since each ρ_C is a homomorphism (again Theorem 9), we can check the overall π' is also a homomorphism:

- Consider $\mathrm{A}(v) \in q$. If $v \in C$ for some $C \in \mathcal{C}$, then ρ_C being a homomorphism gives $\pi'(v) \in \mathrm{A}^{\mathcal{I}'}$. Otherwise $\pi(v) = \overline{e} \in \overline{\Delta^*}$, but since π is a homomorphism we have $\pi(v) \in \mathrm{A}^{\mathcal{J}}$. Since $\overline{e} = \{e\}$ and by definition of $\mathrm{A}^{\mathcal{J}}$, it ensures $e \in \mathrm{A}^{\mathcal{I}'}$, that is $\pi'(v) \in \mathrm{A}^{\mathcal{I}'}$.
- Consider $\mathrm{R}(u, v) \in q$.
  - If both $\pi(u), \pi(v) \notin \overline{\Delta^*}$, then we can find $C \in \mathcal{C}$ such that $u, v \in C$, and then we use ρ_C being a homomorphism.
  - If both $\pi(u), \pi(v) \in \overline{\Delta^*}$, then the definition of $\mathrm{R}^{\mathcal{J}}$ provides $(u_0, v_0) \in \mathrm{R}^{\mathcal{I}'}$ with $\overline{u_0} = \pi(u) \in \overline{\Delta^*}$ and $\overline{v_0} = \pi(v) \in \overline{\Delta^*}$. Hence $\overline{u_0} = \{u_0\}$ and $\overline{v_0} = \{v_0\}$, which gives $(\pi'(u), \pi'(v)) \in \mathrm{R}^{\mathcal{I}'}$.
  - If $\pi(u) \notin \overline{\Delta^*}$ and $\pi(v) \in \overline{\Delta^*}$, then we have $\pi'(u) = \rho_C(\pi(u))$ for some $C \in \mathcal{C}$. Theorem 9 ensures $\pi'(u) \sim_1 \pi(u)$, and since π is a homomorphism, we also have $(\pi(u), \pi(v)) \in \mathrm{R}^{\mathcal{J}}$. Therefore we can apply Lemma 15 and we obtain $(\pi'(u), \pi'(v)) \in \mathrm{R}^{\mathcal{I}'}$.

In particular, π' is a match, hence $\pi'(\mathbf{z}) \subseteq \Delta^*$. Using property $\rho_C^{-1}(\Delta^*) \subseteq \overline{\Delta^*}$ for each $C \in \mathcal{C}$, provided by Theorem 9 along with definition of π', we obtain that $\pi(\mathbf{z}) \subseteq \overline{\Delta^*}$. Since $\rho_{|\overline{\Delta^*}} = \mathsf{Id}$, we have that the application $\pi_{|\mathbf{z}} \mapsto \pi'_{|\mathbf{z}}$ is injective. Therefore $\mathcal{J}$ contains at most as much matches as $\mathcal{I}'$ does.

Size of the model. Finally, an equivalence class $\overline{d}$ is characterized by: $|d| \bmod 2|q| + 3$, that is one equivalence class among $2|q| + 3$ possible classes; $w^d_{|q|+1,d}$, that is a word over an alphabet with at most $|\mathcal{T}|$ symbols and a length at most $|q| + 1$; and $\chi_{|q|+1,d}$, that is a function from words over an alphabet with at most $|\mathcal{T}|$ symbols and length at most $2|q| + 1$ to a set with size at most $|\Delta^*|+2^{|\mathsf{sig}(\mathcal{T})|} + 1$. Therefore, the number of possibly different equivalence classes, that is $|\Delta^{\mathcal{J}}|$, is at most $(2|q| + 3) \times |\mathcal{T}|^{|q|+2} \times (|\Delta^*|+2^{|\mathsf{sig}(\mathcal{T})|} + 1)^{|\mathcal{T}|^{2|q|+2}}$. Recalling that Lemma 3 allows us to assume $|\Delta^*| \leq |\mathsf{Ind}| + (|\mathsf{Ind}(\mathcal{A})| + 3\,|\mathcal{T}|\,2^{|\mathcal{T}|})^{|q|}|q|$, we have the claimed bounds for the size of $\mathcal{J}$, which concludes the proof of Theorem 8. $\square$

3.4.2 DL-Lite$_{\mathsf{core}}$: simpler neighbourhoods

In this section, we refine the preceding construction of optimal models with double-exponential size to the case of DL-Lite$_{\mathsf{core}}$ KBs, in which models with single-exponential size can be obtained, yielding the following complexity refinement.

Theorem 10. *CCQ answering in DL-Lite$_{\mathsf{core}}$ is in* coNEXP *w.r.t. combined complexity.*

The key idea is to explore a more restricted notion of neighbourhoods, yielding in particular exponentially smaller neighbourhoods that are still sufficient to capture the counting matches in the quotient model. To do so, we recall the definition of *interleaving* as first introduced in Kostylev and Reutter [2015], which starts from the canonical model rather than from the existential extraction we designed for the $\mathcal{ALCHI}$ case. Let us also recall that for every DL-Lite$^{\mathcal{H}}_{\mathsf{core}}$ KB $\mathcal{K}$, it is well-known the set of concept names M occurring in an element $w \cdot \exists \mathrm{R.M} \in \Delta^{\mathcal{C}_\mathcal{K}}$ of the canonical model of $\mathcal{K}$ contains exactly those concept names entailed by the concept $\exists \mathrm{R}^-$ [Calvanese et al., 2007b]. We will hence omit such sets of concept names M for the remainder of this section. Let us also fix a DL-Lite$_{\mathsf{core}}$ KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ and a CCQ q.

Definition 37. *Let $\mathcal{I}$ be a model of $\mathcal{K}$. We recall Δ^* denotes the subset of $\Delta^\mathcal{I}$ containing $\mathsf{Ind}(\mathcal{A})$ and the images of counting matches of q in $\mathcal{I}$. Let $f : \mathcal{C}_\mathcal{K} \to \mathcal{I}$ be a homomorphism from the canonical model of $\mathcal{K}$ to $\mathcal{I}$. The interleaving $\mathcal{I}^\flat$ of a model $\mathcal{I}$ is the image of $\mathcal{C}_\mathcal{K}$ by the function $f^\flat$ defined as follows:*

$$f^\flat : \Delta^{\mathcal{C}_\mathcal{K}} \to \Delta^* \uplus (\Delta^{\mathcal{C}_\mathcal{K}} \setminus \Delta^*)$$
$$w \mapsto \begin{cases} f(w) & \text{if } f(w) \in \Delta^* \\ w & \text{otherwise} \end{cases}$$

Remark 16. *The interleaving function $f^\flat$ is essentially the same function than the interlacing function f^* as used in Definition 23, where Δ° is replaced by $\Delta^{\mathcal{C}_\mathcal{K}}$.*

While interpretations of roles in interleavings and interlacings are defined in a similar manner, that is purely syntactically w.r.t. the domain elements, there are two differences we need to stress. First, only concepts entailed by $\exists \mathrm{R}^-$ are satisfied by an element $w \cdot \mathrm{R}$ in the interleaving. This is in contrast with interlacings in which an element $w \cdot \mathrm{R.M}$ satisfies all concepts satisfied by $f(w \cdot \mathrm{R.M})$ in $\mathcal{I}$. Second, the interleaving inherits the parsimonious introduction of fresh elements in DL-Lite$_{\mathsf{core}}$ canonical models, ensuring it also satisfies the following lemma.

Lemma 16. *For any role $\mathrm{R} \in \mathsf{N}^\pm_\mathsf{R}$ and anonymous element d_1 in the canonical model $\mathcal{C}_\mathcal{K}$ of $\mathcal{K}$, there is at most one element $d_2 \in \mathcal{C}_\mathcal{K}$ such that $(d_1, d_2) \in \mathrm{R}^{\mathcal{C}_\mathcal{K}}$.*

Proof. From the definition of $\mathrm{R}^{\mathcal{C}_\mathcal{K}}$, if d_1 is an anonymous domain element and $(d_1, d_2) \in \mathrm{R}^{\mathcal{C}_\mathcal{K}}$, then either:

- $d_1 = d_2 \cdot \mathrm{S}^-$ for some role S such that $\mathcal{T} \models \mathrm{S} \sqsubseteq \mathrm{R}$, or
- $d_2 = d_1 \cdot \mathrm{S}$ for some role S such that $\mathcal{T} \models \mathrm{S} \sqsubseteq \mathrm{R}$.

In both cases, since $\mathcal{T}$ is a DL-Lite$_{\mathsf{core}}$ TBox, the condition on S holds only if $\mathrm{S} = \mathrm{R}$. Moreover, we observe that if the first case holds, i.e., $d_1 = d_2 \cdot \mathrm{R}^-$, then the definition of $\Delta^{\mathcal{C}_\mathcal{K}}$ prevents the creation of an element $d_1 \cdot \mathrm{R}$. It follows that only one of the preceding cases can hold, and so there can be at most one d_2 with $(d_1, d_2) \in \mathrm{R}^{\mathcal{C}_\mathcal{K}}$. □

Repeated applications of Lemma 16 ensure that each partial match of a query q in the non-Δ^* parts of interleavings can be completed uniquely in a maximal such partial match (see further Lemma 17). This motivates a refined notion of the neigbhourhoods of an element c, restricting the (usual) neighbourhood to those elements e that can be reached by a match of some connected sub-query of q involving both c and e.

Definition 38. *Consider an interpretation $\mathcal{I}$ and an element $c \in \Delta^\mathcal{I}$. Its n-core-neighbourhood $\mathcal{N}^{\mathcal{I},\Delta}_{n,\mathsf{core}}(c)$ w.r.t. a subdomain $\Delta \subseteq \Delta^\mathcal{I}$ is defined as:*

$$\begin{array}{rcl} \mathcal{N}^{\mathcal{I},\Delta}_{0,\mathsf{core}}(c) & := & \{c\} \\ \mathcal{N}^{\mathcal{I},\Delta}_{n+1,\mathsf{core}}(c) & := & \left\{ e \;\middle|\; \begin{array}{l} \exists p \subseteq q \text{ connected} \\ \exists \pi : p \to \mathcal{I}_{|(\mathcal{N}^{\mathcal{I},\Delta}_{n,\mathsf{core}}(c) \setminus \Delta) \cup \{e\}} \text{ match} \\ c, e \in \pi(\mathsf{terms}(p)) \end{array} \right\} \end{array}$$

Remark 17. *By contrast with previous neighbourhoods, core-neighbourhoods are query dependent. Furthermore, since the subquery p must be connected, the inclusion $\mathcal{N}^{\mathcal{I},\Delta}_{n,\mathsf{core}}(c) \subseteq \mathcal{N}^{\mathcal{I},\Delta}_{n}(c)$ is straightforward.*

The central property allowing core-neighbourhoods to improve our construction is the following polynomial bound on their size in interleavings.

Lemma 17. *Let $\mathcal{I}$ be a model of $\mathcal{K}$ and $\mathcal{I}^\flat$ its interleaving. Consider $c \in \Delta^{\mathcal{I}^\flat} \setminus \Delta^*$, then $\left|\mathcal{N}^{\mathcal{I}^\flat,\Delta^*}_{n,\mathsf{core}}(c)\right| \leq |q|^2(|\mathcal{T}| + 1)$.*

Proof. Let $c \in \Delta^{\mathcal{I}^\flat} \setminus \Delta^*$ We proceed in two steps. We first prove that the number of elements in $\mathcal{N}^{\mathcal{I}^\flat,\Delta^*}_{n,\mathsf{core}}(c) \setminus \Delta^*$ is at most $|q|^2$. In a second step, we notice that each element $e \in \mathcal{N}^{\mathcal{I}^\flat,\Delta^*}_{n,\mathsf{core}}(c) \cap \Delta^*$ must be connected to an element $d \in \mathcal{N}^{\mathcal{I}^\flat,\Delta^*}_{n,\mathsf{core}}(c) \setminus \Delta^*$ by construction of the core-neighbourhoods. However, by construction of interleavings, each such element d is connected to at most $|\mathcal{T}|$ elements (a property directly inherited from anonymous elements of the canonical model), and we know from the first step that there are at most $|q|^2$ such elements d. This ensures there are at most $|q|^2 \cdot |\mathcal{T}|$ elements in $e \in \mathcal{N}_{n,\mathsf{core}}^{\mathcal{I}^\flat,\Delta^*}(c) \cap \Delta^*$, hence the claimed total bound of $|q|^2 + |q|^2 \cdot |\mathcal{T}| = |q|^2(|\mathcal{T}| + 1)$.

Henceforth, we focus on the first step. We start by proving that if the connected subquery $p \subseteq q$ and the term t_0 that shall map on c are fixed, then all matches $p \to \mathcal{I}_{|(\mathcal{N}_{n,\mathsf{core}}^{\mathcal{I},\Delta}(c) \setminus \Delta)}$ indeed mapping t_0 on c are equal. Consider two such matches π_1 and π_2. We proceed by induction on the terms t of p being connected. For $t = t_0$, we have $\pi_1(t_0) = \pi_2(t_0)$ by definition. For a further term t, we use the induction hypothesis, that is the existence of an atom $\mathrm{R}(t', t) \in p$ (or the other way around) such that $\pi_1(t') = \pi_2(t')$. Recall π_1 and π_2 are matches for p in $\mathcal{I}_{|(\mathcal{N}_{n,\mathsf{core}}^{\mathcal{I},\Delta}(c) \setminus \Delta)}$, in particular $\pi_1(t'), \pi_2(t') \notin \Delta^*$, hence we can apply Lemma 26, yielding $\pi_1(t) = \pi_2(t)$.

This proves that, for a fixed $t_0 \in \mathsf{terms}(q)$, each connected subquery $p \subseteq q$ admitting a match in $\mathcal{I}_{|(\mathcal{N}_{n,\mathsf{core}}^{\mathcal{I},\Delta}(c) \setminus \Delta^*}$ defines at most $|p|$ new neighbours, but also that if $p \subseteq p' \subseteq q$ are two such subqueries, then the neighbours defined by p are subsumed by those defined by p' (the restriction to the terms of p of the unique match of p' mapping t_0 on c must coincide with the unique match of p mapping t_0 on c). Still for a fixed t_0, consider now two connected subqueries $p_1, p_2 \subseteq q$, each admitting a (unique) match π_1 resp. π_2, to $\mathcal{I}_{|(\mathcal{N}_{n,\mathsf{core}}^{\mathcal{I},\Delta}(c) \setminus \Delta^*}$ mapping t_0 to c, and each maximal, w.r.t. the inclusion, for this property. By the previous property, we know π_1 and π_2 coincide on $\mathsf{terms}(p_1) \cap \mathsf{terms}(p_2)$. Therefore, $p_1 \cup p_2$ admits a match $\mathcal{I}_{|(\mathcal{N}_{n,\mathsf{core}}^{\mathcal{I},\Delta}(c) \setminus \Delta^*}$ mapping t_0 to c, being $\pi_1 \cup \pi_2$. But since p_1 and p_2 are assumed maximal for this property, we must have $p_1 = p_2$.

Therefore, for a fixed $t_0 \in \mathsf{terms}(q)$, there is a unique maximal connected subquery $p_{\max} \subseteq q$ admitting a match in $\mathcal{I}_{|(\mathcal{N}_{n,\mathsf{core}}^{\mathcal{I},\Delta}(c) \setminus \Delta^*}$ and mapping t_0 to c. As previously seen, the neighbours defined by p_{max} subsume those defined by other such subqueries, and since the match for $p_{\max}$ is unique, it defines at most $|q|$ neighbours. This holds for each possible choices of term t_0, hence a total number of possible neighbours bounded by $|q|^2$ as claimed. □

As for the general $\mathcal{ALCHI}$ case, this leads us to characterize the n-core-neighbourhood of an element $c \in \mathcal{I}^\flat$ via a subset $\Sigma_{n,c}$ of Ω_n, (we recall Ω_n is the set of words over Ω with length $\leq 2n$) and by the following function $\chi_{n,c}$.

$$\begin{aligned} \chi_{n,c} : \Sigma_{n,c} &\to \Delta^* \cup \{\emptyset\} \\ w &\mapsto \begin{cases} \emptyset & \text{if } f^\flat(r_{n,c}w) \notin \Delta^* \\ f^\flat(r_{n,c}w) & \text{if } f^\flat(r_{n,c}w) \in \Delta^* \end{cases} \end{aligned}$$

Notice Lemma 17 ensures the set $\Sigma_{n,c}$ has size at most $|q|^2(|\mathcal{T}| + 1)$, that is polynomial, while we kept track of the full Ω_n in the $\mathcal{ALCHI}$ setting. This will ensure that the following equivalence relation, used to merge elements, only admits an exponential number of equivalent classes.

Definition 39. *The equivalence relation* $\sim_n^{core}$ *on* $\mathcal{I}^\flat$ *is defined as follows: an element* $e \in \Delta^*$ *is* $\sim_n$*-equivalent only to itself; elements* c_1, c_2 *from* $\Delta^{\mathcal{I}^\flat} \setminus \Delta^*$ *are* $\sim_n^{core}$*-equivalent iff* $w_{n,c_1}^{c_1} = w_{n,c_2}^{c_2}$, $\chi_{n,c_1} = \chi_{n,c_2}$, $\Sigma_{n,c_1} = \Sigma_{n,c_2}$, *and* $|c_1| = |c_2| \mod 2|q|+3$.

We obtain a finite model of the required size by merging elements with respect to $\sim_{|q|+1}^{\text{core}}$.

Theorem 11. *The interpretation* $\mathcal{J} := \mathcal{I}^\flat/\sim_{|q|+1}^{core}$ *is a model of* $\mathcal{K}$ *that has at most as many c-matches for* q *as* $\mathcal{I}$. *Its size is polynomial w.r.t. data complexity, simply-exponential w.r.t. combined complexity.*

Proof. Modelhood is known from Kostylev and Reutter [2015], but can also be easily verified based on the DL-Lite$_{\mathsf{core}}$ subparts from the proof of Theorem 8.

To obtain an injective mapping of counting matches of q in $\mathcal{J}$ to counting matches of q in $\mathcal{I}^\flat$, one follows a similar proof strategy to the one used in Section 3.4.1, building homomorphisms from core-neighbourhoods in the quotient to core-neighbourhoods in the interleaving. We shall not go into the full details, but instead mention a significant point. In the proof of Theorem 9, one can verify the image of inductively built homomorphisms $\rho_{n,c}$ belongs to the n-core-neighbourhood (and not simply to the usual n-neigbourhood!). Indeed, when building the image of an element $\overline{e}$ belonging to $\mathcal{N}_{n,\mathsf{core}}^{\mathcal{J},\overline{\Delta^*}}(\overline{c})$, hence being reached by some match $\pi : p \to \mathcal{J}_{|(\mathcal{N}_{n-1,\mathsf{core}}^{\mathcal{J},\overline{\Delta^*}}(\overline{c}) \setminus \overline{\Delta^*}) \cup \{\overline{e}\}}$, the resulting element $\rho_{n,c}(\overline{e})$ is reached by the match $\rho_{n,c} \circ \pi : p \to \mathcal{I}^\flat_{|(\mathcal{N}_{n-1,\mathsf{core}}^{\mathcal{I}^\flat,\Delta^*}(c) \setminus \Delta^*) \cup \{e\}}$, ensuring $\rho_{n,c}(\overline{e})$ belongs to $\mathcal{N}_{n,\mathsf{core}}^{\mathcal{I}^\flat,\Delta^*}(c)$.

Finally, regarding the size of $\mathcal{J}$, we remark that an equivalence class $\overline{d}$ is now characterized by: $|d| \mod 2|q|+3$, that is one equivalence class among $2|q|+3$ possible classes; $w_{|q|+1,d}^d$, that is a word over an alphabet with at most $|\mathcal{T}|$ symbols and a length at most $|q|+1$; $\Sigma_{|q|+1,d}$, that is a subset, with size at most $|q|^2(|\mathcal{T}|+1)$ (Lemma 17), of the exponentially large set Ω_n; and $\chi_{|q|+1,d}$, that is a function from a set with at most $|q|^2(|\mathcal{T}|+1)$ elements to a set with at most $|\Delta^*|+1$ elements. Therefore, the number of possibly different equivalence classes, that is $|\Delta^{\mathcal{J}}|$, is at most:

$$(2|q|+3) \times |\mathcal{T}|^{|q|+2} \times (|\mathcal{T}|^{2|q|+3})^{|q|^2(|\mathcal{T}|+1)} \times (|\Delta^*|+1)^{|q|^2(|\mathcal{T}|+1)}.$$

Recall Lemma 3 allows to assume $|\Delta^*| \leq |\mathsf{Ind}| + (|\mathsf{Ind}(\mathcal{A})| + 3\,|\mathcal{T}|\,2^{|\mathcal{T}|})^{|q|}|q|$, we have the claimed bounds for the size of $\mathcal{J}$, which concludes the proof of Theorem 11. □

We conclude this section by closing an open question regarding UCQ answering over DL-Lite$_{\mathsf{core}}$ KBs with closed predicates. This problem is known to be coNEXP-hard from Ngo et al. [2016], but, to the best of our knowledge, no matching upper bound has yet been found. We hereby close this question by showing our construction easily adapts to this related setting.

Theorem 12. *(Boolean) UCQ answering over DL-Lite$_{\mathsf{core}}$ KBs with closed predicates is in* coNEXP *w.r.t. combined complexity.*

The proof follows from the following remark, which states that our construction still holds for subqueries p of the original CCQ q as long as the counting matches for p are already captured by the set Δ^* defined to handle q. Notice this condition has no chance to hold in general, as being a subquery makes p "easier" to map in a model than q.

Lemma 18. *Consider a DL-Lite$_{\mathsf{core}}$ KB (without closed predicates), a CCQ q and a model $\mathcal{I}$ of $\mathcal{K}$. Let $\mathcal{J}$ be the model with polynomial size w.r.t. data complexity, simply-exponential size w.r.t. combined complexity, obtained in Theorem 11. We recall that Δ^* contains the individuals from $\mathsf{Ind}(\mathcal{A})$ and the elements reached by counting matches of q in the original model $\mathcal{I}$. Consider a sub-query $p \subseteq q$. If the counting matches for p in $\mathcal{I}$ are also contained in Δ^*, then the model $\mathcal{J}$ has at most as many counting matches for p as $\mathcal{I}$.*

Proof. This follows from the various homomorphisms connecting the intermediate models, and by Δ^* being preserved all along the construction. □

We can now prove Theorem 12.

Proof of Theorem 12. Let $Q(\mathbf{x}) := \bigcup_{k=1}^{l} q_k(\mathbf{x})$ be a UCQ and $\mathcal{K} := (\mathcal{T}, \Sigma, \mathcal{A})$ a DL-Lite$_{\mathsf{core}}$ KB with closed predicates. Without loss of generality, we can assume that $\mathbf{x} = \emptyset$, that is Q is Boolean, and that no existential variable occurs in two distinct CQs q_k. Consider the Boolean CCQ $q := \bigwedge_{k=1}^{l} q_k$ in which all existential variables have been replaced by counting variables. If a countermodel exists for Q over $\mathcal{K}$, that is a model $\mathcal{I}$ in which no q_k matches, then it provides a model of $(\mathcal{T}, \mathcal{A})$ in which each q_k, hence the whole q, admits 0 counting matches. In particular, it yields $\Delta^* = \mathsf{Ind}(\mathcal{A})$ and the counting matches for each subquery q_k are contained in Δ^*. We can therefore use Lemma 18 for each q_k, which ensures that the model $\mathcal{J}$ obtained from the whole q has 0 counting match for each q_k, hence is a countermodel for Q over $(\mathcal{T}, \mathcal{A})$. Finally, it is easily verified that since $\mathsf{Ind}(\mathcal{A}) \subseteq \Delta^*$, model $\mathcal{J}$ complies with the closed predicates from Σ, hence is a countermodel for Q over $(\mathcal{T}, \Sigma, \mathcal{A})$, that is $\mathcal{K}$. □

3.5 Matching lower bounds

We proceed to exhibit matching lower bounds for each previous upper bound. For combined complexity, there are three main results: two 2EXP-hardness proofs for $\mathcal{EL}$ and DL-Lite$^{\mathcal{H}}_{\mathsf{pos}}$, both obtained by reducing UCQ answering over KBs with closed predicates, and a coNEXP-hardness proof for DL-Lite$_{\mathsf{core}}$, relying on a more classical reduction from a tiling problem. For data complexity, two coNP lower bounds follow from the subclass of *rooted* CCQs which is further investigated in Chapter 4, and the DP-hardness of the tightest variant is also proved.

Interestingly, all these reductions to deciding if $[m, +\infty]$ is a certain answer for a CCQ over a KB involve at most polynomially large such integers m. As a consequence, the complexity of answering CCQs over the considered family of KBs does not decrease if one enforces a unary encoding of the input integer m.

3.5.1 Two reductions from closed predicates

We now provide 2EXP lower bounds for $\mathcal{EL}$ and DL-Lite$^{\mathcal{H}}_{\mathsf{pos}}$, which together with Theorem 6, establish the 2EXP-completeness of CCQ answering for $\mathcal{ALCHI}$ and every sublogic that extends $\mathcal{EL}$ or DL-Lite$^{\mathcal{H}}_{\mathsf{pos}}$. The proofs are by reduction from the problem of answering Boolean union of conjunctive queries (BUCQs) over KBs with closed predicates, proven 2EXP-hard in Ngo et al. [2016].

Theorem 13. *CCQ answering in $\mathcal{EL}$ is* 2EXP*-hard w.r.t. combined complexity.*

Proof. Consider an $\mathcal{EL}$ KB $\mathcal{K} = (\mathcal{T}, \mathcal{A}, \Sigma)$ with closed predicates and a BUCQ $q = \bigvee_{k=1}^{l} q_k$. Examining the 2EXP-hardness proof from Ngo et al. [2016], we may assume that Σ consists only of concept names and each q_k is connected and has only variables as terms.

Pick a fresh individual aux not used in $\mathcal{A}$, and let $\mathcal{A}'$ be obtained from $\mathcal{A}$ by adding $\mathrm{A}(\mathsf{aux})$ for every concept name A from $\mathsf{sig}(\mathcal{K})$ and $\mathrm{P}(\mathsf{aux}, \mathsf{aux})$ for every role name P from $\mathsf{sig}(\mathcal{K})$. Consider the KB $\mathcal{K}' = (\mathcal{T}, \mathcal{A}')$ and the CCQ q' built as the conjunction of (i) all of the CQs q_k in q (with all variables treated as counting variables), (ii) the query $q_{\mathrm{A}} = \exists z_{\mathrm{A}}\ \mathrm{A}(z_{\mathrm{A}})$ for each $A \in \Sigma$, and (iii) the queries $q^+_{\mathrm{P}} = \exists z^+_{\mathrm{P}}\ \mathrm{P}(z^+_{\mathrm{P}}, \mathsf{aux})$ and $q^-_{\mathrm{P}} = \exists z^-_{\mathrm{P}}\ \mathrm{P}(\mathsf{aux}, z^-_{\mathrm{P}})$ for each role name P from $\mathcal{K}$. For each $\mathrm{A} \in \Sigma$, let n_{A} be the number of individuals a such that $\mathrm{A}(\mathsf{a}) \in \mathcal{A}$, and set $N := \prod_{\mathrm{A} \in \Sigma}(n_{\mathrm{A}} + 1)$. To complete the proof, we prove the following claim: $N + 1$ is a certain answer to q' over $\mathcal{K}'$ iff $\mathcal{K}$ entails q.

First assume that $N + 1$ is certain answer to q' over $\mathcal{K}'$, and consider a model $\mathcal{I}$ of $\mathcal{K}$. Add aux and all the associated facts from $\mathcal{A}' \setminus \mathcal{A}$ to obtain a model $\mathcal{I}'$ of $\mathcal{K}'$. Observe that $\mathcal{I}'$ must contain at least N matches: the disjuncts q_k and the queries q^+_{P} and q^-_{P} all have a match sending all variables to aux, and each q_{A} has n matches due to $\mathcal{A}$, plus one more sending z_{A} to aux. Since $N + 1$ is a certain answer, there must exist some additional match for q' in $\mathcal{I}'$. As $\mathcal{I}$ is a model of $\mathcal{K}$, it interprets each $A \in \Sigma$ as $\{\mathsf{a} \mid \mathrm{A}(\mathsf{a}) \in \mathcal{A}\}$, so there are no further matches for q_A. Next note that since aux is disconnected from the rest of $\mathcal{I}'$, there is no extra match for each $q^{\pm}_\mathrm{P}$. The only possibility then is that must be an extra match for one of the q_k, aside from the one mapping all variables aux. Since q_k is connected, this extra match is fully contained in $\Delta^{\mathcal{I}'} \setminus \{\mathsf{aux}\}$. Hence, $\mathcal{I}$ contains a match for q_k. We may thus conclude that $\mathcal{K}$ entails q.

For the other direction, suppose that $\mathcal{K}$ entails q, and consider a model $\mathcal{I}'$ of $\mathcal{K}'$. There are at least N trivial matches for q' in $\mathcal{I}'$. If there is an extra match for one of the q_A or one of the $q^{\pm}_\mathrm{P}$, then we are done. Otherwise, removing aux from $\mathcal{I}'$ yields a model $\mathcal{I}$ of $\mathcal{K}$. Since $\mathcal{K}$ entails q, there must be a match for one of the q_k in $\mathcal{I}$. This yields a new match for q_k in $\mathcal{I}'$ and concludes. □

Theorem 14. *CCQ answering in DL-Lite$^{\mathcal{H}}_{\mathsf{pos}}$ is* 2EXP-*hard w.r.t. combined complexity.*

Proof. As the 2EXP-hardness proof for DL-Lite$^{\mathcal{H}}_{\mathsf{core}}$ from Ngo et al. [2016] does not involve negative inclusions, we can employ the same approach as for $\mathcal{EL}$ (the added aux assertions cannot lead to inconsistency). □

We thus close the open question of the combined complexity of CCQ answering in DL-Lite$^{\mathcal{H}}_{\mathsf{core}}$. Note that our lower bound applies even to the subclass of CCQs whose every variable is a counting variable, as considered in Kostylev and Reutter [2015]; Calvanese et al. [2020a].

3.5.2 A tiling problem for DL-Lite$_{\mathsf{core}}$

The preceding 2EXP lower bound does not apply to DL-Lite$_{\mathsf{pos}}$, for which coNEXP membership has been shown (Theorem 10). We pinpoint the exact complexity by giving a matching lower bound, via a reduction from the exponential grid tiling problem. Here again the lower bound holds even when restricted to CCQs with only counting variables.

Theorem 15. *CCQ answering in DL-Lite$_{\mathsf{pos}}$ is* coNEXP-*hard w.r.t. combined complexity.*

Proof. The proof is by reduction from the exponential grid tiling problem ExpTiling. We recall that an instance of this problem consists of a set $\mathcal{C}$ of colors, two relations $\mathcal{H}, \mathcal{V} \subseteq \mathcal{C} \times \mathcal{C}$ that give the horizontal and vertical tiling conditions, and a number n. The task is to decide whether there exists a valid $(\mathcal{H}, \mathcal{V})$-tiling of an $2^n \times 2^n$ grid, i.e., a mapping $\tau : \{0, \ldots, 2^n - 1\} \times \{0, \ldots, 2^n - 1\} \mapsto \mathcal{C}$ such that $(\tau(i, j), \tau(i + 1, j)) \in \mathcal{H}$ for every $0 \leq i < 2^n - 1$ and $(\tau(i, j), \tau(i, j + 1)) \in \mathcal{V}$ for every $0 \leq j < 2^n - 1$. In what follows, we consider an instance $(n, \mathcal{C}, \mathcal{H}, \mathcal{V})$ of the ExpTiling problem.

To be able to test for the existence of a tiling of a $2^n \times 2^n$ grid, we must start by ensuring we can find such a grid in each model. Furthermore, we will need to detect horizontal and vertical adjacency in this grid, it is thus appropriate to use horizontal/vertical coordinates. To ensure a polynomial reduction, we need to use a binary encoding of these coordinates. We start from an initial element a and use TBox axioms to generate all possible coordinates of the horizontal coordinates:

$$\mathrm{A}(\mathsf{a}) \qquad \mathrm{A} \sqsubseteq \exists \mathrm{R}_{\mathrm{h,n-1,b}} \qquad \exists \mathrm{R}^-_{\mathrm{h,i,b}} \sqsubseteq \exists \mathrm{R}_{\mathrm{h,i-1,b'}} \qquad \begin{pmatrix} i = 1, \dots n \\ b, b' \in \{0,1\} \end{pmatrix}$$

We proceed similarly with the vertical coordinates, until we generate all possible pairs of coordinates:

$$\exists \mathrm{R}^-_{\mathrm{h,0,b}} \sqsubseteq \exists \mathrm{R}_{\mathrm{v,n-1,b'}} \qquad \exists \mathrm{R}^-_{\mathrm{v,i,b}} \sqsubseteq \exists \mathrm{R}_{\mathrm{v,i-1,b'}} \qquad \begin{pmatrix} i = 1, \dots n \\ b, b' \in \{0,1\} \end{pmatrix}$$

The preceding axioms will generate a binary tree of height $2n$ in the canonical model, whose leaves represent all possible grid positions. We use the following axiom to assign a color to each of the points representing a grid position:

$$\exists \mathrm{R}^-_{\mathrm{v,0,b}} \sqsubseteq \exists \mathrm{HasCol} \qquad (b \in \{0,1\})$$

To help us compare positions, we will include the following TBox axioms:

$$\exists \mathrm{R}^-_{\mathrm{d,i,b}} \sqsubseteq \exists \mathrm{HasBit}_{\mathrm{d,j}} \qquad \begin{pmatrix} 0 \leq i < j \leq n-1 \\ b \in \{0,1\} \\ d \in \{h, v\} \end{pmatrix}$$

and:

$$\exists \mathrm{R}^-_{\mathrm{v,i,b}} \sqsubseteq \exists \mathrm{HasBit}_{\mathrm{h,j}} \qquad \begin{pmatrix} 0 \leq i, j \leq n-1 \\ b \in \{0,1\} \end{pmatrix}$$

To keep track of elements used as color or bits, we also add:

$$\exists \mathrm{HasCol}^- \sqsubseteq \mathrm{Color} \qquad \exists \mathrm{HasBit}^-_{\mathrm{d,i}} \sqsubseteq \mathrm{Bit} \qquad \begin{pmatrix} 0 \leq i \leq n-1 \\ d \in \{h, v\} \end{pmatrix}$$

This completes our description of the TBox. We will finish our description of the ABox later in the proof, but it will be useful to know that it will contain an ABox individual c for every color $c \in \mathcal{C}$ and two ABox individuals $(\mathsf{one}, \mathsf{zero})$ to represent bits.

Let us now define the query q. In what follows, we build q step by step, providing several subqueries. For the sake of readability, we omit subscript/superscripts that would allow to decide which variable occurs in which subquery. The reason is simple: *in what follows, no variable is shared by different subqueries.*

To keep track of the colors used in a candidate tiling, we use the following subquery:

$$q_{\text{Color}} := \exists z\, \text{Color}(z)$$

We also need to detect if other bits than the intended ones ($\mathsf{one}, \mathsf{zero}$) are being used to satisfy the right hand sides $\exists \text{HasBit}_{d,i}$. For this purpose, we introduce the following subquery:

$$q_{\text{Bit}} := \exists z\, \text{Bit}(z)$$

To detect if the i^{th} bit of the coordinate in direction d is one when it should be zero:

$$q_{d,i,\mathsf{one}} := \exists z_1 \exists z_2\ \text{R}_{\text{d,i,0}}(z_1, z_2) \wedge \text{HasBit}_{\text{d,i}}(z_2, \mathsf{one}) \qquad \begin{pmatrix} 0 \leq i \leq n-1 \\ d \in \{h, v\} \end{pmatrix}$$

And the other way around:

$$q_{d,i,\mathsf{zero}} := \exists z_1 \exists z_2\ \text{R}_{\text{d,i,1}}(z_1, z_2) \wedge \text{HasBit}_{\text{d,i}}(z_2, \mathsf{zero}) \qquad \begin{pmatrix} 0 \leq i \leq n-1 \\ d \in \{h, v\} \end{pmatrix}$$

To detect if the j^{th} bit of the coordinate in direction d isn't carried from the i^{th} level to the next:

$$\begin{aligned} q_{d,i,b,j} := \exists z_1\, \exists z_2\, \exists z_1'\, \exists z_2'\ \text{R}_{\text{d,i,b}}(z_1, z_2)\ \wedge\ \text{HasBit}_{\text{d,j}}(z_1, z_1') \\ \wedge\, \text{HasBit}_{\text{d,j}}(z_2, z_2')\ \wedge\ \text{Bit}^{\neq}(z_1', z_2') \end{aligned} \qquad \begin{pmatrix} 0 \leq i < j \leq n-1 \\ b \in \{0, 1\} \\ d \in \{h, v\} \end{pmatrix}$$

To detect if the j^{th} bit of the horizontal coordinate isn't carried through the i^{th} vertical level:

$$\begin{aligned} q_{i,b,j} := \exists z_1\, \exists z_2\, \exists z_1'\, \exists z_2'\ \text{R}_{\text{v,i,b}}(z_1, z_2)\ \wedge\ \text{HasBit}_{\text{h,j}}(z_1, z_1') \\ \wedge\, \text{HasBit}_{\text{h,j}}(z_2, z_2')\ \wedge\ \text{Bit}^{\neq}(z_1', z_2') \end{aligned} \qquad \begin{pmatrix} 0 \leq i, j \leq n-1 \\ b \in \{0, 1\} \end{pmatrix}$$

To detect if part of the model is collapsing on the auxiliary individual:

$$q_{\mathsf{aux},\text{R}} := \exists z\ \text{R}(z, \mathsf{aux}) \qquad (\text{R} = \text{R}_{\text{d,i,b}}, \text{HasBit}_{\text{d,i}}, \text{HasCol})$$

We next discuss the parts of the query that are used to check the tiling conditions. To detect adjacency, we remark that two grid positions $(h_1, v_1), (h_2, v_2) \in \{0, \ldots, 2^n - 1\} \times \{0, \ldots, 2^n - 1\}$ are vertically adjacent iff:

- $h_1 = h_2$, so the binary encodings of h_1 and h_2 are the same;

- $v_2 = v_1 + 1$, so the binary encodings of v_2 and v_1 are the same until, at some point, v_2 ends with $1 \cdot 0^k$ while v_1 ends with $0 \cdot 1^k$.

To detect a violation of the vertical tiling condition (i.e. two vertically adjacent tiles with colors c and c' such that $(c, c') \notin \mathcal{V}$), we need n queries, one for each possible position where the bit from the vertical coordinates differ. For each $1 \leq k \leq n$, we create a subquery $q^{\mathcal{V},(c,c'),k}$ defined as follows.

$$
\begin{aligned}
q^{\mathcal{V},(c,c'),k} = \; & \exists z_l \, \exists z_r \, \exists z_{h,0} \ldots \exists z_{h,n-1} \, \exists z_{v,k+1} \ldots \exists z_{v,n-1} \\
& \bigwedge_{i=0}^{n-1} (\mathrm{HasBit}_{\mathrm{h,i}}(z_l, z_{h,i}) \wedge \mathrm{HasBit}_{\mathrm{h,i}}(z_r, z_{h,i})) \\
& \wedge \bigwedge_{i=k+1}^{n-1} (\mathrm{HasBit}_{\mathrm{v,i}}(z_l, z_{v,i}) \wedge \mathrm{HasBit}_{\mathrm{v,i}}(z_r, z_{v,i})) \\
& \wedge \mathrm{HasBit}_{\mathrm{v,k}}(z_l, \mathsf{zero}) \wedge \mathrm{HasBit}_{\mathrm{v,k}}(z_r, \mathsf{one}) \\
& \wedge \bigwedge_{i=0}^{k-1} (\mathrm{HasBit}_{\mathrm{v,i}}(z_l, \mathsf{one}) \wedge \mathrm{HasBit}_{\mathrm{v,i}}(z_r, \mathsf{zero})) \\
& \wedge \mathrm{HasCol}(z_l, \mathsf{c}) \wedge \mathrm{HasCol}(z_r, \mathsf{c}')
\end{aligned}
$$

We can similarly define a set of subqueries $q^{\mathcal{H},(c,c'),k}$ that detect violations of the horizontal tiling conditions (see *e.g.* Figure 3.15).

Finally, we let q be the conjunction of the all of the preceding subqueries. We can now define the ABox, which introduces individuals for the intended colors and bits and a further individual d that serves to ensure that all parts of the query can be matched:

$$
\begin{aligned}
\mathcal{A} = \; & \{\mathrm{Root}(\mathsf{a}), \mathrm{Bit}(\mathsf{zero}), \mathrm{Bit}(\mathsf{one}), \mathrm{Bit}^{\neq}(\mathsf{zero}, \mathsf{one}), \mathrm{Bit}^{\neq}(\mathsf{one}, \mathsf{zero})\} \\
& \cup \{\mathrm{Color}(\mathsf{c}) \mid c \in \mathcal{C}\} \\
& \cup \{\mathrm{Root}(\mathsf{aux}), \mathrm{Bit}(\mathsf{aux}), \mathrm{Color}(\mathsf{aux}), \mathrm{Bit}^{\neq}(\mathsf{aux}, \mathsf{aux}), \mathrm{HasCol}(\mathsf{aux}, \mathsf{aux})\} \\
& \cup \{\mathrm{R}_{\mathrm{d,i,b}}(\mathsf{aux}, \mathsf{aux}) \mid d \in \{h, v\}, i \in \{0, \ldots n-1\}, b \in \{0, 1\}\} \\
& \cup \{\mathrm{HasBit}_{\mathrm{d,i}}(\mathsf{aux}, \mathsf{aux}) \mid d \in \{h, v\}, i \in \{0, \ldots n-1\}\}
\end{aligned}
$$

Let $p = |\mathcal{C}|$, and let $\mathcal{K}$ be the KB with the preceding TBox and ABox. To complete the proof, it suffices to establish the following claim:

Claim $[3p{+}4, +\infty]$ is a certain answer for q over $\mathcal{K} \iff (n, \mathcal{C}, \mathcal{H}, \mathcal{V}) \notin$ ExpTiling.

First observe that there are always at least $3(p+1)$ c-matches given by: $p+1$ mappings for q_{Color} (on each color-individual c and on aux), times 3 mappings for q_{Bit} (on zero, one and aux), times 1 mapping for each other subquery (collapse on aux).

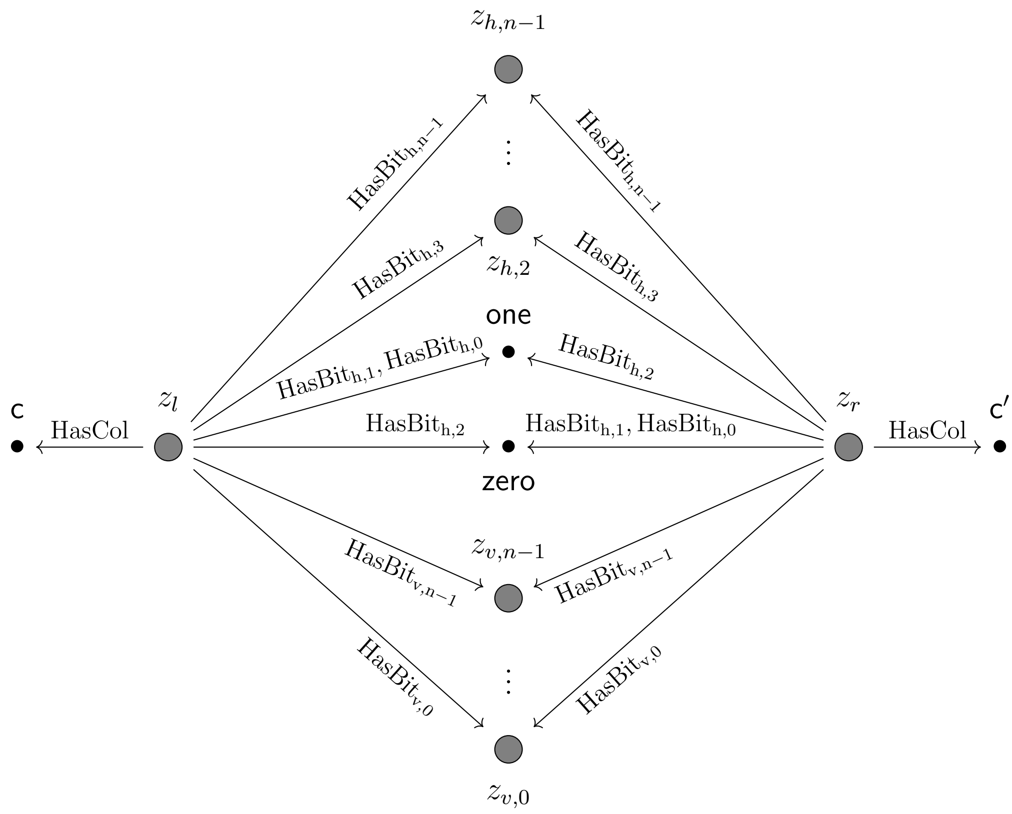


Figure 3.15: The subquery $q^{\mathcal{H},(c,c'),2}$ to check an horizontal tiling condition.

($\Rightarrow$) Assume $[3p+4,+\infty]$ is a certain answer, and take some candidate tiling $\tau : \{0,\dots 2^n-1\} \times \{0,\dots 2^n-1\} \to \{\mathsf{c} \mid c \in \mathcal{C}\}$. Let $\mathcal{I}_\tau$ be the model of $\mathcal{K}$ that is obtained from $\mathcal{C}_\mathcal{K}$ as follows:

- $\Delta^{\mathcal{I}_\tau}$ contains all elements from $\Delta^{\mathcal{C}_\mathcal{K}}$ except those anonymous elements whose last symbol is HasCol or HasBit$_{d,i}$ (i.e. witnesses for axioms involving $\exists$HasCol or $\exists$HasBit$_{d,i}$);

- the roles HasCol and HasBit$_{d,i}$ are interpreted as follows:

$$
\begin{aligned}
\mathrm{HasBit}_{d,i}^{\mathcal{I}_\tau} :=\ & \{(\mathsf{aux},\mathsf{aux})\} \\
& \cup \{(\mathsf{a}w\mathrm{R}_{d,i,0}w', \mathsf{zero}) \mid \mathsf{a}w\mathrm{R}_{d,i,0}w' \in \Delta^{\mathcal{I}_\tau}\} \\
& \cup \{(\mathsf{a}w\mathrm{R}_{d,i,1}w', \mathsf{one}) \mid \mathsf{a}w\mathrm{R}_{d,i,1}w' \in \Delta^{\mathcal{I}_\tau}\} \\
\mathrm{HasCol}^{\mathcal{I}_\tau} :=\ & \{(\mathsf{aux},\mathsf{aux})\} \\
& \cup \{(\mathsf{a}\mathrm{R}_{\mathrm{h,n-1,h_{n-1}}} \dots \mathrm{R}_{\mathrm{h,0,h_0}} \mathrm{R}_{\mathrm{v,n-1,v_{n-1}}} \dots \mathrm{R}_{\mathrm{v,0,v_0}}, \tau(h,v)) \\
& \qquad \mid h := h_{n-1},\dots h_0 \in [0,2^n-1], v := v_{n-1},\dots v_0 \in [0,2^n-1]\}
\end{aligned}
$$

where $h := h_{n-1} \dots h_0$ and $v := v_{n-1} \dots v_0$ mean h and v are the numbers whose binary encodings are $h_{n-1} \dots h_0$ and $v_{n-1} \dots v_0$ respectively;

- the remaining roles are interpreted exactly as in $\mathcal{C}_{\mathcal{K}}$.

Recall our assumption that there is an additional c-match π for q in $\mathcal{I}_\tau$. It is easily verified that the additional match can only result from one of the queries $q^{h,(c,c'),k}$ or $q^{v,(c,c'),k}$. From the definition of $\mathcal{I}_\tau$, this implies that there are two horizontally (or vertically) adjacent tiles, which positions are encoded on $\pi(z_l)$ and $\pi(z_r)$ by the endpoints of their respective roles $\mathrm{HasBit}_{d,i}$, whose respective colors c and c' violate either $\mathcal{H}$ or $\mathcal{V}$. Thus τ is not an $(\mathcal{H}, \mathcal{V})$-tiling. As this construction holds for any possible tiling τ, we infer that $(n, \mathcal{C}, \mathcal{H}, \mathcal{V}) \notin$ ExpTiling.

($\Leftarrow$) Assume $(n, \mathcal{C}, \mathcal{H}, \mathcal{V}) \notin$ ExpTiling, and take some model $\mathcal{I}$ of $\mathcal{K}$. There is a homomorphism $f : \mathcal{C}_{\mathcal{K}} \to \mathcal{I}$. If there exists $\mathsf{a}w \in \Delta^{\mathcal{C}_{\mathcal{K}}}$ such that $f(\mathsf{a}w) = \mathsf{aux}$, then there exists a new c-match for the subquery $q_{\mathsf{aux},\mathrm{R}}$, where R is the last letter of the shortest prefix w' of w such that $f(\mathsf{a}w') = \mathsf{aux}$. Otherwise, we define $\tau : \{0, \dots, 2^n - 1\} \times \{0, \dots 2^n - 1\} \to \Delta^{\mathcal{I}}$ as follows: $\tau(h_{n-1} \dots h_0, v_{n-1} \dots v_0) := f(\mathsf{a}\mathrm{R}_{h,n-1,h_{n-1}} \dots \mathrm{R}_{h,0,h_0}\mathrm{R}_{v,n-1,v_{n-1}} \dots \mathrm{R}_{v,0,v_0}\mathrm{HasCol})$ (again slightly abusing notation by working with binary encodings of numbers). There are five cases to consider:

- If there exists $(h_{n-1} \dots h_0, v_{n-1} \dots v_0)$ such that $\tau(h_{n-1} \dots h_0, v_{n-1} \dots v_0) \notin \{\mathsf{c} \mid c \in \mathcal{C}\}$, then this provides a new c-match of q in $\mathcal{I}$ in which the subquery q_{Color} is mapped as $z \mapsto \tau(h_{n-1} \dots h_0, v_{n-1} \dots v_0)$.

- Otherwise, suppose there exists an element that is in the range of Bit that is not zero nor one, then this also provides a new c-match of q, in which the subquery q_{Bit} is mapped on this element.

- Otherwise, suppose there exists an inconsistent choice of bit, that is $\mathsf{a}w\mathrm{R}_{d,i,0}$ and $f(\mathsf{a}w\mathrm{R}_{d,i,0}\mathrm{HasBit}_{d,i}) = \mathsf{one}$ (respectively: $\mathsf{a}w\mathrm{R}_{d,i,1}$ and $f(\mathsf{a}w\mathrm{R}_{d,i,0}\mathrm{HasBit}_{d,i}) = \mathsf{zero}$), then it provides a new c-match for the subquery $q_{d,i,\mathsf{one}}$ (resp: $q_{d,i,\mathsf{zero}}$).

- Otherwise, suppose there exists an non-propagated coordinate, that is $\mathsf{a}w\mathrm{R}_{d,i,b}$ such that $f(\mathsf{a}w\mathrm{HasBit}_{d',k}) \neq f(\mathsf{a}w\mathrm{R}_{d,i,b}\mathrm{HasBit}_{d',k})$, then it provides a new c-match either for the subquery $q_{d,i,b,j}$ or for the subquery $q_{i,b,j}$.

- Else, since $(n, \mathcal{C}, \mathcal{H}, \mathcal{V}) \notin$ ExpTiling, there exist two adjacent positions with coordinates $p := (h_{n-1} \dots h_0, v_{n-1} \dots v_0)$ and $p' := (h'_{n-1} \dots h'_0, v'_{n-1} \dots v'_0)$ such that $(\tau(p), \tau(p')) \in (\mathcal{C} \times \mathcal{C}) \setminus \mathcal{D}$, for $\mathcal{D}$ either $\mathcal{H}$ or $\mathcal{V}$. Letting k be the bit from which the encoding of the non-$\mathcal{D}$ coordinate differs, we obtain a new c-match for q, in which the subquery $q^{\mathcal{D},(\tau(p),\tau(p')),k}$ is satisfied by mapping z_l to $f(\mathsf{a}\mathrm{R}_{h,n-1,h_{n-1}} \dots \mathrm{R}_{h,0,h_0}\mathrm{R}_{v,n-1,v_{n-1}} \dots \mathrm{R}_{v,0,v_0})$ and z_r to $\mathsf{a}\mathrm{R}_{h,n-1,h'_{n-1}} \dots \mathrm{R}_{h,0,h'_0}\mathrm{R}_{v,n-1,v'_{n-1}} \dots \mathrm{R}_{v,0,v'_0})$ (or the converse).

In every case, there is an additional c-match for q. We thus obtain that $[p+1, +\infty]$ is a certain answer to q over $\mathcal{K}$. □

3.5.3 Data complexity

We move to lower bounds for data complexity, consisting of a coNP-hardness result for all investigated DLs and DP-hardness if we consider tight CCQ answering. For coNP-hardness, only a brief proof sketch is provided here as the result follows from later results in the context of the more restricted class of rooted CCQ (see Theorem 21 in Chapter 4). Note that a coNP-hardness result for DL-Lite$_{\mathsf{pos}}$ was already proved in Kostylev and Reutter [2015], from which our reduction borrows the main ideas.

Theorem 16. *CCQ answering in DL-Lite*$_{\mathsf{pos}} \cap \mathcal{EL}$ *is* coNP-*hard w.r.t. data complexity.*

Proof sketch. We reduce the complement of the graph 3-colorability problem (3-Col) to answering the CCQ which is the conjunction of the subqueries

$$q_{col} := \exists y \, \exists z \; \mathrm{HasCol}(y, z)$$
$$q_{edge} := \exists y_c \, \exists z_1 \, \exists z_2 \; \mathrm{Edge}(z_1, z_2) \wedge \mathrm{HasCol}(z_1, y_c) \wedge \mathrm{HasCol}(z_2, y_c)$$

w.r.t. the TBox $\mathcal{T}$ containing the single axiom $\mathrm{Vertex} \sqsubseteq \exists \mathrm{HasCol}.\top$. □

From the coNP membership of CCQ answering, it is easily seen that tight CCQ answering can be done in DP by making a call to a coNP oracle (is $[m, +\infty]$ a certain answer?) and an NP oracle (is $[m+1, +\infty]$ not a certain answer?). The DP-hardness of this problem was left as an open question by Kostylev and Reutter.

Based on the preceding reduction from 3-Col, we give a reduction from the following problem (DP-complete due to Garey et al. [1976]): given *planar* graphs $\mathcal{G}_1$ and $\mathcal{G}_2$, decide if $\mathcal{G}_1 \in$ 3-Col and $\mathcal{G}_2 \notin$ 3-Col. Here again, the result follows from an analogous result for the subclass of rooted CCQs (see Theorem 22 in Chapter 4).

Theorem 17. *Tight-CCQ answering in DL-Lite*$_{\mathsf{pos}} \cap \mathcal{EL}$ *is* DP-*hard w.r.t. data complexity.*

If one drops the rootedness restriction and focuses on either $\mathcal{EL}$ or DL-Lite$_{\mathsf{pos}}$, then the preceding reduction can be adapted to show DP-hardness also for the two kinds of CCQs from Kostylev and Reutter [2015], closing their open question.

Theorem 18. *Tight CCQ answering in DL-Lite*$_{\mathsf{pos}}$ *and in* $\mathcal{EL}$ *for Count()-queries as defined in Kostylev and Reutter [2015], that are CCQs with* $\mathbf{y} = \emptyset$, *is* DP-*hard w.r.t. data complexity.*

Proof. We focus on DL-Lite$_{\mathsf{pos}}$ and mention along the proof of to adapt to $\mathcal{EL}$. Consider two planar graphs $\mathcal{G}_1 := (\mathcal{V}_1, \mathcal{E}_1)$ and $\mathcal{G}_2 := (\mathcal{V}_2, \mathcal{E}_2)$. Each vertex is described in the ABox with a specific concept, either Vertex$_1$ or Vertex$_2$ depending on which graph it appears in, and each edge by a simple Edge role. Three colors are also provided for each graph, and identified with concepts Color$_1$ and Color$_2$. We also introduce a auxiliary vertex a^i_v for each graph, equipped with a monochromatic red edge so that each upcoming subquery is ensured to match. Formally, we consider the following ABox:

$$\begin{aligned}\mathcal{A} := &\{\text{Vertex}_1(\mathsf{u}) \mid u \in \mathcal{V}_1\} \cup \{\text{Vertex}_2(\mathsf{u}) \mid u \in \mathcal{V}_1\}\\ &\{\text{Edge}(\mathsf{u}_1, \mathsf{u}_2) \mid (u_1, u_2) \in \mathcal{E}_1 \cup \mathcal{E}_2\}\\ &\{\text{Color}_1(\mathsf{c}) \mid \mathsf{c} \in \{\mathsf{r}_1, \mathsf{g}_1, \mathsf{b}_1\}\} \cup \{\text{Color}_2(\mathsf{c}) \mid \mathsf{c} \in \{\mathsf{r}_2, \mathsf{g}_2, \mathsf{b}_2\}\}\\ &\{\text{Edge}(\mathsf{a}^1_\mathsf{v}, \mathsf{a}^1_\mathsf{v}), \text{HasColor}(\mathsf{a}^1_\mathsf{v}, \mathsf{r}_1), \text{Edge}(\mathsf{a}^2_\mathsf{v}, \mathsf{a}^2_\mathsf{v}), \text{HasColor}(\mathsf{a}^2_\mathsf{v}, \mathsf{r}_2)\}\end{aligned}$$

The TBox contains the following axioms, requiring each vertex to get a color identified with the correct concept.

$$\begin{array}{ll}\text{Vertex}_1 \sqsubseteq \exists\text{HasCol}_1 & \exists\text{HasCol}_1^- \sqsubseteq \text{Color}_1\\ \text{Vertex}_2 \sqsubseteq \exists\text{HasCol}_2 & \exists\text{HasCol}_2^- \sqsubseteq \text{Color}_2\end{array}$$

For $\mathcal{EL}$, we consider instead axioms Vertex$_1 \sqsubseteq \exists$HasCol$_1$.Color$_1$ and Vertex$_2 \sqsubseteq \exists$HasCol$_2$.Color$_2$.

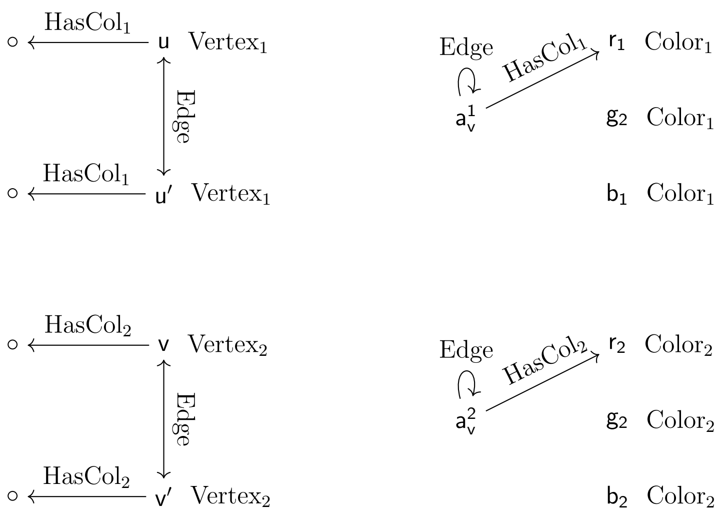


Figure 3.16: A part of $\mathcal{C}_\mathcal{K}$ with $(u, u') \in \mathcal{E}_1$ and $(v, v') \in \mathcal{E}_2$.

Subqueries q_i^{edge} and q_i^{col} are then defined to detect monochromatic edges or use

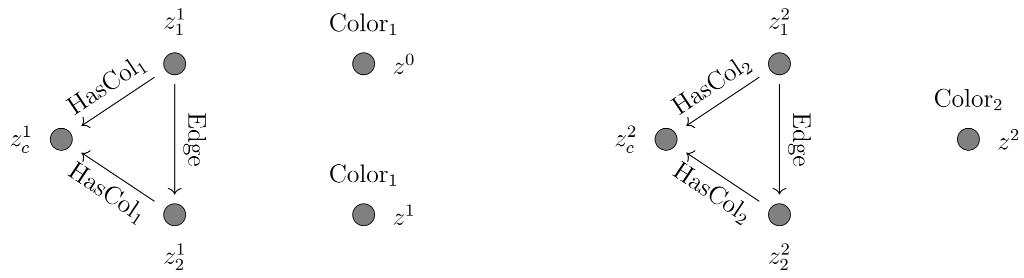


Figure 3.17: The $Count()$-CCQ q, which is the conjunction of $q_1^{edge}, q_1^{col}, q_0^{col}$ (left part) and q_2^{edge}, q_2^{col} (right part).

of new colors for each graph, that is, for $i \in \{1, 2\}$:

$$q_i^{edge} = \exists z_c^i \; \exists z_1^i \; \exists z_2^i \; \text{Edge}(z_1^i, z_2^i) \wedge \text{HasCol}_\text{i}(z_1^i, z_c^i) \wedge \text{HasCol}_\text{i}(z_2^i, z_c^i)$$
$$q_i^{col} = \exists z^i \; \text{Color}_\text{i}(z^i)$$

The main challenge is however to make sure that we can determine the 3-colorability status of the two graphs solely by looking at the number of counting matches of the query. To be able to distinguish $\mathcal{G}_1$ from $\mathcal{G}_2$, we introduce an asymmetry by duplicating the color counter query for $\mathcal{G}_1$, i.e., create a copy q_0^{col} of q_1^{col} that uses a fresh variable: $q_0^{col} = \exists z^0 \; \text{Color}_1(z^0)$. We now let q be the conjunction of these 5 subquery, and $\mathcal{K} := (\mathcal{T}, \mathcal{A})$.

It is easily verified that the query q now corresponds to a Count query as defined in Kostylev and Reutter [2015]. The query q is displayed in Figure 3.17, and the canonical model $\mathcal{C}_\mathcal{K}$ of $\mathcal{K}$ is displayed in Figure 3.16.

We now claim $[36, +\infty]$ is the tightest certain answer to q over $\mathcal{K}$ iff $\mathcal{G}_1 \in$ 3-Col and $\mathcal{G}_2 \notin$ 3-Col. This is proven by a case analysis, summarized here:

	$\mathcal{G}_1 \in$ 3-Col	$\mathcal{G}_1 \notin$ 3-Col
$\mathcal{G}_2 \in$ 3-Col	27 $(= 3 \times 3 \times 3)$	48 $(= 4 \times 4 \times 3)$
$\mathcal{G}_2 \notin$ 3-Col	36 $(= 3 \times 3 \times 4)$	64 $(= 4 \times 4 \times 4)$

Each of the four cells displays the largest value of m such that $[m, +\infty]$ is a certain answer of q over $\mathcal{K}$, under different assumptions on the 3-colorability of $\mathcal{G}_1$ and $\mathcal{G}_2$. To establish these values, one must first prove that every model has at least this many c-matches, and then exhibit a model that realizes the exact number. For the latter, we utilize our assumption that the graphs are planar, hence 4-colorable [Gonthier, 2008], which we use to show that the minimal number of c-matches is realized in a model that encodes proper 3- or 4-colorings of the graphs. We refer to 22 for the proof of this case analysis, as the number involved in the four cells are strictly the same. □

Theorem 19. *Tight CCQ answering over DL-Lite$_{\mathsf{pos}} \cap \mathcal{EL}$ ontologies for Cntd(z)-queries as defined in Kostylev and Reutter [2015], that are CCQs with $|\mathbf{z}| = 1$ is* DP*-hard w.r.t. data complexity.*

Proof. We recall that the Cntd queries from Kostylev and Reutter [2015] correspond to CCQs with exactly one counting variable. As in the previous reductions, we aim to force additional matches whenever an input graph is not 3-colorable, and the challenge is to track of the number of colors used to color the two graphs.

Having only a single counting variable forces us to count colors used for $\mathcal{G}_1$ in exactly the same as we count those used for $\mathcal{G}_2$. In particular, the asymmetry we introduced in the query must now be introduced into the ABox. This is done by considering a copy of our first graph. However, this is not enough as two different graphs could use the same additional color, making it impossible to detect with our single counting variable that both graphs are using more than three colors. Therefore, we will provide a set of basic colors *for each graph* and additionally check whether a graph uses a color that is intended for another graph. Concretely, we achieve this by connecting vertices from different graphs using a new role Diff, and by adding a new subquery that will generate new c-matches whenever two vertices connected by Diff use the same color.

Let us now give a more formal description of the construction. As mentioned earlier, we will introduce a copy $\mathcal{G}_0 = (\mathcal{V}_0, \mathcal{E}_0)$ of the graph $\mathcal{G}_1$. Without loss of generality, we can assume that $\mathcal{V}_0 \cap \mathcal{V}_1 \cap \mathcal{V}_2 = \emptyset$. As ABox individuals, we will use:

- an individual name u for each $u \in \mathcal{V}_0 \cup \mathcal{V}_1 \cup \mathcal{V}_2$, to represent our graphs;
- individuals $\mathsf{r}_0, \mathsf{g}_0, \mathsf{b}_0$ (resp. $\mathsf{r}_1, \mathsf{g}_1, \mathsf{b}_1$ and $\mathsf{r}_2, \mathsf{g}_2, \mathsf{b}_2$), intended to color $\mathcal{G}_0$ (resp. $\mathcal{G}_1$ and $\mathcal{G}_2$);
- auxiliary individuals for vertices (a_0, a_1, a_2, c, d, e) and auxiliary individuals for colors (r, g, b).

We then consider the following ABox:

$$
\begin{aligned}
\mathcal{A}_{(\mathcal{G}_1,\mathcal{G}_2)} = &\{\text{Vertex}(\mathsf{u}) \mid u \in \mathcal{V}_0 \cup \mathcal{V}_1 \cup \mathcal{V}_2\} \\
&\cup \{\text{Edge}(\mathsf{u}_1, \mathsf{u}_2) \mid (u_1, u_2) \in \mathcal{E}_0 \cup \mathcal{E}_1 \cup \mathcal{E}_2\} \\
&\cup \{\text{Edge}(\mathsf{a}_0, \mathsf{a}_0), \text{Edge}(\mathsf{a}_1, \mathsf{a}_1), \text{Edge}(\mathsf{a}_2, \mathsf{a}_2), \text{Edge}(\mathsf{c}, \mathsf{c}), \text{Edge}(\mathsf{d}, \mathsf{d})\} \\
&\cup \{\text{Diff}(\mathsf{u}_1, \mathsf{u}_2) \mid u_1 \in \mathcal{V}_i, u_2 \in \mathcal{V}_j, i \neq j\} \\
&\cup \{\text{Diff}(\mathsf{u}, \mathsf{a}_i) \mid u \in \mathcal{V}_j, i, j \in \{0,1,2\}, i \neq j\} \\
&\cup \{\text{Diff}(\mathsf{a}_0, \mathsf{a}_0), \text{Diff}(\mathsf{a}_1, \mathsf{a}_1), \text{Diff}(\mathsf{a}_2, \mathsf{a}_2), \text{Diff}(\mathsf{c}, \mathsf{c}), \text{Diff}(\mathsf{e}, \mathsf{e})\} \\
&\cup \{\text{Aux}_\text{e}(\mathsf{a}_0, \mathsf{a}_0), \text{Aux}_\text{e}(\mathsf{a}_1, \mathsf{a}_1), \text{Aux}_\text{e}(\mathsf{a}_2, \mathsf{a}_2), \text{Aux}_\text{e}(\mathsf{d}, \mathsf{d})\} \\
&\cup \{\text{Aux}_\text{e}(\mathsf{e}, \mathsf{u}) \mid u \in \mathcal{V}_0 \cup \mathcal{V}_1 \cup \mathcal{V}_2\} \\
&\cup \{\text{Aux}_\text{e}(\mathsf{u}, \mathsf{c}) \mid u \in \mathcal{V}_0 \cup \mathcal{V}_1 \cup \mathcal{V}_2\} \\
&\cup \{\text{Aux}_\text{d}(\mathsf{a}_0, \mathsf{a}_0), \text{Aux}_\text{d}(\mathsf{a}_1, \mathsf{a}_1), \text{Aux}_\text{d}(\mathsf{a}_2, \mathsf{a}_2), \text{Aux}_\text{d}(\mathsf{e}, \mathsf{e})\} \\
&\cup \{\text{Aux}_\text{d}(\mathsf{d}, \mathsf{u}) \mid u \in \mathcal{V}_0 \cup \mathcal{V}_1 \cup \mathcal{V}_2\} \\
&\cup \{\text{Aux}_\text{d}(\mathsf{u}, \mathsf{c}) \mid u \in \mathcal{V}_0 \cup \mathcal{V}_1 \cup \mathcal{V}_2\} \\
&\cup \{\text{HasCol}(\mathsf{a}_i, t) \mid t \in \{\mathsf{r}_i, \mathsf{g}_i, \mathsf{b}_i\}, i \in \{0,1,2\}\} \\
&\cup \{\text{HasCol}(\mathsf{d}, \mathsf{r}), \text{HasCol}(\mathsf{d}, \mathsf{g}), \text{HasCol}(\mathsf{d}, \mathsf{b})\} \\
&\cup \{\text{HasCol}(\mathsf{e}, \mathsf{r}), \text{HasCol}(\mathsf{e}, \mathsf{g}), \text{HasCol}(\mathsf{e}, \mathsf{b})\} \\
&\cup \{\text{HasCol}(\mathsf{c}, \mathsf{r})\}.
\end{aligned}
$$

and the TBox $\mathcal{T} := \{\text{Vertex} \sqsubseteq \exists\text{HasCol}\}$. We denote by $\mathcal{K}_\mathcal{G} = (\mathcal{T}, \mathcal{A})$ the resulting KB.

We consider the three following subqueries:

$$
\begin{aligned}
q^{diff}(y) &= \exists y_1^d\ \exists y_2^d\ \exists y_c^d\ \text{Aux}_\text{d}(y, y_1^d) \wedge \text{Diff}(y_1^d, y_2^d) \wedge \text{HasCol}(y_1^d, y_c^d) \wedge \text{HasCol}(y_2^d, y_c^d) \\
q^{edge}(y) &= \exists y_1^e\ \exists y_2^e\ \exists y_c^e\ \text{Aux}_\text{e}(y, y_1^e) \wedge \text{Edge}(y_1^e, y_2^e) \wedge \text{HasCol}(y_1^e, y_c^e) \wedge \text{HasCol}(y_2^e, y_c^e) \\
q^{col}(y) &= \exists z\ \text{HasCol}(y, z)
\end{aligned}
$$

and let $q = \exists y\ q^{diff}(y) \wedge q^{edge} \wedge q^{col}$ be the complete CCQ, which corresponds to a Cntd query class as there is only one counting variable z. The query q is displayed in Figure 3.18.

Claim: $(\mathbf{a}_\emptyset, [10, +\infty]) \in q^\mathcal{K}$ iff $\mathcal{G}_1 \in$ 3-Col and $\mathcal{G}_2 \notin$ 3-Col.

We prove this claim using the following case analysis:

	$\mathcal{G}_1 \in$ 3-Col	$\mathcal{G}_1 \notin$ 3-Col
$\mathcal{G}_2 \in$ 3-Col	9 $(= 3 + 3 + 3)$	11 $(= 4 + 4 + 3)$
$\mathcal{G}_2 \notin$ 3-Col	10 $(= 3 + 3 + 4)$	12 $(= 4 + 4 + 4)$

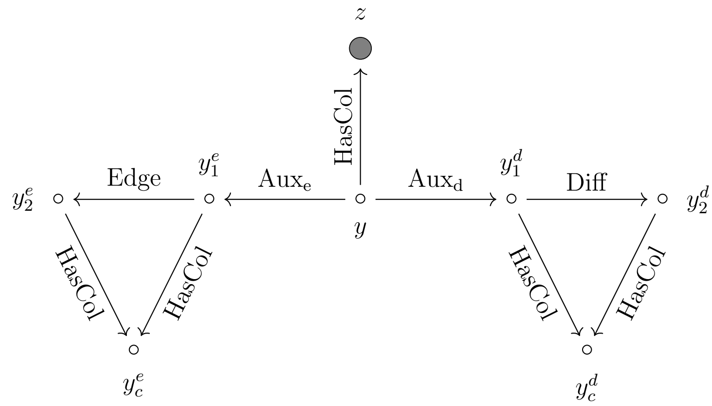


Figure 3.18: The $Cntd(z)$-CCQ q, which is the conjunction of q^{edge} (left part), q^{diff} (right part) and q^{col} (upper part).

To obtain the values in the preceding table, consider an arbitrary model $\mathcal{I}$ of $\mathcal{K}$, along with a homomorphism $f : \mathcal{C}_{\mathcal{K}} \to \mathcal{I}$. First observe that there are always 9 c-matches, which are obtained from the matches given by:

$$z \mapsto \mathsf{r}_i \mid \mathsf{g}_i \mid \mathsf{b}_i \qquad y, y_1^e, y_2^e, y_1^d, y_2^d \mapsto \mathsf{a}_i \qquad y_c^e, y_c^d \mapsto \mathsf{r}_i \qquad (i \in \{0, 1, 2\})$$

Hence $q_{\emptyset}^{\mathcal{I}} \geq 3 + 3 + 3 = 9$.

Furthermore, let us define $\tau_{\mathcal{I}} : \mathcal{V}_0 \cup \mathcal{V}_1 \cup \mathcal{V}_2 \to \Delta^{\mathcal{I}}$ as follows: $\tau_{\mathcal{I}}(u) = f(\mathsf{u}\text{HasCol})$. We'll use the notation $\tau_{\mathcal{I}}(\mathcal{V}_i)$ to refer to the set $\{\tau_{\mathcal{I}}(u) \mid u \in \mathcal{V}_i\}$. Notice that, if $\tau_{\mathcal{I}}(\mathcal{V}_i) \cap \tau_{\mathcal{I}}(\mathcal{V}_j) \neq \emptyset$ with $i \neq j$, that is, we have $u \in \mathcal{G}_i, v \in \mathcal{G}_j$ with $i \neq j$ and $\tau_{\mathcal{I}}(u) = \tau_{\mathcal{I}}(v)$, then we have 3 additional c-matches corresponding to the matches given by:

$$z, y_c^e \mapsto \mathsf{r} \mid \mathsf{g} \mid \mathsf{b} \qquad y, y_1^e, y_2^e \mapsto \mathsf{d} \qquad y_1^d \mapsto \mathsf{u} \qquad y_2^d \mapsto \mathsf{v} \qquad y_c^d \mapsto \tau_{\mathcal{I}}(u)$$

Therefore, in such a model $\mathcal{I}$, we have $q_{\emptyset}^{\mathcal{I}} \geq 9 + 3 = 12$, and thus sufficiently many c-matches w.r.t. the numbers in the table. We will therefore assume in the following that $\tau_{\mathcal{I}}(\mathcal{V}_i) \cap \tau_{\mathcal{I}}(\mathcal{V}_j) = \emptyset$ for $i \neq j$ (assumption (i)).

The same applies in the case where $\tau_{\mathcal{I}}(\mathcal{V}_i) \cap \{\mathsf{r}_j, \mathsf{g}_j, \mathsf{b}_j\} \neq \emptyset$ for $i \neq j$, as one can exhibit the same three additional c-matches by replacing the individual v by a_j in the latter definition of matches. Therefore, we can also assume in what follows that $\tau_{\mathcal{I}}(\mathcal{V}_i) \cap \{\mathsf{r}_j, \mathsf{g}_j, \mathsf{b}_j\} = \emptyset$ for all $i \neq j$ (assumption (ii)).

Finally, notice that if $\tau_{\mathcal{I}}$ introduces a monochromatic edge, i.e. an edge $(u, v) \in \mathcal{E}_0 \cup \mathcal{E}_1 \cup \mathcal{E}_2$ such that $\tau_{\mathcal{I}}(u) = \tau_{\mathcal{I}}(v)$, we again have 3 additional c-matches obtained from the matches given by:

$$z, y_c^d \mapsto \mathsf{r} \mid \mathsf{g} \mid \mathsf{b} \qquad y, y_1^d, y_2^d \mapsto \mathsf{e} \qquad y_1^e \mapsto \mathsf{u} \qquad y_2^e \mapsto \mathsf{v} \qquad y_c^e \mapsto \tau_{\mathcal{I}}(u)$$

Therefore, we can also restrict our attention to models without monochromatic edges (assumption (*iii*)). Any model that satisfies properties (*i*), (*ii*) and (*iii*) will be called *non-trivial*.

We now proceed to consider the four cases. In each case, the minimal number of c-matches is obtained by exhibiting a model built from colorings for each graph that use a minimal number of colors. The only important difference w.r.t preceding reductions is that when more than one graph utilizes a fourth color, we need to use distinct fourth colors for each graph. We now complete the proof by showing that every non-trivial model has at least the number of c-matches as listed in the table.

- $\mathcal{G}_1, \mathcal{G}_2 \in$ 3-Col: We have already seen that every model contains at least 9 c-matches.

- $\mathcal{G}_1 \notin$ 3-Col$, \mathcal{G}_2 \in$ 3-Col: Since $\mathcal{G}_0$ and $\mathcal{G}_1$ are not 3-colorable, any non-trivial model $\mathcal{I}$ must satisfy $\tau_\mathcal{I}(\mathcal{V}_0) \geq 4$ and $\tau_\mathcal{I}(\mathcal{V}_1) \geq 4$, due to assumption (*iii*). In particular, we have a vertex $u_0 \in \mathcal{V}_0$ (resp. $u_1 \in \mathcal{V}_1$) such that $\tau_\mathcal{I}(u_0) \notin \{\mathsf{r}_0, \mathsf{g}_0, \mathsf{b}_0\}$ (resp. $\tau_\mathcal{I}(u_1) \notin \{\mathsf{r}_1, \mathsf{g}_1, \mathsf{b}_1\}$). This yields the following matches:

  $$z \mapsto \tau_\mathcal{I}(u_i) \qquad y \mapsto \mathsf{u_i} \qquad y_1^e, y_2^e, y_1^d, y_2^d \mapsto \mathsf{c} \qquad y_c^e, y_c^d \mapsto \mathsf{r} \qquad (i \in \{0,1\})$$

  which give rise to two new c-matches because of assumptions (*i*) (ensuring the two colors $\tau_\mathcal{I}(u_0)$ and $\tau_\mathcal{I}(u_1)$ are different) and (*ii*) (ensuring $\tau_\mathcal{I}(u_0)$ and $\tau_\mathcal{I}(u_1)$ are different from the colors in the 9 basic c-matches). Hence, every non-trivial model contains at least 11 c-matches.

- $\mathcal{G}_1 \in$ 3-Col$, \mathcal{G}_2 \notin$ 3-Col: Since $\mathcal{G}_2$ is not in 3-Col, any non-trivial model $\mathcal{I}$ must satisfy $\tau_\mathcal{I}(\mathcal{V}_2) \geq 4$ because of assumption (*iii*). In particular, we have a vertex $u_2 \in \mathcal{V}_2$ such that $\tau_\mathcal{I}(u_2) \notin \{\mathsf{r}_2, \mathsf{g}_2, \mathsf{b}_2\}$. This provides a new match given by:

  $$z \mapsto \tau_\mathcal{I}(u_2) \qquad y \mapsto \mathsf{u}_2 \qquad y_1^e, y_2^e, y_1^d, y_2^d \mapsto \mathsf{c} \qquad y_c^e, y_c^d \mapsto \mathsf{r}$$

  which gives rise to a new c-match because of the assumption (*ii*) (which ensures $\tau_\mathcal{I}(u_2)$ is different from the colors in the 9 basic c-matches). Hence, every non-trivial model contains at least 10 c-matches.

- $\mathcal{G}_1, \mathcal{G}_2 \notin$ 3-Col: We can proceed similarly to the two previous cases to exhibit $u_0 \in \mathcal{V}_0, u_1 \in \mathcal{V}_1, u_2 \in \mathcal{V}_2$ that are assigned new colors, providing three new matches given by:

  $$z \mapsto \tau_\mathcal{I}(u_i) \qquad y \mapsto \mathsf{u_i} \qquad y_1^e, y_2^e, y_1^d, y_2^d \mapsto \mathsf{c} \qquad y_c^e, y_c^d \mapsto \mathsf{r} \qquad (i \in \{0,1,2\})$$

which give rise to three new c-matches because of assumptions (*i*) (the colors $\tau_{\mathcal{I}}(u_0), \tau_{\mathcal{I}}(u_1), \tau_{\mathcal{I}}(u_2)$ are all different) and (*ii*) (they are also different from the colors in the 9 basic c-matches). Hence, we have that every non-trivial model contains at least 12 matches. □

4 Rooted CCQs

In this chapter, we explore whether a first structural restriction on CCQs allows us to lower the complexity. The reductions used to prove lower complexity bounds in Chapter 3 mostly rely on disconnected CCQs admitting fully anonymous matches in some models. A natural idea is thus to restrict to the class of *rooted* CCQs, for which each match must involve at least one individual per connected component of the query. Rootedness is already known to lower the complexity in several settings (see *e.g.* Lutz [2008]; Calvanese et al. [2020a]), notably leading to tractable data complexity for CQ answering under bag semantics in DL-Lite$_{\mathsf{core}}$ (Nikolaou et al. [2019]).

In the case of CCQs, we show that this restriction does not lead to lower complexity than in the general case. Intuitively, one can use existential variables to connect a root of a query to its counting variables, thereby making it possible to bypass the rootedness restriction. Rooted CCQs are investigated in Section 4.2 and lead to the very same complexity results as in the general case, recalled in Table 4.1.

	Exhaustive rooted		Rooted	
	Data	Combined	Data	Combined
DL-Lite$^{\mathcal{H}}_{\mathsf{pos}}$, $\mathcal{ELI}$, $\mathcal{ALCHI}$	coNP-c	coNEXP-c	coNP-c	2EXP-c
$\mathcal{ALC}$, $\mathcal{ALCH}$	coNP-c	EXP-c	coNP-c	2EXP-c
$\mathcal{EL}$, $\mathcal{ELH}_\bot$	coNP-c	PSPACE-c	coNP-c	2EXP-c
DL-Lite$_{\mathsf{pos}}$, DL-Lite$_{\mathsf{core}}$	TC^0-c	PP-c	coNP-c	coNEXP-c

Table 4.1: (Exhaustive) rooted CCQs answering: worst-case complexity.

These results lead us to consider an additional constraint: disallowing existential variables. We term such CCQs *exhaustive* as this further restriction means that we count every match for the query, since (plain) matches now coincide with counting matches. As our exploration reveals, exhaustive rooted CCQ answering enjoys lower complexity bounds, summarized in Table 4.1. In data complexity, we obtain a tractable TC^0 case for DL-Lite$_{\mathsf{core}}$, by showing that a carefully defined notion of the canonical model always yields the tightest certain answer. In combined complexity, we exhibit four situations ranging from PP-completeness to coNEXP-completeness. All four results are obtained by refinements of interlacing functions. In Section 4.3, a first refinement for $\mathcal{ALCHI}$ KBs allows us to ensure existence of countermodels with exponential size, if a countermodel does exist, hence yielding the coNEXP upper bound. In Section 4.4, we construct an interlacing function tailored to the absence of inverse role in $\mathcal{ALCH}$, from which we derive a EXP procedure, dropping to PSPACE when considering $\mathcal{ELH}_\bot$. In Section 4.5 we focus on DL-Lite$_{\mathsf{core}}$ and obtain a PP completeness result relying, as mentioned for data complexity, on the canonical model yielding the tightest certain answer.

Interestingly, *all* of the lower bounds for combined complexity explored in the present chapter rely on a binary encoding of the input integer. This is in contrast with the general case (Chapter 3), and whether the complexity of (exhaustive) rooted CCQ answering drops when requiring m to be written in unary remains an open question.

Organization of Chapter 4

4.1 Preliminaries

To define rootedness, we first recall the definition of the labeled directed graph associated to a CCQ q given in Chapter 3: each term t is represented by a vertex v_t labeled by t and by concept names A such that $\mathrm{A}(t) \in q$, and an directed edge (v_{t_1}, v_{t_2}) labeled with P is added for each atom $\mathrm{P}(t_1, t_2) \in q$. To easily distinguish the status of each term (and often to omit the name of the term), the node v_t is depicted as $\bullet$ if $t \in \mathsf{Ind} \cup \mathbf{x}$, as $\circ$ if $t \in \mathbf{y}$, and as ⬤ if $t \in \mathbf{z}$. Rootedness is then defined as follows:

Definition 40. *A CCQ q is* rooted *if every connected component of the graph associated to q contains at least one answer variable or individual name.*

We may focus on Boolean CCQs without loss of generality, since replacing an answer variable by the individual of interest preserves rootedness. Observe that for Boolean CCQs, the rootedness restriction enforces that every connected component of the graph contains at least one individual name.

As mentioned in the introduction of this chapter, we also consider the subclass of exhaustive queries, which basically corresponds to the class of $Count()$-CQs considered in Kostylev and Reutter [2015].

Definition 41. *A CCQ $q(\mathbf{x}) := \exists \mathbf{y}\, \exists \mathbf{z}\ \psi(\mathbf{x}, \mathbf{y}, \mathbf{z})$ is* exhaustive *if $\mathbf{y} = \emptyset$.*

For exhaustive CCQs, the notions of matches and counting matches coincide. When it is clear we are considering such queries, we will use the term matches for simplicity, and hence be interest in the number of matches.

4.2 A weak notion of rootedness

In this section, we prove that rootedness is not sufficient to lower the complexity of CCQ answering compared to the general case. For combined complexity, we show how to directly reduce CCQ answering to rooted CCQ answering. For data complexity, we proceed by reduction from (the complement of) the 3-Col problem to obtain coNP-hardness. Notice this also closes the pending lower bounds from Section 3.5.3.

4.2.1 Combined complexity: from CCQs to rooted CCQs

For the combined complexity measure, we exhibit a direct reduction from CCQ answering to rooted CCQ answering. This is achieved by a slight modification of the ontology, essentially requiring that each (relevant) element in a model must connect to a root-like element via a dedicated role. Such root-like elements are further counted by a subquery, which is duplicated enough times so that each unknown instance of a root-like element causes an exponential number of new counting matches. Interestingly, the number of copies of this subquery to introduce is data-dependent, which is why this reduction doesn't work for data complexity. The key ingredient is hence to be able to count the root-like elements of a model without counting *all* elements of the model. Indeed, as our constructed CCQ must be rooted, the only way to count a root-like element to which an arbitrary element e in the model is connected is by considering a path from a fixed individual (the root of our query) to e. This is precisely what existential variables allow us to do: to consider such a path without counting it.

The precise result, sufficient for our purposes, is as follows.

Theorem 20. *Let $\mathcal{L}$ be a sublogic of $\mathcal{ELHI}$ extending either $\mathcal{EL}$ or DL-Lite$_{\mathsf{pos}}$. Then CCQ answering over $\mathcal{L}$ KBs can be polynomially reduced to rooted CCQ answering over $\mathcal{L}$ KBs.*

Proof. Let $\mathcal{L}$ be a sublogic of $\mathcal{ELHI}$ extending either $\mathcal{EL}$ or DL-Lite$_{\mathsf{pos}}$. Let $\mathcal{K} := (\mathcal{T}, \mathcal{A})$ be an $\mathcal{L}$ KB, q be a CCQ and m a candidate integer. We begin with a slight reformulation, by noticing that we can assume without loss of generality that q doesn't contain any individual name. Indeed, by replacing each occurrence of an individual a in q by an existential variable y_{a}, adding the atom $\mathrm{IsInd}_{\mathsf{a}}(y_{\mathsf{a}})$ to q and the assertion $\mathrm{IsInd}_{\mathsf{a}}(\mathsf{a})$ to $\mathcal{A}$, we obtain such an individual-free query q' and extended ABox $\mathcal{A}'$ such that n is a certain answer to q over $\mathcal{K}$ iff n is a certain answer to q' over $\mathcal{K}' := (\mathcal{T}, \mathcal{A}')$. Henceforth the terms of q are either existential or counting variables.

We now proceed to the main reduction and build a new $\mathcal{L}$ KB $\mathcal{K}' := (\mathcal{T}', \mathcal{A}')$, a rooted CCQ q' and a new integer m' such that m is a certain answer to q over $\mathcal{K}$ iff m' is a certain answer to q' over $\mathcal{K}'$. The built KB $\mathcal{K}'$ extends the original $\mathcal{K}$, that is $\mathcal{T} \subseteq \mathcal{T}'$ and $\mathcal{A} \subseteq \mathcal{A}'$, with additional assertions and axioms that are further detailed.

The general idea is to ask for relevant elements of a model $\mathcal{I}'$ of $\mathcal{K}'$, those are the elements ensuring modelhood, to be directly connected to one of the two roots that we provide as part of $\mathcal{A}'$, namely a and b. We enforce this behaviour with a variation of the axiom $\top \sqsubseteq \exists \mathrm{toRoot}.\top$ in $\mathcal{T}'$ and by counting the number of root-like elements in $\mathcal{I}'$ with a subquery of q' containing an atom $\mathrm{toRoot}(y, z)$. By considering enough copies of the latter subquery ($\approx \log_{\frac{3}{2}} m$), we make sure the introduction of a third root-like element in $\mathcal{I}'$ disqualifies $\mathcal{I}'$ as an optimal model, and we can hence focus on models that only contain the two provided roots. In such a model, q can easily be rewritten as a rooted CCQ since each relevant element of the model is connected either to a or to b.

The reason why we provide two roots is to allow to distinguish two "sides" in a model $\mathcal{I}'$ of $\mathcal{K}'$: the *main* side containing a and the *auxiliary* side containing

b. The purpose of the main side in $\mathcal{I}'$ is to represent (the relevant elements of) a model $\mathcal{I}$ of the original KB $\mathcal{K}$, while the auxiliary side provides basic matches for the query q'. To this end, a is chosen among individual names from $\mathcal{A}$, or as a fresh new individual if ever $\mathcal{A} = \emptyset$, while b is always chosen as a fresh new individual name. Changing side will be represented by following a fresh new role toSide from a. We add in $\mathcal{A}'$ the two following assertions:

$$\text{toSide}(\mathsf{a}, \mathsf{a}) \qquad \text{toSide}(\mathsf{a}, \mathsf{b})$$

Since a is part of the main side, which aims to represent a model of $\mathcal{K}$, we ask each individual element on this side to be connected to a with the following assertions:

$$\text{toRoot}(\mathsf{a}, \mathsf{a}) \qquad \text{toRoot}(\mathsf{c}, \mathsf{a}) \qquad (\mathsf{c} \in \mathsf{Ind}(\mathcal{A}))$$

On the auxiliary side, we allow b to be its own root and to satisfy every possible fact related to the original KB $\mathcal{K}$:

$$\text{toRoot}(\mathsf{b}, \mathsf{b}) \qquad \mathrm{A}(\mathsf{b}) \qquad \mathrm{P}(\mathsf{b}, \mathsf{b}) \qquad (\mathrm{A}, \mathrm{P} \in \mathsf{sig}(\mathcal{K}))$$

To prevent the main side from reusing facts from the auxiliary side, we introduce M copies of the following subqueries in q' (the value of M will be specified latter) to capture outgoing and incoming roles involving b:

$$q^{(i)}_{\mathrm{P},\mathsf{b}} := \exists z\ \mathrm{P}(\mathsf{b}, z) \quad q^{(i)}_{\mathrm{P}^-,\mathsf{b}} := \exists z\ \mathrm{P}(z, \mathsf{b}) \quad (\mathrm{P} \in \mathsf{sig}(\mathcal{T}) \cup \{\text{toRoot}\}, 1 \leq i \leq M)$$

Notice each copy is rooted and embeds on b in all models of $\mathcal{A}'$. We now aim to make sure that relevant elements on the main side are connected to a. We introduce a fresh new concept $\text{Aux}_\top$ aiming to capture all relevant elements by subsuming all concept that might occur in the original KB $\mathcal{K}$. We hence add in $\mathcal{T}'$ the following axioms, depending on which DL the logic $\mathcal{L}$ is extending:

$\mathcal{L}$ extends $\mathcal{EL}$	$\mathcal{L}$ extends DL-Lite$_{\mathsf{pos}}$
$\top \sqsubseteq \text{Aux}_\top$	$\mathrm{A} \sqsubseteq \text{Aux}_\top \quad (\mathrm{A} \in \mathsf{sig}(\mathcal{T}))$ $\exists \mathrm{P}.\top \sqsubseteq \text{Aux}_\top \quad (\mathrm{P} \in \mathsf{sig}(\mathcal{T}))$ $\exists \mathrm{P}^-.\top \sqsubseteq \text{Aux}_\top \quad (\mathrm{P} \in \mathsf{sig}(\mathcal{T}))$

Notice the use of inverse roles is only needed if the logic $\mathcal{L}$ already allows it. We now require that such elements are connected to a root-like element with the axiom:

$$\text{Aux}_\top \sqsubseteq \exists \text{toRoot}.\top$$

To enforce the root a to be reused in optimal models, we proceed inductively: individuals from $\mathcal{A}$ and individual a already satisfy this condition, and we build a subquery $q_{\mathrm{P},\mathsf{a}}$ which counts the root-like elements used by the P-neighbours of an element already connected to a known root. We consider in fact N copies of this subquery per role P and per inverse role P^- from the signature of $\mathcal{T}$:

$$\begin{aligned} q^{(j)}_{\mathrm{P}^\pm,\mathsf{a}} := \ & \exists y, y_1, y_f, y_2, y'_1, y'_f, y'_2 \ \exists z \ \mathrm{toSide}(\mathsf{a}, y) \\ & \wedge \mathrm{toRoot}(y_1, y) \wedge \mathrm{fetch}(y_1, y_f) \wedge \mathrm{fetch}(y_2, y_f) \\ & \wedge \mathrm{P}^\pm(y_2, y'_2) \\ & \wedge \mathrm{fetch}(y'_2, y'_f) \wedge \mathrm{fetch}(y'_1, y'_f) \wedge \mathrm{toRoot}(y'_1, z) \end{aligned} \qquad \begin{pmatrix} \mathrm{P} \in \mathsf{sig}(\mathcal{T}) \\ 0 \leq j \leq N \end{pmatrix}$$

in which fetch is a fresh new role name whose purpose is to let a be a match for each $q^{(i)}_{\mathrm{P}^\pm,\mathsf{a}}$. It is indeed essential that a is already counted as a basic c-match, but since we don't want to introduce auxiliary facts on a (which would restrict the possible models represented on the main side), the role fetch allows a to borrow facts from b thanks to the following assertions in $\mathcal{A}'$:

$$\mathrm{fetch}(\mathsf{a}, \mathsf{b}) \qquad \mathrm{fetch}(\mathsf{b}, \mathsf{b})$$

The following axiom in $\mathcal{T}'$ is therefore needed to ensure queries $q^{(j)}_{\mathrm{P}^\pm,\mathsf{a}}$ can still map on other relevant elements:

$$\mathrm{Aux}_\top \sqsubseteq \exists \mathrm{fetch}.\top$$

Finally, the original query becomes:

$$q_{\mathrm{rooted}} := \exists y \ \mathrm{toSide}(\mathsf{a}, y) \wedge q \wedge \bigwedge_{v \in terms(q)} \mathrm{toRoot}(v, y)$$

and we let q' be the conjunction of all the above subqueries.

Notice there are always: 1 c-match (on b) for each $q^{(i)}_{\mathrm{P},\mathsf{b}}$, 1 c-match (on b) for q_{rooted} and 2 c-matches (on a or b) for each $q^{(j)}_{\mathrm{P},\mathsf{a}}$ for every model of $\mathcal{A}'$. Together, it yields $1^{2M(|\mathsf{sig}(\mathcal{T})|+1)} \times 1 \times 2^{2N|\mathsf{sig}(\mathcal{T})|}$ basic c-matches for the whole query q'. Therefore, if there are m c-matches in a model of $\mathcal{K}$, we should aim for $1^{2M(|\mathsf{sig}(\mathcal{T})|+1)} \times (m+1) \times 2^{2N|\mathsf{sig}(\mathcal{T})|}$ c-matches in the corresponding model of $\mathcal{K}'$. We hence set $n' := (n+1) \times 2^{2N|\mathsf{sig}(\mathcal{T})|}$. We now discuss how to set M and N. We want any new c-match for a subquery $q^{(i)}_{\mathrm{P},\mathsf{b}}$ to allow for more than m' c-matches for the whole q', that is:

$$2^M \times 1^{2M(|\mathsf{sig}(\mathcal{T})|+1)-M} \times 1 \times 2^{2N|\mathsf{sig}(\mathcal{T})|} > (m+1) \times 2^{2N|\mathsf{sig}(\mathcal{T})|}$$

Hence we set $M := \lfloor \log_2(m+1) \rfloor + 1$. We proceed as well for N, aiming for:

$$1^{2M(|\mathsf{sig}(\mathcal{T})|+1)} \times 1 \times 3^N \times 2^{2N|\mathsf{sig}(\mathcal{T})|-N} > (m+1) \times 2^{2N|\mathsf{sig}(\mathcal{T})|}$$

Hence we set $N := \lfloor \log_{\frac{3}{2}}(m+1) \rfloor + 1$.

It remains to prove that $[m', +\infty]$ is a certain answer to q' over $\mathcal{K}'$ iff $[m, +\infty]$ is a certain answer to q over $\mathcal{K}$.

(⇒). Assume $[m, +\infty]$ is *not* a certain answer to q over $\mathcal{K}$, that is we have a countermodel $\mathcal{I}$ of $\mathcal{K}$ for q and m. Denote m_0 the number of matches in $\mathcal{I}$. We now build a countermodel $\mathcal{I}'$ of $\mathcal{K}'$ for q' and m'. If the initial ABox $\mathcal{A}$ is empty, we assume, up to a renaming, that $\mathsf{a} \in \Delta^{\mathcal{I}}$ (recall that otherwise, that is $\mathcal{A} \neq \emptyset$, we chose a among $\mathsf{Ind}(\mathcal{A})$).

The domain of $\mathcal{I}'$ is $\Delta^{\mathcal{I}} \cup \{\mathsf{a}, \mathsf{b}\}$, and interpretations of concept and role names are given as follow:

$$
\begin{array}{rcll}
\mathrm{A}^{\mathcal{I}'} & := & \mathrm{A}^{\mathcal{I}} \cup \{\mathsf{b}\} & (\mathrm{A} \in \mathsf{sig}(\mathcal{T})) \\
\mathrm{Aux}_{\top}{}^{\mathcal{I}'} & := & \Delta^{\mathcal{I}'} & \\
\mathrm{P}^{\mathcal{I}'} & := & \mathrm{P}^{\mathcal{I}} \cup \{(\mathsf{b}, \mathsf{b})\} & (\mathrm{P} \in \mathsf{sig}(\mathcal{T})) \\
\mathrm{toSide}^{\mathcal{I}'} & := & \{(\mathsf{a}, \mathsf{a}), (\mathsf{a}, \mathsf{b})\} & \\
\mathrm{toRoot}^{\mathcal{I}'} & := & \{(\mathsf{a}, \mathsf{a}), (\mathsf{b}, \mathsf{b})\} \cup \{(e, \mathsf{a}) \mid e \in \Delta^{\mathcal{I}}\} & \\
\mathrm{fetch}^{\mathcal{I}'} & := & \{(e, \mathsf{b}) \mid e \in \Delta^{\mathcal{I}}\} &
\end{array}
$$

It is easily verified that $\mathcal{I}'$ is a model of $\mathcal{K}'$, mainly from the following facts:

- $\mathcal{I}$ is a model of $\mathcal{K}$;
- Facts in $\mathcal{I}'$ involving elements of $\Delta^{\mathcal{I}}$ ensure satisfaction of the additional axioms of $\mathcal{T}'$;
- In the absence of negative inclusions (recall $\mathcal{L}$ belongs to $\mathcal{ELHI}$), all the facts on b do not yield any contradiction;
- Since in any case a already belonged to $\Delta^{\mathcal{I}}$, it ensures all axioms with shapes $\top \sqsubseteq \mathrm{B}$ are already satisfied on a in $\mathcal{I}$.

It is further direct that $\mathcal{I}'$ contains exactly $(m_0 + 1) \times 2^{2N|\mathsf{sig}(\mathcal{T})|} < m'$ matches for q', yielding the desired countermodel.

(⇐). Assume that $[m, +\infty]$ is a certain answer to q over $\mathcal{K}$. Consider a model $\mathcal{I}'$ of $\mathcal{K}'$. If b is reached by any new fact involving a role $\mathrm{P}^{\pm}$ from $\mathsf{sig}(\mathcal{T}) \cup \{\mathrm{toRoot}\}$ in $\mathcal{I}'$, it yields a new c-match for all the corresponding $q_{\mathrm{P}^{\pm},\mathsf{b}}$ and ensures existence of at least m' c-matches for q' in $\mathcal{I}'$. Otherwise we say that b is isolated and we consider the submodel $\mathcal{I}^*$ of $\mathcal{I}'$ obtained by only keeping the connected components containing an element from $\mathsf{Ind}(\mathcal{A}) \cup \{\mathsf{a}\}$. Using the queries $q_{\mathrm{P}^{\pm},\mathsf{a}}$ and b being isolated, we prove by induction on these connected components that if any element from $\mathcal{I}^*$ is not connected by toRoot to a, then there exists at least m' matches in $\mathcal{I}^*$, hence in $\mathcal{I}'$. Otherwise, we consider the model $\mathcal{I}$ of $\mathcal{K}$ obtained from $\mathcal{I}^*$ by dropping b. Modelhood is indeed ensured from modelhood of $\mathcal{I}'$, hence of $\mathcal{I}^*$, and the fact that b was isolated before being dropped. By hypothesis, we have

m c-matches for q in $\mathcal{I}$. It can be verified that these m c-matches correspond to exactly m new c-matches for q_{rooted} in $\mathcal{I}'$, ensuring existence of m' c-matches for the whole q' in $\mathcal{I}'$. □

Using Theorem 20, we can import from Chapter 3 the three lower bounds for combined complexity.

Corollary 1. *Rooted CCQ answering over $\mathcal{EL}$ ontologies is* 2EXP*-hard.*

Proof. Theorem 20 combined with Theorem 13. □

Corollary 2. *Rooted CCQ answering over DL-Lite$^{\mathcal{H}}_{\mathsf{pos}}$ ontologies is* 2EXP*-hard.*

Proof. Theorem 20 combined with Theorem 14. □

Corollary 3. *Rooted CCQ answering over DL-Lite$_{\mathsf{pos}}$ ontologies is* coNEXP*-hard.*

Proof. Theorem 20 combined with Theorem 15. □

4.2.2 Two reductions for data complexity

We now move to data complexity, for which we prove the coNP-hardness of answering rooted CCQs over DL-Lite$_{\mathsf{pos}} \cap \mathcal{EL}$, hence coNP-completeness for all DLs up to $\mathcal{ALCHI}$ (recall that CCQ answering over $\mathcal{ALCHI}$ KBs is in coNP from Theorem 7). This also proves Theorem 16 from Chapter 3.

Theorem 21. *Rooted CCQ answering over DL-Lite$_{\mathsf{pos}} \cap \mathcal{EL}$ ontologies is* coNP*-complete w.r.t. data complexity.*

Proof. The proof borrows some ideas from the proofs of Lemmas 12 and 16 from Kostylev and Reutter [2015]. It proceeds by reduction from the well-known coNP-complete 3-Col problem: given an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, return yes iff $\mathcal{G}$ has *no* 3-coloring, i.e., a mapping from $\mathcal{V}$ to $\{\mathsf{red}, \mathsf{green}, \mathsf{blue}\}$ such that adjacent vertices map to different colors (equivalently: there is no monochromatic edge).

The reduction uses atomic roles Edge and toVertex to encode the graph and HasCol to assign colors. The DL-Lite$_{\mathsf{pos}} \cap \mathcal{EL}$ TBox $\mathcal{T}$ has a single axiom: Vertex $\sqsubseteq \exists$HasCol.$\top$. The ABox $\mathcal{A}_{\mathcal{G}}$ contains an assertion Vertex(v) for each vertex $v \in \mathcal{V}$ and an assertion Edge(u, v) for each edge $\{u, v\} \in \mathcal{E}$. All vertices are connected to a special root individual a: toVertex(a, v) for each $v \in \mathcal{V}$. The three colors are represented by individuals r, g and b. To ensure that the query has matches in every model, we include a 'dummy' vertex individual $\mathsf{a_v}$ and the following assertions: toVertex($\mathsf{a}, \mathsf{a_v}$), Edge($\mathsf{a_v}, \mathsf{a_v}$), HasCol($\mathsf{a_v}, \mathsf{r}$), HasCol($\mathsf{a_v}, \mathsf{g}$), and HasCol($\mathsf{a_v}, \mathsf{b}$).

Let $\mathcal{K}_{\mathcal{G}} := (\mathcal{T}, \mathcal{A}_{\mathcal{G}})$ be the built KB. A part of the canonical model of $\mathcal{K}_{\mathcal{G}}$ is depicted in Figure 4.1.

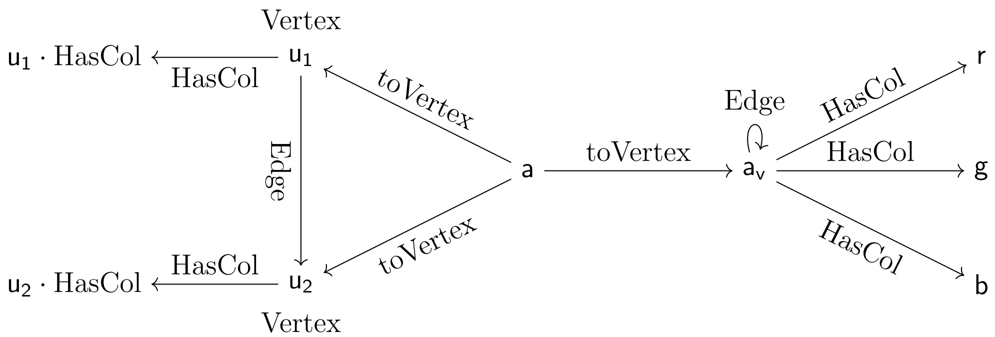


Figure 4.1: A part of $\mathcal{C}_{\mathcal{K}_\mathcal{G}}$ with $(u_1, u_2) \in \mathcal{E}$.

The query q, depicted in Figure 4.2, is the conjunction of the two subqueries:

$$\begin{aligned} q^{edge} &= \exists y_c\ \exists z_1\ \exists z_2\ \text{toVertex}(\mathsf{a}, z_1) \wedge \text{toVertex}(\mathsf{a}, z_2) \wedge \\ &\quad \text{Edge}(z_1, z_2) \wedge \text{HasCol}(z_1, y_c) \wedge \text{HasCol}(z_2, y_c) \\ q^{col} &= \exists y\ \exists z\ \text{toVertex}(\mathsf{a}, y) \wedge \text{HasCol}(y, z) \end{aligned}$$

serving respectively to detect monochromatic edges and to check whether any additional colors have been introduced.

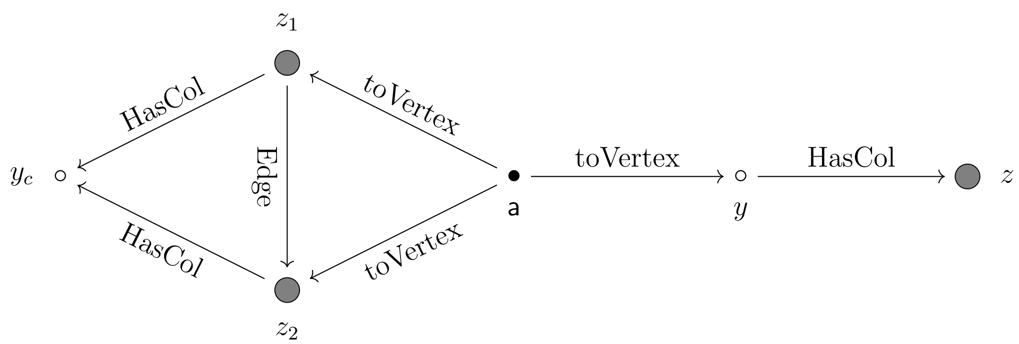


Figure 4.2: The rooted CCQ q, being the conjunction of q^{edge} (left) and q^{col} (right).

It is not hard to see that $[3, +\infty]$ is a certain answer to q over $\mathcal{K}_\mathcal{G}$. Indeed, there are at least 9 matches of q in any model $\mathcal{I}$ of $\mathcal{K}$, given by:

$$z_1, z_2, y \mapsto \mathsf{a_v} \qquad y_c \mapsto \mathsf{r} \mid \mathsf{g} \mid \mathsf{b} \qquad z \mapsto \mathsf{r} \mid \mathsf{g} \mid \mathsf{b}$$

These 9 matches give rise to 3 c-matches for q, corresponding to the three ways of mapping z. To complete the proof, we establish the following claim:

$$[4, +\infty] \text{ is a certain answer to } q \text{ over } \mathcal{K}_\mathcal{G} \text{ iff } \mathcal{G} \notin 3\text{-}\textsc{Col}.$$

($\Rightarrow$) Assume $[4, +\infty]$ is a certain answer to q over $\mathcal{K}_\mathcal{G}$, and take some possible coloring $\tau : \mathcal{V} \to \{\mathsf{r}, \mathsf{g}, \mathsf{b}\}$ of the graph $\mathcal{G}$. Let $\mathcal{I}_\tau^\mathcal{G}$ be the model of $\mathcal{K}_\mathcal{G}$ whose domain is $\mathsf{Ind}(\mathcal{A}_\mathcal{G})$ and which interprets concept Vertex and roles toVertex and Edge exactly following the ABox, and which interprets HasCol according to τ:

$$\text{HasCol}^{\mathcal{I}_\tau^\mathcal{G}} = \{(\mathsf{a_v}, \mathsf{r}), (\mathsf{a_v}, \mathsf{g}), (\mathsf{a_v}, \mathsf{b})\} \cup \{(\mathsf{v}, \tau(v)) \mid v \in \mathcal{V}\}$$

Intuitively, $\mathcal{I}_\tau$ is obtained from the canonical model by replacing the element $\mathsf{v} \cdot \text{HasCol}$ with $\tau(v)$.

By hypothesis, there is a fourth c-match π for q in $\mathcal{I}_\tau^\mathcal{G}$. It is easily verified that the additional match can only result from the atom $\text{Edge}(z_1, z_2)$ being mapped onto an edge $\text{Edge}(\mathsf{u_1}, \mathsf{u_2})$ that is different from $\text{Edge}(\mathsf{a_v}, \mathsf{a_v})$. From the definition of $\mathcal{I}_\tau^\mathcal{G}$, this implies that the edge (u_1, u_2) of $\mathcal{G}$ is monochromatic, both vertices sharing the color $\pi(y_c)$. Thus, τ is not a 3-coloring. As this construction holds for any possible coloring τ, we obtain $\mathcal{G} \notin$ 3-Col.

($\Leftarrow$) Assume $\mathcal{G} \notin$ 3-Col, and take some model $\mathcal{I}$ of $\mathcal{K}_\mathcal{G}$. There is a homomorphism $f : \mathcal{C}_{\mathcal{K}_\mathcal{G}} \to \mathcal{I}$ (which preserves individual names). Define $\tau : \mathcal{V} \to \Delta^\mathcal{I}$ as follows: $\tau(u) = f(\mathsf{u} \cdot \text{HasCol})$. There are two cases to consider:

- If there exists $u \in \mathcal{V}$ such that $\tau(u) \notin \{\mathsf{r}, \mathsf{g}, \mathsf{b}\}$, then this provides a match of q in $\mathcal{I}$ given by $z \mapsto \tau(u)$ and $y \mapsto \mathsf{u}^\mathcal{I}$, whose restriction to the counting variables is a new c-match.

- Else, since $\mathcal{G} \notin$ 3-Col, there exists an edge $(u_1, u_2) \in \mathcal{E}$ such that $\tau(u_1) = \tau(u_2)$. It provides a new match given by:

$$z \mapsto \mathsf{r} \qquad y \mapsto \mathsf{a_v} \qquad z_1 \mapsto \mathsf{u_1} \qquad z_2 \mapsto \mathsf{u_2} \qquad y_c \mapsto \tau(u_1)\ (= \tau(u_2))$$

In both cases, there is a fourth c-match for q. As this holds for any model $\mathcal{I}$ of $\mathcal{K}_\mathcal{G}$, it proves $[4, +\infty]$ is certain answer to q over $\mathcal{K}_\mathcal{G}$. □

To conclude with this form of rooted queries, we turn to the tightest certain answer variant. Based on the previous proof, we can adapt the reduction to obtain a DP lower bound for this tightest variant. This also proves Theorem 17 from Chapter 3.

Theorem 22. *Tight-rooted CCQ answering in DL-Lite*$_{\mathsf{pos}} \cap \mathcal{EL}$ *is* DP*-hard w.r.t. data complexity.*

Proof. We give a reduction from the following problem (DP-complete due to Garey et al. [1976]): given *planar* graphs $\mathcal{G}_1$ and $\mathcal{G}_2$, decide if $\mathcal{G}_1 \in$ 3-Col and $\mathcal{G}_2 \notin$ 3-Col.

Let the TBox $\mathcal{T}$ and ABoxes $\mathcal{A}_{\mathcal{G}_1}, \mathcal{A}_{\mathcal{G}_2}$ be defined as in the proof of Theorem 21. Rename the individuals to ensure $\mathsf{Ind}(\mathcal{A}_{\mathcal{G}_1}) \cap \mathsf{Ind}(\mathcal{A}_{\mathcal{G}_2}) = \emptyset$, then set $\mathcal{K} = (\mathcal{T}, \mathcal{A}_{\mathcal{G}_1} \cup \mathcal{A}_{\mathcal{G}_2})$. Let q_1^{color} and q_1^{edge} (resp. q_2^{color} and q_2^{edge}) be defined as before, but using disjoint variables and the root individual from the $\mathcal{A}_{\mathcal{G}_1}$ (resp. $\mathcal{A}_{\mathcal{G}_2}$). The challenge is to make sure that we can determine the 3-colorability status of the two graphs solely by looking at the number of c-matches of the query. To be able to distinguish $\mathcal{G}_1$ from $\mathcal{G}_2$, we introduce an asymmetry by duplicating the color counter query for $\mathcal{G}_1$, i.e., create a copy q_0^{color} of q_1^{color} that uses fresh variables but the same root individual. We then take the query:

$$q := q_0^{color} \;\wedge\; q_1^{color} \;\wedge\; q_1^{edge} \;\wedge\; q_2^{color} \;\wedge\; q_2^{edge}.$$

We claim $[36, +\infty]$ is the tightest certain answer to q over $\mathcal{K}$ iff $\mathcal{G}_1 \in$ 3-Col and $\mathcal{G}_2 \notin$ 3-Col. This is proven by a case analysis, summarized here:

	$\mathcal{G}_1 \in$ 3-Col	$\mathcal{G}_1 \notin$ 3-Col
$\mathcal{G}_2 \in$ 3-Col	27 $(= 3 \times 3 \times 3)$	48 $(= 4 \times 4 \times 3)$
$\mathcal{G}_2 \notin$ 3-Col	36 $(= 3 \times 3 \times 4)$	64 $(= 4 \times 4 \times 4)$

Each of the four cells displays the largest value of m such that $[m, +\infty]$ is a certain answer of q over $\mathcal{K}$, under different assumptions on the 3-colorability of $\mathcal{G}_1$ and $\mathcal{G}_2$. To establish these values, one must first prove that every model has at least this many c-matches, and then exhibit a model that realizes the exact number. For the latter, we utilize our assumption that the graphs are planar, hence 4-colorable [Gonthier, 2008], which we use to show that the minimal number of c-matches is realized in a model that encodes proper 3- or 4-colorings of the graphs.

We now provide more details on the case analysis. In what follows, $\mathcal{I}$ denotes an arbitrary model of $\mathcal{K} = (\mathcal{T}, \mathcal{A}_{\mathcal{G}_1} \cup \mathcal{A}_{\mathcal{G}_2})$. We first remark that every model contains the c-matches given by:

$$z^0, z^1 \mapsto \mathsf{r}^1 \mid \mathsf{g}^1 \mid \mathsf{b}^1 \qquad z_1^1, z_2^1 \mapsto \mathsf{a_v}^1 \qquad z^2 \mapsto \mathsf{r}^2 \mid \mathsf{g}^2 \mid \mathsf{b}^2 \qquad z_1^2, z_2^2 \mapsto \mathsf{a_v}^2$$

Hence $\mathcal{I}$ contains at least $3 \times 3 \times 1 \times 3 \times 1 = 27$ c-matches.

In what follows, we will use $\mathcal{I}_\tau^\mathcal{G}$ to denote a minimal model of $\mathcal{K}_\mathcal{G}$ complying with a given coloring τ of a graph $\mathcal{G}$, constructed as in the proof of Theorem 21. We observe that if τ_1 and τ_2 are respectively colorings for the graphs $\mathcal{G}_1$ and $\mathcal{G}_2$, then the interpretation $\mathcal{I}_{\tau_1}^{\mathcal{G}_1} \cup \mathcal{I}_{\tau_2}^{\mathcal{G}_2}$ which is the disjoint union of $\mathcal{I}_{\tau_1}^{\mathcal{G}_1}$ and $\mathcal{I}_{\tau_2}^{\mathcal{G}_2}$ is a model of the considered KB $\mathcal{K}$. We use such models to establish the minimum number of c-matches in the four different cases:

- $\mathcal{G}_1, \mathcal{G}_2 \in$ 3-Col: We have already seen that every model of $\mathcal{K}$ contains at least 27 c-matches. Let τ_1 (resp. τ_2) be a 3-coloring for $\mathcal{G}_1$ (resp. $\mathcal{G}_2$). Then the model $\mathcal{I}_{\tau_1}^{\mathcal{G}_1} \cup \mathcal{I}_{\tau_2}^{\mathcal{G}_2}$ has exactly 27 c-matches.

- $\mathcal{G}_1 \in$ 3-Col$, \mathcal{G}_2 \notin$ 3-Col: As $\mathcal{G}_2$ is not 3-colorable, the part of $\mathcal{I}$ describing $\mathcal{G}_2$ must either introduce a fourth color, providing a new value for z^2 (hence at least $3 \times 3 \times 1 \times 4 \times 1 = 36$ c-matches), or contain a monochromatic edge, providing another possible value for (z_1^2, z_2^2) (hence at least $3^2 \times 1 \times 3 \times 2 = 54$ c-matches). Therefore, every model contains at least 36 c-matches for q. To show we cannot ensure more than 36 c-matches, let τ_1 (resp. τ_2) be a 3-coloring (resp. 4-coloring) for $\mathcal{G}_1$ (resp $\mathcal{G}_2$). Then $\mathcal{I}_{\tau_1}^{\mathcal{G}_1} \cup \mathcal{I}_{\tau_2}^{\mathcal{G}_2}$ has exactly 36 c-matches.

- $\mathcal{G}_1 \notin$ 3-Col$, \mathcal{G}_2 \in$ 3-Col: The part of $\mathcal{I}$ describing $\mathcal{G}_1$ must introduce either a fourth color, providing a new value for z^0 and z^1 (hence at least $4 \times 4 \times 1 \times 3 \times 1 = 48$ c-matches), or contain a monochromatic edge, providing another possible value for (z_1^1, z_2^1) (hence at least $3 \times 3 \times 2 \times 3 \times 1 = 54$ c-matches). It follows that every model contains at least 48 c-matches. To show this is the best value that can be attained, let τ_1 (resp. τ_2) be a 4-coloring (resp. 3-coloring) for $\mathcal{G}_1$ (resp. $\mathcal{G}_2$). Then $\mathcal{I}_{\tau_1}^{\mathcal{G}_1} \cup \mathcal{I}_{\tau_2}^{\mathcal{G}_2}$ has exactly 48 c-matches.

- $\mathcal{G}_1, \mathcal{G}_2 \notin$ 3-Col: For each of the two graphs, $\mathcal{I}$ must introduce either a fourth color or a monochromatic edge. There are four cases to consider:

	Fourth color for $\mathcal{G}_1$	Monochrom. edge for $\mathcal{G}_1$
Fourth color for $\mathcal{G}_2$	$4^2 \times 1 \times 4 \times 1 = 64$	$3^2 \times 2 \times 4 \times 1 = 72$
Monochrom. edge for $\mathcal{G}_2$	$4^2 \times 1 \times 3 \times 2 = 96$	$3^2 \times 2 \times 3 \times 2 = 108$

  We therefore see that every model contains at least 64 c-matches of q. To realize the minimal number, we let τ_1 (resp. τ_2) be a 4-coloring (resp. 4-coloring) for $\mathcal{G}_1$ (resp. $\mathcal{G}_2$) and observe that $\mathcal{I}_{\tau_1}^{\mathcal{G}_1} \cup \mathcal{I}_{\tau_2}^{\mathcal{G}_2}$ has exactly 64 c-matches.

This completes the case analysis and the proof. □

4.3 Exhaustive rooted CCQs over $\mathcal{ALCHI}$

For the remainder of this chapter, we turn to exhaustive rooted CCQs and hence no longer distinguish matches from counting matches. In this section, we investigate a refinement of the interlacing function allowing us to obtain countermodels with exponential size, if a countermodel does exist, yielding the coNEXP upper bound. We proceed in two steps: in Section 4.3.1 we introduce the interlacing function $f^\diamond$ and verify the corresponding $f^\diamond$-interlacings have at most as many matches as the initial models, in Section 4.3.2 we quotient these latter $f^\diamond$-interlacings to obtain (counter)models with the claimed size. The equivalence relation used in this process is very similar, but slightly more careful, than the one used in Section 3.2.3 to obtain "naive" finite models. In Section 4.3.3 we exhibit two matching lower bounds for exhaustive rooted CCQ answering over $\mathcal{EL}$ KBs and DL-Lite$^{\mathcal{H}}_{\mathsf{pos}}$ KBs respectively. These two reductions are strongly inspired by a reduction in Lutz [2008] in the context of rooted CQ answering over $\mathcal{ALCI}$ KBs, known to be coNEXP-complete.

4.3.1 The interlacing function $f^\diamond$

We begin with a new refinement of the interlacing function. To introduce it, let $\mathcal{K} := (\mathcal{T}, \mathcal{A})$ be an $\mathcal{ALCHI}$ KB, $\mathcal{I}$ a model of $\mathcal{K}$, and q an exhaustive rooted CCQ. We recall that Ω denotes the set of heads of existential rules from $\mathcal{T}$, that Δ° denotes the existential extraction of $\mathcal{I}$ and that $f : \Delta^\circ \to \Delta^\mathcal{I}$ is the mapping used to build this existential extraction (see Definition 19). We also recall that Δ^* is the subset of $\Delta^\mathcal{I}$ containing all individuals from $\mathcal{A}$ and all elements reached by matches of q in $\mathcal{I}$ (see Definition 22).

We now explain the intuition underlying the new function $f^\diamond$. Consider the Id-interlacing of $\mathcal{I}$, that is essentially its existential extraction equipped with a basic interpretation. Notice f is a homomorphism from the Id-interlacing to $\mathcal{I}$, ensuring the counting matches for q in the Id-interlacing are contained in $f^{-1}(\Delta^*)$. In the general setting, we considered an interlacing function collapsing back in place all these elements from $f^{-1}(\Delta^*)$. However since q is exhaustive rooted, it is sufficient to put back in place only those connected components of $f^{-1}(\Delta^*)$ which contain some individuals from the ABox. Indeed, elements from the Id-interlacing involved in matches of q must be connected by some path of variables in the query to an individual name since q is rooted, and the intermediate elements reached by the match all belong to $f^{-1}(\Delta^*)$ since q is exhaustive. This motivates the following definition of $f^\diamond$ which inductively starts from individuals and stops collapsing elements back in place as soon as we leave $f^{-1}(\Delta^*)$.

Definition 42. *The interlacing function $f^\diamond$ is defined inductively as:*

$$
\begin{array}{rcll}
f^\diamond : \Delta^\circ & \to & \Delta^* \cdot \Omega^* & \\
\mathsf{a} & \mapsto & \mathsf{a} & \diamond_0 \\
w \cdot h & \mapsto & \begin{cases} f(w \cdot h) & \text{if } f^\diamond(w) \in \Delta^* \text{ and } f(w \cdot h) \in \Delta^* \\ f^\diamond(w) \cdot h & \text{otherwise} \end{cases} & \begin{array}{l}\diamond_1 \\ \diamond_2\end{array}
\end{array}
$$

Remark 18. *Notice that in both Cases $\diamond_0$ and $\diamond_1$, we have $f^\diamond(w) = f(w)$.*

The very first thing to do is to verify $f^\diamond$ is pseudo-injective, as this gives modelhood by Theorem 4.

Lemma 19. *$f^\diamond$ is pseudo-injective.*

Proof. We need to prove that for all u and all v in Δ°, if $f^\diamond(u) = f^\diamond(v)$, then $f(u) = f(v)$. We proceed by induction on u.

$u \in \mathsf{Ind}(\mathcal{A})$. By definition of f and $f^\diamond$ (case $\diamond_0$), we have $f(u) = u$ and $f^\diamond(u) = u$. Let $v \in \Delta^\circ$. We distinguish the 3 possible cases for $f^\diamond(v)$:

- **$\diamond_0$ $\diamond_1$.** Based on Remark 18, we have $f^\diamond(v) = f(v)$. Therefore assuming $f^\diamond(u) = f^\diamond(v)$ gives $f(u) = f(v)$.
- **$\diamond_2$.** We have $f^\diamond(v) = f^\diamond(w) \cdot h$. In particular $f^\diamond(v) \notin \Delta^*$. Assuming $f^\diamond(u) = f^\diamond(v)$ yields a contradiction as $f^\diamond(u) = u \in \mathsf{Ind}(\mathcal{A}) \subseteq \Delta^*$.

$u = u_0 \cdot h$. If $f^\diamond(u)$ is in Case $\diamond_1$, then Remark 18 and the same arguments as in the base case conclude (notice $f^\diamond(u) \in \Delta^*$ still holds). Otherwise, $f^\diamond(u)$ is in Case $\diamond_2$, that is $f^\diamond(u) = f^\diamond(u_0) \cdot h \notin \Delta^*$. Let $v \in \Delta^\circ$. If v is in Case $\diamond_0$ or in Case $\diamond_1$, then $f^\diamond(v) \in \Delta^*$, which yields a contradiction. Otherwise, v is in Case $\diamond_2$, that is $f^\diamond(v) = f^\diamond(v_0) \cdot h'$, with $v = v_0 \cdot h'$. Assuming $f^\diamond(u) = f^\diamond(v)$ yields $f^\diamond(u_0) = f^\diamond(v_0)$ and $h = h'$. Induction hypothesis gives $f(u_0) = f(v_0)$. And from $h = h'$ we obtain $f(u_0 \cdot h) = f(v_0 \cdot h')$, that is $f(u) = f(v)$. □

We now turn to the number of matches in the $f^\diamond$-interlacing, which is at most the number of matches from the original model. The proof proceeds by a formal explanation of the intuition previously presented.

Lemma 20. *$\mathcal{I}^\diamond$ has at most as many matches for q than $\mathcal{I}$.*

Proof. Consider a counting match $\pi : q \to \mathcal{I}^\diamond$ of q in $\mathcal{I}^\diamond$.

Let us first suppose that there is a counting variable $z \in \mathbf{z}$ such that $\pi(z) \notin \Delta^*$, in which case we must have $\pi(z) = \mathsf{t} \cdot w$ for some $\mathsf{t} \in \Delta^*$ and some non-empty word $w \in \Omega^*$. Since q is exhaustive rooted, all intermediate elements $\mathsf{t} \cdot w'$ with w' a prefix of w, must be reached by some other counting variables. In particular, one of these counting variables, say z_0, must map onto $\mathsf{t} \cdot h$, with h the first symbol of w. From the definition of $f^\diamond$, we also have a word w_t such that $f^\diamond(w_\mathsf{t}) = f(w_\mathsf{t}) = \mathsf{t}$. However, via the homomorphism σ (see Theorem 4), we can transform π into a match $\sigma \circ \pi : q \to \mathcal{I}$ in the original model $\mathcal{I}$. In particular, we have $\sigma(\pi(z_0)) = \sigma(\mathsf{t} \cdot h) = \sigma(f^\diamond(w_\mathsf{t})) \cdot h) = \sigma(f(w_\mathsf{t} \cdot h)) = f(w_\mathsf{t} \cdot h)$. Thus, $f(w_\mathsf{t} \cdot h)$ belongs to the image of the match $\sigma \circ \pi$ in $\mathcal{I}$. From the definition of Δ^*, we can thus infer that $f(w_\mathsf{t} \cdot h) \in \Delta^*$. But since $f^\diamond(w_\mathsf{t}) = \mathsf{t} \in \Delta^*$ and $f(w_\mathsf{t} \cdot h) \in \Delta^*$, we have $f^\diamond(w_\mathsf{t} \cdot h) = f(w_\mathsf{t} \cdot h) \in \Delta^*$. This contradicts z_0 mapping onto $\mathsf{t} \cdot h \notin \Delta^*$. Therefore, there is no counting variable $z \in \mathbf{z}$ mapping outside Δ^*.

Hence, we have $\pi(\mathbf{z}) \subseteq \Delta^*$. Then since $\sigma_{|\Delta^*} = \mathsf{Id}$, we have $\sigma \circ \pi = \pi$, which shows that the mapping $\pi \mapsto \sigma \circ \pi$ is injective. In particular, $\mathcal{I}$ contains at least as many c-matches as $\mathcal{I}^*$. □

The obtained $f^\diamond$-interlacing $\mathcal{I}^\diamond$ has a particular structure: it is essentially Δ^* enriched by tree-shaped structures. In contrast to the general case of CCQs, these tree-shaped structures do not contain any edges pointing back to Δ^*, which simplifies the structure of interlacings.

4.3.2 Quotients of $f^\diamond$-interlacings: a coNEXP upper bound

In this section, we briefly show how to quotient $f^\diamond$-interlacings to obtain optimal models with exponential size. This immediately yields the following result.

Theorem 23. *Exhaustive rooted CCQ answering over $\mathcal{ALCHI}$ ontologies is in* coNEXP *w.r.t. combined complexity.*

It thus suffices to focus on the following.

Lemma 21. *Let $\mathcal{K}$ be an $\mathcal{ALCHI}$ KB, q an exhaustive rooted CCQ and m a candidate integer. If a countermodel exists for m, then there exists such a countermodel whose domain has an exponential number of elements w.r.t. combined complexity.*

Proof. Recall that from Theorem 5, this is trivial if m is greater than the exponential bound (w.r.t. combined complexity) exposed in Lemma 3. Henceforth we assume m to be at most exponential w.r.t. combined complexity. Consider $\mathcal{I}$ a countermodel for m and $\mathcal{I}^\diamond$ its $f^\diamond$-interlacing. Notice Δ^* also has exponential size due to our assumption on m. For each element of $\Delta^{\mathcal{I}^\diamond}$, we define its size: the size $|\mathsf{a}|$ of an individual a is 1, the size $|w \cdot \mathrm{R.M}|$ of a non-individual element $w \cdot \mathrm{R.M}$ is $|w| + 1$. We now equip $\Delta^{\mathcal{I}^\diamond}$ with the following equivalence relation $\sim$: an element with size less than $|q| + 1$ is only equivalent to itself, while two elements $w_1 \cdot h_1$ and $w_2 \cdot h_2$ with size greater than $|q| + 1$ are equivalent iff they satisfy the same concepts, $h_1 = h_2$ and $|w_1| = |w_2| \mod 3$. Let $\widetilde{u}$ denote the equivalence class of the element u w.r.t $\sim$ and $\rho : d \mapsto \widetilde{d}$ the canonical projection.

We claim that the interpretation $\mathcal{M} := \mathcal{I}^\diamond / \sim$ with domain $\Delta^{\mathcal{I}^\diamond} / \sim$ and interpretation of atomic concepts and roles given by $\cdot^{\mathcal{M}} := \rho \circ \cdot^{\mathcal{I}^\diamond}$ is a model. Notice it has the desired number of elements as each equivalence class is: either an element from $\mathcal{I}^\diamond$ being at distance less than $|q| + 1$ from Δ^* (there can be exponentially many such elements in the tree structure issuing from an element of Δ^*, and Δ^* itself is exponential), or fully characterized by a set of concepts from $\mathsf{sig}(\mathcal{T})$, a role or its inverse from $\mathsf{sig}(\mathcal{T})$ and another set of concepts from $\mathsf{sig}(\mathcal{T})$, and an integer modulo 3. The full verification follows the proof of Theorem 5, to which we refer.

Furthermore, $\mathcal{M}$ has exactly as many matches as $\mathcal{I}^\diamond$ since $\mathcal{M}$ and $\mathcal{I}^\diamond$ both coincide when restricted to their respective set of elements being at distance at most $|q| + 1$ from an element of Δ^*. This concludes the proof that $\mathcal{M}$ is indeed a countermodel with exponential size w.r.t. combined complexity. □

4.3.3 Two matching lower bounds with inverse roles

In this section, we exhibit two lower bounds matching the coNEXP upper bounds from the previous section. These coNEXP-hardness results strongly rely on inverse roles, together with role inclusions and DL-Lite concepts (DL-Lite$^{\mathcal{H}}_{\mathsf{pos}}$), or with $\mathcal{EL}$ concepts ($\mathcal{ELI}$). Both proofs proceed by reduction from the exponential grid tiling problem ExpTiling, as in Theorem 15, and borrow ideas from the reduction for rooted CQ entailment over $\mathcal{ALCI}$ ontologies developed in Lutz [2008]. We recall that an instance of ExpTiling consists of a set $\mathcal{C}$ of colors, two relations $\mathcal{H}, \mathcal{V} \subseteq \mathcal{C} \times \mathcal{C}$ that give the horizontal and vertical tiling conditions, and a number n (given in unary). The task is to decide whether there exists a valid $(\mathcal{H}, \mathcal{V})$-tiling of a $2^n \times 2^n$ grid, i.e., a mapping $\tau : \{0, \ldots, 2^n - 1\} \times \{0, \ldots, 2^n - 1\} \mapsto \mathcal{C}$ such that $(\tau(i,j), \tau(i+1,j)) \in \mathcal{H}$ for every $0 \leq i < 2^n - 1$ and $(\tau(i,j), \tau(i,j+1)) \in \mathcal{V}$ for every $0 \leq j < 2^n - 1$.

To reuse ideas from Lutz [2008], which works with the more expressive $\mathcal{ALCI}$ ontologies, we first need to simulate very basic axioms such as $\mathrm{GridPosition} \sqsubseteq \mathrm{C_1} \sqcup ... \sqcup \mathrm{C_p}$, where each $C_1, \ldots C_p$ are the colors from $\mathcal{C}$, assigning a color to an element being a grid position. In the general CCQ setting and, say, with $\mathcal{EL}$, such a task is fairly easy: simply require in the TBox that $\mathrm{GridPosition} \sqsubseteq \exists\mathrm{hasColor.Color}$ and $\mathrm{C_k} \sqsubseteq \mathrm{Color}$ for each $C_k \in \mathcal{C}$, provide an instance of each color $\mathrm{C_k}(\mathsf{c_k})$ in the ABox, and count instances of the concept Color with the CCQ. To minimize the query, elements satisfying GridPosition will need to reuse the existing instances of Color, hence satisfy $\exists\mathrm{hasCol.C_1} \sqcup ... \sqcup \exists\mathrm{hasCol.C_p}$ which is a good enough simulation of the previous $\mathcal{ALC}$ axiom.

This approach does not easily transfer with exhaustive rooted CCQs as the latter require to count each intermediate element satisfying GridPosition prior to reaching the Color instance of interest. In particular, we cannot enforce given instances of Color to be reused as endpoints for the role hasCol. Instead, we can enforce in the TBox that each instance of GridPosition must be connected to one instance of each color, one of them being "Used", as follows: $\mathrm{GridPosition} \sqsubseteq \exists\mathrm{hasColor.C_k}$ for each $C_k \in \mathcal{C}$ and $\mathrm{GridPosition} \sqsubseteq \exists\mathrm{hasColor.Used}$. If we are interested by m elements satisfying GridPosition, we can count instances of the role hasColor with a exhaustive rooted CCQ and expect the result to be $m \times |\mathcal{C}|$ if the head $\exists\mathrm{hasColor.Used}$ indeed collapses on one of the $\exists\mathrm{hasColor.C_k}$. However, since we are working with $\mathcal{EL}$, hence without any form of disjointness, it could happen that $\exists\mathrm{hasColor.C_1}$ collapses on $\exists\mathrm{hasColor.C_2}$, hence reducing the number of matches.

Therefore, we also need to simulate axioms such as $\mathrm{C_1} \sqcap \mathrm{C_2} \sqsubseteq \bot$. This is achieved by another part of the exhaustive rooted CCQ which has the property of admitting a big enough number of matches, that is more than $m \times |\mathcal{C}|$, as soon as it detects the concept $\exists\mathrm{hasColor.}(\mathrm{C_1} \sqcap \mathrm{C_2})$ on a GridPosition-element of interest.

The proofs of the two following theorems implement the preceding intuitions.

We choose to focus on the case of DL-Lite$^{\mathcal{H}}_{\mathsf{pos}}$ as it is arguably a more distant setting from $\mathcal{ALCI}$, explored in Lutz [2008], than $\mathcal{ELI}$. For the latter, we only give indications on how to adapt the proof of the DL-Lite$^{\mathcal{H}}_{\mathsf{pos}}$ situation but we don't redo all the arguments.

Remark 19. *Readers familiar with the proofs from [Lutz, 2008] will notice several differences with the original construction. This is due to rather non-trivial issues we detected in those proofs when adapting their ideas to our counting queries. Briefly, some collapsings of the constructed queries are not treated in the proofs of Theorems 1 and 2 in the reference, while they actually provide matches for the query. These latter matches however violate the key property used in the reduction: they* do not *connect G-nodes agreeing on the interpretation of a given concept. For example, such a collapsing of the query on Figure 2 in the reference can be obtained by identifying variables v_{m+1} and v_{m+2}, an option not considered in the proof argument.*

This issue has been reported in a personal communication with the author, who addressed it with a rather elegant fix. The idea is to break the excessively symmetric ways in which the query can match by requiring that it navigates from one tree structure to another. This hence requires a duplication of the (exponentially large) original tree structure, both being identified with dedicated concepts also added to the query. Navigating from one tree to another is made possible via a single dedicated edge, which provides the missing ingredient to fully control the collapsing of the query using the intuition underlying the original construction.

Theorem 24. *Exhaustive rooted CCQ answering over DL-Lite$^{\mathcal{H}}_{\mathsf{pos}}$ KBs is* coNEXP*-hard.*

Proof. In what follows, we consider an instance $(n, \mathcal{C}, \mathcal{H}, \mathcal{V})$ of the ExpTiling problem. To be able to test for the existence of a tiling of a $2^n \times 2^n$ grid, we must start by ensuring we can find (the encoding of) such a grid in each model. To easily detect horizontal and vertical adjacency in this grid, it is appropriate to use horizontal/vertical coordinates, and to ensure the reduction remains polynomial, we need to use a binary encoding of these coordinates.

Knowledge Base. We first generate an exponentially-large tree T_0 with $2^n \times 2^n$ leaves, each representing a possible horizontal-vertical coordinate (u, v) identified by the $m := 2n$ branchings leading to this leaf from an individual a_0. For the reasons explained in Remark 19, we also build a similar tree T_1 rooted at an individuala_1. As we'll further need to navigate these trees in both directions (symmetry) but also not to go anywhere (reflexivity), we ask that branchings in the trees happen with the composition of two roles. More precisely, facts $\mathrm{Next}^{\mathrm{b}}_{\mathrm{t,k}}(e_1, e_2)$ and $\mathrm{AltNext}^{\mathrm{b}}_{\mathrm{t,k}}(e_2, e_3)$ represents a branching in the tree T_t from e_1 assigning b to the k^{th} bit in the binary encoding of (u, v). We require $\mathrm{Next}^{\mathrm{b}}_{\mathrm{t,k}}$ to be a subrole of Next, and $\mathrm{AltNext}^{\mathrm{b}}_{\mathrm{t,k}}$ to be a subrole of Next^-, which allows us to move from e_1 to e_3, or vice-versa, by the composition of Next followed by Next^-. Notice that this composition also allows us to move from e_1 to e_1 or from e_3 to e_3. This is achieved with the following facts and axioms (for each $t, b \in \{0, 1\}$ and $0 \leq i \leq m - 1$):

$$\mathrm{Node}_{\mathrm{t,0}}(\mathsf{a_t})$$

$$\mathrm{Node}_{\mathrm{t,i}} \sqsubseteq \exists \mathrm{Next}^{\mathrm{b}}_{\mathrm{t,i+1}} \qquad \exists(\mathrm{Next}^{\mathrm{b}}_{\mathrm{t,i+1}})^- \sqsubseteq \mathrm{Node}^{\mathrm{b}}_{\mathrm{t,i}} \qquad \mathrm{Next}^{\mathrm{b}}_{\mathrm{t,i+1}} \sqsubseteq \mathrm{Next}$$

$$\mathrm{Node}^{\mathrm{b}}_{\mathrm{t,i}} \sqsubseteq \exists \mathrm{AltNext}^{\mathrm{b}}_{\mathrm{t,i+1}} \qquad \exists(\mathrm{AltNext}^{\mathrm{b}}_{\mathrm{t,i+1}})^- \sqsubseteq \mathrm{Node}_{\mathrm{t,i+1}} \qquad \mathrm{AltNext}^{\mathrm{b}}_{\mathrm{t,i+1}} \sqsubseteq \mathrm{Next}^-$$

A node satisfying $\mathrm{Node}_{\mathrm{t,m}}$ shall hence represent the encoding of one pair (u, v) (seen as the concatenation of the binary encodings of u and v) in the tree T_t. Note that, due to our two-step-branching procedure, each element satisfying $\mathrm{Node}_{\mathrm{t,k}}$, which we henceforth term a $\mathrm{Node}_{\mathrm{t,k}}$-element, is actually at depth $2k$ in the tree T_t (and an element satisfying $\mathrm{Node}^{\mathrm{b}}_{\mathrm{t,k}}$ is at depth $2k + 1$).

We desire three properties to hold when reaching a $\mathrm{Node}_{\mathrm{t,m}}$-element e:

1. e is required to satisfy the concept $\mathrm{F_t}$ and the concept $\exists \mathrm{HasBit}_{\mathrm{t,k}}.\mathrm{Bit}_{1-\mathrm{b}}$ if the branch leading to e picks b as k^{th} bit;
2. e is followed through a composition of roles Next and Next^- by a node e' satisfying the concept $\mathrm{G_t}$ and the concept $\exists \mathrm{HasBit}_{\mathrm{k}}.\mathrm{Bit}_{\mathrm{b}}$ if the branch leading to e picks b as k^{th} bit;
3. e' is required to be assigned a color $c \in \mathcal{C}$, encoded as satisfying the concept $\exists \mathrm{HasCol}.\mathrm{Color}_{\mathrm{c}}$;

Notice that the concepts satisfied by a $\mathrm{F_t}$-node shall encode the converse of the branchings used to reach this node, while those satisfied by $\mathrm{G_t}$-node shall match these branchings.

So far, all this latter part about $\mathrm{F_t}$-nodes and $\mathrm{G_t}$-nodes is only a declaration of intent. Let us clarify how to enforce all this. For Property 1, we add the following axioms (for each $t \in \{0, 1\}$):

$$\mathrm{Node}_{\mathrm{t,m}} \sqsubseteq \mathrm{F_t} \qquad \begin{array}{lll} \mathrm{F_t} \sqsubseteq \exists \mathrm{ToBit}_0 & \exists(\mathrm{ToBit}_0)^- \sqsubseteq \mathrm{Bit}_0 & \mathrm{ToBit}_0 \sqsubseteq \mathrm{ToBit} \\ \mathrm{F_t} \sqsubseteq \exists \mathrm{ToBit}_1 & \exists(\mathrm{ToBit}_1)^- \sqsubseteq \mathrm{Bit}_1 & \mathrm{ToBit}_1 \sqsubseteq \mathrm{ToBit} \end{array}$$

and also axioms (for each $t \in \{0, 1\}$ and $1 \leq i \leq m$):

$$\mathrm{F_t} \sqsubseteq \exists \mathrm{HasBit_i} \qquad \exists(\mathrm{HasBit_i})^- \sqsubseteq \mathrm{ChosenBit_i} \qquad \mathrm{HasBit_i} \sqsubseteq \mathrm{ToBit}$$

Ensuring that each role HasBit_i reuses the correct bit Bit_0 or Bit_1 will be further achieved *via* the query.

For Property 2, we use the following axioms (for each $t \in \{0,1\}$):

$$\mathrm{F_t} \sqsubseteq \exists \mathrm{GNext_t} \quad \exists \mathrm{GNext_t}^- \sqsubseteq \exists \mathrm{AltGNext_t} \quad \exists \mathrm{AltGNext_t}^- \sqsubseteq \mathrm{G_t}$$

$$\mathrm{GNext_t} \sqsubseteq \mathrm{Next} \quad \mathrm{AltGNext_t} \sqsubseteq \mathrm{Next}^-$$

and we then assign bits in the very same manner has for $\mathrm{F_t}$-nodes (for each $t \in \{0,1\}$ and $1 \leq i \leq m$):

$$\mathrm{G_t} \sqsubseteq \exists \mathrm{ToBit_0} \qquad \mathrm{G_t} \sqsubseteq \exists \mathrm{ToBit_1} \qquad \mathrm{G_t} \sqsubseteq \exists \mathrm{HasBit_i}$$

For Property 3, we proceed essentially as for bits, but here relying on the colors available in the input set $\mathcal{C}$. This is achieved with the following axioms (for each $t \in \{0,1\}$ and $c \in \mathcal{C}$)

$$\mathrm{G_t} \sqsubseteq \exists \mathrm{ToCol_c} \qquad \exists (\mathrm{ToCol_c})^- \sqsubseteq \mathrm{Color_c} \qquad \mathrm{ToCol_c} \sqsubseteq \mathrm{ToCol}$$

$$\mathrm{G_t} \sqsubseteq \exists \mathrm{HasCol} \qquad \exists \mathrm{HasCol}^- \sqsubseteq \mathrm{ChosenColor} \qquad \mathrm{HasCol} \sqsubseteq \mathrm{ToCol}$$

And here again, ensuring the role HasCol points to a valid color will further be achieved *via* the query.

This completes the description of the DL-Lite$^{\mathcal{H}}_{\mathsf{pos}}$ TBox $\mathcal{T}$, consisting of all the preceding axioms. It remains to introduce the root r that will be used in the query, an element $\mathsf{a}_{1/2}$ connecting individuals a_0 and a_1, and two elements l_0 and l_1 to increase drastically the number of matches of some subqueries. Consider the following assertions (for each $t \in \{0,1\}$):

$$\mathrm{toStart}(\mathsf{r}, \mathsf{a_t}) \qquad \mathrm{Next}(\mathsf{a_t}, \mathsf{a}_{1/2}) \qquad \mathrm{toLoader}(\mathsf{a_t}, \mathsf{l}_0) \qquad \mathrm{toLoader}(\mathsf{a_t}, \mathsf{l}_1)$$

To conclude the construction of the KB, we introduce an auxiliary individual b whose purpose is to ensure that each subquery can map at least once. It satisfies: *all concepts concept assertions* $\mathrm{B}(\mathsf{b})$ *with* B *a concept name previously mentioned, all role assertions* $\mathrm{P}(\mathsf{b}, \mathsf{b})$ *with* P *a role name previously mentioned, and the fact* $\mathrm{toStart}(\mathsf{r}, \mathsf{b})$. We let $\mathcal{A}$ be the ABox consisting of the previous facts and $\mathcal{K} := (\mathcal{T}, \mathcal{A})$ the KB obtained from $\mathcal{T}$ and $\mathcal{A}$. The intended structure of models of $\mathcal{K}$ is depicted in Figure 4.3.

Query. We distinguish two main kinds of subqueries: structural subqueries and consistency subqueries. Structural subqueries ensure that each model contains the desired tree-shaped structures or yields too many matches to be optimal. Consistency subqueries ensure models with the desired tree-shaped structures either represent a valid tiling or yield at least one additional match.

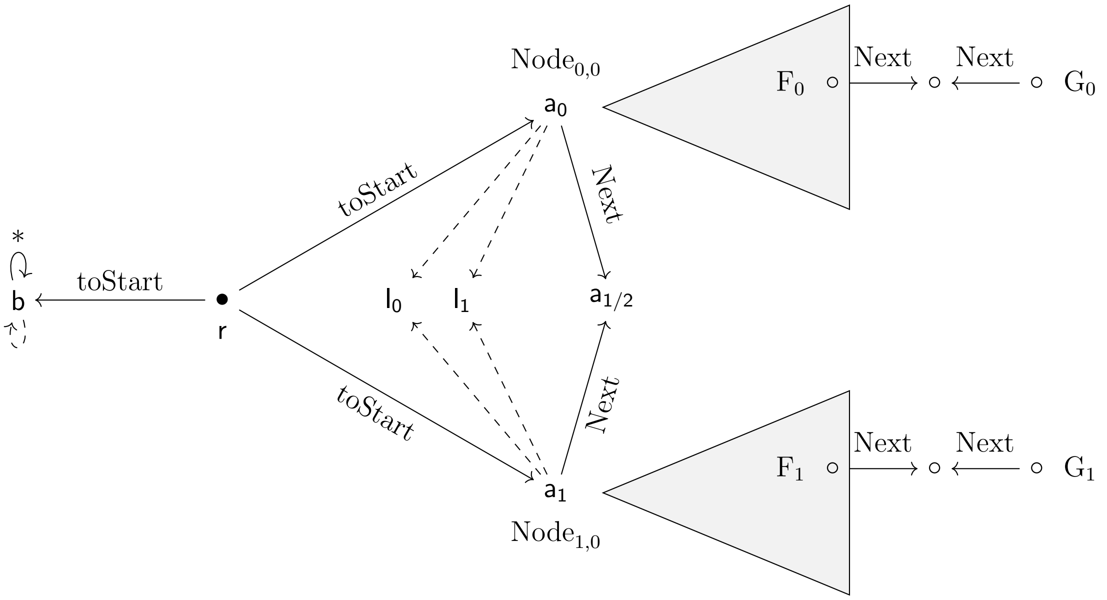


Figure 4.3: The intended structure of models of $\mathcal{K}$. Dashed edges represent toLoader-roles; the label $*$ witnesses that all roles are satisfied on the loop (b, b). We omitted the concepts and roles related to bits and colors.

We begin with the loading subquery q_{load}, which contains a free variable z so that copies of q_{load} will be instantiated for building the other structural subqueries.

$$q_{\text{load}}(z) := \exists z_1, \ldots z_M \ \text{toStart}(\mathsf{r}, z) \wedge \bigwedge_{i=1}^{M} \text{toLoader}(z, z_i)$$

Notice that if z is mapped onto b, then there is in general only one way to map the remaining variables from q_{load} (all onto b as well). On the other hand, if z is mapped onto a_0 or onto a_1, then there are at least 2^M ways to map these remaining variables (each variable can be mapped either onto l_0 or l_1). The exact value of M will be specified later in the construction.

To ensure the tree-shaped structure is preserved, we first require the branchings (leading to either $\text{Node}^0_{\text{t,i}}$ or $\text{Node}^1_{\text{t,i}}$) to be indeed branching, meaning we don't want these two concepts being witnessed by the same element. We proceed with the following subqueries, each detecting a non-branching node at depth $2d$, with $1 \leq d \leq m$ and $t \in \{0, 1\}$:

$$\begin{aligned} q^{t,d}_{\text{branch}} := \ & \exists z_0, \ldots z_{2d} \ q_{\text{load}}(z_0) \wedge \bigwedge_{i=0}^{d-1} \text{Next}(z_{2i}, z_{2i+1}) \wedge \text{Next}(z_{2i+2}, z_{2i+1}) \\ & \wedge \text{Node}^0_{\text{t,i}}(z_{2d}) \wedge \text{Node}^1_{\text{t,i}}(z_{2d}) \end{aligned}$$

Let us emphasize that $q^{t,d}_{\mathrm{branch}}$ is rooted *via* its copy of q_{load} and that existentially-quantified variables from q_{load} are not shared with those of $q^{t,d}_{\mathrm{branch}}$.

We proceed as well with the Bit_0 and Bit_1 branchings at each $\mathrm{F_t}$ and $\mathrm{G_t}$-nodes, with the following subqueries, each detecting collapsed Booleans coming from a node at depth $2m$ (that is a $\mathrm{F_t}$-node) or $2m+2$ (that is a $\mathrm{G_t}$-node), with $d \in \{2m, 2m+2\}$:

$$\begin{aligned} q^{d}_{\mathrm{bool}} := \ & \exists z_0, \ldots z_{2d}, z \ q_{\mathrm{load}}(z_0) \land \bigwedge_{i=0}^{d-1} \mathrm{Next}(z_{2i}, z_{2i+1}) \land \mathrm{Next}(z_{2i+2}, z_{2i+1}) \\ & \land \mathrm{ToBit}(z_{2d}, z) \land \mathrm{Bit}_0(z) \land \mathrm{Bit}_1(z) \end{aligned}$$

Similar subqueries $q^{c_1,c_2}_{\mathrm{color}}$ can detect if two different colors $c_1, c_2 \in \mathcal{C}$ issuing from a $\mathrm{G_t}$-node are collapsed together.

We further detect if a branch loops back on itself, which can be captured by detecting nodes satisfying concepts corresponding to different depths. Consider B_1 and B_2 two such concepts and consider the following subqueries, with $0 \leq d \leq m+1$:

$$\begin{aligned} q^{2d}_{\mathrm{loop}} := \ & \exists z_0, \ldots z_{2d}, z \ q_{\mathrm{load}}(z_0) \land \bigwedge_{i=0}^{d-1} \mathrm{Next}(z_{2i}, z_{2i+1}) \land \mathrm{Next}(z_{2i+2}, z_{2i+1}) \\ & \land \mathrm{B}_1(z_{2d}) \land \mathrm{B}_2(z_{2d}) \end{aligned}$$

and for $1 \leq d \leq m+1$:

$$\begin{aligned} q^{2d-1}_{\mathrm{loop}} := \ & \exists z_0, \ldots z_{2d-1}, z \ q_{\mathrm{load}}(z_0) \land \bigwedge_{i=0}^{d-2} \mathrm{Next}(z_{2i}, z_{2i+1}) \land \mathrm{Next}(z_{2i+2}, z_{2i+1}) \\ & \land \mathrm{Next}(z_{2d-2}, z_{2d-1}) \land \mathrm{B}_1(z_{2d-1}) \land \mathrm{B}_2(z_{2d-1}) \end{aligned}$$

Similar subqueries q^{d}_{tree} can detect if two branches issuing from the two different trees collapse together.

To preserve the tree-shaped structure, it remains to detect two branches separating at depth $2d$ and collapsing further together at depth $2d+2p+1$. Consider two concept name B_1 and B_2, eventually equal, witnessing depth $2d+2p+1$ and consider the subquery with $0 \leq d \leq n-1$ and $t \in \{0,1\}$:

$$\begin{aligned} q_{\text{cycle}}^{t,2d,\mathrm{B}_1,\mathrm{B}_2} \quad := \quad & \exists z_0, \dots z_{2d}, z_1^0, \dots z_{2p}^0, z_1^1, \dots z_{2p}^1, z \; q_{\text{load}}(z_0) \\ & \wedge \textstyle\bigwedge_{i=0}^{d-1} \mathrm{Next}(z_{2i}, z_{2i+1}) \wedge \mathrm{Next}(z_{2i+2}, z_{2i+1}) \\ & \wedge \mathrm{Next}(z_{2d}, z_1^0) \wedge \mathrm{Node}_{\mathrm{t,2d}}^0(z_0^0) \wedge \mathrm{Next}(z_2^0, z_1^0) \\ & \wedge \textstyle\bigwedge_{i=1}^{p-1} \mathrm{Next}(z_{2i}^0, z_{2i+1}^0) \wedge \mathrm{Next}(z_{2i+2}^0, z_{2i+1}^0) \\ & \wedge \mathrm{Next}(z_{2d}, z_1^1) \wedge \mathrm{Node}_{\mathrm{t,2d}}^1(z_0^1) \wedge \mathrm{Next}(z_2^1, z_1^1) \\ & \wedge \textstyle\bigwedge_{i=1}^{p-1} \mathrm{Next}(z_{2i}^1, z_{2i+1}^1) \wedge \mathrm{Next}(z_{2i+2}^1, z_{2i+1}^1) \\ & \wedge \mathrm{Next}(z_{2p}^0, z) \wedge \mathrm{Next}(z_{2p}^1, z) \end{aligned}$$

and this can be also obtained for even depths $(2d+2p)$ in a manner similar to the preceding subqueries q_{loop}^{2d-1}.

Our next structural subqueries detect if a head HasBit_k from a $\mathrm{F_t}$-nodes collapses on the wrong head $\mathrm{ToBit_b}$ in the sense of the Property 1 previously exposed, that is if it collapses on the bit-value *corresponding* to the branching leading to the $\mathrm{F_t}$ of interest. This is achieved with the following subqueries, with $t \in \{0,1\}$, $1 \leq k \leq m$, $b \in \{0,1\}$:

$$\begin{aligned} q_{\text{F-bit}}^{t,k,b} := \; & \exists z_0, \dots z_{2m}, z \; q_{\text{load}}(z_0) \wedge \bigwedge_{i=0}^{m-1} \mathrm{Next}(z_{2i}, z_{2i+1}) \wedge \mathrm{Next}(z_{2i+2}, z_{2i+1}) \\ & \wedge \mathrm{F_t}(z) \wedge \mathrm{Node}_{\mathrm{t,k}}^{\mathrm{b}}(z_{2k}) \wedge \mathrm{ToBit}(z_{2m}, z) \wedge \mathrm{Bit}_b(z) \wedge \mathrm{ChosenBit}_k(z) \end{aligned}$$

Similarly, we detect if a head HasBit_k from a $\mathrm{G_t}$-nodes collapses on the wrong head $\mathrm{ToBit_b}$ in the sense of the Property 2 previously exposed, that is, if it collapses on the bit-value *not corresponding* to the branching leading to the $\mathrm{G_t}$ of interest. This is achieved with the following subqueries, with $t \in \{0,1\}$, $1 \leq k \leq m$, $b \in \{0,1\}$:

$$\begin{aligned} q_{\text{G-bit}}^{t,k,b} := \; & \exists z_0, \dots z_{2m+2}, z \; q_{\text{load}}(z_0) \wedge \bigwedge_{i=0}^{m} \mathrm{Next}(z_{2i}, z_{2i+1}) \wedge \mathrm{Next}(z_{2i+2}, z_{2i+1}) \\ & \wedge \mathrm{G_t}(z) \wedge \mathrm{Node}_{\mathrm{t,k}}^{\mathrm{b}}(z_{2k}) \wedge \mathrm{ToBit}(z_{2m}, z) \wedge \mathrm{Bit}_{1-b}(z) \wedge \mathrm{ChosenBit}_k(z) \end{aligned}$$

To conclude with the structural subqueries, we detect if there is a $\mathrm{G_t}$-node at depth less than expected, that is less than $2m+2$. Consider the subqueries, with $1 \leq d \leq m$ and $t \in \{0,1\}$:

$$\begin{aligned} q_{\text{G-depth}}^{t,d} := \; & \exists z_0, \dots z_{2d} \; q_{\text{load}}(z_0) \wedge \bigwedge_{i=0}^{d-1} \mathrm{Next}(z_{2i}, z_{2i+1}) \wedge \mathrm{Next}(z_{2i+2}, z_{2i+1}) \\ & \wedge \mathrm{G_t}(z_{2d}) \end{aligned}$$

We now move to consistency subqueries. Our first consistency subquery keeps track of the elements used as bits, coming from either a $\mathrm{F_t}$ or from a $\mathrm{G_t}$, node at depth d. Consider the two following subqueries with $d \in \{2m, 2m+1\}$:

$$q^{d}_{\#\ \mathrm{bool}} := \exists z_0, \ldots z_{2d}, z \ \mathrm{toStart}(\mathsf{r}, z_0) \wedge \bigwedge_{i=0}^{d-1} \mathrm{Next}(z_{2i}, z_{2i+1}) \wedge \mathrm{Next}(z_{2i+2}, z_{2i+1})$$
$$\wedge\ \mathrm{ToBit}(z_{2d}, z)$$

A similar subquery $q^{2m+2}_{\#\ \mathrm{color}}$ counts the number of colors issuing from $\mathrm{G_t}$-nodes.

Notice that if the tree-shaped structures are preserved in a model, we know that there should be at least $2 \times 2 \times 2^m + 1$ matches for $q^{2m}_{\#\ \mathrm{bool}}$, from the two bits of each of the 2^m $\mathrm{F_t}$-nodes of each tree T_t and from the individual b. The same holds for $q^{2m+2}_{\#\ \mathrm{bool}}$. For $q^{2m+2}_{\#\ \mathrm{color}}$, we shall similarly expect $|\mathcal{C}| \times 2 \times 2^m + 1$ matches. From the combination of these three subqueries, one shall hence expect the product N of these three numbers of matches, which is essentially $|\mathcal{C}| \times 2^{2m}$. Recalling that $m = 2n$ and that n is given in unary, we can find an integer M with a polynomially large binary encoding and such that $2^M > N$. This is how we set M in the subquery q_{load}.

It can now be verified that if $\mathcal{I}$ is a model minimizing the number of matches for the (yet not fully-defined) query q and that $f : \mathcal{C}_\mathcal{K} \to \mathcal{I}$ is a homomorphism from the canonical model of $\mathcal{K}$, then $\mathcal{I}$ satisfies the following property:
Property ($\star$): f is injective for the tree-shaped structures of $\mathcal{C}_\mathcal{K}$ issuing from a_0 and from a_1, except for those elements ending by either HasCol *or by* HasBit_i *for any* $1 \leq i \leq m$.

We next discuss the parts of the query that are used to check the tiling conditions. The general idea is as follows: say we have a model $\mathcal{I}$ satisfying Property ($\star$), then either there is an element with shape $\mathsf{a}_t \cdot w \cdot \mathrm{HasCol}$ such that

$$f(\mathsf{a}_t \cdot w \cdot \mathrm{HasCol}) \notin \{f(\mathsf{a}_t \cdot w \cdot \mathrm{ToCol_c}) \mid c \in \mathcal{C}\},$$

in which case $f(\mathsf{a}_t \cdot w \cdot \mathrm{HasCol})$ provides a new match for $q^{2m+2}_{\#\ \mathrm{color}}$, or there is no such element, in which case we can define two tilings τ_0 and τ_1 (maybe non-valid), associated with the trees issuing from a_0 and a_1, given by setting $\tau_t(u, v)$ to the color $c \in \mathcal{C}$ such that $f(\mathsf{a}_t \cdot w \cdot \mathrm{HasCol}) = f(\mathsf{a}_t \cdot w \cdot \mathrm{ToCol_c})$, and where w corresponds to the branchings encoding the coordinate (u, v). We first require that $\tau_0 = \tau_1$ and further test if τ_0, hence also τ_1 is valid. We hence need to check whether a $\mathrm{G_0}$-node and a $\mathrm{G_1}$-node correspond to the same coordinates. To do so, we borrow a (slightly patched version of) the query used in Lutz [2008], which allows us to check if two such nodes agree on the interpretation of a given bit $1 \leq km$. This is achieved with the following subquery $q^{k}_{\mathrm{same\ bit}}(z^{(0)}, z^{(1)})$, where $z^{(0)}$ and $z^{(1)}$ are the two variables intended to map on the nodes of interest, and which is depicted in Figure 4.4:

$$
\begin{aligned}
&\exists z_{-1}, \dots z_{4m+5}, z'_{-1}, \dots z'_{4m+5}, z_{0,val}, z'_{0,val}, z_{4m+4,val}, z'_{4m+4,val}\\
&\mathrm{toStart}(\mathsf{r}, z_{2m+2}) \land \mathrm{toStart}(\mathsf{r}, z'_{2m+2}) \land \mathrm{G}_0(z^{(0)}) \land \mathrm{G}_1(z^{(1)})\\
&\land \textstyle\bigwedge_{i=0}^{2m+1} \mathrm{Next}(z_{2i}, z_{2i+1}) \land \mathrm{Next}(z_{2i+2}, z_{2i+1}) \land \mathrm{Next}(z'_{2i}, z'_{2i+1}) \land \mathrm{Next}(z'_{2i+2}, z'_{2i+1})\\
&\land \mathrm{Next}(z_0, z''_0) \land \mathrm{Next}(z''_0, z'_0) \land \mathrm{Next}(z_{4m+4}, z''_{4m+4}) \land \mathrm{Next}(z''_{4m+4}, z'_{4m+4})\\
&\land \mathrm{Next}(z_0, z_{-1}) \land \mathrm{Next}(z_{-1}, z^{(0)}) \land \mathrm{Next}(z'_0, z'_{-1}) \land \mathrm{Next}(z'_{-1}, z^{(0)})\\
&\land \mathrm{Next}(z_{4m+4}, z_{4m+5}) \land \mathrm{Next}(z_{4m+5}, z^{(1)}) \land \mathrm{Next}(z'_{4m+4}, z'_{4m+5}) \land \mathrm{Next}(z'_{4m+5}, z^{(1)})\\
&\land \mathrm{ToBit}(z_0, z_{0,val}) \land \mathrm{ChosenBit}_k(z_{0,val}) \land \mathrm{Bit}_0(z_{0,val})\\
&\land \mathrm{ToBit}(z'_0, z'_{0,val}) \land \mathrm{ChosenBit}_k(z'_{0,val}) \land \mathrm{Bit}_1(z'_{0,val})\\
&\land \mathrm{ToBit}(z_{4m+4}, z_{4m+4,val}) \land \mathrm{ChosenBit}_k(z_{4m+4,val}) \land \mathrm{Bit}_1(z_{4m+4,val})\\
&\land \mathrm{ToBit}(z'_{4m+4}, z'_{4m+4,val}) \land \mathrm{ChosenBit}_k(z'_{4m+4,val}) \land \mathrm{Bit}_0(z'_{4m+4,val})
\end{aligned}
$$

It can be verified that, in a model minimizing the number of matches, hence satisfying Property ($\star$), if $q^k_{\text{same bit}}(z^{(0)}, z^{(1)})$ admits a non-trivial match, that is not a simple collapse on the individual b, then $z^{(0)}$ maps onto a G_0-node and $z^{(1)}$ maps onto a G_1-node such that both these nodes agree on the k^{th} bit of the coordinates they encode. Indeed, to eliminate the 6-cycle in the query going through $z^{(0)}$, z_0 and z'_0, one has to collapse $z^{(0)}$ on either z_0 or z'_0 (the case of z_0 collapsing on z'_0 is excluded as it would trigger a structural subquery $q^{t,k,b}_{\text{F-bit}}$ or $q^{t,k,b}_{\text{G-bit}}$ due to the image of z_0 having admitting the two possible values for its k^{th} bit, hence violating either Property 1 or Property 2). Say $z^{(0)}$ collapse with z_0. Since $z^{(0)}$ maps onto a G_0-node and that there exists no path shorter than $2m+2$ to reach the root of the tree (otherwise one structural subquery $q^{0,d}_{\text{G-depth}}$ would trigger), it enforces z_{2m+2} to map on a_0. But in that case, it is therefore impossible for $z^{(1)}$ to similarly collapse on z_{4m+4} as, for the same reason, it would enforce z_{2m+2} to map in a_1. However, if $z^{(1)}$ collapses on z'_{4m+4}, and hence z_{4m+4} collapses with the corresponding F_1-node, then a match becomes possible as the 2 moves made available in the path bridging z_0 and z_{4m+4} in query can now be used to move from a_0 to a_1. Therefore, either $z^{(0)}$ collapses on z_0 and $z^{(1)}$ on z'_{4m+4}, agreeing on a value of 0 for the k^{th}-bit, or $z^{(0)}$ collapses on z'_0 and $z^{(1)}$ on z_{4m+4} agreeing on a value of 1 for the k^{th} bit.

We therefore detect if the two tilings τ_1 and τ_2 differ with the following subqueries

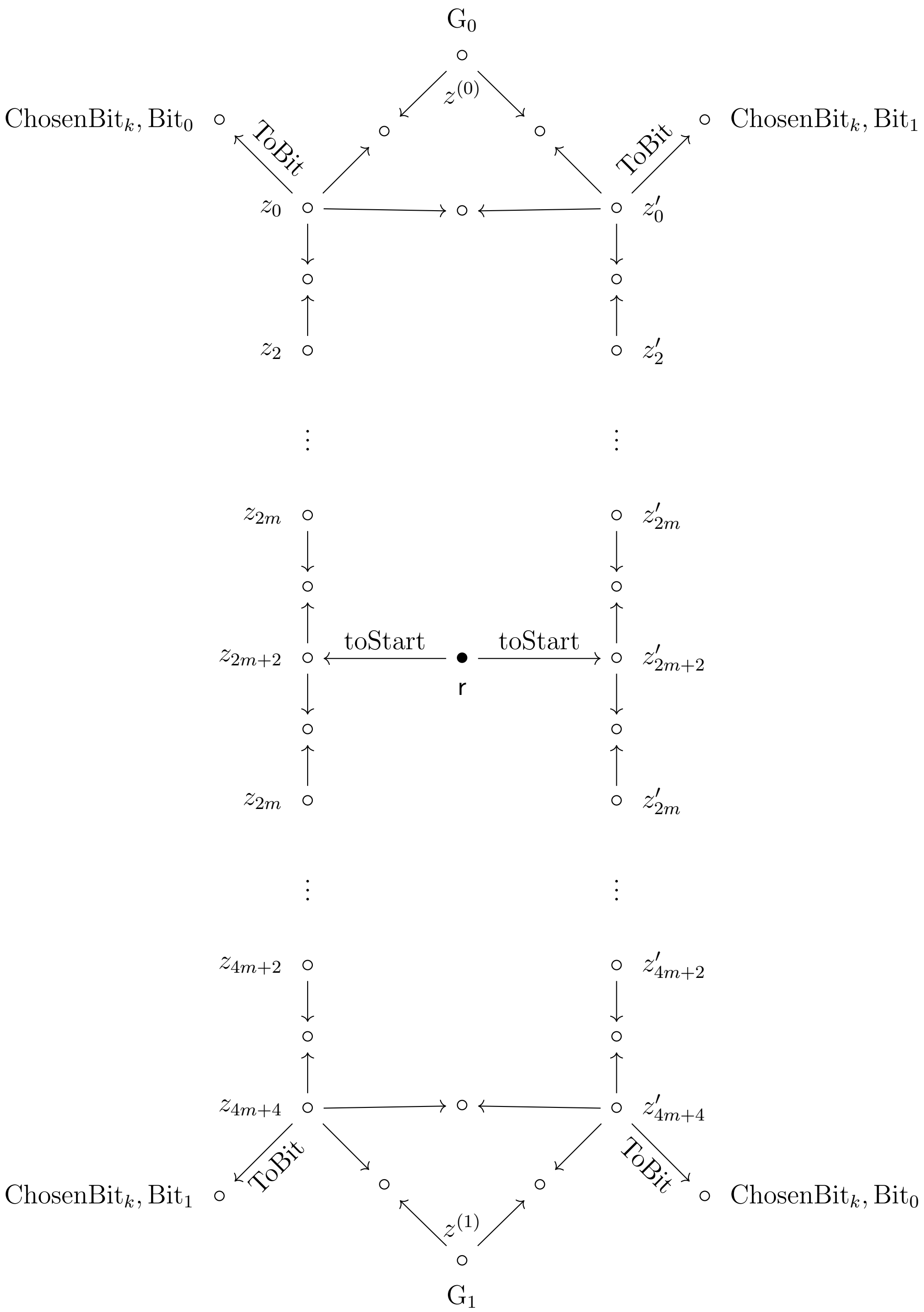


Figure 4.4: The query $q^{k}_{\text{same bit}}(z^{(0)}, z^{(1)})$. For readability, several variable names have been omitted and non-labeled edges depict Next-atoms.

$q^{c,c'}$ defined for each $c, c' \in \mathcal{C}$ such that $c \neq c'$ as follows.

$$\begin{aligned} q^{c,c'} = \exists z^{(0)}, z^{(1)}, z^{(0)}_{col}, z^{(1)}_{col} & \bigwedge_{i=1}^{m} q^{i}_{\text{same bit}}(z^{(0)}, z^{(1)}) \\ & \wedge \text{ToCol}(z^{(0)}, z^{(0)}_{col}) \wedge \text{ChosenColor}(z^{(0)}_{col}) \wedge \text{Color}_{\text{c}}(z^{(0)}_{col}) \\ & \wedge \text{ToCol}(z^{(1)}, z^{(1)}_{col}) \wedge \text{ChosenColor}(z^{(1)}_{col}) \wedge \text{Color}_{\text{c}'}(z^{(1)}_{col}) \end{aligned}$$

To detect adjacency, we remark that two grid positions $(h_1, v_1), (h_2, v_2) \in \{0, \ldots, 2^n - 1\} \times \{0, \ldots, 2^n - 1\}$ are vertically adjacent iff:

- $h_1 = h_2$, so the binary encodings of h_1 and h_2 are the same;
- $v_2 = v_1 + 1$, so the binary encodings of v_2 and v_1 are the same until, at some point, v_2 ends with $1 \cdot 0^k$ while v_1 ends with $0 \cdot 1^k$.

To detect a violation of the vertical tiling condition (i.e. two vertically adjacent tiles with colors c and c' such that $(c, c') \notin \mathcal{V}$), we need n queries, one for each possible position where the bit from the vertical coordinates differ. For each $1 \leq k \leq n$, we create a subquery $q^{\mathcal{V},(c,c'),k}$ defined as follows.

$$\begin{aligned} q^{\mathcal{V},(c,c'),k} = & \exists z^{(0)}, z^{(1)}, z^{(0)}_{n-k} \ldots z^{(0)}_{2n}, z^{(1)}_{n-k} \ldots z^{(1)}_{2n}, z^{(0)}_{col}, z^{(1)}_{col} \\ & \bigwedge_{i=1}^{n+k-2} q^{i}_{\text{same bit}}(z^{(0)}, z^{(1)}) \\ & \wedge \text{ToBit}(z^{(0)}, z^{(0)}_{n-k}) \wedge \text{ChosenBit}_{n-k}(z^{(0)}_{n-k}) \wedge \text{Bit}_0(z^{(0)}_{n-k}) \\ & \wedge \bigwedge_{i=n-k+1}^{2n} \text{ToBit}(z^{(0)}, z^{(0)}_{i}) \wedge \text{ChosenBit}_{i}(z^{(0)}_{i}) \wedge \text{Bit}_1(z^{(0)}_{i}) \\ & \wedge \text{ToBit}(z^{(1)}, z^{(1)}_{n-k}) \wedge \text{ChosenBit}_{n-k}(z^{(1)}_{n-k}) \wedge \text{Bit}_1(z^{(1)}_{n-k}) \\ & \wedge \bigwedge_{i=n-k+1}^{2n} \text{ToBit}(z^{(1)}, z^{(1)}_{i}) \wedge \text{ChosenBit}_{i}(z^{(1)}_{i}) \wedge \text{Bit}_0(z^{(1)}_{i}) \\ & \wedge \text{ToCol}(z^{(0)}, z^{(0)}_{col}) \wedge \text{ChosenColor}(z^{(0)}_{col}) \wedge \text{Color}_{\text{c}}(z^{(0)}_{col}) \\ & \wedge \text{ToCol}(z^{(1)}, z^{(1)}_{col}) \wedge \text{ChosenColor}(z^{(1)}_{col}) \wedge \text{Color}_{\text{c}'}(z^{(1)}_{col}) \end{aligned}$$

We can proceed as well to detect the horizontal violations, and we now let q be the conjunction of all the preceding subqueries.

To conclude from here, it suffices to prove the following claim:

$[N + 1, +\infty]$ is a certain answer for q over $\mathcal{K}$ iff $(n, \mathcal{C}, \mathcal{H}, \mathcal{V}) \notin$ ExpTiling.

($\Rightarrow$) Assume $[N+1,+\infty]$ is a certain answer for q over $\mathcal{K}$ and consider a tiling τ of exponential grid. One can build a model $\mathcal{I}_\tau$of $\mathcal{K}$ satisfying Properties 1, 2, 3 and ($\star$) to represent this tiling τ in a both trees issuing from a_0 and a_1. By construction, there shall be no matches for structural subqueries, hence *a priori* the only N matches in $\mathcal{I}_\tau$ from the three subqueries counting bit-like and color-like elements. However, since $[N+1,+\infty]$ is a certain answer for q over $\mathcal{K}$, there exists an additional match in $\mathcal{I}_\tau$, distinct from the N above. It can be verified that this can only comes from one of the subquery checking the validity of the tiling (such as $q^{\mathcal{V},(c,c'),k}$). By construction of $\mathcal{I}_\tau$, the G_0 and G_1 yielding this extra match correspond to a violation of the tiling τ, hence not being valid. As this holds for any initial choice of τ, it ensures that $(n,\mathcal{C},\mathcal{H},\mathcal{V}) \notin$ ExpTiling.

($\Leftarrow$) If $[N+1,+\infty]$ is not a certain answer for q over $\mathcal{K}$, then there exists a countermodel $\mathcal{I}$ for $N+1$. Therefore no structural subquery triggers in the structures issuing from a_0 and a_1 as that would lead to 2^M matches, being at least $N+1$, hence contradicting $\mathcal{I}$ being a countermodel for $N+1$. We hence have the N basics matches counted by the three subqueries $q^{2m}_{\#\text{ bool}}$, $q^{2m+2}_{\#\text{ bool}}$ and $q^{2m+2}_{\#\text{ color}}$. If the $\mathrm{G_t}$ are not reusing colors already counted by $q^{2m+2}_{\#\text{ color}}$, then it yields a new match for this latter subquery and again contradicts $\mathcal{I}$ being a countermodel for $N+1$. Therefore we can extract the two encoded tilings τ_1 and τ_2, and the subqueries $q^{c,c'}$ not admitting extra matches (again, that would contradict the countermodelhood of $\mathcal{I}$) ensures $\tau_1 = \tau_2$. From the subqueries $q^{\mathcal{V},(c,c'),k}$ not admitting extra matches, we derive that τ_1 is a valid tiling, hence $(n,\mathcal{C},\mathcal{H},\mathcal{V}) \in$ ExpTiling. □

We now explain how to adapt the construction to $\mathcal{ELI}$ ontologies, essentially by replacing each combination of DL-Lite$^{\mathcal{H}}_{\mathsf{pos}}$ axioms $\mathrm{A} \sqsubseteq \exists\mathrm{R}$, $\exists\mathrm{R}^- \sqsubseteq \mathrm{B}$ and $\mathrm{R} \sqsubseteq \mathrm{S}$ by the $\mathcal{ELI}$ axiom $\mathrm{A} \sqsubseteq \exists\mathrm{S.B}$.

Theorem 25. *Exhaustive rooted CCQ answering over $\mathcal{ELI}$ KBs is* coNEXP*-hard.*

Proof. As already mentioned, we replace each combination of DL-Lite$^{\mathcal{H}}_{\mathsf{pos}}$ axioms $\mathrm{A} \sqsubseteq \exists\mathrm{R}$, $\exists\mathrm{R}^- \sqsubseteq \mathrm{B}$ and $\mathrm{R} \sqsubseteq \mathrm{S}$ by the $\mathcal{ELI}$ axiom $\mathrm{A} \sqsubseteq \exists\mathrm{S.B}$. For example, the first block of axioms, generating most of the tree structures, becomes (for the relevant values of t, b and i):

$$\mathrm{Node_{t,i}} \sqsubseteq \exists\mathrm{Next.Node^b_{t,i}}$$

$$\mathrm{Node^b_{t,i}} \sqsubseteq \exists\mathrm{Next^-.Node_{t,i+1}}$$

Proceeding similarly for the subroles creating the $\mathrm{G_t}$-nodes, the subroles creating the bits and those creating the colors, we end up with an $\mathcal{ELI}$ TBox. If suffices now to notice that the rest of proof can remain unchanged as, in particular, we made sure not to use any subsumed roles in the query (which could have slightly simplified the DL-Lite$^{\mathcal{H}}_{\mathsf{pos}}$ construction), hence q doesn't require any update. □

4.4 Further refinements for $\mathcal{ALCH}$

In this section, we devise a procedure to compute the tightest certain answer to exhaustive rooted CCQs over $\mathcal{ALCH}$ KBs. The key ingredient is another refinement of interlacings leading to optimal models consisting of an enriched version of the ABox, that is, only involving individuals from the input ABox, completed as a model by *directed* tree-shaped structures (intuitively, the lack of inverse roles make all roles point in the same direction: deeper in the tree). Such directionality drastically restricts how the rooted CCQ can map into the tree-shaped structures, which allows for local characterizations of matches that require only polynomial space, while still being sufficient to capture the global number of matches. Verifying these local characterizations can be pieced together leads to an essentially PSPACE algorithm, similar in spirit to fork-rewriting approaches [Lutz, 2008] but also to the pattern-based approach developed in Chapter 3. For $\mathcal{ALC}$ and its extension $\mathcal{ALCH}$, the satisfiability check of the interpretation underlying each local characterization however requires an EXP procedure (see Theorem 3), hence an overall complexity of EXP for these two sublogics. The complexity drops to PSPACE for $\mathcal{ELH}_\bot$ ontologies for which the satisfiability check can be performed in P. The EXP matching lower bound is inherited from the analogous lower bound for the satisfiability problem, while the PSPACE lower bound proceeds by reduction from the Quantified Boolean Formula problem (QBF) using some original tricks also used in the preceding proof of Theorem 25.

4.4.1 The interlacing function $f^\star$

We begin with a new refinement $f^\star$ of the interlacing function. To introduce it, let $\mathcal{K} := (\mathcal{T}, \mathcal{A})$ be an $\mathcal{ALCH}$ KB, $\mathcal{I}$ a model of $\mathcal{K}$, and q an exhaustive rooted CCQ. We recall again that Ω denotes the set of heads of existential rules from $\mathcal{T}$, that Δ° denotes the existential extraction of $\mathcal{I}$ and that $f : \Delta^\circ \to \Delta^\mathcal{I}$ is the mapping used to build this existential extraction (see Definition 19). We also recall that Δ^* is the subset of $\Delta^\mathcal{I}$ containing all individuals from $\mathcal{A}$ and all elements reached by matches of q in $\mathcal{I}$ (see Definition 22).

The idea underlying $f^\star$ is to obtain a forest model of $\mathcal{K}$, that is, a model consisting of an interpretation over $\mathsf{Ind}(\mathcal{A})$ extended with tree-shaped structures rooted on individuals from $\mathsf{Ind}(\mathcal{A})$, with at most as many matches as in $\mathcal{I}$. The major improvement of such a model compared to our previous approaches is that the central domain of the forest-model is $\mathsf{Ind}(\mathcal{A})$ instead of Δ^*, the latter being eventually exponentially large. Existence of such forest models is already used to answer CQs over expressive DLs such as $\mathcal{SH}$ [Lutz, 2008]. However, existing constructions essentially consist in considering the Id-interlacing of $\mathcal{I}$, which may contain more matches than the original model $\mathcal{I}$. Indeed, apart from several ABox

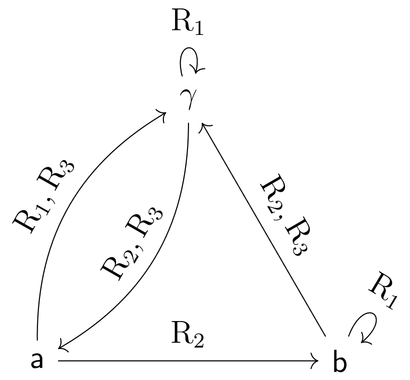


Figure 4.5: A model $\mathcal{I}_e$ of $\mathcal{K}_e$ from Example 16. Labels for concept A (on each visible vertex) and role S (on each visible edge) are omitted.

facts not being reused, we may need to merge together two elements $w \cdot \mathrm{R.B}_1$ and $w \cdot \mathrm{R.B}_2$ from the Id-interlacing to minimize the number of matches for any role subsuming R (*e.g.* R itself). Surprisingly, for exhaustive rooted CCQs and in the absence of inverse roles, this is all we need to modify in the Id-interlacing to obtain a model with at most as many matches than in the original model.

To identify which such elements should be identified, we simply mimic the initial model $\mathcal{I}$ and capture its behavior through the following equivalence relations on heads from Ω.

Definition 43. *Let e be an element of $\mathcal{I}$. We define an equivalence relation $\sim_e$ on elements of Ω: two elements h_1 and h_2 of Ω are equivalent w.r.t. e iff $\mathsf{succ}^{\mathcal{I}}_{\mathrm{h}_1}(e)$ and $\mathsf{succ}^{\mathcal{I}}_{\mathrm{h}_2}(e)$ are both defined and equal. We denote by $\overline{h}^{e}$ the equivalence class of h for relation $\sim_e$.*

We can now define the new interlacing function $f^\star$, which, as explained, reuses facts from $\mathcal{A}$ when $\mathcal{I}$ does (only as long as we remain among the individuals $\mathsf{Ind}(\mathcal{A})$, for the same reasons as explained in Section 4.3.1), and further identifies elements from the existential extraction according to the equivalence relations defined just above.

Definition 44. *The interlacing function $f^\star$ is defined inductively as:*

$$\begin{array}{rcll} f^\star : \Delta^\circ & \to & \mathsf{Ind}(\mathcal{A}) \cdot (2^\Omega)^* & \\ \mathsf{a} & \mapsto & \mathsf{a} & \star_0 \\ w \cdot h & \mapsto & \begin{cases} f(w \cdot h) & \textit{if } f^\star(w) \in \mathsf{Ind}(\mathcal{A}) \textit{ and } f(w \cdot h) \in \mathsf{Ind}(\mathcal{A}) \\ f^\star(w) \cdot \overline{h}^{f(w)} & \textit{otherwise} \end{cases} & \begin{array}{l}\star_1 \\ \star_2\end{array} \end{array}$$

Remark 20. *Notice that in both Cases $\star_0$ and $\star_1$, we have $f^\star(w) = f(w)$.*

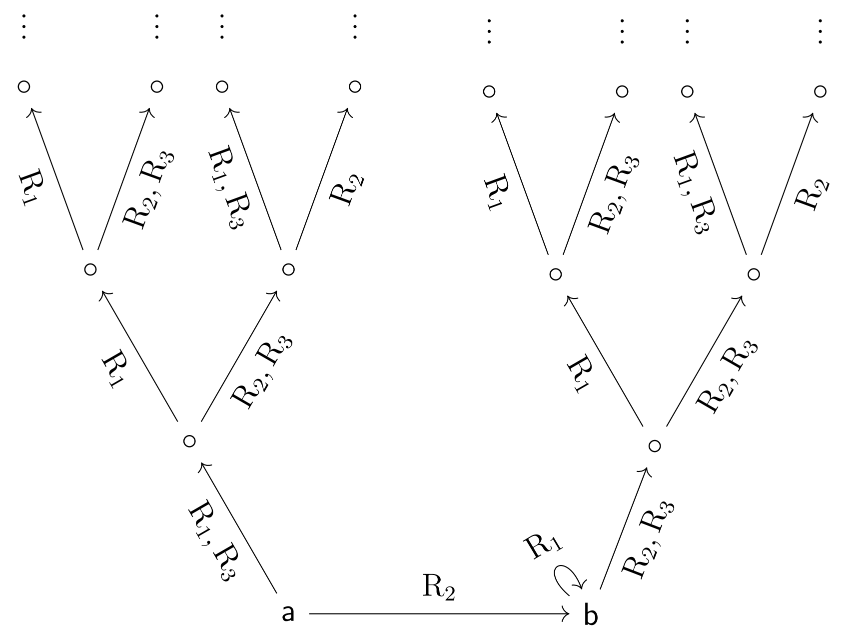


Figure 4.6: Initial portion of the $f^\star$-interlacing of $\mathcal{I}_e$. Labels for concept A (on each visible vertex) and role S (on each visible edge) have been omitted.

Example 16. *We illustrate this construction with the KB $\mathcal{K}_e$ given by the following ABox $\mathcal{A}_e := \{\mathrm{A}(\mathsf{a}),\ \mathrm{A}(\mathsf{b}),\ \mathrm{S}(\mathsf{a},\mathsf{b})\}$ and the following TBox $\mathcal{T}_e$:*

$$\mathrm{A} \sqsubseteq \exists \mathrm{R}_1.\mathrm{A} \quad \mathrm{A} \sqsubseteq \exists \mathrm{R}_2.\mathrm{A} \quad \mathrm{A} \sqsubseteq \exists \mathrm{R}_3.\mathrm{A}$$

$$\mathrm{R}_1 \sqsubseteq \mathrm{S} \qquad \mathrm{R}_2 \sqsubseteq \mathrm{S} \qquad \mathrm{R}_3 \sqsubseteq \mathrm{S}$$

$$\mathrm{R}_1 \sqcap \mathrm{R}_2 \sqsubseteq \bot$$

A model $\mathcal{I}_e$ of $\mathcal{K}_e$ is depicted in Figure 4.5. Its $f^\star$-interlacing $\mathcal{I}^\star$ is depicted in Figure 4.6. Notice the two directed tree-shaped structures issuing from a *and* b*.*

We now verify that $f^\star$ is pseudo-injective to ensure modelhood of the resulting $f^\star$-interlacing $\mathcal{I}^\star$ *via* Theorem 4.

Lemma 22. *$f^\star$ is pseudo-injective.*

Proof. We need to prove that for all u and all v in Δ°, if $f^\star(u) = f^\star(v)$, then $f(u) = f(v)$. We proceed by induction on u.

$u \in \mathsf{Ind}(\mathcal{A})$. By definition of f and $f^\star$ (case $\star_0$), we have $f(u) = u$ and $f^\star(u) = u$. Let $v \in \Delta^\circ$. We distinguish the 3 possible cases for $f^\star(v)$:

$\star_0$ $\star_1$. Based on Remark 20, we have $f^\star(v) = f(v)$. Therefore assuming $f^\star(u) = f^\star(v)$ gives $f(u) = f(v)$.

$\star_2$. We have $f^\star(v) = f^\star(w) \cdot \overline{h}^{w}$. In particular $f^\star(v) \notin \mathsf{Ind}(\mathcal{A})$. Assuming $f^\star(u) = f^\star(v)$ yields a contradiction as $f^\star(u) = u \in \mathsf{Ind}(\mathcal{A})$.

$u = u_0 \cdot h$. If $f^\star(u)$ is in Case $\star_1$, then Remark 20 and the same arguments as in the base case conclude (notice $f^\star(u) \in \mathsf{Ind}(\mathcal{A})$ still holds). Otherwise, $f^\star(u)$ is in Case $\star_2$, that is $f^\star(u) = f^\star(u_0) \cdot \overline{h}^{f(u_0)}$. Let $v \in \Delta^\circ$. If v is in Case $\star_0$ or in Case $\star_0$, then $f^\star(v) \in \mathsf{Ind}(\mathcal{A})$, which yields a contradiction. Otherwise, v is in Case $\star_2$, that is $f^\star(v) = f^\star(v_0) \cdot \overline{h'}^{f(v_0)}$, with $v = v_0 \cdot h'$. Assuming $f^\star(u) = f^\star(v)$ yields $f^\star(u_0) = f^\star(v_0)$ and $\overline{h}^{f(u_0)} = \overline{h'}^{f(v_0)}$. Induction hypothesis gives $f(u_0) = f(v_0)$. In particular, it ensures $h \sim_{f(u_0)} h'$, that is $\mathsf{succ}^{\mathcal{I}}_{\mathrm{h}}(f(u_0)) = \mathsf{succ}^{\mathcal{I}}_{\mathrm{h'}}(f(u_0))$. The definition of f hence gives $f(u_0 \cdot h) = f(v_0 \cdot h')$, that is $f(u) = f(v)$. □

It remains to consider the matches of q in the $f^\star$-interlacing $\mathcal{I}^\star$. Importantly, all the construction above is query independent (observe that we do not rely on Δ^* to define $f^\star$), which leads to the following statement.

Lemma 23. *Let $\mathcal{I}^\star$ be the $f^\star$-interlacing of $\mathcal{I}$. For every exhaustive rooted CCQ q, there are at most as many matches of q in $\mathcal{I}^\star$ as there are in $\mathcal{I}$.*

Proof. Let q be an exhaustive rooted CCQ. We prove that σ injects matches of $\mathcal{I}^\star$ into matches of $\mathcal{I}$ (we recall $\sigma : \mathcal{I}^\star \to \mathcal{I}$ is the homomorphism obtained in Theorem 4). Let π_1 and π_2 be two matches $q \to \mathcal{I}^\star$ such that $\sigma \circ \pi_1 = \sigma \circ \pi_2$.We prove by induction on each connected component p of q that $(\pi_1)_{|p} = (\pi_2)_{|p}$ on p. It is therefore sufficient to focus on the case of a connected exhaustive rooted CCQ q (so that $p = q$ and $(\pi_1)_{|p} = \pi_1$ and $(\pi_2)_{|p} = \pi_2$).

Base case: individual terms of q. Consider an individual a occurring in q. Since π_1 and π_2 are matches, they satisfy in particular that $\pi_1(\mathsf{a}) = \pi_2(\mathsf{a})$.

Induction case. Consider a counting variable z of q which is connected by a role P to a term t being closer to an individual than z is. The induction hypothesis hence ensures that $\pi_1(t) = \pi_2(t)$. There are two main cases to distinguish based on the direction of the connection, that is either $\mathrm{P}(z, t) \in q$ or $\mathrm{P}(t, z) \in q$.

Subcase $\mathrm{P}(t, z) \in q$. Since π_1 is a match, we have $(\pi_1(t), \pi_1(z)) \in \mathrm{P}^{\mathcal{I}^\star}$. In the absence of inverse roles, it only yields Cases ∇_0 and ∇_+ from the definition of $\mathrm{P}^{\mathcal{I}^\star}$ (see Definition 20). The same holds for π_2, yielding 4 subcases:

$\triangledown_0 \cdot \triangledown_0$. In particular, we have $\pi_1(z), \pi_2(z) \in \mathsf{Ind}(\mathcal{A})$. Therefore $\sigma(\pi_1(z)) = \pi_1(z)$ and $\sigma(\pi_2(z)) = \pi_2(z)$. Recalling the assumption $\sigma \circ \pi_1 = \sigma \circ \pi_2$ yields $\pi_1(z) = \pi_2(z)$.

$\triangledown_0 \cdot \triangledown_+$. From Case $\triangledown_0$ on π_1 we obtain $\pi_1(t), \pi_1(z) \in \mathsf{Ind}(\mathcal{A})$ and $\sigma(\pi_1(z)) = \pi_1(z)$. From Case $\triangledown_+$ on π_2, we can write $(\pi_2(t), \pi_2(z))$ as $(f^\star(w), f^\star(w \cdot h))$. On the first hand, from $\pi_1(t) = \pi_2(t)$ we get $f^\star(w) \in \mathsf{Ind}(\mathcal{A})$. On the second hand, from $\sigma \circ \pi_1 = \sigma \circ \pi_2$ and $f = \sigma \circ f^\star$, we obtain $f^\star(w \cdot h) \in \mathsf{Ind}(\mathcal{A})$. Therefore, $f^\star(w \cdot h)$ is in Case $\star_1$ and we have $f^\star(w \cdot h) = f(w \cdot h)$, that is $\pi_2(z) = \sigma(\pi_2(z))$ by reusing $f = \sigma \circ f^\star$. Recalling $\sigma \circ \pi_1 = \sigma \circ \pi_2$, we now obtain $\pi_1(z) = \pi_2(z)$.

$\triangledown_+ \cdot \triangledown_0$. Symmetric to the previous case $\triangledown_0 \cdot \triangledown_+$.

$\triangledown_+ \cdot \triangledown_+$. From both cases $\triangledown_+$, we can write $(\pi_1(t), \pi_1(z))$ as $(f^\star(w_1), f^\star(w_1 \cdot h_1))$ and $(\pi_2(t), \pi_2(z))$ as $(f^\star(w_2), f^\star(w_2 \cdot h_2))$. From $\pi_1(t) = \pi_2(t)$ we get $f^\star(w_1) = f^\star(w_2)$. From $\sigma \circ \pi_1 = \sigma \circ \pi_2$ and $f = \sigma \circ f^\star$, we obtain $f(w_1) = f(w_2)$ and $f(w_1 \cdot h_1) = f(w_2 \cdot h_2)$. Combining the latter with the definition of f gives us $\mathsf{succ}^{\mathcal{I}}_{\mathrm{h}_1}(f(w_1)) = \mathsf{succ}^{\mathcal{I}}_{\mathrm{h}_2}(f(w_2))$, which, when further combined with the former equality, yields $h_1 \sim_{f(w_1)} h_2$. Altogether, this ensures $f^\star(w_1 \cdot h_1) = f^\star(w_2 \cdot h_2)$, that is $\pi_1(z) = \pi_2(z)$.

Subcase $\mathrm{P}(z, t) \in q$. Since π_1 is a match, we have $(\pi_1(z), \pi_1(t)) \in \mathrm{P}^{\mathcal{I}^\star}$. In the absence of inverse roles, it only yields Cases $\triangledown_0$ and $\triangledown_+$ from the definition of $\mathrm{P}^{\mathcal{I}^\star}$. The same holds for π_2, yielding 4 subcases:

$\triangledown_0 \cdot \triangledown_0$. In particular, we have $\pi_1(z), \pi_2(z) \in \mathsf{Ind}(\mathcal{A})$. Therefore $\sigma(\pi_1(z)) = \pi_1(z)$ and $\sigma(\pi_2(z)) = \pi_2(z)$. Recalling the assumption $\sigma \circ \pi_1 = \sigma \circ \pi_2$ yields $\pi_1(z) = \pi_2(z)$.

$\triangledown_0 \cdot \triangledown_+$. From Case $\triangledown_0$ on π_1 we obtain $\pi_1(z), \pi_1(t) \in \mathsf{Ind}(\mathcal{A})$ and $\sigma(\pi_1(t)) = \pi_1(t)$. From Case $\triangledown_+$ on π_2, we can write $(\pi_2(z), \pi_2(t))$ as $(f^\star(w), f^\star(w \cdot h))$. From $\pi_1(t) = \pi_2(t)$ we now get $f^\star(w \cdot h) \in \mathsf{Ind}(\mathcal{A})$. Hence $f^\star(w) \in \mathsf{Ind}(\mathcal{A})$, that is $\pi_2(z) \in \mathsf{Ind}(\mathcal{A})$, and we conclude as in case $\triangledown_0 \cdot \triangledown_0$.

$\triangledown_+ \cdot \triangledown_0$. Symmetric to the previous case $\triangledown_0 \cdot \triangledown_+$.

$\triangledown_+ \cdot \triangledown_+$. From both cases $\triangledown_+$, we can write $(\pi_1(t), \pi_1(z))$ as $(f^\star(w_1), f^\star(w_1 \cdot h_1))$ and $(\pi_2(t), \pi_2(z))$ as $(f^\star(w_2), f^\star(w_2 \cdot h_2))$. From $\pi_1(t) = \pi_2(t)$ we get $f^\star(w_1 \cdot h_1) = f^\star(w_2 \cdot h_2)$. If this latter common value belongs to $\mathsf{Ind}(\mathcal{A})$, then so do $f^\star(w_1)$ and $f^\star(w_2)$ and we conclude as in the subcase $\triangledown_0 \cdot \triangledown_0$. Otherwise, we must have $f^\star(w_1 \cdot h_1) = f^\star(w_1) \cdot \overline{h}_1^{f(w_1)}$ and $f^\star(w_2 \cdot h_2) =$

$f^\star(w_2) \cdot \overline{h_2}^{f(w_2)}$. Recall $f^\star(w_1 \cdot h_1) = f^\star(w_2 \cdot h_2)$, hence $f^\star(w_1) = f^\star(w_2)$, that is $\pi_1(z) = \pi_2(z)$. □

4.4.2 A PSPACE algorithm, up to satisfiability

The goal of this section is to establish the following result, relying on the structure of $f^\star$-interlacings.

Theorem 26. *Let $\mathcal{L}$ be a subclass of $\mathcal{ALCH}$ KBs and denote by* SAT($\mathcal{L}$) *the satisfiability problem of $\mathcal{L}$ KBs. There exists a* PSPACE *algorithm with access to a* SAT($\mathcal{L}$) *oracle for answering exhaustive rooted CCQ over $\mathcal{L}$.*

We obtain the following two corollaries for fragments of $\mathcal{ALCH}$.

Corollary 4. *Exhaustive rooted CCQ answering over $\mathcal{ALCH}$ ontologies is in* EXP.

Proof. Since SAT($\mathcal{ALCH}$) $\in$ EXP (see Theorem 3) and that PSPACE $\subseteq$ EXP, Theorem 26 yields an overall EXP procedure. □

Corollary 5. *Exhaustive rooted CCQ answering over $\mathcal{ELH}_\bot$ ontologies is in* PSPACE.

Proof. Since SAT($\mathcal{ELH}_\bot$) $\in$ P and that P $\subseteq$ PSPACE, Theorem 26 yields an overall PSPACE procedure. □

The remainder of this section is devoted to the proof of Theorem 26. Let $\mathcal{K} := (\mathcal{T}, \mathcal{A})$ be an $\mathcal{ALCH}$ KB, q be an exhaustive rooted CCQ and m be a candidate integer. We prove that if a countermodel for m exists, then its existence can be checked in NPSPACE (up to satisfiability checks), which yields a coNPSPACE (again, up to satisfiability checks) procedure. Savitch's theorem [Savitch, 1970] and closure of PSPACE under complement then concludes the proof. We start by giving a general intuition of our approach.

Assume that there exists a countermodel $\mathcal{I}$ for m, and let $\mathcal{I}^\star$ be its $f^\star$-interlacing. Consider a match π of an exhaustive rooted CCQ q in $\mathcal{I}^\star$. The query q can be decomposed into a subquery $p_0 \subseteq q$ mapped by π onto $\mathsf{Ind}(\mathcal{A})$ and other subqueries $p_1, \dots p_k \subseteq q$ mapping to directed-tree shaped structures $T_1, \dots T_k$ of $\mathcal{I}^\star$, each T_i being a set of words admitting $\mathsf{a_i} \cdot h_i$ as a prefix for some individual $\mathsf{a_i} \in \mathsf{Ind}(\mathcal{A})$ and some equivalence class $h_i \subseteq \Omega$. If we fix $\pi_{|p_0}$, subqueries $p_1, \dots p_k \subseteq q$, and the $\mathsf{a_i} \cdot h_i$, then the number of matches of q mapping p_0 as $\pi_{|p_0}$ and each p_i to each T_i can simply be obtained by multiplying the number of matches of each p_i in each T_i that can be consistently assembled with $\pi_{|p_0}$. The number of matches of q in $\mathcal{I}^\star$ can then be obtained by forming the sum of these products over each possible choice of $\pi_{|p_0}$, subqueries $p_1, \dots p_k \subseteq q$ and $\mathsf{a_i} \cdot h_i$.

The key ingredient is hence the number of matches of p_i in T_i that are consistent with $\pi_{|p_0}$. Since q is rooted, p_0 cannot be empty, hence p_i has to reach $\mathsf{a_i} \cdot h_i$. Since there are no inverse roles, the variables of p_i that shall map onto $\mathsf{a_i} \cdot h_i$ are fully decided and further variables must map further in T_i. Since q is exhaustive there can be no "blank" steps, hence the number of such further matches can be decomposed as we did with q: as the sum, over ways to split p_i into further structures of T_i, of the number of matches induced by each such split, which itself can be obtained as a product.

Our procedure starts by guessing an interpretation corresponding to $\mathcal{I}^\star$ restricted to $\mathsf{Ind}(\mathcal{A})$ and all elements with shape $\mathsf{a} \cdot h$ (the first layer of anonymous elements in $\mathcal{I}^\star$), together with a promise function χ indicating the number of matches one can expect for each relevant $p \subseteq q$ in the directed tree-structure following each $\mathsf{a} \cdot h$. Importantly, the number of such p to be considered is polynomial (due to the absence of inverse roles in the directed tree-structure, each relevant p is essentially characterized by which variable of q maps onto $\mathsf{a} \cdot h$), so that χ has polynomial size as a function. The procedure further checks the consistency of the promise χ by guessing how each $\mathsf{a} \cdot h$ further extends (that is, guessing elements of $\mathcal{I}^\star$ with shape $\mathsf{a} \cdot h \cdot h'$ and an extension of the promise χ to these elements). This is performed in a depth-first manner so that we can drop information each time we reach a depth greater than $|q|$, hence only using a polynomial amount of space. Satisfiability is checked at each step, hence the need for an oracle for $\textsc{Sat}(\mathcal{L})$, so that if no inconsistency is detected at the end, the union of all guessed branches can be extended to a model (recall we only guess the first $|q|$ layers of $\mathcal{I}^\star$ along the procedure), whose number of matches is encoded in the initial interpretation and promise.

We now move to a proper proof. Again, let $\mathcal{K} := (\mathcal{T}, \mathcal{A})$ be a $\mathcal{ALCH}$ KB, q an exhaustive rooted CCQ and m be a candidate integer. Let us recall that if m is greater than some exponential bound M depending only on $\mathcal{K}$ and q (see Theorem 3), then $[m, +\infty]$ is not a certain answer and we can return false. We henceforth assume $m \leq M$. To identify how constrained are the mappings q in directed tree structures, we introduce the following notions enlightening the dependencies between terms of a query.

Definition 45. *Let $s \in \mathsf{terms}(q)$. The possible depths $\Delta_s(t)$ of terms $t \in \mathsf{terms}(q)$ relative to s are defined inductively as the smallest sets such that for all $t, u \in$ $\mathsf{terms}(q)$, all role names* P *and all integers $n \geq 0$, we have:*

$$\begin{cases} 0 \in \Delta_s(s) \\ n \in \Delta_s(t) \ \wedge\ P(t,u) \in q \ \rightarrow\ n+1 \in \Delta_s(u) \\ n+1 \in \Delta_s(t) \ \wedge\ P(u,t) \in q \ \rightarrow\ n \in \Delta_s(u) \end{cases}$$

We say that a term $t \in \mathsf{terms}(q)$ depends on s if $\Delta_s(t) \neq \emptyset$. We denote $\mathsf{dep}(s)$ *the*

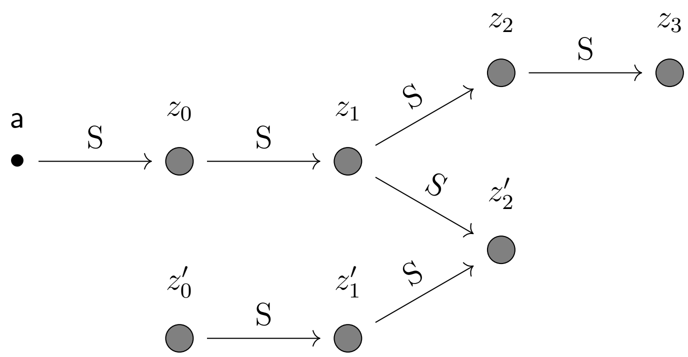


Figure 4.7: The query q_e from Example 17.

set of terms depending on s and denote $q_s := q_{|\mathsf{dep}(s)}$ *the query restricted to terms depending on s.*

We say a term s from q is tree-mappable if $\mathsf{dep}(s) \cap \mathsf{Ind} = \emptyset$ *and for all* $t \in \mathsf{dep}(s)$ *the set* $\Delta_s(t)$ *is a singleton. We denote by* $\mathcal{V}_{\mathsf{anon}}$ *by the set of tree-mappable terms. If* $s \in \mathcal{V}_{\mathsf{anon}}$*, we define* $\delta_s : \mathsf{dep}(s) \to \mathbb{N}$ *the function that maps a term t depending on s to the single element of* $\Delta_s(t)$ *and call it* the *depth of t relative to s.*

Remark 21. *Notice that for all* $s \in \mathsf{terms}(q)$*, we have* $s \in \mathsf{dep}(s)$*. The two conditions for a term to be tree-mappable reflect the facts that: all individuals can only match on themselves; if a term depending on s has two possible depths, then there exists a cycle in the query that cannot be collapsed onto a directed tree-shaped structure. In particular, individuals are not tree-mappable.*

One situation is of particular interest, namely when two terms of q have relative depth to each other of 0, *i.e.* two terms t_1 and t_2 such that $\delta_{t_1}(t_2) = 0$ (which implies $\delta_{t_2}(t_1) = 0$). These are two terms satisfying that if t_1 maps in an directed tree-shaped structure, then t_2 must map on the same element from the structure. Keeping track of these "equivalent" variables is primordial to avoid counting as different the matches of q_{t_1} and of q_{t_2}.

Lemma 24. *Let* $s_1, s_2 \in \mathcal{V}_{\mathsf{anon}}$*, we denote* $s_1 \sim s_2$ *iff* $\delta_{s_1}(s_2) = 0$*. The relation* $\sim$ *defines an equivalence relation on* $\mathcal{V}_{\mathsf{anon}}$*.*

Proof. Reflexivity is trivial. To prove symmetry, that is $\delta_{s_1}(s_2) = 0$ implies $\delta_{s_2}(s_1) = 0$, one simply follows backwards the sequence of atoms leading from s_1 to s_2. Transitivity is obtained by concatenating the two intermediate sequences. More generally, one can prove that if $\delta_{s_1}(s_2) = k$ and $\delta_{s_2}(s_3) = k'$, then $\delta_{s_1}(s_3) = k + k'$. □

Example 17. *Consider the exhaustive rooted CCQ q_e, depicted in Figure 4.7, and defined as follows:*

$$q_e \;:=\; \exists z_0, z_1, z_2, z_3, z_0', z_1', z_2' \;\; \mathrm{S}(\mathsf{a}, z_0) \;\wedge\; \mathrm{S}(z_0, z_1) \;\wedge\; \mathrm{S}(z_1, z_2) \;\wedge\; \mathrm{S}(z_2, z_3) \\ \wedge\; \mathrm{S}(z_0', z_1') \;\wedge\; \mathrm{S}(z_1', z_2') \;\wedge\; \mathrm{S}(z_1, z_2')$$

One can verify that $\delta_{z_1}(z_3) = 2$ while $\delta_{z_3}(z_1)$ is undefined. Furthermore, since $\delta_{z_1}(z_1') = 0$, we have $z_1 \sim z_1'$. Similarly $z_0 \sim z_0'$ holds, while $z_2 \not\sim z_2'$. Continuing Example 16, the model $\mathcal{I}_e$ has 144 matches (among which 36 map z_1 to a, 36 map z_1 to b, and 72 map z_1 to γ) for the query q_e, while its $f^\star$-interlacing $\mathcal{I}^\star$ only has 40 matches for q_e.

We now abstract the branches one can encounter in $f^\star$-interlacings as follows.

Definition 46. *Let W be a word from $\mathcal{W} := \{\varepsilon\} \cup \{\mathsf{a} \cdot h_1 \cdots h_n \mid \mathsf{a} \in \mathsf{Ind}(\mathcal{A}), h_i \in 2^\Omega, n \geq 0\}$, where ε denotes the empty word. A W-branch $\mathbb{B}$ is an interpretation whose domain $\Delta^\mathbb{B}$ is divided into two disjoints sets of elements:*

- *a set $\mathsf{inner}(\mathbb{B})$ of inner elements being all individuals from $\mathsf{Ind}(\mathcal{A})$ and all prefixes of W;*
- *a set $\mathsf{front}(\mathbb{B})$ of frontier elements with shapes $w \cdot h$ such that w is an inner element of $\mathbb{B}$ and h a set of heads.*

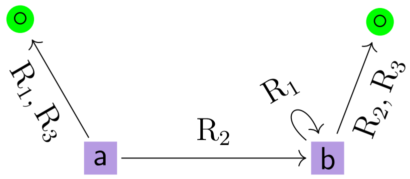
and such that:

1. *$\mathbb{B} \models \mathcal{A}$;*
2. *$\mathbb{B}$ is $\mathcal{T}$-satisfiable;*
3. *For all role names P, we have: $(u, v) \in \mathrm{P}^\mathbb{B}$ iff either $u, v \in \mathsf{Ind}(\mathcal{A})$ or $v = u \cdot h$ and h contains some head $\mathrm{R.B}$ such that $\mathcal{T} \models \mathrm{R} \sqsubseteq \mathrm{P}$;*
4. *No rule apply on inner elements of $\mathbb{B}$,* i.e. *there are no $\mathrm{R.B} \in \Omega$ and $e \in \mathsf{inner}(\mathbb{B})$ such that there exists $\mathrm{A} \sqsubseteq \exists \mathrm{R.B} \in \mathcal{T}$ with $e \in \mathrm{A}^\mathbb{B}$ and $e \notin (\exists \mathrm{R.B})^\mathbb{B}$.*

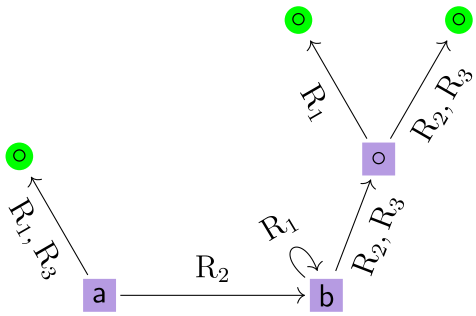
Example 18. *Figures 4.8a, 4.8b and 4.8c depict respectively an ε-branch $\mathbb{B}^{(0)}$, a $\mathsf{b} \cdot \{\mathrm{R}_2.\mathrm{A}, \mathrm{R}_3.\mathrm{A}\}$-branch $\mathbb{B}^{(1)}$, and a $\mathsf{b} \cdot \{\mathrm{R}_2.\mathrm{A}, \mathrm{R}_3.\mathrm{A}\} \cdot \{\mathrm{R}_1.\mathrm{A}\}$-branch $\mathbb{B}^{(2)}$. Inner elements are indicated using square-purple and frontier elements by circle-green.*

As explained before, we now introduce a promise to specify how many matches for each q_t can be found in the directed tree-structure following each non-individual element.

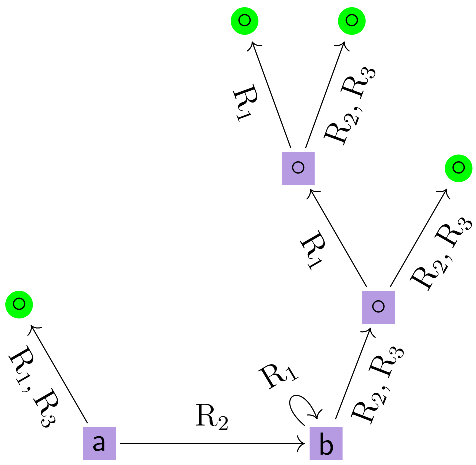
Definition 47. *A weighted branch $(\mathbb{B}, \chi)$ is a branch $\mathbb{B}$ along with a promise χ, that is a family $\chi := (\chi_w)_{w \in \Delta^\mathbb{B} \setminus \mathsf{Ind}}$ where each χ_w is a function $\mathsf{terms}(q) \to \{0, \ldots M\}$.*



(a) The ε-branch $\mathbb{B}^{(0)}$.



(b) The $\mathsf{b} \cdot \{R_2.A, R_3.A\}$-branch $\mathbb{B}^{(1)}$.



(c) The $\mathsf{b} \cdot \{R_2.A, R_3.A\} \cdot \{R_1.A\}$-branch $\mathbb{B}^{(2)}$.

Figure 4.8: The tree branches $\mathbb{B}^{(0)}$, $\mathbb{B}^{(1)}$ and $\mathbb{B}^{(2)}$ from Example 18. Labels for concept A (on each visible vertex) and role S (on each visible edge) are omitted. Inner elements are indicated using square-purple and frontier elements by circle-green.

Remark 22. *The exponential bound M allows χ to be described in polynomial size. Without such a bound, we would have to consider functions* $\mathsf{terms}(q) \to \mathbb{N}$, *that might be impossible to fully guess in polynomial space.*

As we need to combine weighted branches together, it is useful to define the notion of induced weighted branches.

Definition 48. *Let $(\mathbb{B}, \chi)$ is a weighted-W-branch and that w is a prefix of W. The* induced weighted-w-branch *of $(\mathbb{B}, \chi)$ is the biggest (w.r.t. inclusion of interpretations and inclusions of functions) weighted-w-branch that is included in $(\mathbb{B}, \chi)$.*

Remark 23. *The above definition is slightly abusive as we need in general to modify the inner and frontier sets of elements: if $w \cdot h$ is also a prefix of W, then we must allow $w \cdot h$ to be moved from the inner elements of the weighted-W-branch to the frontier elements of the (induced) weighted-w-branch.*

To check whether a promise is coherent w.r.t. the weighted branch it belongs to, we verify that the encoded numbers of matches promised by χ on inner elements correspond to those that can be inferred from a combination of a partial match on $\mathbb{B}$ and the promised numbers of matches at further elements. To formalize this, we define a *split* of the query q, a notion inspired by the fork-rewriting approach (see e.g. Lutz [2008]).

Definition 49. *Let $p \subseteq q$ be a subquery of q and $\pi : p \to \mathbb{B}$ a match. We say π splits q over $(\mathbb{B}, \chi)$ iff:*

1. *For all v_1, v_2 such that $\pi(v_1), \pi(v_2) \in \Delta^{\mathbb{B}} \setminus \mathsf{Ind}$, if $v_1 \sim v_2$ then $\pi(v_1) = \pi(v_2)$;*
2. *For all v such that $\pi(v) \in \mathsf{front}(\mathbb{B})$, we have $\chi_{\pi(v)}(v) \neq 0$.*
3. *The following is a partition of* $\mathsf{terms}(q)$:

$$(\mathsf{terms}(p) \setminus \pi^{-1}(\mathsf{front}(\mathbb{B}))) \cup \{\mathsf{dep}(v) \mid \overline{v} \in \overline{\pi^{-1}(\mathsf{front}(\mathbb{B}))}\}$$

And we hence restrict ourselves to *valid* weighted branches.

Definition 50. *A weighted branch $(\mathbb{B}, \chi)$ is* valid *if the two following conditions are satisfied:*

1. *For all $w \in \Delta^{\mathbb{B}} \setminus \mathsf{Ind}$ and for all $v_1, v_2 \in \mathcal{V}_{\mathsf{anon}}$, if $v_1 \sim v_2$ then $\chi_w(v_1) = \chi_w(v_2)$;*
2. *For all $w \in \mathsf{inner}(\mathbb{B}) \setminus \mathsf{Ind}$ and all $v \in \mathsf{terms}(q)$, we have:*

$$\chi_w(v) = \sum_{\substack{\pi_{|q_v} \\ \pi \text{ splits } q \text{ over } \mathbb{B} \\ \pi(v) = w}} \prod_{\substack{\overline{v'} \\ v' \in \mathsf{terms}(q_v) \\ \delta_v(v') = 1}} \chi_{\pi(v')}(v')$$

	z_0	z_0'	z_1	z_1'	z_2	z_2'	z_3
$\chi_{\mathsf{a}\cdot\{\mathrm{R_1.A,R_3.A}\}}$	16	16	0	0	0	0	0
$\chi_{\mathsf{b}\cdot\{\mathrm{R_2.A,R_3.A}\}}$	0	0	8	8	2	1	1
$\chi_{\mathsf{b}\cdot\{\mathrm{R_2.A,R_3.A}\}\cdot\{\mathrm{R_2.A,R_3.A}\}}$	0	0	0	0	2	1	1
$\chi_{\mathsf{b}\cdot\{\mathrm{R_2.A,R_3.A}\}\cdot\{\mathrm{R_1.A}\}}$	0	0	0	0	2	1	1
$\chi_{\mathsf{b}\cdot\{\mathrm{R_2.A,R_3.A}\}\cdot\{\mathrm{R_1.A}\}\cdot\{\mathrm{R_1.A}\}}$	0	0	0	0	0	0	1
$\chi_{\mathsf{b}\cdot\{\mathrm{R_2.A,R_3.A}\}\cdot\{\mathrm{R_1.A}\}\cdot\{\mathrm{R_2.A,R_3.A}\}}$	0	0	0	0	0	0	1

Table 4.2: The promise $\chi^{(2)}$ for Example 19.

In that case, we define the number of matches $m_{\mathbb{B},\chi}$ encoded in $(\mathbb{B}, \chi)$ as:

$$m_{\mathbb{B},\chi} := \sum_{\pi \text{ splits } q \text{ over } \mathbb{B}} \prod_{\substack{\overline{v} \\ v \in \mathsf{terms}(q) \\ \pi(v) \in \mathsf{front}(\mathbb{B})}} \chi_{\pi(v)}(v)$$

Remark 24. *The product from Condition 2 is well defined as, despite the fact that we iterate over equivalence classes of terms of q (with relative depth from v equal to 1), the choice of witness doesn't matter due to Condition 1 and the first item of the definition of a split. Furthermore, notice that Condition 2 can be checked in polynomial space by enumerating the splits of q over $\mathbb{B}$ and incrementing a counter, despite the fact that there might exist exponentially many such splits.*

Example 19. *Consider the promise $\chi^{(2)}$ for the branch $\mathbb{B}^{(2)}$ (see Example 18 and Figure 4.8c) given in Table 4.2. It can be verified that $(\mathbb{B}^{(2)}, \chi^{(2)})$ is a valid weighted $\mathsf{b} \cdot \{\mathrm{R_2.A, R_3.A}\} \cdot \{\mathrm{R_1.A}\}$-branch. By restricting $\chi^{(2)}$ to the elements of $\mathbb{B}^{(1)}$, resp. $\mathbb{B}^{(0)}$, we obtain a promise $\chi^{(1)}$, resp. $\chi^{(0)}$, such that $(\mathbb{B}^{(1)}, \chi^{(1)})$ is a valid weighted $\mathsf{b} \cdot \{\mathrm{R_2.A, R_3.A}\}$-branch, resp. $(\mathbb{B}^{(0)}, \chi^{(0)})$ is a valid weighted ε-branch. The latter two are the induced $\mathsf{b} \cdot \{\mathrm{R_2.A, R_3.A}\}$-branch and the induced ε-branch of $(\mathbb{B}^{(2)}, \chi^{(2)})$. All these weighted branches have an encoded number of matches of 40. For example, one can consider the following split of q over $(\mathbb{B}^{(0)}, \chi^{(0)})$:*

$$z_0, z_0' \mapsto \mathsf{a} \cdot \{\mathrm{R_1.A, R_3.A}\}$$

whose contribution in the sum defining $m_{\mathbb{B}^{(0)},\chi^{(0)}}$ is $\chi^{(0)}_{\mathsf{a}\cdot\{\mathrm{R_1.A,R_3.A}\}}(z_0) = 16$ (recall $z_0 \sim z_0'$ hence we consider either $\chi^{(0)}_{\mathsf{a}\cdot\{\mathrm{R_1.A,R_3.A}\}}(z_0)$ or $\chi^{(0)}_{\mathsf{a}\cdot\{\mathrm{R_1.A,R_3.A}\}}(z_0')$, which are equal, but do not combine both). Another split of of q over $(\mathbb{B}^{(0)}, \chi^{(0)})$ is:

$$z_0, z_0', z_1, z_1' \mapsto \mathsf{b} \qquad z_2, z_2' \mapsto \mathsf{b} \cdot \{\mathrm{R_2.A, R_3.A}\}$$

whose contribution is $\chi^{(0)}_{\mathsf{b}\cdot\{\mathrm{R_2.A,R_3.A}\}}(z_2) \times \chi^{(0)}_{\mathsf{b}\cdot\{\mathrm{R_2.A,R_3.A}\}}(z_2') = 2 \times 1$ (recall $z_2 \not\sim z_2'$).

Data: $\mathcal{ALCH}$ KB $\mathcal{K} := (\mathcal{T}, \mathcal{A})$, exhaustive rooted CCQ q, integer m.
Result: May return yes iff $[m, +\infty]$ is not a certain answer of q w.r.t. $\mathcal{K}$.

```
(𝔹, χ) ← Guess a candidate weighted-ε-branch.
W_lim ← ε
if (𝔹, χ) is not a valid weighted ε-branch or m ≤ m_{𝔹,χ} then
  | return no
end
while there exists W ∈ front(𝔹) with |W| < |q| + 2 and W > W_lim do
  | W' ← smallest W ∈ front(𝔹) s.t. |W| < |q| + 2 and W > W_lim
  | (𝔹', χ') ← Guess a candidate weighted W'-branch.
  | w ← longest common prefix of W_lim and W'
  | (𝔹_1, χ_1) ← induced weighted w-branch of (𝔹, χ)
  | (𝔹_2, χ_2) ← induced weighted w-branch of (𝔹', χ')
  | if (𝔹', χ') is not a valid weighted W'-branch or (𝔹_1, χ_1) ≠ (𝔹_2, χ_2) then
  |   | return no
  | end
  | (𝔹, χ) ← (𝔹', χ')
  | W_lim ← W'
end
return yes
```

Algorithm 1: An algorithm for exhaustive rooted CCQ answering in $\mathcal{ALCH}$.

We assume fixed a depth-first ordering on the set of words $\mathcal{W} := \{\varepsilon\} \cup \{\mathsf{a} \cdot h_1 \cdots h_n \mid \mathsf{a} \in \mathsf{Ind}(\mathcal{A}), h_i \in 2^{\Omega}, n \geq 0\}$, where ε denotes the empty word, and consider the coNPSPACE procedure described by Algorithm 1. Observe the $\textsc{Sat}(\mathcal{L})$ oracle is needed to test whether the guessed branches are valid weighted branches.

We now prove the central lemma of this section, which concludes the proof of Theorem 26.

Lemma 25. *There exists a countermodel for m iff there exists an accepting computation for Algorithm 1.*

Proof. We prove the two directions in turn.

(⇒) If there exists a countermodel $\mathcal{I}$, then consider its $f^\star$-interlacing $\mathcal{I}^\star$. An accepting run can be obtained by extracting branches from $\mathcal{I}^\star$ and setting promises as follows:

$$\begin{array}{rcl} \chi_w : \mathsf{terms}(q) & \to & \{0, \dots M\} \\ t & \mapsto & \#\{\pi_{|q_t} \mid \pi : q \to \mathcal{I}^\star \text{ is a match s.t. } \pi(t) = w\} \end{array}$$

($\Leftarrow$) If there exists an accepting run, let us consider $w_0, \dots w_K$ all intermediate values of W_{lim} and $(\mathbb{B}_0, \chi_0), \dots (\mathbb{B}_K, \chi_{K-1})$ the corresponding values of $(\mathbb{B}, \chi)$ (in particular $w_0 := \varepsilon$). Let us consider the interpretation obtained as $\mathcal{I} := \bigcup_{k=0}^{K} \mathbb{B}_k$, along with the promise obtained as the union of all intermediate promises $\chi := (\bigcup_{k=0}^{K} (\chi_k)_w)_w$.

We prove that $\mathcal{I}$ can be completed into a countermodel for m. From the satisfiability checks performed at each step (when verifying that the guessed branches are indeed branches, see Condition 2 of Definition 46),it follows that one can extend $\mathcal{I}$ into a complete model $\mathcal{I}_{ext}$. Importantly, since branches require that no rule applies on inner elements and that each element from $\mathcal{I}$ at distance less than $|q|+1$ from the individuals is an inner element of some encountered branch, this extension $\mathcal{I}_{ext}$ can be obtained without introducing any element at distance less than $|q|+1$ from the individuals. To prove the extension $\mathcal{I}_{ext}$ has less than m matches, it hence suffices to prove that this holds for the interpretation $\mathcal{I}$. To do so, we prove the following property, henceforth referred to as Property ($\star$):

$$\forall w \in \Delta^{\mathcal{I}}\ \forall t \in \mathsf{terms}(q), \chi_w(t) = \#\{\pi_{|q_t} \mid \pi : q \to \mathcal{I} \text{ is a match s.t. } \pi(t) = w\}$$

To see how Property ($\star$) will conclude the proof, it suffices to notice that for all branches $\mathbb{B} \in \{\mathbb{B}_0, \dots \mathbb{B}_K\}$, the set of matches of q in $\mathcal{I}$ can be decomposed as follows:

$$\bigcup_{\substack{\pi \\ \pi \text{ splits } q \text{ over } \mathbb{B}}} \{\pi_{|\Delta^{\mathbb{B}} \setminus \mathsf{front}(\mathbb{B})}\} \times \prod_{\substack{\overline{v} \\ v \in \mathsf{terms}(q) \\ \pi(v) \in \mathsf{front}(\mathbb{B})}} \left\{ \rho_{|q_v} \,\middle|\, \begin{array}{l} \rho : q \to \mathcal{I} \text{ is a match} \\ \text{s.t. } \rho(v) = \pi(v) \end{array} \right\}$$

In particular for $\mathbb{B} = \mathbb{B}_0$ and if Property ($\star$) holds, this is exactly the comparison with m performed at the beginning of the algorithm.

We henceforth focus on proving Property ($\star$). We proceed by a descending induction on elements of $\mathcal{I}$, with respect to the depth-first order chosen on words $\mathcal{W}$. For an element w at depth $|q|+1$, the sum in the definition of χ_w must be empty since q is rooted, hence χ_w is the function always equal to 0. For the same reason of q being rooted, $\{\pi_{|q_t} \mid \pi : q \to \mathcal{I}$ is a match s.t. $\pi(t) = w\}$ must be empty, hence the desired equality. For the induction case, consider an element $w \in \Delta^{\mathcal{I}}$ at depth at most $|q|$ and assume Property ($\star$) holds for deeper elements. By construction of $\mathcal{I}$, there must exists k such that $W_k = w$ and consider then the corresponding weighted w-branch $(\mathbb{B}_k, \chi_k) \subseteq (\mathcal{I}, \chi)$. Fix a term $t \in \mathsf{terms}(q)$ and specialize the decomposition above to the branch $\mathbb{B}_k$ and to matches mapping t onto w. Since w is a inner element of $\mathbb{B}_k$, one simply obtains the following decomposition:

$$\bigcup_{\substack{\pi \\ \pi \text{ splits } q \text{ over } \mathbb{B}_k \\ \pi(t) = w}} \{\pi_{|\Delta^{\mathbb{B}_k} \setminus \mathsf{front}(\mathbb{B}_k)}\} \times \prod_{\substack{\overline{v} \\ v \in \mathsf{terms}(q) \\ \pi(v) \in \mathsf{front}(\mathbb{B}_k)}} \left\{ \rho_{|q_v} \,\middle|\, \begin{array}{l} \rho : q \to \mathcal{I} \text{ is a match} \\ \text{s.t. } \rho(v) = \pi(v) \end{array} \right\}$$

Consider now only the restriction of those matches to q_t, that is exactly the set appearing in the RHS of Property ($\star$), which hence decomposes as:

$$\bigcup_{\substack{\pi \\ \pi \text{ splits } q \text{ over } \mathbb{B}_k \\ \pi(t)=w}} \{(\pi_{|\Delta^{\mathbb{B}_k} \setminus \mathsf{front}(\mathbb{B}_k)})_{|q_t}\} \times \prod_{\substack{v \\ v \in \mathsf{terms}(q) \\ \pi(v) \in \mathsf{front}(\mathbb{B}_k)}} \left\{ (\rho_{|q_v})_{|q_t} \;\middle|\; \begin{array}{l} \rho : q \to \mathcal{I} \text{ is a match} \\ \text{s.t. } \rho(v) = \pi(v) \end{array} \right\}$$

Since $\pi(t) = w$ and that $w \notin \mathsf{front}(\mathbb{B}_k)$, the terms in the product are always empty unless $v \in \mathsf{dep}(t)$, and since frontier elements are exactly 1 step away from an inner element, we must have more precisely $\delta_t(v) = 1$. Hence the simplification:

$$\bigcup_{\substack{\pi \\ \pi \text{ splits } q \text{ over } \mathbb{B}_k \\ \pi(t)=w}} \{(\pi_{|\Delta^{\mathbb{B}_k} \setminus \mathsf{front}(\mathbb{B}_k)})_{|q_t}\} \times \prod_{\substack{v \\ v \in \mathsf{terms}(q_t) \\ \delta_t(v)=1}} \left\{ \rho_{|q_v} \;\middle|\; \begin{array}{l} \rho : q \to \mathcal{I} \text{ is a match} \\ \text{s.t. } \rho(v) = \pi(v) \end{array} \right\}$$

If we consider the above in terms of cardinality, we can apply the induction hypothesis to obtain the following:

$$\#\{\pi_{|q_t} \mid \pi : q \to \mathcal{I} \text{ is a match s.t. } \pi(t) = w\} = \sum_{\substack{\pi \\ \pi \text{ splits } q \text{ over } \mathbb{B}_k \\ \pi(t)=w}} \prod_{\substack{v \\ v \in \mathsf{terms}(q_t) \\ \delta_t(v)=1}} \chi_{\pi(v)}(v)$$

From the Condition 2 of $(\mathbb{B}_k, \chi_k)$ being a valid weighted-branch, we now obtain that the RHS of this equality is exactly $\chi_w(t)$, hence the Property ($\star$) for w. ☐

4.4.3 Matching lower bounds

In this section, we exhibit two lower bounds for exhaustive rooted CCQ answering in $\mathcal{ALCH}$ ontologies. As we have seen in Section 4.4.2, the complexity of our algorithm to answer such queries drops from EXP to PSPACE when moving from $\mathcal{ALC}$ to $\mathcal{ELH}_\bot$, as the corresponding satisfiability problem gets easier. Our first lower bound emphasizes that satisfiability is indeed the limiting factor by trivially reducing the EXP-complete satisfiability problem over $\mathcal{ALC}$ to ours.

Theorem 27. *Exhaustive rooted CCQ answering over $\mathcal{ALC}$ ontologies is* EXP*-hard w.r.t. combined complexity.*

Proof. We proceed by reduction from the EXP-complete concept satisfiability problem with an $\mathcal{ALC}$ TBox (see Schild [1991]). Let $\mathcal{T}$ be an $\mathcal{ALC}$ TBox and C be the concept of interest. We claim that: C is satisfiable w.r.t. $\mathcal{T}$ iff 2 is not a certain answer to $q := \mathrm{C}(\mathsf{a})$ over the KB $(\mathcal{T}, \{\mathrm{C}(\mathsf{a})\})$. If C is satisfiable, then there exists a model for $(\mathcal{T}, \{\mathrm{C}(\mathsf{a})\})$, which obviously yields a single match for q, hence it is a countermodel for 2. Conversely, if 2 is not a certain answer, then there exists a model of $(\mathcal{T}, \{\mathrm{C}(\mathsf{a})\})$, witnessing the satisfiability of C w.r.t. $\mathcal{T}$. ☐

For restrictions of $\mathcal{ELH}_\bot$, we obtain a PSPACE lower bound for $\mathcal{EL}$ by a more involved reduction from the Quantified Satisfiability (QBF) problem. Interestingly, this construction strongly relies on a binary encoding of the input integer. Whether the complexity drops if the integer is encoded in unary remains an open question.

Theorem 28. *Exhaustive rooted CCQ answering over $\mathcal{EL}$ ontologies is* PSPACE-*hard w.r.t. combined complexity.*

Proof. We reduce (the complement of) the Quantified Boolean Formula problem (QBF) known to be PSPACE-complete (see *e.g.* Sipser [1996]). QBF takes as input a Boolean formula with shape $\forall x_1 \exists y_1 \ldots \forall x_n \exists y_n \; \phi(x_1, y_1, \ldots, x_n, y_n)$, where ϕ is a quantifier-free Boolean formula, and decides if this formula is true.

Let $F := \forall x_1 \exists y_1 \ldots \forall x_n \exists y_n \; \phi(x_1, y_1, \ldots, x_n, y_n)$ be an instance of QBF. We build a knowledge base $\mathcal{K}$, an exhaustive rooted CCQ q and an integer N such that $N + 1$ is a certain answer to q over $\mathcal{K}$ iff F is false. We are interested in *optimal models*, that are models minimizing the number of matches for q.

Intuitively, we want each model $\mathcal{I}$ to generate *all* possible assignments for the universally quantified variables, which is represented as a tree-shaped structure in $\mathcal{I}$, and pick *along each branch* one assignment for the existentially quantified variables. At each leaf the formula ϕ is evaluated and a part q_ϕ of the query q detects if this returns False. If the whole formula F is true, there exists a model without such a match for q_ϕ and it gives a countermodel for $N + 1$. Otherwise each model must contain this additional match and $N + 1$ is a certain answer.

The main challenge is to ensure assignments represented at each node in the tree uses only two Boolean values, True or False. With exhaustive rooted queries we cannot simply count the number of elements used as Boolean in our models and set $N = 2$ so that optimal models only use two Booleans. Indeed, to detect such a Boolean-like element, the query q has to keep track of the whole path between its root and the element of interest. Since we want our tree to contain 2^n leaves, that is, a total of $2^{n+1} - 1$ nodes (including the root), we should consider $2 \times (2^{n+1} - 2)$ Boolean-like elements (2 per anonymous node). In particular, it means that we want optimal models to contains at least that many matches coming from their tree-shaped structure, and therefore, if this structure collapses in a model, then it must come at the cost of at least that many matches (so that we can focus on models in which the tree-shaped structure does not collapse).

This tree-shaped structure is rooted on an individual, namely a and further generated with the TBox and the use of *a single role*, namely "next" ($\star$). Since we need a single-exponential number of nodes, each node of interest can be generated within polynomial depth n, which can be reached by our rooted query ($\dagger$). To prevent an optimal model to collapse the tree-shaped structure, we take advantage of ($\star$) and ($\dagger$) which ensures the number of shapes of cycles to exclude is polynomial. Each such shape is excluded from optimal model with the help of a subquery q'

which itself contains a "loading subquery", namely q_{load}. This loading subquery satisfies the property that if q' maps in the tree-shaped structure of $\mathcal{I}$, that is, if q' detects the shape of cycle in this structure, then it allows q_{load} to map in 2^M different manners in the ABox, with M big enough to make $\mathcal{I}$ non-optimal.

Knowledge Base. We first generate the exponentially large tree. The branching containing concept name $\mathrm{Node}_\mathrm{i}^\mathrm{t}$, resp. $\mathrm{Node}_\mathrm{i}^\mathrm{f}$, represents an assignment in which the universally quantified variable x_i should be true, resp. false:

$$\mathrm{Node_0}(\mathsf{a}) \qquad \begin{array}{ll} \mathrm{Node_i} \sqsubseteq \exists \mathrm{next}.\mathrm{Node}^\mathrm{t}_\mathrm{i+1} & \mathrm{Node}^\mathrm{t}_\mathrm{i+1} \sqsubseteq \mathrm{Node_{i+1}} \\ \mathrm{Node_i} \sqsubseteq \exists \mathrm{next}.\mathrm{Node}^\mathrm{f}_\mathrm{i+1} & \mathrm{Node}^\mathrm{f}_\mathrm{i+1} \sqsubseteq \mathrm{Node_{i+1}} \end{array} \qquad (0 \leq i \leq n-1)$$

We now require that a node at depth i assigns a valuation to each variable x_j and y_j with $j \leq i$, and that it provides at least the two usual Boolean values represented by concepts True and False:

$$\begin{array}{cc} \mathrm{Node_i} \sqsubseteq \exists \mathrm{toBool}.\mathrm{True} & \mathrm{Node_i} \sqsubseteq \exists \mathrm{toBool}.\mathrm{EVar_j} \\ \mathrm{Node_i} \sqsubseteq \exists \mathrm{toBool}.\mathrm{False} & \mathrm{Node_i} \sqsubseteq \exists \mathrm{toBool}.\mathrm{UVar_j} \end{array} \qquad (1 \leq j \leq i \leq n)$$

We now evaluate ϕ inductively at each node. The concept IsTrue_ψ, resp. $\mathrm{IsFalse}_\psi$, indicates that the subformula ψ occurring in ϕ evaluates to True, resp. False. We start with the case of a single variable:

$$\begin{array}{cc} \exists \mathrm{tobool}.(\mathrm{True} \sqcap \mathrm{EVar_i}) \sqsubseteq \mathrm{IsTrue_{x_i}} & \exists \mathrm{tobool}.(\mathrm{True} \sqcap \mathrm{UVar_i}) \sqsubseteq \mathrm{IsTrue_{y_i}} \\ \exists \mathrm{tobool}.(\mathrm{False} \sqcap \mathrm{EVar_i}) \sqsubseteq \mathrm{IsFalse_{x_i}} & \exists \mathrm{tobool}.(\mathrm{False} \sqcap \mathrm{UVar_i}) \sqsubseteq \mathrm{IsFalse_{y_i}} \end{array}$$

Case of a conjunction:

$$\begin{array}{r} \mathrm{IsTrue}_{\psi_1} \sqcap \mathrm{IsTrue}_{\psi_2} \sqsubseteq \mathrm{IsTrue}_\psi \\ \mathrm{IsFalse}_{\psi_1} \sqsubseteq \mathrm{IsFalse}_\psi \\ \mathrm{IsFalse}_{\psi_2} \sqsubseteq \mathrm{IsFalse}_\psi \end{array} \qquad (\psi = \psi_1 \wedge \psi_2 \text{ occurs in } \phi)$$

Case of a disjunction:

$$\begin{array}{r} \mathrm{IsTrue}_{\psi_1} \sqsubseteq \mathrm{IsTrue}_\psi \\ \mathrm{IsTrue}_{\psi_2} \sqsubseteq \mathrm{IsTrue}_\psi \\ \mathrm{IsFalse}_{\psi_1} \sqcap \mathrm{IsFalse}_{\psi_2} \sqsubseteq \mathrm{IsFalse}_\psi \end{array} \qquad (\psi = \psi_1 \vee \psi_2 \text{ occurs in } \phi)$$

Case of a negation:

$$\begin{array}{r} \mathrm{IsFalse}_{\psi'} \sqsubseteq \mathrm{IsTrue}_\psi \\ \mathrm{IsTrue}_{\psi_2} \sqsubseteq \mathrm{IsFalse}_\psi \end{array} \qquad (\psi = \neg\psi' \text{ occurs in } \phi)$$

The preceding axioms together ensure that each leaf, that is, a node satisfying $\mathrm{Node_n}$, satisfies IsTrue_ϕ or $\mathrm{IsFalse}_\phi$. Note that it is possible that internal nodes, satisfying $\mathrm{Node_i}$ for some $i < n$, can already satisfy these two concepts. This won't interfere with the number of matches for the query as the corresponding subquery will ask for a path of length exactly n from the root.

It remains to introduce the root that will be used in the query and the two elements allowing for the loading subquery to increase drastically the number of matches:

$$\mathrm{toStart}(\mathsf{r}, \mathsf{a}) \qquad \mathrm{toLoader}(\mathsf{a}, \mathsf{l_0}) \qquad \mathrm{toLoader}(\mathsf{a}, \mathsf{l_1})$$

To conclude the construction of the KB, we introduce an auxiliary individual b whose purpose is to let each subquery map at least once. In particular, it satisfies *all concepts names previously mentioned*, which we don't recap here, and the following role assertions:

$$\mathrm{toStart}(\mathsf{r}, \mathsf{b}) \qquad \mathrm{toLoader}(\mathsf{b}, \mathsf{b}) \qquad \mathrm{next}(\mathsf{b}, \mathsf{b}) \qquad \mathrm{toBool}(\mathsf{b}, \mathsf{b})$$

Query. We distinguish two main kinds of subqueries: structural subqueries and consistency subqueries. Structural subqueries ensure each model contains the desired tree-shape structured or yields too many matches to be optimal. Consistency subqueries ensure models in the first case either represent proper assignments for variables or yields at least one additional match.

We begin with the loading subquery q_{load}, which contains a free variable z so that copies of q_{load} will be instantiated for building the other structural subqueries.

$$q_{\mathrm{load}}(z) := \exists z_1, \ldots, z_M \ \mathrm{toStart}(\mathsf{r}, z) \wedge \bigwedge_{i=1}^{M} \mathrm{toLoader}(z, z_i)$$

Notice that if z is mapped onto b, then there is in general only one way to map the remaining variables from q_{load} (all onto b as well). On the other hand, if z is mapped on a, then there are at least 2^M ways to map these remaining variables (each variable can be mapped onto either $\mathsf{l_0}$ or $\mathsf{l_1}$). The exact value of M will be specified later in the construction.

To ensure the desired tree-shaped structure, we first require the branching (leading to either $\mathrm{Node_i^t}$ or $\mathrm{Node_i^f}$) to be indeed branching, meaning we don't want these two concepts being witnessed by the same element. We proceed with the following subqueries, each detecting a non-branching node at depth d, with $1 \leq d \leq n$:

$$q^d_{\mathrm{branch}} := \exists z_0, \ldots, z_d \ q_{\mathrm{load}}(z_0) \wedge \bigwedge_{i=0}^{d-1} \mathrm{next}(z_i, z_{i+1}) \wedge \mathrm{Node_d^t}(z_d) \wedge \mathrm{Node_d^f}(z_d)$$

We proceed as well with the True and False branching at each node, with the following subqueries, each detecting collapsed Booleans coming from a node at depth d, with $1 \leq d \leq n$:

$$q^d_{\text{bool}} := \exists z_0, \ldots, z_d, z\; q_{\text{load}}(z_0) \wedge \bigwedge_{i=0}^{d-1} \text{next}(z_i, z_{i+1}) \wedge \text{toBool}(z_d, z) \wedge \text{True}(z) \wedge \text{False}(z)$$

We further detect if a branch loops back on itself. We proceed with the following subqueries, each detecting a branch whose d^{th} and $(d+p+1)^{\text{th}}$ nodes are merged, with $0 \leq d \leq n-1$ and $0 \leq p \leq n-1-d$:

$$q^d_{\text{loop}} := \exists z_0, \ldots, z_{d+p}\; q_{\text{load}}(z_0) \wedge \bigwedge_{i=0}^{d+p-1} \text{next}(z_i, z_{i+1}) \wedge \text{next}(z_{d+p}, z_d)$$

To ensure a tree-shaped structure, it remains to detect two branches separating at depth d and collapsing further together, the first after p additional nodes and the second after q additional nodes, with $0 \leq d \leq n-1$ and $0 \leq p, q \leq n-1-d$:

$$\begin{aligned} q^{d,p,q}_{\text{cycle}} \;:=\; & \exists z_0, \ldots, z_d, z^t_0, \ldots, z^t_p, z^f_0, \ldots, z^f_q, z\; q_{\text{load}}(z_0) \wedge \textstyle\bigwedge_{i=0}^{d-1} \text{next}(z_i, z_{i+1}) \\ & \wedge \text{next}(z_d, z^t_0) \wedge \text{Node}^{\text{t}}_{\text{d+1}}(z^t_0) \wedge \textstyle\bigwedge_{i=0}^{p-1} \text{next}(z^t_i, z^t_{i+1}) \wedge \text{next}(z^t_p, z) \\ & \wedge \text{next}(z_d, z^f_0) \wedge \text{Node}^{\text{f}}_{\text{d+1}}(z^f_0) \wedge \textstyle\bigwedge_{i=0}^{q-1} \text{next}(z^f_i, z^f_{i+1}) \wedge \text{next}(z^f_q, z) \end{aligned}$$

We now move to the consistency subqueries. Our first consistency subquery keeps track of the elements used as Boolean values. We count elements used as Boolean values coming from a node at depth d, with $1 \leq d \leq n$:

$$q^d_{\#\text{ bool}} := \exists z_0, \ldots, z_d, z\; \text{toStart}(\mathsf{r}, z_0) \wedge \bigwedge_{i=0}^{d-1} \text{next}(z_i, z_{i+1}) \wedge \text{toBool}(z_d, z)$$

Notice that due to the subqueries q^d_{bool}, we know that there should be at least 2 such elements for each node at depth d in the tree-shaped structure. Therefore, in a model exhibiting the desired tree-shaped structure, $q^d_{\#\text{ bool}}$ yields at least 2^{d+1} matches plus 1 on b. For the combination of all $q^d_{\#\text{ bool}}$ for $1 \leq d \leq n$, one shall hence expect $\prod_{d=1}^{n}(2^{d+1}-1)$ matches, which is essentially $2^{\mathcal{O}(n^2)}$. Since n is given in unary (it is the essentially the number of variables in F), we can find an integer M with a polynomially large binary encoding and such that $2^M > N$. This is how we set M in the subquery q_{load}.

We now detect if the assignment for an existentially quantified variable y_d (hence chosen for some node at depth d), here True, isn't preserved at some further node at depth $d+k$, with $1 \leq d \leq n-1$ and $1 \leq k \leq n-d$:

$$\begin{aligned} q^{y_d,\text{True},k}_{\text{assign}} \;:=\; & \exists z_0, \ldots, z_d, z_{d+k}, z, z'\; \text{toStart}(\mathsf{r}, z_0) \wedge \textstyle\bigwedge_{i=0}^{d+k-1} \text{next}(z_i, z_{i+1}) \\ & \wedge \text{toBool}(z_d, z) \wedge \text{EVar}_{\text{k}}(z) \wedge \text{True}(z) \\ & \wedge \text{toBool}(z_{d+k}, z') \wedge \text{EVar}_{\text{k}}(z') \wedge \text{False}(z') \end{aligned}$$

We do the same in case the chosen assignment is False:

$$\begin{aligned} q_{\text{assign}}^{y_d,\text{False},k} \quad := \quad & \exists z_0, \ldots, z_d, z_{d+k}, z, z' \ \text{toStart}(\mathsf{r}, z_0) \wedge \textstyle\bigwedge_{i=0}^{d+k-1} \text{next}(z_i, z_{i+1}) \\ & \wedge \text{toBool}(z_d, z) \wedge \text{EVar}_\text{k}(z) \wedge \text{False}(z) \\ & \wedge \text{toBool}(z_{d+k}, z') \wedge \text{EVar}_\text{k}(z') \wedge \text{True}(z') \end{aligned}$$

We proceed as well with universally quantified variable x_d, whose valuation, here True, should be decided by the branching following a node at depth $d-1$ and then be preserved at further nodes, here at depth $d+k$, with $1 \leq d \leq n$ and $0 \leq k \leq n-d$:

$$\begin{aligned} q_{\text{assign}}^{x_d,\text{True},k} \quad := \quad & \exists z_0, \ldots, z_d, z_{d+k}, z \ \text{toStart}(\mathsf{r}, z_0) \wedge \textstyle\bigwedge_{i=0}^{d+k-1} \text{next}(z_i, z_{i+1}) \\ & \wedge \text{Node}_\text{d}^\text{t}(z_d) \wedge \text{toBool}(z_{d+k}, z) \wedge \text{UVar}_\text{k}(z) \wedge \text{False}(z) \end{aligned}$$

We do the same if the chosen assignment is False:

$$\begin{aligned} q_{\text{assign}}^{x_d,\text{False},k} \quad := \quad & \exists z_0, \ldots, z_d, z_{d+k}, z \ \text{toStart}(\mathsf{r}, z_0) \wedge \textstyle\bigwedge_{i=0}^{d+k-1} \text{next}(z_i, z_{i+1}) \\ & \wedge \text{Node}_\text{d}^\text{f}(z_d) \wedge \text{toBool}(z_{d+k}, z) \wedge \text{UVar}_\text{k}(z) \wedge \text{True}(z) \end{aligned}$$

Finally, we detect if an assignment at a leaf evaluates the formula to false:

$$q_\phi := \exists z_0, \ldots, z_n, z \ \text{toStart}(\mathsf{r}, z_0) \wedge \bigwedge_{i=0}^{n-1} \text{next}(z_i, z_{i+1}) \wedge \text{IsFalse}_\phi(z_n)$$

We now set q to be the conjunction of all the above subqueries and complete the reduction by proving the following claim:

$$F \text{ is false iff } N+1 \text{ is a certain answer to } q \text{ over } \mathcal{K}.$$

(⇒). Assume F is false. Consider a model $\mathcal{I}$ of $\mathcal{K}$, and let f be a homomorphism from the canonical model $\mathcal{C}_\mathcal{K}$ of $\mathcal{K}$ to $\mathcal{I}$. We focus on how the tree-shaped structure consisting of elements of $\Delta^{\mathcal{C}_\mathcal{K}}$ with prefix a embeds into $\mathcal{I}$. If an element $\mathsf{a} \cdot w \in \Delta^{\mathcal{C}_\mathcal{K}}$ is such that $f(\mathsf{a} \cdot w) \in (\text{Node}_\text{d}^\text{t} \cap \text{Node}_\text{d}^\text{f})^\mathcal{I}$, then it yields 2^M matches for the subquery $q_{\text{branch}}^{|w|}$. The other subqueries can independently collapse on b, ensuring the whole q admits at least 2^M matches. Since we set M and N such that $2^M \geq N$, we obtain at least the desired $N+1$ matches in $\mathcal{I}$. Similarly, one can eliminate models containing loops (2^M matches for a subquery q_{loop}^d), cycles (2^M matches for a subquery $q_{\text{cycle}}^{d,p,q}$), or elements representing both true and false (2^M matches for q_{bool}^d).

In the remaining models of $\mathcal{K}$, there are at least N matches for $q_{\#\text{bool}}$ as explained when setting the correct value for M. In a model $\mathcal{I}$, if an element $f(\mathsf{a} \cdot w)$ does not reuse its two already counted True and False Booleans, that is there exists some concept EVar_k (or UVar_k) with $k \leq |w|$ such that $\mathsf{a} \cdot w \cdot \text{toBool}.\{\text{EVar}_\text{k}\} \in \Delta^{\mathcal{C}_\mathcal{K}}$

but $f(\mathsf{a} \cdot w \cdot \text{toBool}.\{\text{EVar}_k\}) \notin \{f(\mathsf{a} \cdot w \cdot \text{toBool}.\{\text{True}\}), f(\mathsf{a} \cdot w \cdot \text{toBool}.\{\text{False}\})\}$, then it yields a new match for $q^{|w|}_{\#\text{bool}}$ and we are done. Otherwise, each such node $f(\mathsf{a} \cdot w)$ is connected to an element satisfying $\text{EVar}_k \sqcap \text{True}$ or to an element satisfying $\text{EVar}_k \sqcap \text{False}$, for each $k \leq |w|$. This defines a valuation at this node for the $|w|$ first existential variables $y_1, \dots y_d$ which we denote $\tau_w : \{y_1, \dots y_d\} \to \{\text{True}, \text{False}\}$. We proceed as well with the $|w|$ first universal variables to extend τ_w as $\tau_w : \{x_1, y_1 \dots x_d, y_d\} \to \{\text{True}, \text{False}\}$.

If $w, w' \in \Delta^{\mathcal{C}_\mathcal{K}}$ with w a prefix of w' but that $\tau_e \subsetneq \tau_{e'}$, then it yields an additional match for $q^{v_k,\text{True},p}_{\text{assign}}$ or for $q^{v_k,\text{False},p}_{\text{assign}}$, where v_k is the existential or universal variable on which τ_w and $\tau_{w'}$ disagree, and where p is the integer such that $|w'| = |w| + p$.

Recall now that F is false, hence there must exists a valuation, say True, of x_1 such that whatever the valuation of y_1 the remaining F_1 of the formula is false. Consider hence the element $f(\mathsf{a} \cdot \text{next}.\{\text{Node}^{\text{t}}_1, \text{Node}_1\})$ in $\mathcal{I}$, and remark we can assume $\tau_{\mathsf{a}\cdot\text{next}.\{\text{Node}^{\text{t}}_1,\text{Node}_1\}}(x_1) = \text{True}$ (otherwise it would yield an additional match for $q^{x_1,\text{True},0}_{\text{assign}}$) and that this valuation and the one for y_1 is now fixed for further elements. But since F_1 is false, we can iterate until we reach depth n, for which we have an element $\mathsf{a} \cdot w \in \Delta^{\mathcal{C}_\mathcal{K}}$ with $|w| = d$ whose valuation τ_w is such that F_n, that is ϕ, must be false. Therefore the element $f(\mathsf{a} \cdot w)$ yields a new match for q_ϕ and we are done.

($\Leftarrow$). Assume F is true. We briefly explain how to obtain a model with exactly N matches from the canonical model $\mathcal{C}_\mathcal{K}$ of $\mathcal{K}$. Since F is true, for each case $x_1 = 0$ or $x_1 = 1$, there exists a valuation of y_1 such that the remainder of the formula is true. Say that for $x_1 = 1$ we need to set $y_1 = 0$. Then we consider $\mathcal{C}_\mathcal{K}$ in which we identify all elements with shape $\mathsf{a} \cdot \text{next}.\{\text{Node}^{\text{t}}_1, \text{Node}_1\} \cdot w \cdot \text{toBool}.\{\text{UVar}_1\}$ with the element $\mathsf{a} \cdot \text{next}.\{\text{Node}^{\text{t}}_1, \text{Node}_1\} \cdot w \cdot \text{toBool}.\{\text{True}\}$ and all elements with shape $\mathsf{a}\cdot\text{next}.\{\text{Node}^{\text{t}}_1, \text{Node}_1\}\cdot w\cdot\text{toBool}.\{\text{EVar}_1\}$ with the element $\mathsf{a}\cdot\text{next}.\{\text{Node}^{\text{t}}_1, \text{Node}_1\}\cdot w \cdot \text{toBool}.\{\text{False}\}$. We proceed similarly on the side of $x_1 = 0$, according to the required valuation of y_1, and further iterate this construction.

From the tree-shaped structure of the canonical model, even with our slight modifications, it can be verified that no structural subquery can map in the structure issuing from a. The consistency queries $q^d_{\#bool}$ indeed yield N matches from the remaining Booleans in this structure and from their match on b, while other consistency subqueries can only map onto b. In particular, since F is true, such a construction does not trigger the concept IsFalse_ϕ on elements $\mathsf{a} \cdot w$ with $|w| = n$, hence q_ϕ can only map onto b. □

We also close the case of data complexity for $\mathcal{EL}$ with the following theorem.

Theorem 29. *Exhaustive rooted CCQ answering over $\mathcal{EL}$ is* coNP*-complete w.r.t. data complexity.*

Proof. The main idea is the same as in proof of Theorem 21. However, due to the lack of existential variables, we can no longer 'reach' the colors without taking into account the paths leading to them. To address this difficulty, we translate into our context an idea from Kostylev and Reutter [2015], which takes advantage of role inclusions.

Starting from an instance $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ of the decision problem 3-Col, we consider the ABox $\mathcal{A}_\mathcal{G}$ given by:

$$\begin{aligned}\mathcal{A}_\mathcal{G} = &\{\text{toVertex}(\mathsf{a}, \mathsf{u}) \mid u \in \mathcal{V}\} \cup \{\text{Vertex}(\mathsf{u}) \mid u \in \mathcal{V}\} \\ &\cup \{\text{Edge}(\mathsf{u_1}, \mathsf{u_2}) \mid (u_1, u_2) \in \mathcal{E}\} \\ &\cup \{\text{toVertex}(\mathsf{a}, \mathsf{a_v}), \text{Colors}(\mathsf{a_v}, \mathsf{a_v}), \text{Monochrom}(\mathsf{a_v})\} \\ &\cup \bigcup_{u \in \mathcal{V}} \{\text{Colors}(\mathsf{u}, \mathsf{r_u}), \text{R}(\mathsf{r_u}), \text{Colors}(\mathsf{u}, \mathsf{g_u}), \text{G}(\mathsf{g_u}), \text{Colors}(\mathsf{u}, \mathsf{b_u}), \text{B}(\mathsf{b_u})\}\end{aligned}$$

and the TBox $\mathcal{T}$ containing axiom Vertex $\sqsubseteq$ $\exists$Colors.Used, and each following axiom for C $\in$ {R, G, B}:

$$\exists\text{Colors}.(\text{C} \sqcap \text{Used}) \sqcap \exists\text{Edge}.(\exists\text{Colors}.(\text{C} \sqcap \text{Used})) \sqsubseteq \text{Monochrom}.$$

and we denote by $\mathcal{K}_\mathcal{G} = (\mathcal{T}, \mathcal{A}_\mathcal{G})$ the resulting KB. A part of the canonical model of $\mathcal{K}_\mathcal{G}$ is depicted in Figure 4.9. We use Vertex $\sqsubseteq$ $\exists$Colors.Used to assign colors to vertices.

We consider the two following exhaustive rooted CCQs:

$$\begin{aligned} q^{edge} &= \exists z_m \ \text{toVertex}(\mathsf{a}, z_m) \wedge \text{Monochrom}(z_m) \\ q^{col} &= \exists z_v \ \exists z \ \text{toVertex}(\mathsf{a}, z_v) \wedge \text{Colors}(z_v, z)\end{aligned}$$

and let q be the query obtained by taking the conjunction of these two queries and keeping all of the variables existentially quantified. The query q is displayed in Figure 4.10. Observe that compared to the query from the proof of Theorem 21 (see Figure 4.2), the part of the query detectecing monochromatic edges has been internalized into the TBox $\mathcal{T}$.

It is not hard to see that $[3|\mathcal{V}| + 1, +\infty]$ is a certain answer to q over $\mathcal{K}_\mathcal{G}$. Indeed, there are at least $3|\mathcal{V}|$ matches of q in any model $\mathcal{I}$ of $\mathcal{KG}$, obtained as follows:

$$z_m \mapsto \mathsf{a_v} \qquad z_v \mapsto \mathsf{u} \qquad z \mapsto \mathsf{r_u} \mid \mathsf{g_u} \mid \mathsf{b_u} \qquad (u \in \mathcal{V})$$

and one additional match given by:

$$z_m, z_v, z \mapsto \mathsf{a_v}$$

To complete the proof, we establish the following claim:

$$[3|\mathcal{V}| + 2, +\infty] \text{ is a certain answer to } q \text{ over } \mathcal{K}_\mathcal{G} \text{ iif } \mathcal{G} \notin \text{3-Col}.$$

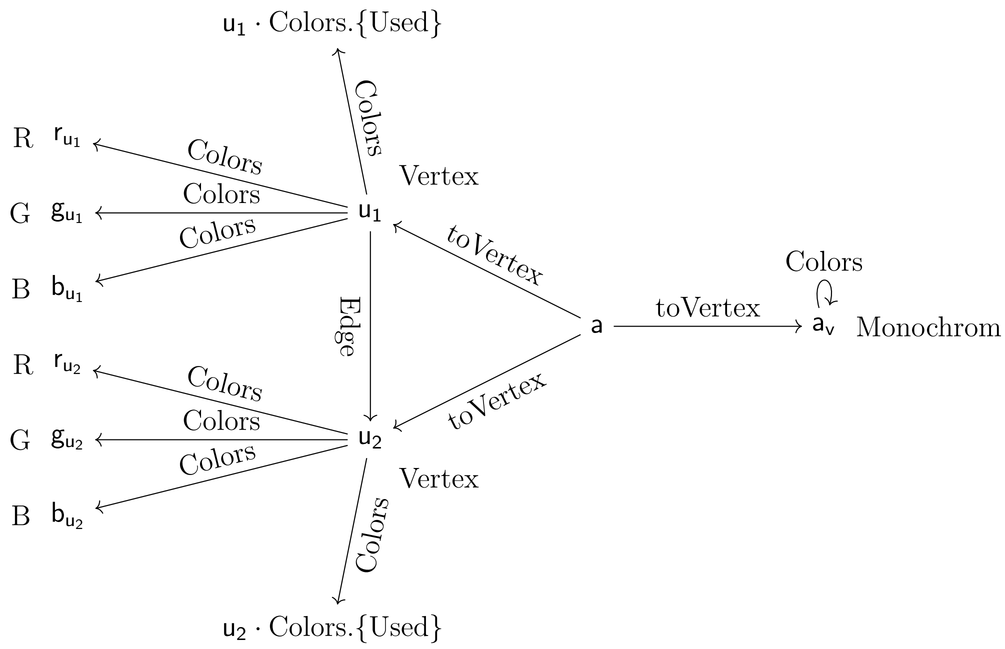


Figure 4.9: A part of $\mathcal{C}_{\mathcal{K}_\mathcal{G}}$ with $(u_1, u_2) \in \mathcal{E}$.

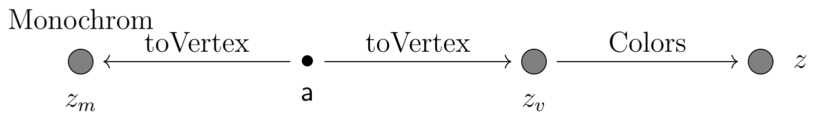


Figure 4.10: The exhaustive rooted CCQ q, which is the conjunction of q^{edge} (left part) and q^{col} (right part).

($\Rightarrow$) This direction is proven in the same manner as the claim in the proof of Theorem 21. We assume $[3|\mathcal{V}| + 2, +\infty]$ is a certain answer and take a possible coloring $\tau : \mathcal{V} \to \{\mathsf{r}, \mathsf{g}, \mathsf{b}\}$. We then use τ to build a model $\mathcal{I}_\tau$ of $\mathcal{K}_\mathcal{G}$ and use the existence of an additional match π to show that τ contains a monochromatic edge (hence $\mathcal{G} \notin$ 3-Col).

($\Leftarrow$) Assume $\mathcal{G} \notin$ 3-Col, and take some model $\mathcal{I}$ of $\mathcal{K}_\mathcal{G}$. There is a homomorphism $f : \mathcal{C}_{\mathcal{K}_\mathcal{G}} \to \mathcal{I}$. Define $\tau : \mathcal{V} \to \Delta^\mathcal{I}$ as follows: $\tau(u) = f(\mathsf{u} \cdot \text{Colors}.\{\text{Used}\})$. Note that τ is well defined, as the inclusion Vertex $\sqsubseteq \exists$Colors.Used ensures that there is an element $\mathsf{u} \cdot \text{Colors}.\{\text{Used}\}$ in $\mathcal{C}_{\mathcal{K}_\mathcal{G}}$. There are two cases to consider:

- If there exists $u \in \mathcal{V}$ such that $\tau(u) \notin \{\mathsf{r_u}, \mathsf{g_u}, \mathsf{b_u}\}$, then it provides an additional match of q^{color} in $\mathcal{I}$ with $z \mapsto \tau(u)$ and $z_v \mapsto \mathsf{u}^\mathcal{I}$.
- Else, since $\mathcal{G} \notin$ 3-Col, there exists an edge $(u_1, u_2) \in \mathcal{E}$ such that $\tau(u_1) = \mathsf{c}_{\mathsf{u}_1}$

and $\tau(u_2) = \mathsf{c}_{\mathsf{u}_2}$ for some $c \in \{\mathsf{r}, \mathsf{g}, \mathsf{b}\}$. The corresponding axiom

$$\exists\text{Colors}.(\text{C} \sqcap \text{Used}) \sqcap \exists\text{Edge}.(\exists\text{Colors}.(\text{C} \sqcap \text{Used})) \sqsubseteq \text{Monochrom}$$

hence triggers and yields a new match for q^{edge} given by:

$$z_m \mapsto \mathsf{u}_1$$

In both cases, there is an additional c-match for q. We thus obtain that $[3|\mathcal{V}| + 2, +\infty]$ is certain answer to q over $\mathcal{K}_\mathcal{G}$. □

We can adapt the previous reduction to prove DP-hardness w.r.t. data complexity of the corresponding problem of tight exhaustive rooted CCQ answering.

Theorem 30. *Tight exhaustive rooted CCQ answering over $\mathcal{EL}$ ontologies is* coNP*-complete w.r.t. data complexity.*

Proof. We give a reduction from the following problem (DP-complete due to Garey et al. [1976]): given *planar* graphs $\mathcal{G}_1 := (\mathcal{V}_1, \mathcal{E}_1)$ and $\mathcal{G}_2 := (\mathcal{V}_2, \mathcal{E}_2)$, decide if $\mathcal{G}_1 \in$ 3-Col and $\mathcal{G}_2 \notin$ 3-Col. The proof proceeds w.r.t. the proof of Theorem 29 exactly as the proof of Theorem 22 proceeds w.r.t. the proof of Theorem 21, that is by introducing an asymmetry in the query on the $\mathcal{G}_1$-side. The only difference is that the basic number of matches on the side of $\mathcal{G}_1$ is $(3|\mathcal{V}_1| + 1) \times (3|\mathcal{V}_1| + 1)$ instead of 3×3, and similarly for $\mathcal{G}_2$, it is $3|\mathcal{V}_2| + 1$ instead of 3. This slightly modifies the case analysis as follows:

	$\mathcal{G}_1 \in$ 3-Col	$\mathcal{G}_1 \notin$ 3-Col
$\mathcal{G}_2 \in$ 3-Col	$(3\|\mathcal{V}_1\| + 1)^2 \times (3\|\mathcal{V}_2\| + 1)$	$(3\|\mathcal{V}_1\| + 2)^2 \times (3\|\mathcal{V}_2\| + 1)$
$\mathcal{G}_2 \notin$ 3-Col	$(3\|\mathcal{V}_1\| + 1)^2 \times (3\|\mathcal{V}_2\| + 2)$	$(3\|\mathcal{V}_1\| + 2)^2 \times (3\|\mathcal{V}_2\| + 2)$

One can easily verify these 4 numbers are always distinct, hence the claim becomes:

$[(3|\mathcal{V}_1| + 1)^2 \times (3|\mathcal{V}_2| + 2), +\infty]$ is the tightest certain answer
iff
$\mathcal{G}_1 \in$ 3-Col and $\mathcal{G}_2 \notin$ 3-Col. □

4.5 Refinements within DL-Lite

In the previous section, we investigated how the absence of inverse roles allows us to lower the combined complexity of answering exhaustive rooted CCQs. We now turn to DL-Lite$^{\mathcal{H}}_{\mathsf{core}}$ KBs, for which we already know that inverse roles coupled with role inclusions, a combination allowed in DL-Lite$^{\mathcal{H}}_{\mathsf{pos}}$, leads to a coNEXP-complete problem (see Theorems 23 and 24).

In Section 4.5.1, we close the case of DL-Lite$^{\mathcal{H}}_{\mathsf{pos}}$ by showing coNP-hardness in data complexity, hence matching the upper bound from the general case (Theorem 6). We further move to DL-Lite$_{\mathsf{core}}$ and exhibit the central property of this section, namely that the canonical model yields the optimal number of matches for exhaustive rooted CCQs over DL-Lite$_{\mathsf{core}}$ KBs. This property has been independently used in Calvanese et al. [2020a], from which an L upper bound is derived. A similar property in the bag semantics counterpart of DL-Lite$_{\mathsf{core}}$ has been exploited in Nikolaou et al. [2019] for CQ entailment, though reducing one setting to the other seems non-trivial (see Example 1 in Calvanese et al. [2020a]). In Sections 4.5.2 and 4.5.3, we explore the consequence of this property on combined, resp. data, complexity for answering exhaustive rooted CCQs in DL-Lite$_{\mathsf{core}}$. More precisely, we prove that the problem becomes PP-complete for combined complexity, hence in PSPACE, while it becomes TC^0 for data complexity.

Let us also recall that for every DL-Lite$^{\mathcal{H}}_{\mathsf{core}}$ KB $\mathcal{K}$, it is well known the set of concept names M occurring in an element $w \cdot \mathrm{R.M} \in \Delta^{\mathcal{C}_\mathcal{K}}$ of the canonical model of $\mathcal{K}$ contains exactly those concept names entailed by the concept $\exists \mathrm{R}^-$ [Calvanese et al., 2007b]. We will hence omit such sets of concept names M within this section.

4.5.1 From DL-Lite$^{\mathcal{H}}_{\mathsf{core}}$ to DL-Lite$_{\mathsf{core}}$

We begin by closing the case of DL-Lite$^{\mathcal{H}}_{\mathsf{pos}}$, hence also of DL-Lite$^{\mathcal{H}}_{\mathsf{core}}$, by proving that exhaustive rooted CCQ answering over such KBs is coNP-hard w.r.t. data complexity, hence coNP-complete from Theorem 7. This is shown by another reduction from 3-Col which involves ideas from our proof of Theorem 21 and the proof of Lemma 16 from Kostylev and Reutter [2015].

Theorem 31. *In DL-Lite$^{\mathcal{H}}_{\mathsf{pos}}$, exhaustive rooted CCQ answering is coNP-complete w.r.t. data complexity.*

Proof. The main idea is the same as in the proof of Theorem 21. However, due to the lack of existential variables, we can no longer 'reach' the colors without taking into account the paths leading to them. To address this difficulty, we translate into our context an idea from Kostylev and Reutter [2015], which takes advantage of role inclusions.

Starting from an instance $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ of the decision problem 3-Col, we consider the ABox $\mathcal{A}_\mathcal{G}$ given by:

$$\begin{aligned}\mathcal{A}_\mathcal{G} = {} & \{\mathrm{toVertex}(\mathsf{a}, \mathsf{u}) \mid u \in \mathcal{V}\} \cup \{\mathrm{Vertex}(\mathsf{u}) \mid u \in \mathcal{V}\} \\ & \cup \{\mathrm{Edge}(\mathsf{u}_1, \mathsf{u}_2) \mid (u_1, u_2) \in \mathcal{E}\} \\ & \cup \{\mathrm{toVertex}(\mathsf{a}, \mathsf{a_v}), \mathrm{Edge}(\mathsf{a_v}, \mathsf{a_v}), \mathrm{HasCol}(\mathsf{a_v}, \mathsf{r})\} \\ & \cup \{\mathrm{Colors}(\mathsf{u}, \mathsf{r}) \mid u \in \mathcal{V}\} \cup \{\mathrm{Colors}(\mathsf{u}, \mathsf{g}) \mid u \in \mathcal{V}\} \cup \{\mathrm{Colors}(\mathsf{u}, \mathsf{b}) \mid u \in \mathcal{V}\}\end{aligned}$$

and the TBox $\mathcal{T} := \{\text{Vertex} \sqsubseteq \exists\text{HasCol}, \text{HasCol} \sqsubseteq \text{Colors}\}$, and we denote by $\mathcal{K}_\mathcal{G} = (\mathcal{T}, \mathcal{A}_\mathcal{G})$ the resulting KB. A part of the canonical model of $\mathcal{K}_\mathcal{G}$ is depicted in Figure 4.11. As in the proof of Theorem 21, we use Vertex $\sqsubseteq$ ∃HasCol to assign colors to vertices, and the more general role Colors will be used to detect colors.

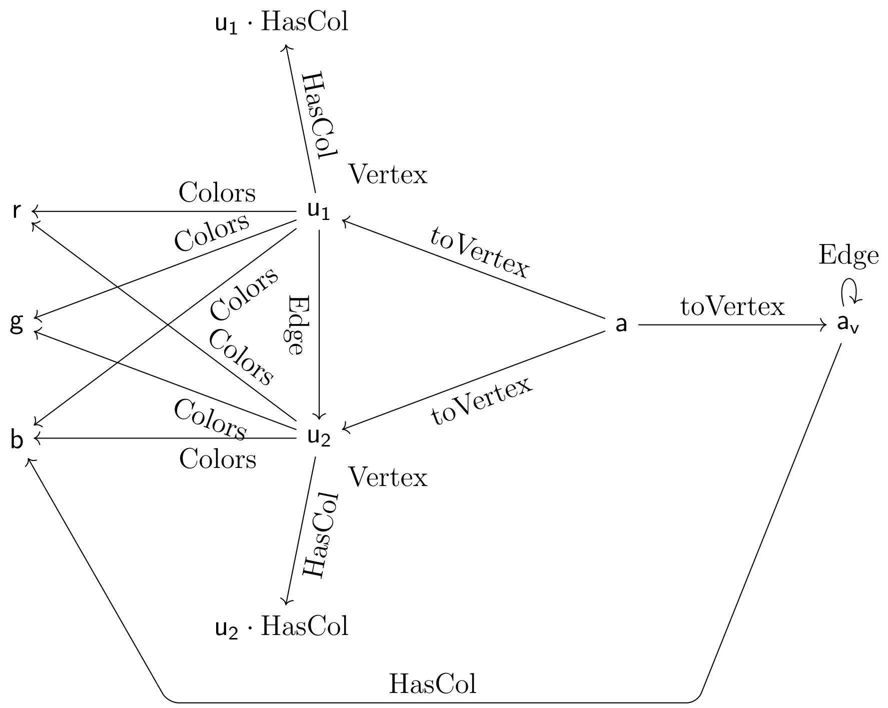


Figure 4.11: A part of $\mathcal{C}_{\mathcal{K}_\mathcal{G}}$ with $(u_1, u_2) \in \mathcal{E}$.

We consider the two following exhaustive rooted CCQs:

$$\begin{aligned} q^{edge} &= \exists z_c\ \exists z_1\ \exists z_2\ \text{toVertex}(\mathsf{a}, z_1) \wedge \text{toVertex}(\mathsf{a}, z_2) \\ &\quad \wedge \text{Edge}(z_1, z_2) \wedge \text{HasCol}(z_1, z_c) \wedge \text{HasCol}(z_2, z_c) \\ q^{col} &= \exists z_v\ \exists z\ \text{toVertex}(\mathsf{a}, z_v) \wedge \text{Colors}(z_v, z) \end{aligned}$$

and let q be the query obtained by taking the conjunction of these two queries and keeping all of the variables existentially quantified. The query q is displayed in Figure 4.12. Observe that while it is similar to the query from the proof of Theorem 21 (see Figure 4.2), the two existential variables in that query (y_c, y) have been replaced with counting variables (z_c, z_v), and one of the HasCol atom has been changed to a Colors atom.
It is not hard to see that $[3|\mathcal{V}| + 1, +\infty]$ is a certain answer to q over $\mathcal{K}_\mathcal{G}$. Indeed, there are at least $3|\mathcal{V}|$ matches of q in any model $\mathcal{I}$ of $\mathcal{KG}$, obtained as follows:

$$z_1, z_2 \mapsto \mathsf{a_v} \qquad z_c \mapsto \mathsf{r} \qquad z_v \mapsto \mathsf{u}\ (u \in \mathcal{V}) \qquad z \mapsto \mathsf{r} \mid \mathsf{g} \mid \mathsf{b}$$

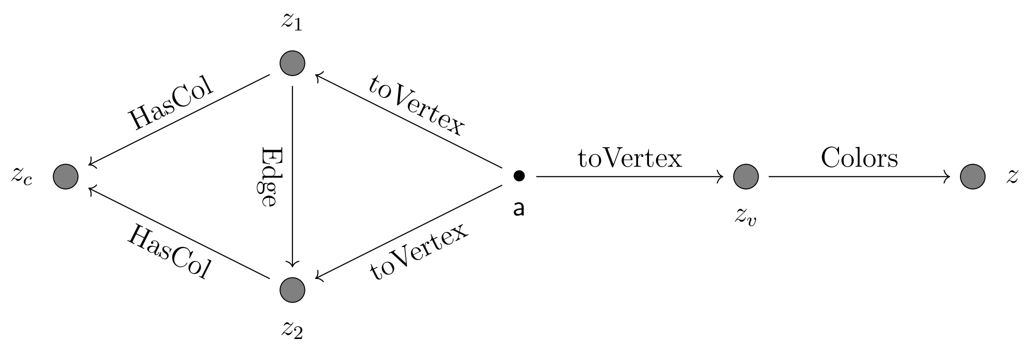


Figure 4.12: The exhaustive rooted CCQ q, which is the conjunction of q^{edge} (left part) and q^{col} (right part).

and one additional match given by:

$$z_1, z_2, z_v \mapsto \mathsf{a_v} \qquad z_c, z \mapsto \mathsf{r}$$

To complete the proof, we establish the following claim:

$$[3|\mathcal{V}| + 2, +\infty] \text{ is a certain answer to } q \text{ over } \mathcal{K}_{\mathcal{G}} \text{ iif } \mathcal{G} \notin \text{3-}\textsc{Col}.$$

($\Rightarrow$) This direction is proven in the same manner as the claim in the proof of Theorem 21. We assume $[3|\mathcal{V}| + 2, +\infty]$ is a certain answer and take a possible coloring $\tau : \mathcal{V} \to \{\mathsf{r}, \mathsf{g}, \mathsf{b}\}$. We then use τ to build a model $\mathcal{I}_\tau$ of $\mathcal{K}_{\mathcal{G}}$ and use the existence of an additional match π to show that τ contains a monochromatic edge (hence $\mathcal{G} \notin$ 3-Col).

($\Leftarrow$) Assume $\mathcal{G} \notin$ 3-Col, and take some model $\mathcal{I}$ of $\mathcal{K}_{\mathcal{G}}$. There is a homomorphism $f : \mathcal{C}_{\mathcal{K}_{\mathcal{G}}} \to \mathcal{I}$. Define $\tau : \mathcal{V} \to \Delta^{\mathcal{I}}$ as follows: $\tau(u) = f(\mathsf{u} \cdot \text{HasCol})$. Note that τ is well defined, as the inclusion Vertex $\sqsubseteq \exists$HasCol ensures that there is an element $\mathsf{u} \cdot$ HasCol in $\mathcal{C}_{\mathcal{K}_{\mathcal{G}}}$. There are two cases to consider:

- If there exists $u \in \mathcal{V}$ such that $\tau(u) \notin \{\mathsf{r}, \mathsf{g}, \mathsf{b}\}$, then the axiom HasCol $\sqsubseteq$ Colors ensures $(\mathsf{u}^{\mathcal{I}}, \tau(u)) \in \text{Colors}^{\mathcal{I}}$, which provides an additional match of q^{color} in $\mathcal{I}$ with $z \mapsto \tau(u)$ and $z_v \mapsto \mathsf{u}^{\mathcal{I}}$.
- Else, since $\mathcal{G} \notin$ 3-Col, there exists an edge $(u_1, u_2) \in \mathcal{E}$ such that $\tau(u_1) = \tau(u_2)$. It yields a new match given by:

$$z \mapsto \mathsf{r} \quad z_v \mapsto \mathsf{a_v} \quad z_1 \mapsto \mathsf{u_1} \quad z_2 \mapsto \mathsf{u_2} \quad z_c \mapsto \tau(u_1)\ (= \tau(u_2))$$

In both cases, there is an additional c-match for q. We thus obtain that $[3|\mathcal{V}| + 2, +\infty]$ is certain answer to q over $\mathcal{K}_{\mathcal{G}}$. □

We can adapt the previous reduction to prove DP-hardness w.r.t. data complexity of the corresponding problem of tight exhaustive rooted CCQ answering.

Theorem 32. *Tight exhaustive rooted CCQ answering over DL-Lite$^{\mathcal{H}}_{\mathsf{pos}}$ ontologies is* coNP*-complete w.r.t. data complexity.*

Proof. Here again we proceed by reduction from the DP-complete problem of deciding if $\mathcal{G}_1 \in$ 3-Col and $\mathcal{G}_2 \notin$ 3-Col. given *planar* graphs $\mathcal{G}_1 := (\mathcal{V}_1, \mathcal{E}_1)$ and $\mathcal{G}_2 := (\mathcal{V}_2, \mathcal{E}_2)$. The proof proceeds w.r.t. the proof of Theorem 31 exactly as the proof of Theorem 22 proceeds w.r.t. the proof of Theorem 21, that is by introducing an asymmetry in the query on the $\mathcal{G}_1$-side. The only salient difference is that the basic number of matches on the side of $\mathcal{G}_1$ is $(3|\mathcal{V}_1| + 1) \times (3|\mathcal{V}_1| + 1)$ instead of 3×3, and similarly for $\mathcal{G}_2$, it is $3|\mathcal{V}_2| + 1$ instead of 3. The case analysis is modified as follows:

	$\mathcal{G}_1 \in$ 3-Col	$\mathcal{G}_1 \notin$ 3-Col
$\mathcal{G}_2 \in$ 3-Col	$(3\|\mathcal{V}_1\| + 1)^2 \times (3\|\mathcal{V}_2\| + 1)$	$(3\|\mathcal{V}_1\| + 2)^2 \times (3\|\mathcal{V}_2\| + 1)$
$\mathcal{G}_2 \notin$ 3-Col	$(3\|\mathcal{V}_1\| + 1)^2 \times (3\|\mathcal{V}_2\| + 2)$	$(3\|\mathcal{V}_1\| + 2)^2 \times (3\|\mathcal{V}_2\| + 2)$

One can easily verify these 4 numbers are always distinct, hence the claim becomes:

$[(3|\mathcal{V}_1| + 1)^2 \times (3|\mathcal{V}_2| + 2), +\infty]$ is the tightest certain answer
iff
$\mathcal{G}_1 \in$ 3-Col and $\mathcal{G}_2 \notin$ 3-Col. ☐

We now move to DL-Lite$_{\mathsf{core}}$ and start by recalling Lemma 16, which highlights an important property of the canonical model construction for DL-Lite$_{\mathsf{core}}$ KBs.

Lemma 26 (Recalling Lemma 16). *For any role $\mathrm{R} \in \mathsf{N}_{\mathsf{R}}^{\pm}$ and anonymous element d_1 in the canonical model $\mathcal{C}_{\mathcal{K}}$ of a DL-Lite$_{\mathsf{core}}$ KB $\mathcal{K}$, there is at most one element $d_2 \in \mathcal{C}_{\mathcal{K}}$ such that $(d_1, d_2) \in \mathrm{R}^{\mathcal{C}_{\mathcal{K}}}$.*

This leads to the following central property, which ensures it is sufficient to compute the number of matches of an exhaustive rooted CCQ over the canonical model of a DL-Lite$_{\mathsf{core}}$ KB to answer it.

Theorem 33. *For every DL-Lite$_{\mathsf{core}}$ KB $\mathcal{K}$ and exhaustive rooted CCQ q, the minimum number of matches of q across models of $\mathcal{K}$ is reached in the canonical model of $\mathcal{K}$.*

Proof. Exploiting the structure of DL-Lite$_{\mathsf{core}}$ canonical models, we show that if π_1, π_2 are distinct matches of an exhaustive rooted CCQ q in $\mathcal{C}_{\mathcal{K}}$, then there exists a variable v such that $\pi_1(v) \neq \pi_2(v)$ and $\pi_1(v), \pi_2(v) \in \mathsf{Ind}(\mathcal{A})$. It follows that if we take an arbitrary model $\mathcal{I}$ of $\mathcal{K}$, and f a homomorphism of $\mathcal{C}_\mathcal{K}$ into $\mathcal{I}$, then f injectively maps query matches in $\mathcal{C}_\mathcal{K}$ to query matches in $\mathcal{I}$.

We hence focus on proving that if π_1, π_2 are distinct matches of a exhaustive rooted CCQ q in $\mathcal{C}_\mathcal{K}$, then there exists a variable v such that $\pi_1(v) \neq \pi_2(v)$ and $\pi_1(v), \pi_2(v) \in \mathsf{Ind}(\mathcal{A})$.

Suppose for a contradiction that this is not the case. There there are distinct matches π_1, π_2 of q in $\mathcal{C}_\mathcal{K}$ such that for all variables v such that $\pi_1(v) \neq \pi_2(v)$, either $\pi_1(v) \notin \mathsf{Ind}(\mathcal{A})$ or $\pi_2(v) \notin \mathsf{Ind}(\mathcal{A})$. As q is exhaustive rooted, every variable v is connected to either an answer variable or individual in the Gaifman graph. Let $d(v)$ denote the length of the shortest path from v to an answer variable of individual. Note that $d(v) = 0$ iff v is an answer variable. Since π_1 and π_2 are distinct, there exists a variable v such that $\pi_1(v) \neq \pi_2(v)$. Choose such a variable v^* with minimal d-value, i.e., if $d(u) < d(v^*)$, then $\pi_1(u) = \pi_2(u)$. By assumption, either $\pi_1(v^*) \notin \mathsf{Ind}(\mathcal{A})$ or $\pi_2(v^*) \notin \mathsf{Ind}(\mathcal{A})$. We'll suppose the former (the other case is treated analogously). Note that v^* cannot be an answer variable (else we would have $\pi_1(v^*) \in \mathsf{Ind}(\mathcal{A})$). It follows that $d(v^*) > 0$, and so we can find another variable u^* and role name $\mathrm{R} \in \mathsf{N}_\mathsf{R}^\pm$, with $d(u^*) = d(v^*) - 1$ and either $\mathrm{R}(u^*, v^*) \in q$ or $\mathrm{R}^-(v^*, u^*) \in q$ (recall that if $\mathrm{R} = \mathrm{P}^-$, then $\mathrm{R}^- = \mathrm{P}$). As π_1 and π_2 are matches of q in $\mathcal{C}_\mathcal{K}$, we therefore have $(\pi_1(u^*), \pi_1(v^*)) \in \mathrm{R}^{\mathcal{C}_\mathcal{K}}$ and $(\pi_2(u^*), \pi_2(v^*)) \in \mathrm{R}^{\mathcal{C}_\mathcal{K}}$. Moreover, since $d(u^*) < d(v^*)$, we have $\pi_1(u^*) = \pi_2(u^*)$. There are two cases to consider:

- Case 1: $\pi_1(u^*) = \pi_2(u^*) = \mathsf{c} \in \mathsf{Ind}(\mathcal{A})$. From the proof of Lemma 26, we know that $\pi_1(v^*) = \mathsf{c} \cdot \mathrm{R}$. The fact that $\mathsf{c} \cdot \mathrm{R} \in \Delta^{\mathcal{C}_\mathcal{K}}$ implies that there is no individual b such that $(\mathsf{c}, \mathsf{b}) \in \mathrm{R}^{\mathcal{C}_\mathcal{K}}$. Thus, we must have $\pi_2(v^*) = \mathsf{c} \cdot \mathrm{R}$, which yields $\pi_1(v^*) = \pi_2(v^*)$, contradicting our earlier assumption.

- Case 2: $\pi_1(u^*) = \pi_2(u^*) \notin \mathsf{Ind}(\mathcal{A})$. By Lemma 26, there is a unique element e such that $(\pi_1(u^*), e) \in \mathrm{R}^{\mathcal{C}_\mathcal{K}}$. We thus obtain $\pi_1(v^*) = e = \pi_2(v^*)$, a contradiction.

As both cases lead to a contradiction, it must therefore be the case that the statement holds. □

In the following sections, we explore how this property of the canonical model being optimal impacts the combined, resp. data, complexity of answering exhaustive rooted CCQs. For both situations, we use the next lemma, implicit in Bienvenu et al. [2013], constraining the possible images of matches in $\mathcal{C}_\mathcal{K}$:

Lemma 27. *For every DL-Lite*$_{\mathsf{core}}$ *TBox $\mathcal{T}$ and CCQ q, we can construct in polynomial time a set of words $\Gamma_{q,\mathcal{T}}$ such that for every KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, match π of q in $\mathcal{C}_\mathcal{K}$, and variable v of q: $\pi(v) = \mathsf{a} \cdot w$ for some $\mathsf{a} \in \mathsf{Ind}(\mathcal{A})$ and $w \in \Gamma_{q,\mathcal{T}}$.*

4.5.2 DL-Lite$_{\mathsf{core}}$ and combined complexity

In this section, we prove that exhaustive rooted CCQ answering over DL-Lite$_{\mathsf{core}}$ KBs is PP-complete w.r.t. combined complexity, and hence in PSPACE. We recall that the class PP contains all decision problems for which there exists a non-deterministic Turing machine (TM) such that, when the input is a 'yes' instance, then at least half of the computation paths accept, while on 'no' instances, less than half of the computation paths accept.

Theorem 34. *Exhaustive rooted CCQ answering over DL-Lite$_{\mathsf{core}}$ KBs is in* PP *w.r.t. combined complexity.*

Proof. We describe the TM used for PP membership, which takes as input a DL-Lite$_{\mathsf{core}}$ KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, an exhaustive rooted CCQ q, and candidate integer m.
Phase 1 The TM deterministically constructs the set $\Gamma_{q,\mathcal{T}}$ of words from Lemma 27.

Phase 2 The TM guesses a mapping π of the variables in q to elements from $\{\mathsf{a} \cdot w \mid \mathsf{a} \in \mathsf{Ind}(\mathcal{A}), w \in \Gamma_{q,\mathcal{T}}\}$. It then compares m with the number $C = |\Gamma_{q,\mathcal{T}}|^{|q|}$ of possible mappings and proceeds accordingly:

- if $m \geq \frac{C}{2} + 1$, the TM guesses an integer i with $0 \leq i \leq 2m - 3$ and accepts iff π is a c-match of q and $i < C$;
- if $m < \frac{C}{2} + 1$, the TM guesses an integer i with $0 \leq i \leq 2C - 2m + 1$ and accepts iff π is c-match for q or $i < C - 2m + 2$.

Let us denote $q^{\mathcal{C}_\mathcal{K}}$ the number of matches for q in $\mathcal{C}_\mathcal{K}$. Due to Theorem 33 and Lemma 27, an input is a 'yes' instance iff $q^{\mathcal{C}_\mathcal{K}} \geq m$. To finish the proof of PP membership, we need to examine the number of accepting computation paths for the described TM and show that when $q^{\mathcal{C}_\mathcal{K}} \geq m$, at least half of the computation paths accept, and when $q^{\mathcal{C}_\mathcal{K}} < m$, less than half of the computation paths accept. Let us consider the two cases from Phase 2:

- If $m \geqslant \frac{C}{2} + 1$, then the number of accepting computation paths is $q^{\mathcal{C}_\mathcal{K}} \times C$, corresponding to cases where the TM guesses a mapping that is a c-match, then guess a number $0 \leq i < C$. The total number of computation paths is $C \times (2m - 2)$, corresponding to a guess of one of the C mappings, then the guess of an integer $0 \leq i \leq 2m - 3$.

- If $m < \frac{C}{2} + 1$, then the number of accepting computation paths is

$$q^{\mathcal{C}_\mathcal{K}} \times (2C - 2m + 2) + (C - q^{\mathcal{C}_\mathcal{K}}) \times (C - 2m + 2) = C(C - 2m + q^{\mathcal{C}_\mathcal{K}} + 2),$$

corresponding to the sum of the number of cases where we guess a c-match followed by an integer $0 \leq i \leq 2C - 2m + 1$ and the number of cases where we guess a mapping that is not a c-match followed by an integer i with $0 \leq i < C - 2m + 2$. The total number of computation paths is $C \times (2C - 2m + 2)$ (guess one of the C mappings, then guess an integer $0 \leq i \leq 2C - 2m + 1$).

In both cases, it is easily verified that:

$$q^{\mathcal{C}_\mathcal{K}} \geq m \iff \frac{\#\text{accepting computation paths}}{\#\text{possible computation paths}} > \frac{1}{2}.$$

(Note that in the first case, we always have $m \geq 2$, so the value $2m - 2$ in the denominator is positive, while in the second case, $C \geq 1$ implies that the value $(2C - 2m + 2)$ in the denominator is positive.) □

The lower bound is obtained by a reduction from the following PP-complete problem [Bailey et al., 2007]: given a propositional formula ψ in CNF and number n, decide whether ψ has at least n satisfying assignments.

Theorem 35. *Exhaustive rooted CCQ answering over DL-Lite$_{\mathsf{pos}}$ KBs is PP-hard w.r.t. combined complexity.*

Proof. Consider an instance of the PP-complete problem mentioned above, given by the formula $\psi := \exists \mathbf{u} \bigwedge_{k=1}^{l} \xi_k$ (with ξ_k is a 3-clause) and number N. We consider the KB $\mathcal{K}_\psi = (\emptyset, \mathcal{A}_\psi)$, which has an empty TBox, and whose ABox $\mathcal{A}_\psi$ contains the following assertions:

- $\mathrm{Clause_k}(\mathsf{a}, \xi_\mathsf{k}^\mathsf{p})$ for each clause ξ_k and each $p \in \{1, ...7\}$, with each $\xi_\mathsf{k}^\mathsf{p}$ representing one of the 7 satisfying assignments for the clause ξ_k;
- $\mathrm{Asn_1}(\xi_\mathsf{k}^\mathsf{p}, \xi_k^p(\omega_k^1))$, $\mathrm{Asn_2}(\xi_\mathsf{k}^\mathsf{p}, \xi_k^p(\omega_k^2))$ and $\mathrm{Asn_3}(\xi_\mathsf{k}^\mathsf{p}, \xi_k^p(\omega_k^3))$ for each $p = 1, ...7$ and each clause ξ_k, where $\xi_k^p(\omega_k^i)$ is the truth value (true or false) assigned by ξ_k^p to the ith variable occurring in the kth clause.

As for the query, we consider the following exhaustive rooted CCQ (an example is depicted in Figure 4.13):

$$q_\psi := \exists z_{\xi_1} \ldots \exists z_{\xi_l} \; \exists z_{u_1} \ldots \exists z_{u_n} \bigwedge_{k=1}^{l} \left(\mathrm{Clause_k}(\mathsf{a}, z_{\xi_k}) \land \bigwedge_{i=1}^{3} \left(\mathrm{Asn_i}(z_{\xi_k}, z_{\omega_k^i}) \right) \right)$$

To complete the proof, we establish the following claim:

$[N, +\infty]$ is a certain answer to q_ψ over $\mathcal{K}_\psi$ iff ψ has at least N satisfying assignments.

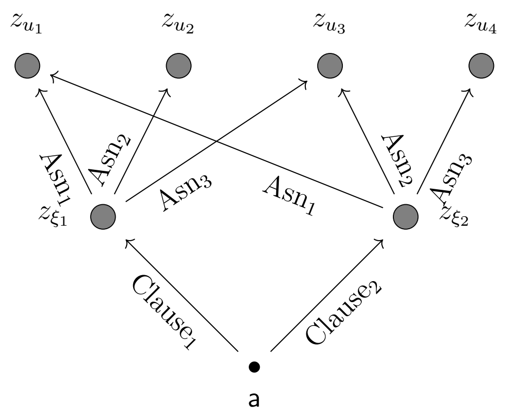


Figure 4.13: The query q_ψ with $\psi = (u_1 \vee \neg u_2 \vee \neg u_3) \wedge (\neg u_1 \vee u_3 \vee u_4)$.

($\Rightarrow$) Assume $(\emptyset, [N, +\infty]) \in q_\psi^{\mathcal{K}_\psi}$. This implies in particular that there are N c-matches for q_ψ in $\mathcal{C}_{\mathcal{K}_\psi}$. Since the TBox is empty, the domain of $\mathcal{C}_{\mathcal{K}_\psi}$ is $\mathsf{Ind}(\mathcal{A}_\psi)$, and $\mathcal{C}_{\mathcal{K}_\psi}$ makes true precisely the assertions in $\mathcal{A}_\psi$. By examining q_ψ and $\mathcal{A}_\psi$, we see that each of the matches of q_ψ in $\mathcal{C}_{\mathcal{K}_\psi}$ maps each of the variables z_{u_i} to either true or false. We can therefore associate with each match π the following truth assignment for the variables $u_1, \ldots, u_n$: $\tau_\pi(u_i) = \pi(z_{u_i})$. By further examining the definition of the individuals $\xi_\mathsf{k}^\mathsf{p}$ and the roles $\mathrm{Asn}_1, \mathrm{Asn}_2, \mathrm{Asn}_3$, it is easy to verify that each τ_π is a satisfying assignment for ψ. Moreover, since we know we have N such assignments, it only remains to show that each match π yields a distinct assignment τ_π. To see why this is the case, observe that once we know the images of all of the variables z_{u_i}, there is a unique way of mapping the variables z_{ξ_p}. It follows that ψ has at least N satisfying assignments.

($\Leftarrow$) Assume ψ has at least N satisfying assignments. Therefore, we have $\tau_1, \ldots \tau_N$ distinct assignments for $u_1, \ldots u_n$ satisfying ψ. This ensures that, if we define $\pi_{\tau_m}(z_{u_i}) = \tau_m(u_i)$, we can always extend the mapping $\pi_{\tau_m}(z_{u_i})$ into a match for q_ψ, yielding N distinct matches. This holds in any model since we only need the 'ABox part' of the model, hence $[N, +\infty]$ is a certain answer to q_ψ over $\mathcal{K}_\psi$. □

4.5.3 DL-Lite$_{\mathsf{core}}$ and data complexity

We now turn to the data complexity of answering exhaustive rooted CCQs over DL-Lite$_{\mathsf{pos}}$ ontologies. With Theorem 33 and Lemma 27 in hand, we prove that this problem is TC^0-complete. We recall that TC^0 is a circuit complexity class defined similarly to AC^0 but additionally allowing threshold gates. It is known that $\mathsf{AC}^0 \subsetneq \mathsf{TC}^0 \subseteq \mathsf{NC}^1 \subseteq \mathsf{LogSpace} \subseteq \mathsf{PTime}$.

Theorem 36. *Exhaustive rooted CCQ answering in DL-Lite*$_{\mathsf{core}}$ *is in* TC^0 *w.r.t. data complexity.*

Proof. We need a family of circuits in order to be able to handle ABoxes of different sizes. More precisely, we will create one circuit for each possible number ℓ of individual names. We can assume w.l.o.g. that the same set of individuals, denoted Ind_ℓ, is used for all of the ABoxes having ℓ individuals. Let us now explain how to represent an input $(\mathcal{A}^*, \mathbf{a}^*, m^*)$ to the circuit that handles ℓ-individual ABoxes.

- Each atomic role P appearing in $\mathcal{T}$ and/or q is represented by input gates $?_{\mathrm{P}(\mathsf{a},\mathsf{b})\in\mathcal{A}?}$ for $\mathsf{a}, \mathsf{b} \in \mathsf{Ind}_\ell$. The gate $?_{\mathrm{P}(\mathsf{a},\mathsf{b})\in\mathcal{A}?}$ is set to 1 iff $\mathrm{P}(\mathsf{a}, \mathsf{b}) \in \mathcal{A}^*$.
- Each atomic concept A appearing in $\mathcal{T}$ and/or q is represented by input gates $?_{\mathrm{A}(\mathsf{a})\in\mathcal{A}?}$ for $\mathsf{a} \in \mathsf{Ind}_\ell$. The gate $?_{\mathrm{A}(\mathsf{a})\in\mathcal{A}?}$ is set to 1 iff $\mathrm{A}(\mathsf{a}) \in \mathcal{A}^*$.
- The tuple $\mathbf{a}^*$ is represented by input gates $?_{\mathsf{a}_k=\mathsf{a}}$ for $1 \leq k \leq |\mathbf{x}|$ and $\mathsf{a} \in \mathsf{Ind}_\ell$. The gate is set to 1 iff $\mathbf{a}_k^* = \mathsf{a}$.
- The integer m^* is represented in binary by input gates $?_{b_k=1}$ for each $0 \leq k < \log_2(|\mathsf{Ind}(\mathcal{A}^*)| + |\mathcal{T}|)^{|q|})$. The gate $?_{b_k=1}$ is set to 1 iff the k^{th} bit of m^* is 1 (with 0^{th}-bit being the least significant bit).

Regarding the last point, we use the observation from Kostylev and Reutter [2015] that if $(\mathbf{a}^*, [m^*, +\infty]) \in q^{(\mathcal{T},\mathcal{A}^*)}$, then m^* cannot exceed $(|\mathsf{Ind}(\mathcal{A}^*)| + |\mathcal{T}|)^{|q|} = (|\mathsf{Ind}_\ell| + |\mathcal{T}|)^{|q|}$. This is a direct consequence of the fact that every satisfiable DL-Lite$_\mathcal{R}$ KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ has a model with at most $|\mathsf{Ind}(\mathcal{A})| + |\mathcal{T}|$ elements.

We now describe the other parts of the circuit. We introduce, for each relevant positive concept C (i.e., atomic concept or existential concept $\exists R$ that uses concept and role names from $\mathcal{T}$ and/or q) and each individual name $\mathsf{a} \in \mathsf{Ind}_\ell$, a disjunctive gate $\vee_{\mathcal{K}\models \mathrm{C}(\mathsf{a})?}$ taking as inputs:

- $?_{\mathrm{A}(\mathsf{a})\in\mathcal{A}?}$ for each atomic concept A such that $\mathcal{T} \models \mathrm{A} \sqsubseteq \mathrm{C}$.
- $?_{\mathrm{P}(\mathsf{a},\mathsf{b})\in\mathcal{A}?}$ for all $\mathsf{b} \in \mathsf{Ind}(\mathcal{A})$ such that $\mathcal{T} \models \exists \mathrm{P} \sqsubseteq \mathrm{C}$.
- $?_{\mathrm{P}(\mathsf{b},\mathsf{a})\in\mathcal{A}?}$ for all $\mathsf{b} \in \mathsf{Ind}(\mathcal{A})$ such that $\mathcal{T} \models \exists \mathrm{P}^- \sqsubseteq \mathrm{C}$.

The preceding gates determine the ABox part of the canonical model. We next need to decide the existence of each element of the form $\mathsf{a}w$, where $a \in \mathsf{Ind}(\mathcal{A})$ and $w \in \Gamma_{q,\mathcal{T}} \setminus \varepsilon$ (by Lemma 27, these are the only anonymous elements that can occur in a match for q). For each such $\mathsf{a}w$, we denote by R_w the first role name of w and introduce a conjunctive gate $\wedge_{\mathsf{a}w\in\Delta^{\mathcal{C}_\mathcal{K}}?}$ which takes as input:

- The negation $\neg_{\forall \mathsf{b}\in\mathsf{Ind}(\mathcal{A})\ \neg\mathrm{R}(\mathsf{a},\mathsf{b})?}$ of a disjunctive gate $\vee_{\exists \mathsf{b}\in\mathsf{Ind}(\mathcal{A})\ \mathrm{R}(\mathsf{a},\mathsf{b})?}$ taking as inputs the gates:

- $?_{\mathrm{P}(\mathsf{a},\mathsf{b})}$ for each $\mathsf{b} \in \mathsf{Ind}_\ell$, if $\mathrm{R} = \mathrm{P} \in \mathsf{N_R}$.
- $?_{\mathrm{P}(\mathsf{b},\mathsf{a})}$ for each $\mathsf{b} \in \mathsf{Ind}_\ell$ if $\mathrm{R} = \mathrm{P}^-$ with $\mathrm{P} \in \mathsf{N_R}$.

which verifies that there is not already a R_w-successor to a.

- The gate $\vee_{\mathcal{K}\models\exists \mathrm{R}_w(\mathsf{a})?}$ that checks that a witnessing R_w-successor is needed.

The circuit next determines for each mapping $\pi : \mathbf{x} \cup \mathbf{z} \mapsto \{\mathsf{a}w | \mathsf{a} \in \mathsf{Ind}_\ell, w \in \Gamma_{q,\mathcal{T}}\}$, whether π is a match for $q(\mathbf{a}^*)$. Notice that, regardless of the input ABox, we can restrict to a set of relevant mappings by keeping only those which map the answer variables $\mathbf{x}$ to individuals from Ind_ℓ and which map variables v_1, v_2 occurring in a role atom $\mathrm{R}(v_1, v_2)$ from q onto either:

- a pair of individual names, or
- a pair w_1, w_2 such that $w_2 = w_1\mathrm{R}$ or $w_1 = \mathrm{R}^- w_2$.

Similarly, we can restrict the set of relevant mappings by keeping only those which map variable v occuring in a concept atom $\mathrm{A}(v)$ from q onto either an individual name, or an element $aw\mathrm{R}$, where $\mathcal{K} \models \exists \mathrm{R}^- \sqsubseteq \mathrm{A}$. Clearly, any mapping π that does not respect these conditions cannot be a match, due to the definition of $\mathrm{R}^{\mathcal{C}_\mathcal{K}}$. This restriction simplifies the process of checking if a mapping is a match for $q(\mathbf{a}^*)$: we are only left with verifying the existence of the anonymous elements in its image, as well as the validity of the atoms mapped onto the ABox part of the canonical model.

For each relevant mapping π, we introduce a conjunctive gate $\wedge_{\pi \text{ match } ?}$ taking as inputs all gates:

- $?_{\mathbf{a}_k = \pi(x_k)?}$ for each $1 \leq k \leq |\mathbf{x}|$ (to check $\mathbf{x}$ is mapped on $\mathbf{a}^*$).
- $\wedge_{\pi(z) \in \Delta^{\mathcal{C}_\mathcal{K}}?}$ for each $z \in \mathbf{z}$ such that $\pi(z) \notin \mathsf{Ind}_\ell$ (to check for existence of $\pi(z)$ under input $\mathcal{A}^*$).
- $?_{\mathrm{R}(\pi(v_1),\pi(v_2)) \in \mathcal{A}?}$ for each $v_1, v_2 \in \mathbf{x} \cup \mathbf{z}$ such that $\mathrm{R}(v_1, v_2) \in q$ and $\pi(v_1), \pi(v_2) \in \mathsf{Ind}(\mathcal{A})$ (to check the validity of the mapping for pairs of variables mapped on individual names).
- $\vee_{\mathcal{K}\models \mathrm{A}(\pi(v))?}$ for each $v \in \mathbf{x} \cup \mathbf{z}$ such that $\mathrm{A}(v) \in q$ and $\pi(v) \in \mathsf{Ind}(\mathcal{A})$ (to check the validity of the mapping for variables mapped on individual names).

We will next use threshold gates in order to compute the total number of matches. Introduce, for each $k = 0, \ldots, (\mathsf{Ind}_\ell \times \Gamma_{q,\mathcal{T}})^{|q|}$, a threshold gate $\mathsf{T}^{(k)}_{q_\mathbf{a}^{\mathcal{C}_\mathcal{K}} \geq k?}$ taking as input every $\wedge_{\pi \text{ match } ?}$. The gate $\mathsf{T}^{(k)}_{q_\mathbf{a}^{\mathcal{C}_\mathcal{K}} \geq k?}$ returns 1 iff at least k of its inputs are 1. By construction, the latter holds iff there are at least k matches for $q(\mathbf{a}^*)$.

In parallel, we introduce a conjunctive gate $\wedge_{m=k?}$ for each $k = 0, \ldots, (\mathsf{Ind}_\ell \times \Gamma_{q,\mathcal{T}})^{|q|}$ taking as inputs:

- the input gates $?_{b_j=1?}$ such that the j^{th} bit of the binary encoding of k is 1
- the negation of each input gate $?_{b_j=1?}$ such that the j^{th} bit of the binary encoding of k is 0

The gate $\wedge_{m=k?}$ returns 1 iff $m^* = k$.

We combine the preceding two types of gates to compare m^* and the computed number of matches. For each $k = 0, \ldots, (\mathsf{Ind}_\ell \times \Gamma_{q,\mathcal{T}})^{|q|}$, we introduce a conjunctive gate $\wedge_{q_{\mathbf{a}}^{\mathcal{C}_\mathcal{K}} \geq m?}$ taking as input $\mathsf{T}^{(k)}_{q_{\mathbf{a}}^{\mathcal{C}_\mathcal{K}} \geq k?}$ and $\wedge_{m=k?}$.

Finally, our output gate is a disjunctive gate $\vee_{\mathsf{output}}$ taking as inputs all gates $\wedge_{q_{\mathbf{a}}^{\mathcal{C}_\mathcal{K}} \geq m?}$. By construction, this gate outputs 1 iff there are at least m^* matches of $q(\mathbf{a}^*)$ in the canonical model of the considered KB.

The depth of the circuit is 7, hence constant, showing membership in TC^0. □

A matching lower bound can be shown by a simple reduction (using an empty TBox) from the TC^0-complete problem that asks, given a binary string s and number k, whether the number of 1-bits in s exceeds k [Aehlig et al., 2007].

Theorem 37. *Exhaustive rooted CCQ answering in DL-Lite*$_{\mathsf{pos}}$ *is* TC^0*-hard w.r.t. data complexity.*

Proof. The reduction from the NumOnes problem works as follows: given an instance (s, k), we create an ABox $\mathcal{A}_s := \{\mathrm{R}(\mathsf{a}, \mathsf{s}_k) \mid s_k \in s \wedge s_k = 1\}$, along with the empty TBox $\mathcal{T} = \emptyset$ and exhaustive rooted CCQ $q := \exists z\ \mathrm{R}(\mathsf{a}, z)$. It is clear that $[k, +\infty]$ is a certain answer of q over $(\mathcal{T}, \mathcal{A}_s)$ iff $(s, k) \in$ NumOnes. It can be verified that this simple reduction can be implemented by AC^0 circuits (so constitutes an AC^0-reduction, as required). □

5 Cardinality Queries

In this chapter, we focus on Boolean atomic counting queries of the form $\exists z\ \mathrm{A}(z)$ and $\exists z_1\, \exists z_2\ \mathrm{R}(z_1, z_2)$. We shall refer to such restricted CCQs as *cardinality queries* as they correspond to the natural task of determining (bounds on) the cardinality of a given concept or role name.

The data complexity of answering such basic counting queries has been briefly explored in Calvanese et al. [2020a], remaining completely open for DL-Lite$_{\mathsf{core}}$ ontologies, whilst for DL-Lite$^{\mathcal{H}}_{\mathsf{core}}$, the problem is known to be P-hard and in coNP. The main results of our investigation of data complexity are displayed Table 5.1. Due to our results for general CCQs from Chapter 3, membership in coNP holds for the more expressive DL $\mathcal{ALCHI}$ and all of its sublogics. Moreover, we shall prove corresponding lower bounds for concept cardinality queries evaluated over $\mathcal{EL}$ or DL-Lite$^{\mathcal{H}}_{\mathsf{core}}$ ontologies. We show that when ontologies are expressed in DL-Lite$_{\mathsf{core}}$, cardinality query answering is tractable in data complexity and enjoys the lowest possible complexity (TC^0-complete). For cardinality queries based upon a concept atom, TC^0 membership holds even for the fragment of DL-Lite$^{\mathcal{H}}_{\mathsf{core}}$ obtained by disallowing negative role inclusions. By contrast, for role cardinality queries, we show that coNP-hard situations arise in DL-Lite$^{\mathcal{H}}_{\mathsf{pos}}$, which allows only positive concept and role inclusions.

In fact, we obtain a complete data complexity classification for DL-Lite$^{\mathcal{H}}_{\mathsf{pos}}$, showing that every ontology-mediated query is either TC^0-complete, coNP-complete, or is in P and logspace-equivalent to the complement of Perfect Matching (whose precise complexity is a longstanding open problem). The preceding classification does not extend to DL-Lite$^{\mathcal{H}}_{\mathsf{core}}$: we identify new sources of coNP-hardness and further exhibit L-complete cases. We find it intriguing that such complex behaviour arises in what appears at first glance to be a simple OMQA setting. Moreover, in all of the tractable cases we identify, the canonical model may not yield the

	Concept cardinality	Role cardinality
DL-Lite$_{\mathsf{pos}}$, DL-Lite$_{\mathsf{core}}$	TC^0-c	TC^0-c
DL-Lite$^{\mathcal{H}}_{\mathsf{pos}}$	TC^0-c	coNP-c
DL-Lite$^{\mathcal{H}}_{\mathsf{core}}$, $\mathcal{EL}$, $\mathcal{ALCHI}$	coNP-c	coNP-c

Table 5.1: Cardinality query answering: worst-case data complexity.

minimum cardinality, and query answering involves solving non-trivial optimization problems. This leads us to devise an entirely new approach based upon exploring a space of strategies to find the optimal way of merging witnesses for existential axioms.

In addition to data complexity, we also obtain a complete picture of the combined complexity of answering cardinality queries in $\mathcal{ALCHI}$ and its various sublogics. The combined complexity ranges from NL to coNP in DL-Lite logics and from EXP to coNEXP for $\mathcal{EL}$ and its extensions. We achieve these results using a variety of techniques: refinements of our approach for general CCQs, adaptations of existing constructions, and further reductions involving closed predicates. Figure 5.1 summarizes these latter complexity results for cardinality queries.

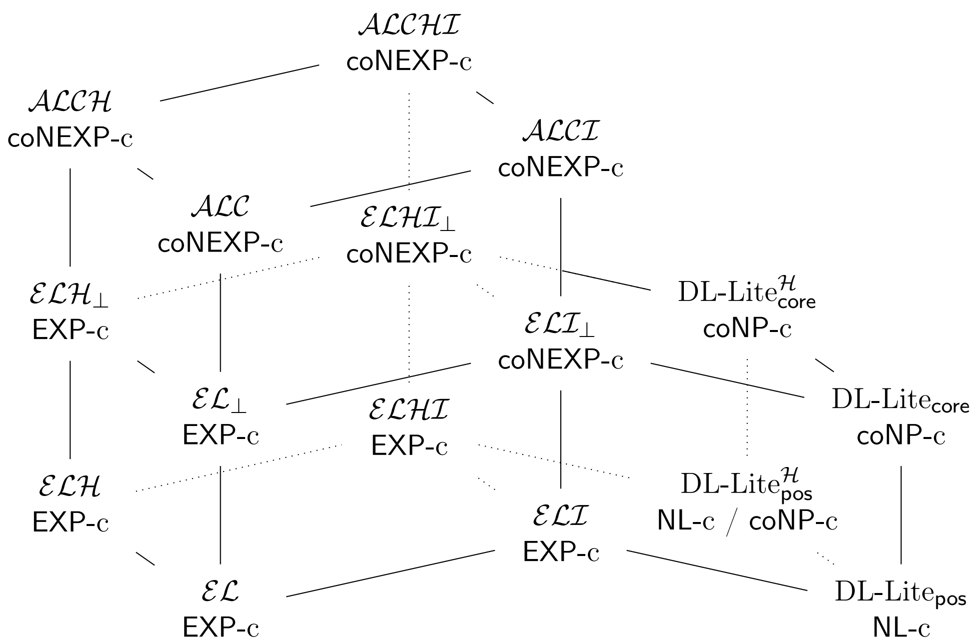


Figure 5.1: Concept / Role cardinality answering: worst-case combined complexity.

Organization of Chapter 5

5.1 Preliminaries

As mentioned in the introduction, this chapter will focus on Boolean atomic counting queries, which correspond to determining bounds on the cardinality of a given concept or role name. Such queries come in two flavours, depending upon whether the query predicate is a concept or a role name.

Definition 51. Concept cardinality queries *are Boolean CCQs of the form* $\exists z\ \mathrm{A}(z)$ *with* A *an atomic concept from* $\mathsf{N_C}$*, while* role cardinality queries *have the form* $\exists z_1\, \exists z_2\ \mathrm{R}(z_1, z_2)$ *with* R *an atomic role from* $\mathsf{N_R}$*. The* query predicate *refers to this concept name* A *or role name* R *occurring in the cardinality query of interest. We denote* q_{P} *the cardinality query whose query predicate is* P*.*

The next theorem illustrates how to reduce one setting to the other. Despite the rich concept constructors allowed in $\mathcal{ALCHI}$ compared to those allowed for roles, our result proves that concept cardinality queries can be reduced to role cardinality queries. While most of our results show no difference between the two settings in term of complexity, a notable exception arises with DL-Lite$^{\mathcal{H}}_{\mathsf{pos}}$ ontologies, for which all concept cardinality queries can be answered in TC^0 (resp. in NL) while there exists coNP-hard (resp. coNP-hard) role cardinality queries w.r.t. data complexity (resp. combined complexity).

Theorem 38. *Let $\mathcal{L}$ be a sublogic of $\mathcal{ALCHI}$ that can express* $\mathrm{A} \sqsubseteq \exists \mathrm{P}.\top$ *(*$\mathrm{A} \in \mathsf{N_C}$*,* $\mathrm{P} \in \mathsf{N_R}$*). Then concept cardinality query answering over $\mathcal{L}$ KBs can be polynomially reduced to role cardinality query answering over $\mathcal{L}$ KBs.*

Proof. Consider a concept cardinality query $q_\mathrm{A} = \exists z\ \mathrm{A}(z)$ and a KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$. We pick a fresh role name $\mathrm{P} \notin \mathsf{sig}(\mathcal{K})$, and consider the role cardinality query $q_\mathrm{P} = \exists z_1\, \exists z_2\ \mathrm{P}(z_1, z_2)$ and modified TBox $\mathcal{T}' := \mathcal{T} \cup \{\mathrm{A} \sqsubseteq \exists \mathrm{P}.\top\}$.

Any model $\mathcal{I}$ of $\mathcal{K}$ can be extended to a model $\mathcal{I}'$ of $\mathcal{K}' = (\mathcal{T}', \mathcal{A})$ by setting $\mathrm{P}^{\mathcal{I}'} := \{(e, e) \mid e \in \mathrm{A}^\mathcal{I}\}$. Indeed, this ensures satisfaction of the additional axiom $\mathrm{A} \sqsubseteq \exists \mathrm{P}.\top$. Moreover, as no new domain elements were introduced, axioms $\top \sqsubseteq \mathrm{B}$ from $\mathcal{T}$ remain satisfied, and all other axioms are not affected since $\mathrm{P} \notin \mathsf{sig}(\mathcal{T})$.

Notice that q_A has *exactly* as many matches in $\mathcal{I}$ as q_P has in $\mathcal{I}'$, hence an interval $[m, +\infty]$ is a certain answer to q_A over $\mathcal{K}$ iff it is a certain answer to q_P over $\mathcal{K}'$. □

Due to our results for general CCQs investigated in Chapter 3, we know that cardinality query answering can be performed in 2EXP. A closer look at the size of optimal models exhibited in Section 3.4, coupled with the trivial bound on the size of cardinality queries, containing at most one atom with two variables, allows us to refine this upper bound as follows.

Theorem 39. *Role cardinality query answering in $\mathcal{ALCHI}$ is in* coNEXP *w.r.t. combined complexity.*

Proof. Theorem 8 from Chapter 3 proves that the minimal number of matches is reached with a model of exponential size. □

In the next section, we prove that this coNEXP upper bound is optimal for extensions of either $\mathcal{ELI}_\bot$ or $\mathcal{ALC}$, but can be refined *via* connections to closed predicates for some of the other considered sublogics of $\mathcal{ALCHI}$.

5.2 Combined complexity and closed predicates

5.2.1 Extensions of $\mathcal{EL}$

The next two results, together with Theorem 38, establish that cardinality query answering is coNEXP-complete w.r.t. combined complexity in extensions of $\mathcal{ELI}_\bot$ and $\mathcal{ALC}$. The proof relies upon the existence of KBs that only admit exponentially large models. These are easily obtained through a combination of atomic concepts, disjointness axioms, and a feature allowing to propagate these atomic concepts forward along roles. This is performed by existential-restrictions involving inverse roles in $\mathcal{ELI}_\bot$ and by universal restrictions in $\mathcal{ALC}$.

Theorem 40. *Concept cardinality query answering in $\mathcal{ELI}_\bot$ and in $\mathcal{ALC}$ is* coNEXP*-hard w.r.t. combined complexity.*

Proof. This proof focuses on the case of $\mathcal{ELI}_\bot$ and involves inverse roles only in axioms of the form $\exists R^-.A \sqsubseteq B$. Therefore, the proof for $\mathcal{ALC}$ can be obtained by replacing each such $\mathcal{ELI}_\bot$ axiom $\exists R^-.A \sqsubseteq B$ by the $\mathcal{ALC}$ axiom $A \sqsubseteq \forall R.B$. In particular, one doesn't need the full expressive power of $\mathcal{ALC}$.

The proof proceeds by reduction from the complement of the NEXP-complete Succinct-3-Col problem. An instance of Succinct-3-Col consists of a Boolean circuit $\mathcal{C}$ with $2n$ input gates. The graph $\mathcal{G}_\mathcal{C}$ encoded by $\mathcal{C}$ has 2^n vertices, identified by binary encodings on n bits. Two vertices u and v, with respective binary encodings $u_1 \dots u_n$ and $v_1 \dots v_n$, are adjacent in $\mathcal{G}_\mathcal{C}$ iff $\mathcal{C}$ returns True when given as input $u_1 \dots u_n$ on its first n gates and $v_1 \dots v_n$ on the second half. The problem of deciding if $\mathcal{G}_\mathcal{C}$ is 3-colorable has been proven to be NEXP-complete in Papadimitriou and Yannakakis [1986].

Let $\mathcal{C}$ be an instance of Succinct-3-Col having $2n$ input gates. We start by generating an exponential tree, henceforth referred to as the *reference tree*, to assign a color to each vertex, that is a binary identifier (k ranges from 1 to n):

$$\mathrm{U_0}(\mathsf{a}) \quad \begin{matrix} \mathrm{U_{k-1}} \sqsubseteq \exists \mathrm{R}.\mathrm{A_k^0} & \mathrm{A_k^0} \sqsubseteq \mathrm{U_k} & \exists \mathrm{R}^-.\mathrm{A_k^0} \sqsubseteq \mathrm{A_k^0} \\ \mathrm{U_{k-1}} \sqsubseteq \exists \mathrm{R}.\mathrm{A_k^1} & \mathrm{A_k^1} \sqsubseteq \mathrm{U_k} & \exists \mathrm{R}^-.\mathrm{A_k^1} \sqsubseteq \mathrm{A_k^1} \end{matrix} \quad \mathrm{A_k^0} \sqcap \mathrm{A_k^1} \sqsubseteq \bot$$

At the end of a branch, we ask for a color to be chosen among three provided options. The color can actually be chosen elsewhere, but at the cost of a new c-match for our query q_{Goal}.

$$\mathrm{U_n} \sqsubseteq \exists \mathrm{HasCol.Color} \quad \mathrm{Color} \sqsubseteq \mathrm{Goal} \quad \mathrm{Color}(\mathsf{c_1}) \quad \mathrm{Color}(\mathsf{c_2}) \quad \mathrm{Color}(\mathsf{c_3})$$

We now generate all possible pairs of vertex identifiers, starting from the first identifier (k ranges from 1 to n):

$$\mathrm{V_0}(\mathsf{b}) \quad \begin{matrix} \mathrm{V_{k-1}} \sqsubseteq \exists \mathrm{R}.\mathrm{B_k^0} & \mathrm{B_k^0} \sqsubseteq \mathrm{V_k} & \exists \mathrm{R}^-.\mathrm{B_k^0} \sqsubseteq \mathrm{B_k^0} \\ \mathrm{V_{k-1}} \sqsubseteq \exists \mathrm{R}.\mathrm{B_k^1} & \mathrm{B_k^1} \sqsubseteq \mathrm{V_k} & \exists \mathrm{R}^-.\mathrm{B_k^1} \sqsubseteq \mathrm{B_k^1} \end{matrix} \quad \mathrm{B_k^0} \sqcap \mathrm{B_k^1} \sqsubseteq \bot$$

and followed by the second identifier (k ranges from 1 to n):

$$\mathrm{V_n} \sqsubseteq \mathrm{W_0} \quad \begin{matrix} \mathrm{W_{k-1}} \sqsubseteq \exists \mathrm{R}.\mathrm{C_k^0} & \mathrm{C_k^0} \sqsubseteq \mathrm{W_k} & \exists \mathrm{R}^-.\mathrm{C_k^0} \sqsubseteq \mathrm{C_k^0} \\ \mathrm{W_{k-1}} \sqsubseteq \exists \mathrm{R}.\mathrm{C_k^1} & \mathrm{C_k^1} \sqsubseteq \mathrm{W_k} & \exists \mathrm{R}^-.\mathrm{C_k^1} \sqsubseteq \mathrm{C_k^1} \end{matrix} \quad \mathrm{C_k^0} \sqcap \mathrm{C_k^1} \sqsubseteq \bot$$

At the end of a branch, we ask for each node to be connected to the two corresponding nodes from the reference tree.

$$\begin{matrix} \mathrm{W_n} \sqsubseteq \exists \mathrm{Fst.Goal} & \exists \mathrm{Fst}^-.\mathrm{B_k^0} \sqsubseteq \mathrm{A_k^0} & \exists \mathrm{Snd}^-.\mathrm{C_k^0} \sqsubseteq \mathrm{A_k^0} & \mathrm{U_n} \sqsubseteq \mathrm{Goal} \\ \mathrm{W_n} \sqsubseteq \exists \mathrm{Snd.Goal} & \exists \mathrm{Fst}^-.\mathrm{B_k^1} \sqsubseteq \mathrm{A_k^1} & \exists \mathrm{Snd}^-.\mathrm{C_k^1} \sqsubseteq \mathrm{A_k^1} & \mathrm{U_n} \sqcap \mathrm{Color} \sqsubseteq \bot \end{matrix}$$

Notice axioms $\mathrm{U_n} \sqsubseteq \mathrm{Goal}$ and $\mathrm{U_n} \sqcap \mathrm{Color} \sqsubseteq \bot$ act as an incentive to reuse elements from the reference tree, otherwise it would come at the cost of a new c-match for our query q_{Goal}. We also note that, at this point, there are always at least $2^n + 3$ matches in every model given by the three possible colors $\mathsf{c}_1, \mathsf{c}_2, \mathsf{c}_3$ and the 2^n instances of $\mathrm{U_n}$, which must all be disjoint. Finally, we import the chosen colors from the reference tree with the following assertions and axioms:

$$\begin{array}{lll} \mathrm{Col_1}(\mathsf{c}_1) & \exists\mathrm{Fst}.(\exists\mathrm{HasCol}.\mathrm{Col_1}) \sqsubseteq \mathrm{Col_1^{fst}} & \exists\mathrm{Snd}.(\exists\mathrm{HasCol}.\mathrm{Col_1}) \sqsubseteq \mathrm{Col_1^{snd}} \\ \mathrm{Col_2}(\mathsf{c}_2) & \exists\mathrm{Fst}.(\exists\mathrm{HasCol}.\mathrm{Col_2}) \sqsubseteq \mathrm{Col_2^{fst}} & \exists\mathrm{Snd}.(\exists\mathrm{HasCol}.\mathrm{Col_2}) \sqsubseteq \mathrm{Col_2^{snd}} \\ \mathrm{Col_3}(\mathsf{c}_3) & \exists\mathrm{Fst}.(\exists\mathrm{HasCol}.\mathrm{Col_2}) \sqsubseteq \mathrm{Col_3^{fst}} & \exists\mathrm{Snd}.(\exists\mathrm{HasCol}.\mathrm{Col_3}) \sqsubseteq \mathrm{Col_3^{snd}} \end{array}$$

It remains to evaluate the circuit to test adjacency for each pair of vertex identifiers. This is handled by the TBox in the following fashion. For the first n input gates g_k^{fst} introduce the axioms:

$$\mathrm{B_k^0} \sqsubseteq \mathrm{IsFalse}_{\mathrm{g_k^{fst}}} \qquad \mathrm{B_k^1} \sqsubseteq \mathrm{IsTrue}_{\mathrm{g_k^{fst}}} \qquad (k = 1, \dots, n)$$

and for the remaining n input gates g_k^{snd} introduce the axioms:

$$\mathrm{C_k^0} \sqsubseteq \mathrm{IsFalse}_{\mathrm{g_k^{snd}}} \qquad \mathrm{C_k^1} \sqsubseteq \mathrm{IsTrue}_{\mathrm{g_k^{snd}}} \qquad (k = 1, \dots, n).$$

For each negation gate g with parent gate g_0, we introduce the two axioms:

$$\mathrm{IsFalse}_{\mathrm{g_0}} \sqsubseteq \mathrm{IsTrue}_{\mathrm{g}} \qquad \mathrm{IsTrue}_{\mathrm{g_0}} \sqsubseteq \mathrm{IsFalse}_{\mathrm{g}}.$$

For each conjunctive gate g with parent gates g_1 and g_2, introduce the three axioms:

$$\mathrm{IsTrue}_{\mathrm{g_1}} \sqcap \mathrm{IsTrue}_{\mathrm{g_2}} \sqsubseteq \mathrm{IsTrue}_{\mathrm{g}} \qquad \begin{array}{l} \mathrm{IsFalse}_{\mathrm{g_1}} \sqsubseteq \mathrm{IsFalse}_{\mathrm{g}} \\ \mathrm{IsFalse}_{\mathrm{g_2}} \sqsubseteq \mathrm{IsFalse}_{\mathrm{g}}. \end{array}$$

For each disjunctive gate g with parent gates g_1 and g_2, introduce the three axioms:

$$\begin{array}{l} \mathrm{IsTrue}_{\mathrm{g_1}} \sqsubseteq \mathrm{IsTrue}_{\mathrm{g}} \\ \mathrm{IsTrue}_{\mathrm{g_2}} \sqsubseteq \mathrm{IsTrue}_{\mathrm{g}} \end{array} \qquad \mathrm{IsFalse}_{\mathrm{g_1}} \sqcap \mathrm{IsFalse}_{\mathrm{g_2}} \sqsubseteq \mathrm{IsFalse}_{\mathrm{g}}.$$

Finally, to detect monochromatic edges, consider the three axioms where g_{out} denotes the output gate of $\mathcal{C}$:

$$\begin{array}{c} \mathrm{IsTrue}_{\mathrm{g_{out}}} \sqcap \mathrm{Col_1^{fst}} \sqcap \mathrm{Col_1^{snd}} \sqsubseteq \mathrm{Goal} \\ \mathrm{IsTrue}_{\mathrm{g_{out}}} \sqcap \mathrm{Col_2^{fst}} \sqcap \mathrm{Col_2^{snd}} \sqsubseteq \mathrm{Goal} \\ \mathrm{IsTrue}_{\mathrm{g_{out}}} \sqcap \mathrm{Col_3^{fst}} \sqcap \mathrm{Col_3^{snd}} \sqsubseteq \mathrm{Goal} \end{array}$$

To ensure this case indeed creates a new match for q_{Goal} we make sure that it cannot be an already existing match with the two negative concept inclusions:

$$\mathrm{W_n} \sqcap \mathrm{Color} \sqsubseteq \bot \qquad \mathrm{U_n} \sqcap \mathrm{W_n} \sqsubseteq \bot$$

Claim: $\mathcal{C} \notin$ SUCCINCT-3-COL iff $2^n + 4$ is a certain answer for q_{Goal} over $\mathcal{K}$.

Note: Both the constructed $\mathcal{ELI}_\bot$ and $\mathcal{ALC}$ KBs admit a canonical model, hence we allow ourselves to refer to the canonical model in what follows. For readability, we omit the concepts associated with the evaluation of the circuit when considering elements of $\mathcal{C}_\mathcal{K}$.

($\Longrightarrow$). Assume $\mathcal{C} \notin$ SUCCINCT-3-COL and consider a model $\mathcal{I}$ of $\mathcal{K}$. There exists a homomorphism from the canonical model of $\mathcal{K}$ to this $\mathcal{I}$, say we choose one such $f : \mathcal{C}_\mathcal{K} \to \mathcal{I}$.

If any of the $f(\mathsf{a} \cdots \text{R}.\{\text{U}_\text{n}, \text{Goal}, \text{A}_1^{\text{a}_1}, \dots \text{A}_\text{n}^{\text{a}_\text{n}}\} \cdot \text{HasCol}.\{\text{Color}, \text{Goal}\})$ does not belong to $\{\mathsf{c}_1, \mathsf{c}_2, \mathsf{c}_3\}$, then it provides a new c-match for q_{Goal} and we are done.

Otherwise, denote by τ the coloring induced by the reference tree in $\mathcal{I}$, defined by setting:

$$\tau(a_1 \dots a_n) := f(\mathsf{a} \cdots \text{R}.\{\text{U}_\text{n}, \text{Goal}, \text{A}_1^{\text{a}_1}, \dots \text{A}_\text{n}^{\text{a}_\text{n}}\} \cdot \text{HasCol}.\{\text{Color}, \text{Goal}\}).$$

Since $\mathcal{C} \notin$ SUCCINCT-3-COL and τ only uses the 3 colors c_1, c_2 and c_3, there must exist a monochromatic edge $\{u, v\}$. Denote by $b_1, \dots, b_n$ the identifier of u, by $c_1, \dots, c_n$ the identifier of v, and by k the number of the shared color c_k. Since u and v are adjacent, the concept $\text{IsTrue}_{\text{g}_\text{out}}$ is satisfied on the element $e := f(\mathsf{b} \cdots \text{R}.\{\text{W}_\text{n}, \text{B}_1^{\text{b}_1}, \dots \text{B}_\text{n}^{\text{b}_\text{n}}, \text{C}_1^{\text{c}_1}, \dots \text{C}_\text{n}^{\text{c}_\text{n}}\})$ of $\mathcal{I}$.

If $f(\mathsf{b} \cdots \text{R}.\{\text{W}_\text{n}, \text{B}_1^{\text{b}_1}, \dots \text{B}_\text{n}^{\text{b}_\text{n}}, \text{C}_1^{\text{c}_1}, \dots \text{C}_\text{n}^{\text{c}_\text{n}}\} \cdot \text{Fst}.\{\text{U}_\text{n}, \text{Goal}, \text{A}_1^{\text{b}_1}, \dots, \text{A}_\text{n}^{\text{b}_\text{n}}\})$ (notice the first vertex identifier is converted into an identifier in the reference tree) is not equal to $f(\mathsf{a} \cdots \text{R}.\{\text{U}_\text{n}, \text{Goal}, \text{A}_1^{\text{b}_1}, \dots \text{A}_\text{n}^{\text{b}_\text{n}}\})$ being the corresponding element from the reference tree, then it yields a new c-match and we are done.

Otherwise, axiom $\exists\text{Fst}.(\exists\text{HasCol}.\text{Col}_\text{k}) \sqsubseteq \text{Col}_\text{k}^{\text{fst}}$ ensures $\text{C}_\text{k}^{\text{fst}}$ holds on e. Similarly, $f(\mathsf{b} \cdots \text{R}.\{\text{W}_\text{n}, \text{B}_1^{\text{b}_1}, \dots \text{B}_\text{n}^{\text{b}_\text{n}}, \text{C}_1^{\text{c}_1}, \dots \text{C}_\text{n}^{\text{c}_\text{n}}\} \cdot \text{Snd}.\{\text{U}_\text{n}, \text{Goal}, \text{A}_1^{\text{c}_1}, \dots, \text{A}_\text{n}^{\text{c}_\text{n}}\})$ either yields a new c-match, in which case we are done, or $\text{C}_\text{k}^{\text{snd}}$ holds on e. In the latter case, axiom $\text{IsTrue}_{\text{g}_\text{out}} \sqcap \text{Col}_\text{k}^{\text{fst}} \sqcap \text{Col}_\text{k}^{\text{snd}} \sqsubseteq \text{Goal}$ triggers a new c-match on e.

In all cases, we exhibit an additional c-match, which proves $2^n + 4$ is a certain answer for q_{Goal} over $\mathcal{K}$.

($\Longleftarrow$). Assume $\mathcal{C} \in$ SUCCINCT-3-COL and pick a 3-coloring τ of the underlying graph of $\mathcal{C}$, using as colors c_1, c_2 and c_3. From the canonical model of $\mathcal{K}$, identify each element of the form $\mathsf{a} \cdots \text{R}.\{\text{U}_\text{n}, \text{Goal}, \text{A}_1^{\text{a}_1}, \dots \text{A}_\text{n}^{\text{a}_\text{n}}\} \cdot \text{HasCol}.\{\text{Color}, \text{Goal}\}$ with the individual $\tau(a_1 \dots a_n)$. Additionally, identify each element of the form $\mathsf{b} \cdots \text{R}.\{\text{W}_\text{n}, \text{B}_1^{\text{b}_1}, \dots \text{B}_\text{n}^{\text{b}_\text{n}}, \text{C}_1^{\text{c}_1}, \dots \text{C}_\text{n}^{\text{c}_\text{n}}\} \cdot \text{Fst}.\{\text{U}_\text{n}, \text{Goal}, \text{A}_1^{\text{b}_1}, \dots, \text{A}_\text{n}^{\text{b}_\text{n}}\}$ with the element $\mathsf{a} \cdots \text{R}.\{\text{U}_\text{n}, \text{Goal}, \text{A}_1^{\text{b}_1}, \dots \text{A}_\text{n}^{\text{b}_\text{n}}\}$, and similarly identify each element of the form $\mathsf{b} \cdots \text{R}.\{\text{W}_\text{n}, \text{B}_1^{\text{b}_1}, \dots \text{B}_\text{n}^{\text{b}_\text{n}}, \text{C}_1^{\text{c}_1}, \dots \text{C}_\text{n}^{\text{c}_\text{n}}\} \cdot \text{Snd}.\{\text{U}_\text{n}, \text{Goal}, \text{A}_1^{\text{c}_1}, \dots, \text{A}_\text{n}^{\text{c}_\text{n}}\}$ with the element $\mathsf{a} \cdots \text{R}.\{\text{U}_\text{n}, \text{Goal}, \text{A}_1^{\text{c}_1}, \dots \text{A}_\text{n}^{\text{c}_\text{n}}\}$.

Saturate the obtained interpretation to obtain a model $\mathcal{I}_\tau$ of $\mathcal{K}$. Because τ is a 3-coloring, there is no monochromatic edge, hence it can be verified that $\mathcal{I}_\tau$ has exactly the $2^n + 3$ original c-matches. This provides a model of $\mathcal{K}$ with less than $2^n + 4$ c-matches for q_{Goal}, ensuring $2^n + 4$ is not a certain answer for q_{Goal} over $\mathcal{K}$. □

As already mentioned, the latter coNEXP-hardness proof relies on KBs that only admit exponentially large models. We now turn to $\mathcal{ALCHI}$ KBs admitting polysize models. The key observation is that, for logics with polysize models and single-atom queries, the optimal number of matches is bounded polynomially in the size of the KB. We can thus iterate over all polynomial-sized ABoxes that could represent the restriction of an optimal model to the ABox and elements in matches. We test whether such an ABox extends to a model without new matches by performing a satisfiability check, taking the query role as closed predicate. This gives a deterministic single-exponential time procedure, since satisfiability of $\mathcal{ALCHI}$ KBs with closed predicates is in EXP, as proven in Ngo et al. [2016].

Theorem 41. *Let $\mathcal{L}$ be a subclass of $\mathcal{ALCHI}$ KBs for which every satisfiable KB admits a polynomial-sized model. Then role cardinality query answering over $\mathcal{L}$ KBs is in* EXP*.*

Proof. Let $\mathcal{L}$ be a sublogic of $\mathcal{ALCHI}$ for which every satisfiable KB admits a polynomial-sized model. Then proceeding similarly to Lemma 3, we can exhibit a polynomial p such that for every satisfiable KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ and cardinality query q, there exists a model of $\mathcal{K}$ having at most $p(|\mathcal{K}|)$ matches to q.

With this in mind, let us fix a satisfiable KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ and a role cardinality query $q_{\text{Goal}} = \exists z_1 \exists z_2 \, \text{Goal}(z_1, z_2)$, and let $n_\mathcal{K} = p(|\mathcal{K}|)$. Consider a set of individual names $D \subseteq \mathsf{N_I}$ of size $2n_\mathcal{K} + |\mathsf{Ind}(\mathcal{A})|$ and containing $\mathsf{Ind}(\mathcal{A})$. For each subset $S \subseteq D \times D$, we check whether the following KB with closed predicates is satisfiable (note that Goal is the only closed predicate):

$$\mathcal{K}_S := (\mathcal{T}, \{\text{Goal}\}, \mathcal{A} \cup \{\text{Goal}(\mathsf{a}, \mathsf{b}) \mid (\mathsf{a}, \mathsf{b}) \in S\})$$

If such a KB is satisfiable with Goal a closed predicate, it provides a model of $\mathcal{K}$ with precisely $|S|$ matches. Conversely, if there exists a model $\mathcal{I}$ of $\mathcal{K}$ with $n \leq n_\mathcal{K}$ matches, there exists a subset $S \subseteq D \times D$ such that $\mathcal{K}_S$ is satisfiable: pick S as the pairs $(\varphi(\mathsf{a}), \varphi(\mathsf{b})) \in \text{Goal}^\mathcal{I}$, where φ is an injection from the subset of $\Delta^\mathcal{I}$ appearing in matches of q_{Goal} to D which is the identity on $\mathsf{Ind}(\mathcal{A})$.

By Theorem 7 of Ngo et al. [2016], this check can be performed in exponential time in $\mathcal{K}_S$, which is of polynomial size w.r.t. $\mathcal{K}$. □

Corollary 6. *Role cardinality query answering in $\mathcal{ELH}_\bot$ is in* EXP *w.r.t. combined complexity.*

Proof. Let $\mathcal{K}$ be a satisfiable $\mathcal{ELH}_\bot$ KB, which we may suppose w.l.o.g. to be in normal form, and consider the following interpretation $\mathcal{I}_\mathcal{K}$ (a variation on the one defined in Lutz et al. [2009] for ${\mathcal{ELH}_\bot}^{dr}$ without negative role inclusions):

$$\begin{aligned}
\Delta^{\mathcal{I}_\mathcal{K}} =& \mathsf{Ind}(\mathcal{A}) \cup \{x_{\mathrm{R.B}} \mid \mathrm{A} \sqsubseteq \exists \mathrm{R.B} \in \mathcal{T} \text{ and } \mathcal{T} \not\models B \sqsubseteq \bot\} \\
\mathrm{A}^{\mathcal{I}_\mathcal{K}} =& \{\mathsf{a} \mid \mathcal{K} \models \mathrm{A}(\mathsf{a})\} \cup \{x_{\mathrm{R.B}} \mid \mathcal{T} \models \mathrm{B} \sqsubseteq \mathrm{A}\} \\
\mathrm{P}^{\mathcal{I}_\mathcal{K}} =& \{(\mathsf{a},\mathsf{b}) \mid \mathcal{K} \models \mathrm{P}(\mathsf{a},\mathsf{b})\} \cup \{(\mathsf{a}, x_{\mathrm{R.B}}) \mid \mathcal{K} \models \exists \mathrm{R.B}(\mathsf{a}), \mathcal{T} \models \mathrm{R} \sqsubseteq \mathrm{P}\} \cup \\
& \{(x_{\mathrm{R_1.B_1}}, x_{\mathrm{R_2.B_2}}) \mid \mathcal{T} \models \mathrm{B_1} \sqsubseteq \exists \mathrm{R_2.B_2}, \mathcal{T} \models \mathrm{R_2} \sqsubseteq \mathrm{P}\}
\end{aligned}$$

Note that $|\Delta^{\mathcal{I}_\mathcal{K}}| \leq |\mathcal{K}|$, so we only need to show that $\mathcal{I}_\mathcal{K}$ is a model of $\mathcal{K}$. It is not hard to see that $\mathcal{I}_\mathcal{K}$ satisfies ABox assertions of $\mathcal{A}$ and all concept axioms and positive role inclusions from $\mathcal{T}$. Suppose that $\mathcal{T}$ contains a negative role inclusion $\mathrm{T_1} \sqcap \mathrm{T_2} \sqsubseteq \bot$ and there is a pair $(u,v) \in {\mathrm{T_1}}^{\mathcal{I}_\mathcal{K}} \cap {\mathrm{T_2}}^{\mathcal{I}_\mathcal{K}}$. We cannot have $u, v \in \mathsf{Ind}(\mathcal{A})$, since this would imply that $\mathcal{K}$ is unsatisfiable. If $(u,v) = (\mathsf{a}, x_{\mathrm{R.B}})$, then $\mathcal{K} \models \exists \mathrm{R.B}(\mathsf{a})$, $\mathcal{T} \models \mathrm{R} \sqsubseteq \mathrm{T_1}$, and $\mathcal{T} \models \mathrm{R} \sqsubseteq \mathrm{T_2}$, which again means $\mathcal{K}$ is unsatisfiable. Finally suppose that we have $(u,v) = (x_{\mathrm{R_1.B_1}}, x_{\mathrm{R_2.B_2}})$. Then $\mathcal{T} \models \mathrm{B_1} \sqsubseteq \exists \mathrm{R_2.B_2}$, $\mathcal{T} \models \mathcal{T} \models \mathrm{R_2} \sqsubseteq \mathrm{T_1}$, and $\mathcal{T} \models \mathrm{R_2} \sqsubseteq \mathrm{T_2}$. But that would mean that $\mathcal{T} \models \mathrm{B_1} \sqsubseteq \bot$, contradicting the definition of $\Delta^{\mathcal{I}_\mathcal{K}}$. We thus conclude that $\mathcal{I}_\mathcal{K}$ is indeed a model of $\mathcal{K}$. □

Corollary 7. *Role cardinality query answering in $\mathcal{ELHI}$ is in* EXP *w.r.t. combined complexity.*

Proof. Existence of polynomial-sized models is trivial due to the absence of negative inclusions. For example, extending $\mathcal{A}$ with every possible fact constructed from $\mathsf{Ind}(\mathcal{A})$ and $\mathsf{sig}(\mathcal{K})$ yields a model of $\mathcal{K} = (\mathcal{T}, \mathcal{A})$. □

We conclude this subsection by providing matching lower bounds for concept cardinality queries in $\mathcal{EL}$, which completes the complexity landscape for cardinality queries answering over $\mathcal{ALCHI}$ KBs extending $\mathcal{EL}$.

Theorem 42. *Concept cardinality query answering in $\mathcal{EL}$ is* EXP*-hard w.r.t. combined complexity.*

Proof. The proof proceeds by reduction from the problem of deciding if an $\mathcal{EL}$ KB with closed predicates is satisfiable, known to be EXP-hard from Ngo et al. [2016]. As noticed by the authors in the conclusion of Ngo et al. [2016], we point out that their reduction (Propositions 4 and 5) produces a KB $(\mathcal{T}, \Sigma, \mathcal{A})$ such that the set of closed predicates Σ only contains concept names. Therefore, we assume w.l.o.g. that our starting KB $\mathcal{K} := (\mathcal{T}, \Sigma, \mathcal{A})$ also satisfies this property. Restricting the construction of $\mathcal{ALCHI}$ normal form (see Section 2.1.3) to $\mathcal{EL}$ axioms, we reduce to the case in which every concept inclusion has one of the following restricted shapes:

$$\top \sqsubseteq \mathrm{A} \quad \mathrm{A} \sqcap \mathrm{B} \sqsubseteq \mathrm{C} \quad \mathrm{A} \sqsubseteq \exists \mathrm{R}.\mathrm{B} \quad \exists \mathrm{R}.\mathrm{A} \sqsubseteq \mathrm{B} \qquad \text{with } \mathrm{A}, \mathrm{B}, \mathrm{C} \in \mathsf{N_C}, \mathrm{R} \in \mathsf{N_R}.$$

It can be verified that such a normalization procedure doesn't affect the satisfiability w.r.t. the closed predicates.

We will need to consider two fresh new concept names Goal and $\mathrm{Aux}_\top$, a fresh new role name $\mathrm{R_B}$ for each closed concept name $\mathrm{B} \in \Sigma$, and a fresh individual aux. The concept Goal will be our query predicate and aims to capture excessive uses of the closed predicates.

To capture such uses on non-individual elements, we consider the axiom $\mathrm{B} \sqsubseteq \mathrm{Goal}$ for each $\mathrm{B} \in \Sigma$. Therefore, we also consider the assertion $\mathrm{Goal}(\mathsf{a})$ for each a such that there exists $\mathrm{B}(\mathsf{a}) \in \mathcal{A}$ with $\mathrm{B} \in \Sigma$. To prevent such an assertion $\mathrm{Goal}(\mathsf{a})$ from "hiding" the use of a by a closed concept B such that $\mathrm{B}(\mathsf{a}) \notin \mathcal{A}$, we introduce the axiom $\exists \mathrm{R_B}.\mathrm{B} \sqsubseteq \mathrm{Goal}$ for each $\mathrm{B} \in \Sigma$ and the assertion $\mathrm{R_B}(\mathsf{aux}, \mathsf{a})$ for each $\mathsf{a} \in \mathsf{Ind}(\mathcal{A})$ and each $\mathrm{B} \in \Sigma$ such that $\mathrm{B}(\mathsf{a}) \notin \mathcal{A}$.

Adding such a new individual aux may cause axioms with shape $\top \sqsubseteq \mathrm{A}$ from $\mathcal{T}$ to trigger on aux hence requiring further concepts to hold on aux. To prevent this, we replace each axiom $\top \sqsubseteq \mathrm{A}$ from $\mathcal{T}$ by $\mathrm{Aux}_\top \sqsubseteq \mathrm{A}$, we also add the axiom $\mathrm{A} \sqsubseteq \mathrm{Aux}_\top$ for each $\mathrm{A} \in \mathsf{sig}(\mathcal{T})$ and the assertion $\mathrm{Aux}_\top(\mathsf{a})$ for each $\mathsf{a} \in \mathsf{Ind}(\mathcal{A})$.

To summarize, we define $\mathcal{T}'$ and $\mathcal{A}'$ as follows:

$$\begin{aligned} \mathcal{T}' := {} & (\mathcal{T} \setminus \{\top \sqsubseteq \mathrm{A} \mid \top \sqsubseteq \mathrm{A} \in \mathcal{T}\}) \\ & \cup \{\mathrm{Aux}_\top \sqsubseteq \mathrm{A} \mid \top \sqsubseteq \mathrm{A} \in \mathcal{T}\} \\ & \cup \{\mathrm{A} \sqsubseteq \mathrm{Aux}_\top \mid \mathrm{A} \in \mathsf{sig}(\mathcal{T})\} \\ & \cup \{\mathrm{B} \sqsubseteq \mathrm{Goal} \mid \mathrm{B} \in \Sigma\} \\ & \cup \{\exists \mathrm{R_B}.\mathrm{B} \sqsubseteq \mathrm{Goal} \mid \mathrm{B} \in \Sigma\} \end{aligned} \qquad \begin{aligned} \mathcal{A}' := {} & \mathcal{A} \\ & \cup \{\mathrm{Aux}(\mathsf{a}) \mid \mathsf{a} \in \mathsf{Ind}(\mathcal{A})\} \\ & \cup \{\mathrm{R_B}(\mathsf{aux}, \mathsf{a}) \mid \mathrm{B}(\mathsf{a}) \notin \mathcal{A}, \mathrm{B} \in \Sigma\} \\ & \cup \{\mathrm{Goal}(\mathsf{a}) \mid \mathrm{B}(\mathsf{a}) \in \mathcal{A}, \mathrm{B} \in \Sigma\} \end{aligned}$$

Finally, let $n := |\{\mathrm{Goal}(\mathsf{a}) \mid \mathrm{B}(\mathsf{a}) \in \mathcal{A}, \mathrm{B} \in \Sigma\}|$ be the number the of ABox matches for q_{Goal} in $(\mathcal{T}', \mathcal{A}')$. To complete the proof, we establish the following claim.

$(\mathcal{T}, \Sigma, \mathcal{A})$ is satisfiable iff $n + 1$ is not a certain answer for q_{Goal} over $(\mathcal{T}', \mathcal{A}')$.

($\Rightarrow$). Assume $(\mathcal{T}, \Sigma, \mathcal{A})$ is satisfiable and let $\mathcal{I}$ be one of its models. We build an interpretation $\mathcal{I}'$ of $(\mathcal{T}', \mathcal{A}')$ with domain $\Delta^{\mathcal{I}'} := \Delta^{\mathcal{I}} \cup \{\mathsf{aux}\}$ as follows:

$$\begin{aligned} \mathrm{A}^{\mathcal{I}'} &:= \mathrm{A}^{\mathcal{I}} && (\mathrm{A} \in \mathsf{sig}(\mathcal{T})) \\ \mathrm{Goal}^{\mathcal{I}'} &:= \{\mathrm{Goal}(\mathsf{a}) \mid \mathrm{B}(\mathsf{a}) \in \mathcal{A}, \mathrm{B} \in \Sigma\} \\ \mathrm{Aux}_\top{}^{\mathcal{I}'} &:= \Delta^{\mathcal{I}} \\ \mathrm{P}^{\mathcal{I}'} &:= \mathrm{P}^{\mathcal{I}} && (\mathrm{P} \in \mathsf{sig}(\mathcal{T})) \\ \mathrm{R_B}{}^{\mathcal{I}'} &:= \{\mathrm{R_B}(\mathsf{aux}, \mathsf{a}) \mid \mathrm{B}(\mathsf{a}) \notin \mathcal{A}, \mathrm{B} \in \Sigma\} \end{aligned}$$

Clearly, $\mathcal{I}'$ has exactly n matches for q_{Goal}. We verify it is a model of $(\mathcal{T}', \mathcal{A}')$, concluding this part of the proof as $\mathcal{I}'$ is a counter-model for $n+1$. All axioms from $\mathcal{T}$ are trivially satisfied as interpretations of concept and roles names from $\mathsf{sig}(\mathcal{T})$ are preserved (recall those with shape $\top \sqsubseteq \mathrm{A}$ have been removed!). Assertions in $\mathcal{A}'$ are also trivially satisfied, either by definition We check the other axioms in turn:

$\mathrm{Aux}_\top \sqsubseteq \mathrm{A}$ ($\top \sqsubseteq \mathrm{A} \in \mathcal{T}$). Since $\mathcal{I}$ is a model of $\mathcal{T}$, we obtain: ${\mathrm{Aux}_\top}^{\mathcal{I}'} = \Delta^{\mathcal{I}} = \top^{\mathcal{I}} \subseteq \mathrm{A}^{\mathcal{I}} = \mathrm{A}^{\mathcal{I}'}$.

$\mathrm{A} \sqsubseteq \mathrm{Aux}_\top$ ($\mathrm{A} \in \mathsf{sig}(\mathcal{T})$). Trivial: $\mathrm{A}^{\mathcal{I}'} = \mathrm{A}^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} = {\mathrm{Aux}_\top}^{\mathcal{I}'}$.

$\mathrm{B} \sqsubseteq \mathrm{Goal}$ ($\mathrm{B} \in \Sigma$). Let $e \in \mathrm{B}^{\mathcal{I}'}$. We have $\mathrm{B} \in \Sigma \subseteq \mathsf{sig}(\mathcal{T})$, hence by definition $e \in \mathrm{B}^{\mathcal{I}}$. Since $\mathrm{B} \in \Sigma$ and $\mathcal{I}$ is a model of $\mathcal{K}$, it follows that $\mathrm{B}(e) \in \mathcal{A}$. Hence, by definition: $e \in \mathrm{Goal}^{\mathcal{I}'}$.

$\exists \mathrm{R}_\mathrm{B}.\mathrm{B} \sqsubseteq \mathrm{Goal}$ ($\mathrm{B} \in \Sigma$). Let $e \in (\exists \mathrm{R}_\mathrm{B}.\mathrm{B})^{\mathcal{I}'}$. We hence have an individual $\mathsf{a} \in \mathrm{B}^{\mathcal{I}'}$ such that $\mathrm{B}(\mathsf{a}) \notin \mathcal{A}$ (from the definition of ${\mathrm{R}_\mathrm{B}}^{\mathcal{I}'}$). From the definition of $\mathrm{B}^{\mathcal{I}'}$, we obtain $\mathsf{a} \in \mathrm{B}^{\mathcal{I}}$, which implies, as $\mathcal{I}$ is a model of $\mathcal{K}$, that $\mathrm{B}(\mathsf{a}) \in \mathcal{A}$. Contradiction, hence $(\exists \mathrm{R}_\mathrm{B}.\mathrm{B})^{\mathcal{I}'} = \emptyset$ and the axiom is trivially satisfied.

($\Leftarrow$). Assume $n+1$ is not a certain answer, that is, we have a counter-model $\mathcal{I}$ (in which matches are exactly the n ABox matches). Consider the interpretation $\mathcal{I}'$ obtained by restricting $\mathcal{I}$ to the domain $\Delta^{\mathcal{I}'} := (\mathrm{Aux}_\top)^{\mathcal{I}}$.

Axioms from $\mathcal{A}$ are clearly satisfied in $\mathcal{I}'$ as $\mathcal{A} \subseteq \mathcal{A}'$. We verify that axioms from $\mathcal{T}$ also hold:

$\top \sqsubseteq \mathrm{A}$. In particular $\mathrm{Aux}_\top \sqsubseteq \mathrm{A} \in \mathcal{T}'$. From $\mathcal{I}$ being a model of $\mathcal{T}'$, we have ${\mathrm{Aux}_\top}^{\mathcal{I}} \subseteq \mathrm{A}^{\mathcal{I}}$. Thus, $\mathrm{A}^{\mathcal{I}'} = {\mathrm{Aux}_\top}^{\mathcal{I}} \cap \mathrm{A}^{\mathcal{I}} = \mathrm{A}^{\mathcal{I}}$, which yields: $\top^{\mathcal{I}'} = {\mathrm{Aux}_\top}^{\mathcal{I}} \subseteq \mathrm{A}^{\mathcal{I}} = \mathrm{A}^{\mathcal{I}'}$.

$\mathrm{A} \sqcap \mathrm{B} \sqsubseteq \mathrm{C}$. In particular $\mathrm{A} \sqcap \mathrm{B} \sqsubseteq \mathrm{C} \in \mathcal{T}'$. Using $\mathcal{I}'$ being a model of $\mathcal{T}'$, we obtain: $(\mathrm{A} \cap \mathrm{B})^{\mathcal{I}'} = \mathrm{A}^{\mathcal{I}} \cap \mathrm{B}^{\mathcal{I}} \cap \Delta^{\mathcal{I}'} \subseteq \mathrm{C}^{\mathcal{I}} \cap \Delta^{\mathcal{I}'} = \mathrm{C}^{\mathcal{I}'}$.

$\exists \mathrm{R}.\mathrm{A} \sqsubseteq \mathrm{B}$. In particular $\exists \mathrm{R}.\mathrm{A} \sqsubseteq \mathrm{B} \in \mathcal{T}'$. First notice that $(\exists \mathrm{R}.\mathrm{A})^{\mathcal{I}'} \subseteq (\exists \mathrm{R}.\mathrm{A})^{\mathcal{I}}$ since $\mathrm{R}^{\mathcal{I}'} \subseteq \mathrm{R}^{\mathcal{I}}$ and $\mathrm{A}^{\mathcal{I}'} \subseteq \mathrm{A}^{\mathcal{I}}$. Using $\mathcal{I}'$ being a model of $\mathcal{T}'$, we now obtain: $(\exists \mathrm{R}.\mathrm{A})^{\mathcal{I}'} \subseteq (\exists \mathrm{R}.\mathrm{A})^{\mathcal{I}} \cap \Delta^{\mathcal{I}'} \subseteq \mathrm{B}^{\mathcal{I}} \cap \Delta^{\mathcal{I}'} = \mathrm{B}^{\mathcal{I}'}$.

$\mathrm{A} \sqsubseteq \exists \mathrm{R}.\mathrm{B}$. In particular, both $\mathrm{A} \sqsubseteq \exists \mathrm{R}.\mathrm{B}$ and $\mathrm{B} \sqsubseteq \mathrm{Aux}_\top$ are in $\mathcal{T}'$. Let $e \in \mathrm{A}^{\mathcal{I}'}$. In particular, $e \in \mathrm{A}^{\mathcal{I}}$. Since $\mathcal{I}$ is a model of $\mathcal{T}'$, we have some $(e, e') \in \mathrm{R}^{\mathcal{I}}$ with $e' \in \mathrm{B}^{\mathcal{I}}$. Still from $\mathcal{I}$ being a model of $\mathcal{T}'$, we also have $e' \in {\mathrm{Aux}_\top}^{\mathcal{I}}$, and therefore $b \in \Delta^{\mathcal{I}'}$. Hence $(e, e') \in \mathrm{R}^{\mathcal{I}} \cap \Delta^{\mathcal{I}'}$ and $e' \in \mathrm{B}^{\mathcal{I}} \cap \Delta^{\mathcal{I}'}$, yielding $e \in (\exists \mathrm{R}.\mathrm{B})^{\mathcal{I}'}$.

We now verify that no closed concept has been violated, which concludes the proof. Let $e \in \mathrm{B}^{\mathcal{I}'}$ for some closed concept $\mathrm{B} \in \Sigma$. In particular we have both $\mathrm{B} \sqsubseteq \mathrm{Goal}$ and $\exists \mathrm{R_B}.\mathrm{B} \sqsubseteq \mathrm{Goal}$ in $\mathcal{T}'$. By definition of $\mathrm{B}^{\mathcal{I}'}$ and from $\mathcal{I}$ being a model of $\mathcal{T}'$, we obtain $e \in \mathrm{B}^{\mathcal{I}} \subseteq \mathrm{Goal}^{\mathcal{I}}$.

From $\mathcal{I}$ being a counter-model for $n+1$, we know that $\mathrm{Goal}^{\mathcal{I}} = \{\mathrm{Goal}(\mathsf{a}) \mid \mathrm{B}(\mathsf{a}) \in \mathcal{A}, \mathrm{B} \in \Sigma\}$. In particular $\mathsf{aux} \notin \mathrm{Goal}^{\mathcal{I}}$. But since $\mathcal{I}$ is a model of $\mathcal{T}'$, it ensures that $\mathsf{aux} \notin (\exists \mathrm{R_B}.\mathrm{B})^{\mathcal{I}}$. Recall we have $\mathrm{R_B}(\mathsf{aux}, \mathsf{b}) \in \mathcal{A}'$ for all individuals b such that $\mathrm{B}(\mathsf{b}) \notin \mathcal{A}$, and therefore $\mathsf{b} \notin \mathrm{B}^{\mathcal{I}}$ for such individuals. It follows that $\mathrm{B}(e) \in \mathcal{A}$. □

5.2.2 Extensions of DL-Lite$_{\mathsf{pos}}$

We now proceed to extensions of DL-Lite$_{\mathsf{pos}}$. For DL-Lite$^{\mathcal{H}}_{\mathsf{core}}$ KBs, we proceed as for the extensions of $\mathcal{EL}$ admitting polysize models and establish a connection to closed predicates which allows to obtain coNP membership. One can indeed guess a small countermodel to $[m, +\infty]$ being a certain answer, relying on the existence of small models, atomicity of the query, and Theorem 3 of Ngo et al. [2016].

Theorem 43. *Role cardinality query answering in DL-Lite$^{\mathcal{H}}_{\mathsf{core}}$ is in* coNP *w.r.t. combined complexity.*

Proof. Let q_{P} be a role cardinality query. As DL-Lite$^{\mathcal{H}}_{\mathsf{core}}$ knowledge bases admit models of polynomial size in combined complexity, and the query is atomic, there are at most polynomially many guaranteed matches. To check if $[m, +\infty]$ is a certain answer, we can proceed as follows:

- if m is too big with respect to the polynomial bound, we reject;

- otherwise, we guess an instance $\mathcal{A}'$ containing $\mathcal{A}$ and additional matches (up to m) for q_{P}. We then check whether $(\mathcal{A}', \mathcal{T}, \{\mathrm{P}\})$ is a satisfiable knowledge base with closed predicates. According to the proof of Theorem 3 of Ngo et al. [2016], if this is the case, then there is a model of polynomial size. We guess it, and this provides a counterexample to $[m, +\infty]$ being a certain answer. □

A matching coNP lower bound can easily be obtained for concept cardinality queries as soon as concept disjointness is permitted, that is, for DL-Lite$_{\mathsf{core}}$ KBs. We point out that this holds even if the ABox is consists of a single fact.

Theorem 44. *Concept cardinality query answering in DL-Lite$_{\mathsf{core}}$ is* coNP*-hard w.r.t. combined complexity.*

Proof. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an undirected graph, and consider

$$\mathcal{T}_\mathcal{G} = \bigcup_{v \in \mathcal{V}} \{\mathrm{A} \sqsubseteq \exists \mathrm{V}, \exists \mathrm{V}^- \sqsubseteq \mathrm{C}\} \cup \bigcup_{\{v_1, v_2\} \in \mathcal{E}} \{\exists \mathrm{V}_1^- \sqsubseteq \neg \exists \mathrm{V}_2^-\}.$$

It is easily verified that $\mathcal{G} \in$ 3-Col iff $[4, +\infty] \notin q^{\mathcal{K}_\mathcal{G}}$ for the KB $\mathcal{K}_\mathcal{G} := (\mathcal{T}_\mathcal{G}, \{\mathrm{A}(\mathsf{a})\})$ and query $q = \exists z\, \mathrm{C}(z)$. □

More interestingly, complexity results for role and concept cardinality queries differ when disallowing disjointness axioms but keeping role inclusions. While there exist coNP-complete DL-Lite$^\mathcal{H}_\mathsf{pos}$ KBs for role cardinality answering (see Theorem 50 in the next section, proving it already holds in data complexity), we obtain a NL procedure for all DL-Lite$^\mathcal{H}_\mathsf{pos}$ ontologies and all concept cardinality queries.

Theorem 45. *Concept cardinality query answering in DL-Lite$^\mathcal{H}_\mathsf{pos}$ is in* NL *w.r.t. combined complexity.*

Proof. Let $q_\mathrm{C} = \exists z\, \mathrm{C}(z)$ be a concept cardinality query. Starting from the canonical model $\mathcal{C}_\mathcal{K}$ of a KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, the minimal number of matches can easily be computed.

- If there exists an individual $\mathsf{a} \in \mathsf{Ind}(\mathcal{A})$ such that $\mathcal{K} \models \mathrm{C}(\mathsf{a})$, then we can collapse all anonymous elements onto one such individual (the choice doesn't matter), obtaining a model in which matches are exactly such individuals a, which is clearly minimal (recall we make the UNA). We can check whether $\mathcal{K} \models \mathrm{C}(\mathsf{a})$ in NL, see e.g. Artale et al. [2009].
- Otherwise, if there exists an anonymous match in $\mathcal{C}_\mathcal{K}$, then we collapse all anonymous elements onto a chosen ABox individual, obtaining a model with a single match for q_C, which is clearly optimal. Existence of an anonymous match can be checked in NL [Artale et al., 2009].
- Otherwise, there are no matches in $\mathcal{C}_\mathcal{K}$, hence 0 is the minimal number of matches.

Notice that we do not need to actually compute the model corresponding to the optimal number of matches, and we only need to compare that number to the input integer. □

Relying on the same principle but employing a more sophisticated case analysis, we obtain NL membership for role cardinality queries evaluated over DL-Lite$_\mathsf{pos}$ ontologies.

Theorem 46. *Role cardinality query answering in DL-Lite$_\mathsf{pos}$ is in* NL *w.r.t. combined complexity.*

Proof. Consider the role cardinality query $\exists z_1 \exists z_2 \ \mathrm{P}(z_1, z_2)$, and define the sets $\mathcal{D}^+_\mathcal{K} = \{\mathsf{a} \mid \mathsf{a}\mathrm{P} \in \Delta^{\mathcal{C}_\mathcal{K}}\}$ and $\mathcal{D}^-_\mathcal{K} = \{\mathsf{a} \mid \mathsf{a}\mathrm{P}^- \in \Delta^{\mathcal{C}_\mathcal{K}}\}$ of positive and negative demanding individuals. We assume w.l.o.g. that $|\mathcal{D}^+_\mathcal{K}| \leq |\mathcal{D}^-_\mathcal{K}|$. Let $\mathsf{p} : \mathcal{D}^+_\mathcal{K} \to \mathcal{D}^-_\mathcal{K}$ be an injection.

We partition the generated roles (*i.e.*, the roles such that there is $w\mathrm{T} \in \Delta^{\mathcal{C}_\mathcal{K}}$) into four categories:

1. $\mathcal{T} \models \exists \mathrm{T}^- \sqsubseteq \exists \mathrm{P}$ and $\mathcal{T} \models \exists \mathrm{T}^- \sqsubseteq \exists \mathrm{P}^-$
2. $\mathcal{T} \models \exists \mathrm{T}^- \sqsubseteq \exists \mathrm{P}$ and $\mathcal{T} \not\models \exists \mathrm{T}^- \sqsubseteq \exists \mathrm{P}^-$
3. $\mathcal{T} \not\models \exists \mathrm{T}^- \sqsubseteq \exists \mathrm{P}$ and $\mathcal{T} \models \exists \mathrm{T}^- \sqsubseteq \exists \mathrm{P}^-$
4. $\mathcal{T} \not\models \exists \mathrm{T}^- \sqsubseteq \exists \mathrm{P}$ and $\mathcal{T} \not\models \exists \mathrm{T}^- \sqsubseteq \exists \mathrm{P}^-$

The roles in the first three cases are called *demanding*, and we need to consider which P-edges can be used for them.

We use the term *non-paired critical individual* to designate an individual belonging to $\mathcal{D}^+_\mathcal{K} \cup \mathcal{D}^-_\mathcal{K}$ but not to the domain of p. We then define what constitutes a *solution to a demanding role*:

- A solution to a case-1 demanding role is either a non-paired critical individual, or an individual a such that $\mathcal{A}, \mathcal{T} \models \exists x \mathrm{P}(\mathsf{a}, x)$ and $\mathcal{A}, \mathcal{T} \models \exists x \mathrm{P}(x, \mathsf{a})$.
- A solution to a case-2 demanding role is either a non-paired critical individual, or an individual a such that $\mathcal{A}, \mathcal{T} \models \exists x \mathrm{P}(\mathsf{a}, x)$.
- A solution to a case-3 demanding role is either a non-paired critical individual, or an individual a such that $\mathcal{A}, \mathcal{T} \models \exists x \mathrm{P}(x, \mathsf{a})$.

If a demanding role T has a solution, we let $\mathsf{sol}(\mathrm{T})$ be (an arbitrarily chosen) solution.

If all demanding roles have a solution, then the optimal number of matches is $n_\mathcal{A} + \mathsf{max}(|\mathcal{D}^+_\mathcal{K}|, |\mathcal{D}^-_\mathcal{K}|)$, as witnessed by the model $f(\mathcal{C}_\mathcal{K})$, which is the image of $\mathcal{C}_\mathcal{K}$ under the following *partial* function f:

- $f(\mathsf{a}) = \mathsf{a}$;
- $f(\mathsf{a}\mathrm{P}) = \mathsf{p}(\mathsf{a})$;
- $f(\mathsf{a}\mathrm{P}^-) = \mathsf{p}^{-1}(\mathsf{a})$ if defined, a otherwise;
- $f(w\mathrm{T}) = \mathsf{sol}(\mathrm{T})$ if T is neither P nor P^- and is demanding;
- $f(w\mathrm{T}) = w\mathrm{T}$ if $w\mathrm{T}$ contains no occurrence of P nor of P^- and T is not demanding.

Note that f is not defined on elements from $\mathcal{C}_{\mathcal{K}}$ with shape $\mathsf{a}w\mathrm{TP}w'$ or $\mathsf{a}w\mathrm{TP}^{-}w'$, where w is a possibly-empty word that contains neither P nor P^{-} and w' is a possibly-empty word. In the case of $\mathsf{a}w\mathrm{TP}w'$ (the case of $\mathsf{a}w\mathrm{TP}^{-}w'$ is similar), notice that $\mathsf{a}w\mathrm{T}$ is sent to an element $\mathsf{sol}(\mathrm{T})$, such that $\mathcal{K} \models \exists x\ \mathrm{P}(\mathsf{sol}(\mathrm{T}), x)$ by definition of a solution. Therefore the images of elements $\mathsf{a}w\mathrm{TP}w'$ don't need to be specified to ensure modelhood, as the corresponding facts are already consequences of the P-edge (a, b) (if there exists b such that $(\mathsf{sol}(\mathrm{T}), \mathsf{b}) \in \mathcal{A}$) or of the P-edge $(\mathsf{a}, f(\mathsf{aP}))$ (if no such b exists). It can be verified that $f(\mathcal{C}_{\mathcal{K}})$ is a model with exactly $n_{\mathcal{A}} + \mathsf{max}(|\mathcal{D}^{+}_{\mathcal{K}}|, |\mathcal{D}^{-}_{\mathcal{K}}|)$ matches.

If there is at least one demanding role that does not have a solution, then the optimal number of matches is $m_{\mathcal{A}} + \mathsf{max}(|\mathcal{D}^{+}_{\mathcal{K}}|, |\mathcal{D}^{-}_{\mathcal{K}}|) + 1$, as witnessed by the following model (which we describe by an ABox):

$$\begin{aligned}\mathcal{A} &\cup \{\mathrm{A}(\mathsf{a}) \mid \mathcal{A}, \mathcal{T} \models \mathrm{A}\}\\ &\cup \{\mathrm{P}(\mathsf{a}, \mathsf{p}(\mathsf{a})) \mid \mathsf{a} \in \mathcal{D}^{+}_{\mathcal{K}}\}\\ &\cup \{\mathrm{R}(a, \star) \mid \mathrm{R} \neq \mathrm{P} \wedge \mathsf{a}\mathrm{R} \in \Delta^{\mathcal{C}_{\mathcal{K}}}\}\\ &\cup \{\mathrm{R}(\star, \star) \mid \mathrm{R} \in \mathsf{N_R}\} \cup \{\mathrm{A}(\star) \mid \mathrm{A} \in \mathsf{N_C}\}\end{aligned}$$

The above interpretation is indeed a model, because all elements are paired and disjointness is not expressible in DL-Lite$_{\mathsf{pos}}$. Moreover, its number of matches is $m_{\mathcal{A}} + \mathsf{max}(|\mathcal{D}^{+}_{\mathcal{K}}|, |\mathcal{D}^{-}_{\mathcal{K}}|) + 1$. This is optimal as there are at least $m_{\mathcal{A}} + |\mathcal{D}^{+}_{\mathcal{K}}|$ matches in any model and that there exists T a demanding role having no solution. Indeed, if T is in cases 2 or 3, there cannot be any P-edge in the ABox nor paired elements (as it would provide a solution for T), and 1 is thus the optimal as any model contains at least 1 match given by the image of the pair $(w\mathrm{T}, w\mathrm{TP})$ from the canonical model (T in case 2) or of the pair $(w\mathrm{T}, w\mathrm{TP}^{-})$ (in case 3). Otherwise T belongs to case 1, still without a solution, which means that no individual has both an ingoing and outgoing P. Therefore, in any model, at least one of the image of the pairs $(w\mathrm{T}, w\mathrm{TP})$ and $(w\mathrm{T}, w\mathrm{TP}^{-})$ (both exist in the canonical model, for the same w!) provides an additional match.

Note that each condition can be checked in non-deterministic logarithmic space. The number of optimal matches is thus also computable within the same bound, as is the comparison with the input integer. This shows that role cardinality answering lies in NL. □

Finally, we prove that concept cardinality query answering over DL-Lite$_{\mathsf{pos}}$ KBs is NL-hard by reduction from the ST-CONNECTIVITY problem, known to be NL-complete [Immerman, 1999].

Theorem 47. *Concept cardinality query answering in DL-Lite*$_{\mathsf{pos}}$ *is* NL*-hard w.r.t. combined complexity.*

Proof. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an oriented graph and s, t two vertices from $\mathcal{V}$. For each vertex $v \in \mathcal{V}$, we introduce a concept name V. Consider the KB given by $\mathcal{A} := \{\mathrm{S}(\mathsf{a})\}$ and $\mathcal{T} := \{\mathrm{V}_1 \sqsubseteq \mathrm{V}_2 \mid (v_1, v_2) \in \mathcal{E}\}$. We are interested in the concept cardinality query $q_{\mathrm{T}} := \exists z\ \mathrm{T}(z)$.

It is now straightforward that 1 is a certain answer to q_{T} over $(\mathcal{T}, \mathcal{A})$ iff t is reachable from s in $\mathcal{G}$. □

This concludes our study of cardinality queries in combined complexity, which is summarized in Figure 5.1. The remainder of the present chapter is devoted to the study of data complexity, and more especially in the extensions of DL-Lite$_{\mathsf{pos}}$.

5.3 Hard cases in data complexity

To begin our study of cardinality query answering in data complexity, let us first recall that we inherit the coNP upper bound from the general CCQ setting (see Theorem 8 from Chapter 3) In this section, we exhibit three matching lower bounds relying on three rather different mechanisms.

5.3.1 A reduction from 3-Col

We begin with a reduction from the 3-Col problem to prove the coNP-hardness of answering the concept cardinality query q_{B} over a specific $\mathcal{EL}$ TBox. The key ingredient of the reduction is the ability the detect monochromatic edges with a single $\mathcal{EL}$ axiom, while known reductions from 3-Col to the DL-Lite settings (see e.g. Theorem 16 treating the case of CCQs) achieve this through a more involved query. In other words, $\mathcal{EL}$ TBoxes allow us to internalize sufficient parts of the query so the latter can be restricted to a concept cardinality query.

Theorem 48. *Concept cardinality query answering in $\mathcal{EL}$ is* coNP*-hard w.r.t. data complexity.*

Proof. We reduce the complement of the graph 3-colorability problem to answering the $\mathcal{EL}$ OMQ $(q, \mathcal{T})$, with $q = \exists z\ \mathrm{B}(z)$ and $\mathcal{T}$ containing $\mathrm{A} \sqsubseteq \exists \mathrm{R.B}$ and $\exists \mathrm{R.C_k} \sqcap \exists \mathrm{E}.(\exists \mathrm{R.C_k}) \sqsubseteq \mathrm{B}$ for $k \in \{1, 2, 3\}$.

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an undirected graph, and consider now the ABox given by:

$$\begin{aligned}\mathcal{A} := & \{\mathrm{A}(\mathsf{v}) \mid v \in \mathcal{V}\} \cup \{\mathrm{E}(\mathsf{v_1}, \mathsf{v_2}) \mid \{v_1, v_2\} \in \mathcal{E}\} \\ & \cup \{\mathrm{C}_1(\mathsf{c_1}), \mathrm{C}_2(\mathsf{c_2}), \mathrm{C}_3(\mathsf{c_3}), \mathrm{B}(\mathsf{c_1}), \mathrm{B}(\mathsf{c_2}), \mathrm{B}(\mathsf{c_3})\}\end{aligned}$$

Set $\mathcal{K} = (\mathcal{T}, \mathcal{A})$. Observe that there are 3 ABox matches: $\mathsf{c_1}, \mathsf{c_2}, \mathsf{c_3}$. We claim:

$$[4, +\infty] \text{ is a certain answer of } q \text{ w.r.t. } \mathcal{K} \quad \Leftrightarrow \quad \mathcal{G} \notin \text{3-Col}$$

($\Leftarrow$). Assume $\mathcal{G} \notin$ 3-Col. Let $\mathcal{I}$ be a model of $\mathcal{K}$ and $f : \mathcal{C}_\mathcal{K} \to \mathcal{I}$ a homomorphism. We are interested in the image of elements $\mathsf{v} \cdot \mathrm{R.B}$, with $v \in \mathcal{V}$, whose existence in $\Delta^{\mathcal{C}_\mathcal{K}}$ is ensured by axiom $\mathrm{A} \sqsubseteq \exists \mathrm{R.B}$. If there exists $v \in \mathcal{V}$ such that $f(\mathsf{v} \cdot \mathrm{R.B}) \notin \{\mathsf{c}_1, \mathsf{c}_2, \mathsf{c}_3\}$, then $f(\mathsf{v} \cdot \mathrm{R.B})$ provides a new match. Otherwise, define the colouring induced by $\mathcal{I}$ as $\rho_\mathcal{I}(v) = f(\mathsf{v} \cdot \mathrm{R.B}) \in \{\mathsf{c}_1, \mathsf{c}_2, \mathsf{c}_3\}$. Since $\mathcal{G} \notin$ 3-Col, there exists an edge $\{v_1, v_2\} \in \mathcal{E}$ with both vertices having the same colour c_k for some $k \in \{1, 2, 3\}$. For the corresponding individuals v_1 and v_2, the axiom $\exists \mathrm{R.C_k} \sqcap \exists \mathrm{E.}(\exists \mathrm{R.C_k}) \sqsubseteq \mathrm{B}$ triggers and provides two new matches: v_1 and v_2. In all cases, $[4, +\infty]$ is a certain answer of q w.r.t. $\mathcal{K}$.

($\Rightarrow$). Assume $\mathcal{G} \in$ 3-Col. Take a 3-colouring $\rho : \mathcal{V} \to \{\mathsf{c}_1, \mathsf{c}_2, \mathsf{c}_3\}$, and consider the interpretation $\mathcal{I}_\rho$ obtained from $\mathcal{K}$ in which we add facts $\mathrm{R}(\mathsf{v}, \rho(v))$ for each $v \in \mathcal{V}$, complying with the axiom $\mathrm{A} \sqsubseteq \exists \mathrm{R.B}$. By definition of ρ, there is no monochromatic edge, which ensures the three other axioms don't trigger on individuals v. This interpretation $\mathcal{I}_\rho$ is hence a model. It only has 3 matches, hence $[4, +\infty]$ is not a certain of q w.r.t. $\mathcal{K}$. □

This concludes for the present chapter our study of extensions of $\mathcal{EL}$, for which cardinality query answering is always coNP-complete in the worst case for data complexity.

5.3.2 A reduction from 3-Sat

We now move to extensions of DL-Lite$_{\mathsf{pos}}$, as usual up to DL-Lite$^{\mathcal{H}}_{\mathsf{core}}$ KBs. Interestingly, this setting has already been investigated in Calvanese et al. [2020a]. They proved cardinality query answering over DL-Lite$_{\mathsf{pos}}$ ontologies can be performed in P, an upper bound that we refine further in Section 5.4, and that there exists P-hard DL-Lite$^{\mathcal{H}}_{\mathsf{core}}$ ontologies for role cardinality answering. None of their bounds in that work are tight and in particular the existence of coNP-hard cardinality queries over DL-Lite$^{\mathcal{H}}_{\mathsf{core}}$ ontologies was left open.

Through a reduction from 3-Sat, we exhibit a DL-Lite$^{\mathcal{H}}_{\mathsf{core}}$ ontology and a concept cardinality query for which the answering problem is coNP-hard. The construction relies mainly on role disjointness axioms to constraint the reuse of individuals, when trying to minimize the number of instances for the query concept which is entailed by the endpoints of different roles.

Theorem 49. *Concept cardinality query answering in DL-Lite$^{\mathcal{H}}_{\mathsf{core}}$ is* coNP*-complete w.r.t. data complexity.*

Proof. We reduce the complementary of 3-Sat to the problem of answering the concept cardinality query q_C over the DL-Lite$^{\mathcal{H}}_{\mathsf{core}}$ TBox $\mathcal{T}$ containing the 7 following axioms:

$$\mathrm{A} \sqsubseteq \exists \mathrm{U} \quad \exists \mathrm{U}^- \sqsubseteq \mathrm{C} \quad \mathrm{U} \sqsubseteq \neg \mathrm{U}' \quad \mathrm{B} \sqsubseteq \exists \mathrm{V} \quad \exists \mathrm{V}^- \sqsubseteq \mathrm{C} \quad \mathrm{V} \sqsubseteq \neg \mathrm{V}' \quad \exists \mathrm{U}^- \sqsubseteq \neg \exists \mathrm{V}^-$$

We first explain the general idea. Three individuals are introduced per propositional variable (one for the variable itself with concept A, two for its possible truth values), as well as one individual per clause (with concept B). Each variable should have a truth value given by U (whose possible values in the ABox are restricted through the use of U′), and each clause should have a falsified literal given by V (whose possibles values in the ABox are restricted, according to the input formula, with V′). The input formula is a tautology iff every model introduces a new element marked C (as a witness for either ∃U or ∃V). More formally, consider a 3DNF formula $\phi(x_1, \ldots x_m) = \bigvee_{i=1}^{n} l_i$, with $l_i = \bigwedge_{j=1}^{3} (\neg)^{p_{i,j}} v_{i,j}$. Introduce the following individual names:

$$\mathsf{Ind}_\phi = \{\mathsf{x}_1, \ldots \mathsf{x}_\mathsf{m}, \mathsf{l}_1, \ldots \mathsf{l}_\mathsf{n}, \mathsf{t}_1, \ldots \mathsf{t}_\mathsf{m}, \mathsf{f}_1, \ldots \mathsf{f}_\mathsf{m}\}$$

Consider now the ABox given by:

$$\begin{aligned} \mathcal{A}_\phi = \; & \{\mathrm{A}(\mathsf{x}_1), \ldots \mathrm{A}(\mathsf{x}_\mathsf{m}), \mathrm{B}(\mathsf{l}_1), \ldots \mathrm{B}(\mathsf{l}_\mathsf{n}), \mathrm{C}(\mathsf{t}_1), \ldots \mathrm{C}(\mathsf{t}_\mathsf{m}), \mathrm{C}(\mathsf{f}_1), \ldots \mathrm{C}(\mathsf{f}_\mathsf{m})\} \\ \cup \; & \{\mathrm{U}'(\mathsf{x}_\mathsf{k}, \mathsf{a}) \mid 1 \leq k \leq m, \; \mathsf{a} \in \mathsf{Ind}_\phi \setminus \{\mathsf{t}_\mathsf{k}, \mathsf{f}_\mathsf{k}\}\} \\ \cup \; & \left\{ \mathrm{V}'(\mathsf{l}_\mathsf{i}, \mathsf{a}) \,\middle|\, 1 \leq i \leq n, \; \mathsf{a} \in \mathsf{Ind}_\phi \setminus \left\{ \mathsf{t}_\mathsf{i} \,\middle|\, \begin{array}{c} v_{i,j} = x_i \\ p_{i,j} = 0 \end{array} \right\} \cup \left\{ \mathsf{f}_\mathsf{i} \,\middle|\, \begin{array}{c} v_{i,j} = x_i \\ p_{i,j} = 1 \end{array} \right\} \right\} \end{aligned}$$

Set $\mathcal{K}_\phi = (\mathcal{T}, \mathcal{A}_\phi)$. Notice there are $2m$ ABoxes matches: $\mathsf{t}_1 \ldots \mathsf{t}_\mathsf{m}$ and $\mathsf{f}_1, \ldots \mathsf{f}_\mathsf{m}$. We now prove the following claim:

$$[2m+1, +\infty] \text{ is a certain answer of } q_\mathrm{C} \text{ w.r.t. } \mathcal{K}_\phi \text{ iff } \forall \mathbf{x}\ \phi(\mathbf{x})$$

(⇐). Assume $\forall \mathbf{x}\ \phi(\mathbf{x})$. Let $\mathcal{I}$ be a model of $\mathcal{K}_\phi$ and $f : \mathcal{C}_{\mathcal{K}_\phi} \to \mathcal{I}$ a homomorphism. If there exists a $k \in \{1, \ldots m\}$ such that $f(\mathsf{x}_\mathsf{k}\mathrm{U}) \notin \{\mathsf{t}_\mathsf{k}, \mathsf{f}_\mathsf{k}\}$, then $f(\mathsf{x}_\mathsf{k}\mathrm{U})$ is an anonymous element, since U′ prevents $f(\mathsf{x}_\mathsf{k}\mathrm{U})$ to be equal to other individuals. As $f(\mathsf{x}_\mathsf{k}\mathrm{U}) \in \mathrm{C}^\mathcal{I}$, it provides a new match. Otherwise, define the assignment induced by $\mathcal{I}$ as $\rho_\mathcal{I}(x) = 1$ if $f(\mathsf{x}_\mathsf{k}\mathrm{U}) = \mathsf{t}_\mathsf{k}$, and $\rho_\mathcal{I}(x) = 0$ if $f(\mathsf{x}_\mathsf{k}\mathrm{U}) = \mathsf{f}_\mathsf{k}$. Since $\forall \mathbf{x}\ \phi(\mathbf{x})$, there exists a satisfied clause l_i. For this i, the element $f(\mathsf{l}_\mathsf{i}\mathrm{V})$ cannot be equal to any individual (as V′ and $\exists \mathrm{U}^-$ prevent it), and therefore provides a new match for q_C. In all cases $[2m+1, +\infty]$ is a certain answer of q_C w.r.t. $\mathcal{K}_\phi$.

(⇒). Assume $\exists \mathbf{x}\ \neg\phi(\mathbf{x})$. Consider such a valuation $\rho : \mathbf{x} \to \{0, 1\}$ such that $\neg\phi(\rho(\mathbf{x}))$. For each clause l_i, there exists (at least) a variable x_{k_i} which invalidates l_i. Consider the interpretation $\mathcal{I}_\rho$ obtained from $\mathcal{K}_\phi$ in which we add facts $\mathrm{U}(\mathsf{x}_\mathsf{k}, \mathsf{t}_\mathsf{k})$ iff $\rho(x_k) = 1$, resp $\mathrm{U}(\mathsf{x}_\mathsf{k}, \mathsf{f}_\mathsf{k})$ iff $\rho(x_k) = 0$, and $\mathrm{V}(\mathsf{l}_\mathsf{i}, \mathsf{t}_{\mathsf{k}_\mathsf{i}})$ if $\rho(x_{k_i}) = 0$, resp $\mathrm{V}(\mathsf{l}_\mathsf{i}, \mathsf{f}_{\mathsf{k}_\mathsf{i}})$ if $\rho(x_{k_i}) = 1$. By definition of variables x_{k_i}, we are ensured this interpretation $\mathcal{I}_\rho$ is a model. It only has $2m$ matches, hence $[2m+1, +\infty]$ is not a certain of q_C w.r.t. $\mathcal{K}_\phi$. □

5.3.3 A reduction from Set Cover

While concept cardinality queries over DL-Lite$^{\mathcal{H}}_{\mathsf{pos}}$ ontologies are proven tractable in the next section, our third reduction shows there exist coNP-hard such ontologies for role cardinality query answering. This reduction plays a central role in Section 5.5, as it is the prototype of such coNP-hard ontology-mediated role cardinality queries. It exploits the propagation of some subroles of the query role to construct a reduction from the Set Cover problem, and this propagation schema allows us to separate exactly the coNP-hard ontologies (for a fixed query role), from those that can be solved in P.

Theorem 50. *Role cardinality query answering in DL-Lite$^{\mathcal{H}}_{\mathsf{pos}}$ is* coNP*-complete w.r.t. data complexity.*

Proof. We consider the NP-complete Set Cover problem: given a set $\mathcal{U}$, set of subsets $\mathcal{S} \subseteq 2^{\mathcal{U}}$ whose union is $\mathcal{U}$, and number k, decide whether there exists a k-cover, i.e. a subset $\mathcal{C}$ of $\mathcal{S}$ with $|\mathcal{C}| \leq k$ whose union is $\mathcal{U}$. We reduce the complementary of Set Cover to the problem of answering the role cardinality query q_{S} over the DL-Lite$^{\mathcal{H}}_{\mathsf{pos}}$ TBox $\mathcal{T} := \{\mathrm{B} \sqsubseteq \exists \mathrm{R}_1, \mathrm{R}_1 \sqsubseteq \mathrm{S}, \exists \mathrm{R}_1^- \sqsubseteq \exists \mathrm{R}_2, \mathrm{R}_2 \sqsubseteq \mathrm{S}\}$. From an instance of Set Cover we construct $\mathcal{A} = \{\mathrm{B}(\mathsf{u}) \mid u \in \mathcal{U}\} \cup \{\mathrm{S}(\mathsf{u}, \mathsf{s}) \mid u \in s, s \in \mathcal{S}\}$. Figure 5.2 depicts the ABox built from the following instance of Set Cover:

$$\mathcal{U} = \{1,2,3,4,5\} \qquad \mathcal{S} = \{\ \{1,2\},\ \{3,4\},\ \{4,5\},\ \{1,2,3\}\ \} \qquad k$$

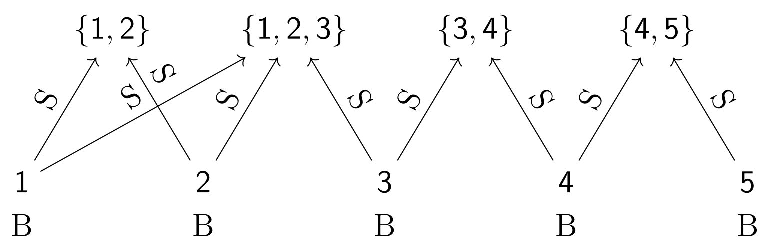


Figure 5.2: The built ABox for the example instance of Set Cover.

We prove the following claim:

$$\text{There exists a } k\text{-cover}$$
$$\text{iff}$$
$$[\Sigma_{s\in\mathcal{S}}|s| + k + 1, +\infty] \text{ is } \textit{not} \text{ a certain answer for } q_{\mathrm{S}} \text{ over } \mathcal{K} := (\mathcal{T}, \mathcal{A}).$$

Intuitively, from a k-cover $\mathcal{C}$, we obtain a countermodel in which role R_1 contains pairs (u, s) such that $u \in s$ and $s \in \mathcal{C}$, and there is one outgoing R_2 role from each $s \in \mathcal{C}$. Notice there are at least $m_{\mathcal{A}} := \Sigma_{s\in\mathcal{S}}|s|$ many matches for q_{S} from the instances of S encoded in the ABox $\mathcal{A}$.

($\Rightarrow$). Assume $(\mathcal{U}, \mathcal{S}, k) \in$ Set Cover. Take some k-cover $F \subseteq \mathcal{S}$ of $\mathcal{U}$. For each $u \in s \in \mathcal{C}$, enrich the ABox $\mathcal{A}$ with the assertions $\mathrm{R}_1(\mathsf{u}, \mathsf{s})$, $\mathrm{R}_2(\mathsf{s}, \mathsf{s})$ and $\mathrm{S}(\mathsf{s}, \mathsf{s})$. The resulting interpretation $\mathcal{I}_{\mathcal{C}}$ (based upon the described enriched ABox) is a model, as we introduced all needed roles. In addition to the $m_{\mathcal{A}}$ ABox matches, each used subset s provides one additional match since the assertion $\mathrm{S}(\mathsf{s}, \mathsf{s})$ has been added. We thus obtain a model with exactly $m_{\mathcal{A}} + k$ matches, that is a countermodel for $[m_{\mathcal{A}} + k + 1, +\infty]$ being a certain answer.

($\Leftarrow$). Assume $(\mathcal{U}, \mathcal{S}, k) \notin$ Set Cover. Consider a model $\mathcal{I}$ of $\mathcal{K}$ and a homomorphism $f : \mathcal{C}_{\mathcal{K}} \to \mathcal{I}$. or each $u \in \mathcal{U}$, we associate a subset $\rho(u) := s$ if $f(\mathsf{u} \cdot \mathrm{R}_1) = \mathsf{s}$ and $u \in s \in \mathcal{S}$, otherwise set $\rho(u) := s_u$, where $s_u \in \mathcal{S}$ in any subset containing u (if no such s_u, then the Set Cover instance is trivially unsatisfiable). The image $\rho(\mathcal{U})$ is a covering of $\mathcal{U}$, hence $|\rho(\mathcal{U})| \geq k+1$. By definition, for each $s \in \rho(\mathcal{U})$ there exists $u \in \mathcal{U}$ such that: either $f(\mathsf{u} \cdot \mathrm{R}_1) = \mathsf{s}$ with $u \in s \in \mathcal{S}$, or $f(\mathsf{u} \cdot \mathrm{R}_1) \neq \mathsf{s}'$ for all $u \in s' \in \mathcal{S}$. In the first case, the pair $(f(\mathsf{u} \cdot \mathrm{R}_1), f(\mathsf{u} \cdot \mathrm{R}_1 \cdot \mathrm{R}_2))$ is an additional match. In the second case, $(f(\mathsf{u}), f(\mathsf{u} \cdot \mathrm{R}_1))$ is a new match. Therefore we can conclude that there are at least $m_{\mathcal{A}} + k + 1$ matches in $\mathcal{I}$. □

5.4 Tractable cases in data complexity

In this section, we identify two settings in which cardinality queries can be answered with the lowest possible complexity. As previously mentioned, a P procedure was already provided in Calvanese et al. [2020a] for the even more restricted class of cardinality queries over DL-Lite$_{\mathsf{pos}}$ ontologies. Our result refines and extends theirs:

Theorem 51. *Answering a cardinality query q over a TBox $\mathcal{T}$ is in TC^0 if either (i) q is a role cardinality query and $\mathcal{T}$ a DL-Lite$_{\mathsf{core}}$ TBox, or (ii) q is a concept cardinality query and $\mathcal{T}$ is a DL-Lite$^{\mathcal{H}}_{\mathsf{core}}$ TBox without negative role inclusions.*

The remainder of this section is devoted to establishing TC^0 membership for case (i) where our query is $q_{\mathrm{S}} = \exists z_1 \exists z_2\, \mathrm{S}(z_1, z_2)$ (see Section 5.4.1). A similar but simpler argument can be used for the membership of case (ii) (see Section 5.4.3). The lower bound, that is, TC^0-hardness, is also discussed in Section 5.4.3 and is easily shown, if the query predicate is satisfiable, by reduction from the TC^0-complete NumOnes problem [Aehlig et al., 2007] asking, given a binary string X and $k \geq 1$, whether X contains at least k 1-bits.

Since we will be focusing on DL-Lite$^{\mathcal{H}}_{\mathsf{core}}$ KBs from the present section to the end of the chapter, we recall the definition of the canonical models we consider, slightly simplified to better fit our restricted setting [Calvanese et al., 2007b].

Definition 52. *Every satisfiable DL-Lite$^{\mathcal{H}}_{\mathsf{core}}$ KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ has a* canonical model *$\mathcal{C}_{\mathcal{K}}$, defined as follows. The domain of $\mathcal{C}_{\mathcal{K}}$ contains $\mathsf{Ind}(\mathcal{A})$ and all words $\mathsf{a}\mathrm{R}_1 \dots \mathrm{R}_n$, with $\mathsf{a} \in \mathsf{Ind}(\mathcal{A})$, $\mathrm{R}_i \in \mathsf{N}^{\pm}_{\mathsf{R}}$, and $n \geqslant 1$, such that:*

- $\mathcal{K} \models \exists \mathrm{R}_1(\mathsf{a})$ *and there is no* $\mathrm{R}_1(\mathsf{a}, \mathsf{b}) \in \mathcal{A}$;
- *for* $1 \leq i < n$, $\mathcal{T} \models \exists \mathrm{R}_i^- \sqsubseteq \exists \mathrm{R}_{i+1}$ *and* $\mathrm{R}_i^- \neq \mathrm{R}_{i+1}$.

Concept and role names are interpreted as follows:

$$\begin{aligned}
\mathrm{A}^{\mathcal{C}_{\mathcal{K}}} = &\{\mathsf{a} \in \mathsf{Ind}(\mathcal{A}) \mid \mathcal{K} \models \mathrm{A}(a)\} \\
& \cup\ \{\mathsf{a}\mathrm{R}_1 \dots \mathrm{R}_n \in \Delta^{\mathcal{C}_{\mathcal{K}}} \setminus \mathsf{Ind}(\mathcal{A}) \mid \mathcal{T} \models \exists \mathrm{R}_n^- \sqsubseteq \mathrm{A}\} \\
\mathrm{P}^{\mathcal{C}_{\mathcal{K}}} = &\{(\mathsf{a}, \mathsf{b}) \mid \mathrm{P}(\mathsf{a}, \mathsf{b}) \in \mathcal{A}\} \\
& \cup\ \{(e_1, e_2) \mid e_2 = e_1\mathrm{R} \text{ and } \mathcal{T} \models \mathrm{R} \sqsubseteq \mathrm{P}\} \\
& \cup\ \{(e_2, e_1) \mid\ e_2 = e_1\mathrm{R} \text{ and } \mathcal{T} \models \mathrm{R} \sqsubseteq \mathrm{P}^-\}
\end{aligned}$$

We use $\mathsf{gen}_{\mathcal{K}}$ to refer to the set of generated roles*, i.e. those $\mathrm{R} \in \mathsf{N}^{\pm}_{\mathsf{R}}$ such that $\Delta^{\mathcal{C}_{\mathcal{K}}}$ contains an element $w\mathrm{R}$.*

5.4.1 Role cardinality over DL-Lite$_{\mathsf{core}}$

Existing proofs of sub-polynomial data complexity for restricted classes of counting queries rely on the canonical model minimizing the number of matches (see Chapter 4 and Calvanese et al. [2020a]). However, for the class of cardinality queries, the canonical model may not yield the minimum value (see e.g. Example 20 below). Therefore, we develop a different approach based upon a systematic exploration of a set of models that is guaranteed to contain an optimal model and whose size depends only on the TBox. This special set of models will be induced from *strategies* that dictate how to merge elements of the canonical model. To show such models contain the optimal value, we show that if we extract a strategy σ from an arbitrary model $\mathcal{I}$ and consider any model $\mathcal{J}$ induced by σ, then $\mathcal{J}$ has at most as many matches as the initial model $\mathcal{I}$.

Example 20. *As a running example, we will consider the KB $\mathcal{K}_e = (\mathcal{T}_e, \mathcal{A}_e)$ whose TBox contains the following inclusions*

$$\begin{array}{llll}
\mathrm{A}_1 \sqsubseteq \exists \mathrm{T}_1 & \mathrm{A}_2 \sqsubseteq \exists \mathrm{T}_2 & \exists \mathrm{T}_1^- \sqsubseteq \exists \mathrm{S} & \exists \mathrm{R}_1^- \sqsubseteq \neg \exists \mathrm{R}_2^- \\
\mathrm{B}_1 \sqsubseteq \exists \mathrm{R}_1 & \mathrm{B}_2 \sqsubseteq \exists \mathrm{R}_2 & \exists \mathrm{R}_1^- \sqsubseteq \exists \mathrm{S}^- & \exists \mathrm{R}_1^- \sqsubseteq \neg \exists \mathrm{T}_1^- \\
\exists \mathrm{T}_2^- \sqsubseteq \exists \mathrm{S} & \exists \mathrm{S}^- \sqsubseteq \exists \mathrm{S} & \exists \mathrm{R}_2^- \sqsubseteq \exists \mathrm{S}^- &
\end{array}$$

and whose ABox contains the assertions

$$\{\mathrm{A}_1(\mathsf{a}_1), \mathrm{A}_2(\mathsf{a}_2), \mathrm{B}_1(\mathsf{b}_1), \mathrm{B}_2(\mathsf{b}_2), \mathrm{R}_1(\mathsf{a}_1, \mathsf{a}_2), \mathrm{S}(\mathsf{b}_2, \mathsf{b}_1)\}$$

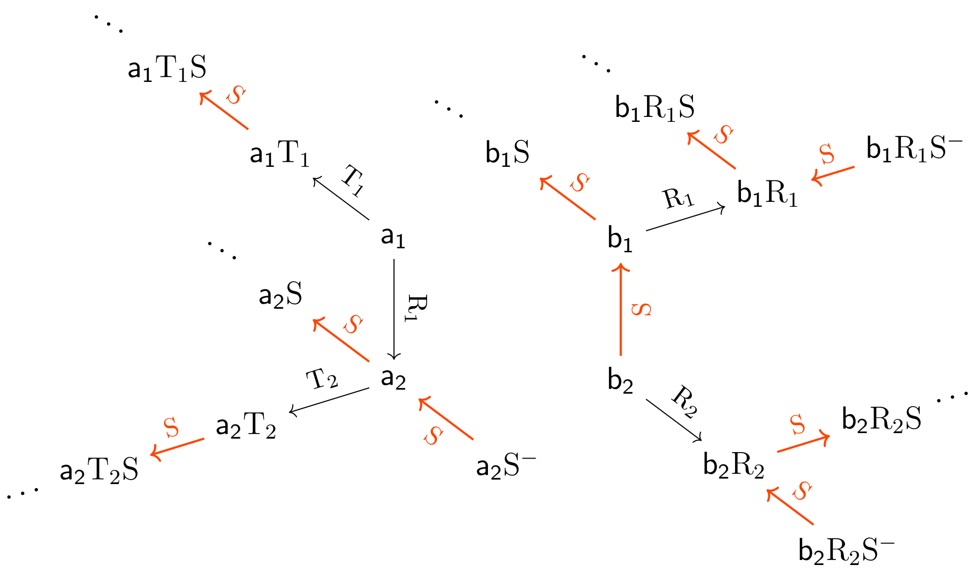


Figure 5.3: Initial portion of the canonical model of $\mathcal{K}_e$. For readability, we have omitted concepts and highlighted the role S from the cardinality query.

Two finite models of $\mathcal{K}_e$ are displayed in Figures 5.4a and 5.4b. An initial portion of (the infinite) canonical model of $\mathcal{K}_e$ is displayed in Figure 5.3. Observe that $\mathsf{gen}_{\mathcal{K}} = \{\mathrm{S}, \mathrm{S}^-, \mathrm{R}_1, \mathrm{R}_2, \mathrm{T}_1, \mathrm{T}_2\}$.

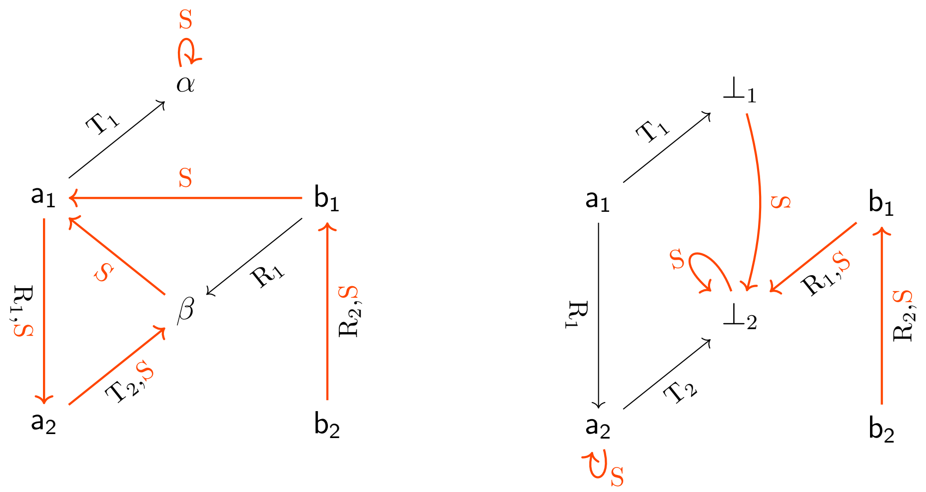
Consider the role cardinality query q_{S}. The answer to q_{S} is $+\infty$ in $\mathcal{C}_{\mathcal{K}_e}$, 6 in the model from Figure 5.4a, and 5 in the model from Figure 5.4b. The latter implies that $[6, +\infty]$ is not *a certain answer. We leave it is an exercise to find a model with 3 matches and show there is no model with fewer matches, which means that $[m, +\infty]$ is a certain answer to q_{S} over $\mathcal{K}_e$ if and only if $m \leq 3$.*

We now formalize the sketched approach. In order to abstract from specific ABox individuals, we introduce types.

Definition 53. *A* type *for a TBox $\mathcal{T}$ is a subset of $\mathsf{sig}(\mathcal{T})^{\pm}_{\mathsf{C}}$. The set of all types is $\Theta_{\mathcal{T}} = 2^{\mathsf{sig}(\mathcal{T})^{\pm}_{\mathsf{C}}}$. We denote by $\theta_{\mathcal{K}}(d)$ the* type of a domain element d w.r.t. $\mathcal{K}$ *and define it by: $\theta_{\mathcal{K}}(d) = \left\{\mathrm{B} \in \mathsf{sig}(\mathcal{T})^{\pm}_{\mathsf{C}} \mid \mathcal{K} \models \mathrm{B}(d)\right\}$ if $d \in \mathsf{Ind}(\mathcal{A})$, else $\theta_{\mathcal{K}}(d) = \emptyset$.*

Example 21. *In our running example, $\theta_{\mathcal{K}_e}(\mathsf{a}_1) = \{\mathrm{A}_1, \exists\mathrm{R}_1, \exists\mathrm{T}_1\}$ and $\theta_{\mathcal{K}_e}(\alpha) = \emptyset$ (since $\alpha \notin \mathsf{Ind}(\mathcal{A}_e)$).*

We use types to define strategies, which indicate for each generated role R the type onto which all elements $w\mathrm{R}$ should merge and whether roles with the same target type should or should not be mapped onto the same element. Several copies of a type might be required to comply with negative inclusions (e.g. R_1 and R_2 associated to the same type but the TBox satifies $\mathcal{T} \models \exists\mathrm{R}_1^- \sqsubseteq \neg\exists\mathrm{R}_2^-$).



(a) Another model of $\mathcal{K}_e$.   (b) Interpretation of strategy σ_e.

Figure 5.4: Finite models of the example KB $\mathcal{K}_e$. For readability, we have omitted concepts and highlighted the role S from the cardinality query.

Definition 54. *A* strategy *σ for the TBox $\mathcal{T}$ is a partial function from $\mathsf{sig}(\mathcal{T})^{\pm}_{\mathsf{R}}$ to $\Theta_{\mathcal{T}} \times \{1, \dots |\mathsf{sig}(\mathcal{T})^{\pm}_{\mathsf{R}}|\}$, satisfying the following two conditions:*

1. *For all $\mathrm{R} \in \mathsf{dom}(\sigma)$, if $\sigma(\mathrm{R}) = (\mathfrak{t}, i)$, then $\mathcal{T} \not\models \exists \mathrm{R}^- \sqsubseteq \neg \mathrm{B}$ for all $\mathrm{B} \in \mathfrak{t}$.*
2. *For all $\mathrm{R}_1, \mathrm{R}_2 \in \mathsf{dom}(\sigma)$, if $\sigma(\mathrm{R}_1) = \sigma(\mathrm{R}_2)$, then $\mathcal{T} \not\models \exists \mathrm{R}_1^- \sqsubseteq \neg \exists \mathrm{R}_2^-$.*

Where $\mathsf{dom}(\sigma)$ denotes the subset of $\mathsf{sig}(\mathcal{T})^{\pm}_{\mathsf{R}}$ on which σ is defined.

This notion only depending on the TBox, the number of possible strategies is *constant* w.r.t. data complexity. However, it also means a given strategy might be irrelevant for a particular ABox, as it may require more copies of a type than the ABox can provide. This motivates the following notion of *legal strategy*.

Definition 55. *Let $\mathcal{A}$ be an ABox and $\mathcal{K} := (\mathcal{T}, \mathcal{A})$. A strategy σ for $\mathcal{T}$ is* legal *for $\mathcal{K}$ if it satisfies the following two conditions:*

1. *Coverage: $\mathsf{gen}_{\mathcal{K}} = \mathsf{dom}(\sigma)$*
2. *Availability: For all $\mathfrak{t} \in \Theta_{\mathcal{T}}$, if $\mathfrak{t} \neq \emptyset$, then*

$$|\{i \mid \exists \mathrm{R} \in \mathsf{gen}_{\mathcal{K}}, \sigma(\mathrm{R}) = (\mathfrak{t}, i)\}| \leq |\{\mathsf{a} \mid \mathsf{a} \in \mathsf{Ind}(\mathcal{A}) \wedge \theta_{\mathcal{K}}(\mathsf{a}) = \mathfrak{t}\}| \, .$$

Condition 1 ensures roles for which the strategy is defined are matching those encountered in $\mathcal{C}_{\mathcal{K}}$, while Condition 2 requires the ABox provides at least as many individuals of a non-empty type as the strategy requires copies of this type.

Example 22. *The following mapping σ_e is a legal strategy for $\mathcal{K}_e$:*

$$\begin{array}{lllllll} \mathrm{T}_1 & \mapsto & (\emptyset, 1) & \quad \mathrm{R}_2 & \mapsto & (\{\mathrm{B}_1, \exists\mathrm{R}_1, \exists\mathrm{S}, \exists\mathrm{S}^-\}, 1) \\ \mathrm{T}_2 & \mapsto & (\emptyset, 2) & \quad \mathrm{S} & \mapsto & (\emptyset, 2) \\ \mathrm{R}_1 & \mapsto & (\emptyset, 2) & \quad \mathrm{S}^- & \mapsto & (\{\mathrm{A}_1, \exists\mathrm{R}_1, \exists\mathrm{T}_1\}, 1) \end{array}$$

To construct a model from a legal strategy σ, the basic idea is to merge elements $w\mathrm{R}$ with an element of type $\sigma(\mathrm{R})$, with the latter selected according to a *choice of well-typed elements*:

Definition 56. *A mapping* $\mathsf{ch} : \mathsf{gen}_{\mathcal{K}} \to \mathsf{Ind}(\mathcal{A}) \uplus \{\bot_i \mid i = 1, \ldots, \left|\mathsf{sig}(\mathcal{T})^{\pm}_{\mathsf{R}}\right|\}$, *is a* choice of well-typed elements for σ over $\mathcal{K}$ *if it satisfies the following conditions:*

1. *For all* $\mathrm{R} \in \mathsf{gen}_{\mathcal{K}}$, *there exists* $1 \leq i \leq \left|\mathsf{sig}(\mathcal{T})^{\pm}_{\mathsf{R}}\right|$ *such that* $\sigma(\mathrm{R}) = (\theta_{\mathcal{K}}(\mathsf{ch}(\mathrm{R})), i)$.

2. *For all* $\mathrm{R}_1, \mathrm{R}_2 \in \mathsf{gen}_{\mathcal{K}}$, *we have* $\mathsf{ch}(\mathrm{R}_1) = \mathsf{ch}(\mathrm{R}_2)$ *iff* $\sigma(\mathrm{R}_1) = \sigma(\mathrm{R}_2)$.

Example 23. *The function* ch_e*, defined as below, is a choice of well-typed elements for* σ_e *over* $\mathcal{K}_e$*:*

$$\begin{array}{lllllllll} \mathrm{T}_1 & \mapsto & \bot_1 & \mathrm{T}_2 & \mapsto & \bot_2 & \mathrm{R}_1 & \mapsto & \bot_2 \\ \mathrm{R}_2 & \mapsto & \mathsf{b}_1 & \mathrm{S} & \mapsto & \bot_2 & \mathrm{S}^- & \mapsto & \mathsf{a}_1 \end{array}$$

When reusing an element w.r.t. a strategy, we often take advantage of possible existing S or S^- edges involving this element. Choosing such an edge, when it exists, motivates the following definitions.

Definition 57. *For every* $\mathrm{R} \in \mathsf{sig}(\mathcal{T})^{\pm}_{\mathsf{R}}$, *pick a function* $\mathsf{succ}^{\mathcal{K}}_{\mathrm{R}}$ *that maps every individual in* $\{\mathsf{a} \mid \mathcal{K} \models \mathrm{R}(\mathsf{a}, \mathsf{b})$ *for some* $\mathsf{b} \in \mathsf{N_I}\}$ *to an individual* $\mathsf{succ}^{\mathcal{K}}_{\mathrm{R}}(\mathsf{a})$ *such that* $\mathcal{K} \models \mathrm{R}(\mathsf{a}, \mathsf{succ}^{\mathcal{K}}_{\mathrm{R}}(\mathsf{a}))$. *The family of functions* $(\mathsf{succ}^{\mathcal{K}}_{\mathrm{R}})_{\mathrm{R}}$ *is called a* certain successor preference. *Similarly for a given interpretation* $\mathcal{I}$, *a family of functions* $(\mathsf{succ}^{\mathcal{I}}_{\mathrm{R}})_{\mathrm{R}}$ *mapping an element* $d \in (\exists\mathrm{R})^{\mathcal{I}}$ *to an element* $e \in \Delta^{\mathcal{I}}$ *such that* $(d, e) \in \mathrm{R}^{\mathcal{I}}$ *is a* successor preference in $\mathcal{I}$.

It turns out however that when $\mathrm{R} = \mathrm{S}$ or $\mathrm{R} = \mathrm{S}^-$, it is useful to depart from the guideline of a strategy in order to reduce the number of query matches, as this stand-alone example illustrates:

Example 24. *Consider the canonical model of the KB formed by the TBox* $\mathcal{T} = \{\mathrm{A} \sqsubseteq \exists\mathrm{S}, \mathrm{B} \sqsubseteq \exists\mathrm{S}^-\}$ *and the ABox* $\mathcal{A} = \{\mathrm{A}(\mathsf{a}_1), \mathrm{A}(\mathsf{a}_2), \mathrm{B}(\mathsf{b}_1), \mathrm{B}(\mathsf{b}_2)\}$. *If we merge* $\mathsf{a}_1\mathrm{S}$ *with* $\mathsf{a}_2\mathrm{S}$, *and* $\mathsf{b}_1\mathrm{S}^-$ *with* $\mathsf{b}_2\mathrm{S}^-$, *then there will be at least three matches of* q_{S}, *no matter which further merges are performed. However, by 'pairing'* a_1 *with* b_1 *and* a_2 *with* b_2, *we can obtain a model with only two matches:* $(\mathsf{a}_1, \mathsf{b}_1), (\mathsf{a}_2, \mathsf{b}_2)$.

The next three definitions serve to identify the *critical elements* for which such a pairing operation is useful. Notice they are similar in spirit to those used within the proof of Theorem 46, though slightly different for roles.

Definition 58. *We define positive (resp. negative) demanding individuals as:*

$$\mathcal{D}^+_\mathcal{K} := \{\mathsf{a} \mid \mathsf{a} \in \mathsf{Ind}(\mathcal{A}) \text{ and } \mathsf{a}\mathrm{S} \in \Delta^{\mathcal{C}_\mathcal{K}}\} \qquad \mathcal{D}^-_\mathcal{K} := \{\mathsf{a} \mid \mathsf{a} \in \mathsf{Ind}(\mathcal{A}) \text{ and } \mathsf{a}\mathrm{S}^- \in \Delta^{\mathcal{C}_\mathcal{K}}\}$$

Definition 59. *Given a strategy σ, we define positive (resp. negative) demanding roles as:*

$$\mathcal{D}^+_\sigma = \{\mathrm{R} \in \mathsf{dom}(\sigma) \setminus \{\mathrm{S}, \mathrm{S}^-\} \mid \mathcal{T} \models \exists \mathrm{R}^- \sqsubseteq \exists \mathrm{S} \text{ and } \exists \mathrm{S} \notin \mathfrak{t} \text{ where } (\mathfrak{t}, k) := \sigma(\mathrm{R})\}$$

and

$$\mathcal{D}^-_\sigma = \{\mathrm{R} \in \mathsf{dom}(\sigma) \setminus \{\mathrm{S}, \mathrm{S}^-\} \mid \mathcal{T} \models \exists \mathrm{R}^- \sqsubseteq \exists \mathrm{S}^- \text{ and } \exists \mathrm{S}^- \notin \mathfrak{t} \text{ where } (\mathfrak{t}, k) := \sigma(\mathrm{R})\}.$$

Definition 60. *Let ch be a choice of well-typed elements for σ. We define positive (resp. negative) critical elements as:*

$$\mathsf{crit}^+ := \mathcal{D}^+_\mathcal{K} \cup \mathsf{ch}(\mathcal{D}^+_\sigma) \qquad \mathsf{crit}^- := \mathcal{D}^-_\mathcal{K} \cup \mathsf{ch}(\mathcal{D}^-_\sigma)$$

Example 25. *For σ_e and ch_e as defined in Examples 22 and 23, we have* $\mathsf{crit}^+ = \{\mathsf{a}_2, \mathsf{b}_1, \bot_1, \bot_2\}$ *and* $\mathsf{crit}^- = \{\mathsf{a}_2, \bot_2\}$.

Intuitively, a pairing matches critical elements from crit^+ (which require an outgoing S) with those from crit^- (which require an incoming S).

Definition 61. *A* pairing for ch and σ *consists of two partial functions* $\mathsf{p}^+ : \mathsf{crit}^+ \to \mathsf{crit}^-$ *and* $\mathsf{p}^- : \mathsf{crit}^- \to \mathsf{crit}^+$ *such that one of the functions is total and injective, and the other is its partial inverse.*

Example 26. *A pairing for ch_e and σ_e is given by* $\mathsf{p}^+_e = \{\mathsf{a}_2 \mapsto \mathsf{a}_2, \mathsf{b}_1 \mapsto \bot_2\}$ *and* $\mathsf{p}^-_e = \{\mathsf{a}_2 \mapsto \mathsf{a}_2, \bot_2 \mapsto \mathsf{b}_1\}$.

We are now ready to define the interpretation of a strategy.

Definition 62. *Consider a strategy σ, choice of well-typed elements ch, certain successor preference $(\mathsf{succ}^\mathcal{K}_\mathrm{R})_\mathrm{R}$ and pairing $(\mathsf{p}^+, \mathsf{p}^-)$ for ch. Define function χ as follows:*

$$\begin{array}{rcl}
\Delta^{\mathcal{C}_\mathcal{K}} & \to & \mathsf{Ind}(\mathcal{A}) \cup \{\bot_i \mid i = 1, \ldots, |\mathsf{sig}(\mathcal{T})^\pm_\mathsf{R}|\} \\
\mathsf{a} & \mapsto & \mathsf{a} \\
w\mathrm{S} & \mapsto & \begin{cases} \mathsf{succ}^\mathcal{K}_\mathrm{S}(\chi(w)) & \text{if } \mathsf{succ}^\mathcal{K}_\mathrm{S}(\chi(w)) \text{ is defined} \\ \mathsf{p}^+(\chi(w)) & \text{else if } \mathsf{p}^+(\chi(w)) \text{ is defined} \\ \mathsf{ch}(\mathrm{S}) & \text{otherwise} \end{cases} \\
w\mathrm{S}^- & \mapsto & \begin{cases} \mathsf{succ}^\mathcal{K}_{\mathrm{S}^-}(\chi(w)) & \text{if } \mathsf{succ}^\mathcal{K}_{\mathrm{S}^-}(\chi(w)) \text{ is defined} \\ \mathsf{p}^-(\chi(w)) & \text{else if } \mathsf{p}^-(\chi(w)) \text{ is defined} \\ \mathsf{ch}(\mathrm{S}^-) & \text{otherwise} \end{cases} \\
w\mathrm{R} & \mapsto & \mathsf{ch}(\mathrm{R})
\end{array}$$

The interpretation of σ *(according to $\mathsf{ch}, (\mathsf{p}^+, \mathsf{p}^-)$ and the $\mathsf{succ}^\mathcal{K}_\mathrm{R}$) has domain $\chi(\Delta^{\mathcal{C}_\mathcal{K}})$ and interpretation function $\chi \circ \cdot^{\mathcal{C}_\mathcal{K}}$.*

Example 27. *With choice* ch_e *and pairing* $(\mathsf{p}_e^+, \mathsf{p}_e^-)$*, we get* $\chi(\mathsf{b}_2\mathrm{R}_2) = \mathsf{ch}(\mathrm{R}_2) = \mathsf{b}_1$*,* $\chi(\mathsf{b}_2\mathrm{R}_2\mathrm{S}) = \mathsf{p}_e^+(\mathsf{b}_1) = \bot_2$*, and* $\chi(\mathsf{b}_2\mathrm{R}_2\mathrm{S}^-) = \mathsf{succ}^{\mathcal{K}}_{\mathrm{S}^-}(\mathsf{b}_1) = \mathsf{b}_2$ *(observe that on our example, the function* $\mathsf{succ}^{\mathcal{K}}_{\mathrm{S}^-}$ *is uniquely defined, and the same is true for the other roles). Figure 5.4b displays the interpretation of* σ_e*.*

Observe that the interpretation of a strategy σ depends not only on σ but also on the functions $\mathsf{ch}, \mathsf{p}^+, \mathsf{p}^-, \mathsf{succ}^{\mathcal{K}}_{\mathrm{R}}$. Importantly, however, the key property of such interpretations (stated in Lemma 29 later in this section) holds for *any* particular choice of these functions.

It remains to prove that a model minimizing the number of matches can be found among the interpretations of strategies. The first step is to extract a strategy from a model.

Definition 63. *Let* $\mathcal{I}$ *be a model of* $\mathcal{K}$*,* $f : \mathcal{C}_{\mathcal{K}} \to \mathcal{I}$ *be a homomorphism, and* repr *be a function mapping each role* $\mathrm{R} \in \mathsf{gen}_{\mathcal{K}}$ *to an element with shape* $w\mathrm{R}$ *from* $\Delta^{\mathcal{C}_{\mathcal{K}}}$*. Then* $\mathcal{P} = \{P_1, \ldots, P_k\}$*, defined by*

$$\{P_1, \ldots, P_k\} = \{(f \circ \mathsf{repr})^{-1}(w) \mid w \in \Delta^{\mathcal{I}}\} \setminus \{\emptyset\}$$

is a partition of $\mathsf{gen}_{\mathcal{K}}$*. The strategy extracted from* $\mathcal{I}$ *(for* f *and* repr*) is defined as:*

$$\begin{array}{rcl} \mathsf{gen}_{\mathcal{K}} & \to & \Theta_{\mathcal{T}} \times \{1, \ldots, |\mathsf{sig}(\mathcal{T})^{\pm}_{\mathsf{R}}|\} \\ \mathrm{R} & \mapsto & ((\theta_{\mathcal{K}} \circ f \circ \mathsf{repr})(\mathrm{R}), i) \text{ with } \mathrm{R} \in P_i \end{array}$$

Example 28. *In our running example, there is a unique homomorphism* f_e *from* $\mathcal{C}_{\mathcal{K}_e}$ *to the model displayed in Figure 5.4a. Let* repr_e *be:*

$$\begin{array}{lcl lcl lcl} \mathrm{T}_1 & \mapsto & \mathsf{a}_1\mathrm{T}_1 & \mathrm{R}_2 & \mapsto & \mathsf{b}_2\mathrm{R}_2 & \mathrm{T}_2 & \mapsto & \mathsf{a}_2\mathrm{T}_2 \\ \mathrm{S} & \mapsto & \mathsf{b}_1\mathrm{SSS} & \mathrm{R}_1 & \mapsto & \mathsf{b}_1\mathrm{R}_1 & \mathrm{S}^- & \mapsto & \mathsf{a}_2\mathrm{S}^- \end{array}$$

The strategy extracted from this model (for f_e *and* repr_e*) is the strategy provided in Example 22.*

We further introduce the following useful lemma, stating that a choice of well-typed elements for a strategy σ extracted from a model provides, as one would expect, elements with the same type as those used in the first place to extract the strategy σ.

Lemma 28. *Let* $\mathcal{A}$ *be an ABox and* $\mathcal{K} := (\mathcal{T}, \mathcal{A})$*. Let* $\mathsf{repr}_{\mathcal{K}}$ *be a function mapping each role* $\mathrm{R} \in \mathsf{gen}_{\mathcal{K}}$ *to an element with shape* $w\mathrm{R}$ *from* $\Delta^{\mathcal{C}_{\mathcal{K}}}$*. Let* $\mathcal{I}$ *be a model of* $\mathcal{K}$*, and let* $f : \mathcal{C}_{\mathcal{K}} \to \mathcal{I}$ *be a homomorphism. Let* $\mathsf{ch}_{\sigma/\mathcal{K}}$ *be a choice of well-typed elements for* $\sigma_{f \circ \mathsf{repr}_{\mathcal{K}}}$ *over* $\mathcal{A}$*. The strategy* $\sigma_{f \circ \mathsf{repr}_{\mathcal{K}}}$ *extracted from* $\mathcal{I}$ *(for* f *and* $\mathsf{repr}_{\mathcal{K}}$*) preserves both:*

1. $\forall \mathrm{R} \in \mathsf{gen}_{\mathcal{K}},\ \theta_{\mathcal{K}}(\mathsf{ch}_{\sigma_{f \circ \mathsf{repr}_{\mathcal{K}}}/\mathcal{K}}(\mathrm{R})) = \theta_{\mathcal{K}}(f(\mathsf{repr}_{\mathcal{K}}(\mathrm{R})))$

2. $\forall \mathrm{R}, \mathrm{T} \in \mathsf{gen}_{\mathcal{K}},\ \mathsf{ch}_{\sigma_{f \circ \mathsf{repr}_{\mathcal{K}}}/\mathcal{K}}(\mathrm{R}) = \mathsf{ch}_{\sigma_{f \circ \mathsf{repr}_{\mathcal{K}}}/\mathcal{K}}(\mathrm{T}) \ \Leftrightarrow\ f(\mathsf{repr}_{\mathcal{K}}(\mathrm{R})) = f(\mathsf{repr}_{\mathcal{K}}(\mathrm{T}))$

Proof. 1. Let $\mathrm{R} \in \mathsf{gen}_{\mathcal{K}}$. By definition of $\sigma_{f \circ \mathsf{repr}_{\mathcal{K}}}$, there exists $i \in \{1, \ldots |\mathsf{sig}(\mathcal{T})|\}$ such that $\sigma_{f \circ \mathsf{repr}_{\mathcal{K}}}(\mathrm{R}) = ((\theta_{\mathcal{K}} \circ f \circ \mathsf{repr}_{\mathcal{K}})(\mathrm{R}), i)$ with $\mathrm{R} \in P_i$. From Condition 1 of Definition 56, we get $\sigma_{f \circ \mathsf{repr}_{\mathcal{K}}}(\mathrm{R}) = (\theta_{\mathcal{K}}(\mathsf{ch}_{\sigma_{f \circ \mathsf{repr}_{\mathcal{K}}}/\mathcal{K}}(\mathrm{R})), i)$, which gives the desired equality of types.

2. From Condition 2 of Definition 56, $\mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{R}) = \mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{T})$ iff $\sigma_{f \circ \mathsf{repr}_{\mathcal{K}}}(\mathrm{R}) = \sigma_{f \circ \mathsf{repr}_{\mathcal{K}}}(\mathrm{T})$. If $f(\mathsf{repr}_{\mathcal{K}}(\mathrm{R})) = f(\mathsf{repr}_{\mathcal{K}}(\mathrm{T}))$, then by the definition of the extracted strategy, we have $\sigma_{f \circ \mathsf{repr}_{\mathcal{K}}}(\mathrm{R}) = \sigma_{f \circ \mathsf{repr}_{\mathcal{K}}}(\mathrm{T})$, so we are done. Conversely, if $\mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{R}) = \mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{T})$, then $\sigma_{f \circ \mathsf{repr}_{\mathcal{K}}}(\mathrm{R}) = \sigma_{f \circ \mathsf{repr}_{\mathcal{K}}}(\mathrm{T})$. This implies in particular that R and T belong to the same P_i, hence $f(\mathsf{repr}_{\mathcal{K}}(\mathrm{R})) = f(\mathsf{repr}_{\mathcal{K}}(\mathrm{T}))$. □

By applying the next lemma to a model $\mathcal{I}$ having the fewest possible number of matches, we obtain the desired conclusion: there is a model minimizing the number of matches among the models obtained by interpreting a strategy.

Lemma 29. *Let $\mathcal{I}$ be a model of $\mathcal{K}$, and $\mathcal{J}$ an interpretation of a strategy extracted from $\mathcal{I}$. $\mathcal{J}$ is a model of $\mathcal{K}$ and $q_{\mathrm{S}}^{\mathcal{J}} \leq q_{\mathrm{S}}^{\mathcal{I}}$.*

We first prove the first point of Lemma 29, stating the interpretation of a strategy extracted from a model is also a model, in the following stronger form, which does not require the strategy to be extracted from a model in the first place.

Lemma 30. *Let $\mathcal{A}$ be an ABox and $\mathcal{K} := (\mathcal{T}, \mathcal{A})$ a satisfiable KB. Let $(\mathsf{succ}_{\mathrm{R}}^{\mathcal{K}})_{\mathrm{R}}$ be a certain successor preference. Let σ be a legal strategy for $\mathcal{K}$. Let $\mathsf{ch}_{\sigma/\mathcal{K}}$ be a choice of well-typed elements for σ over $\mathcal{K}$. Let $\mathsf{pair}_{\mathsf{ch}_{\sigma/\mathcal{K}}} := (\mathsf{pair}^{+}_{\mathsf{ch}_{\sigma/\mathcal{K}}}, \mathsf{pair}^{-}_{\mathsf{ch}_{\sigma/\mathcal{K}}})$ be a pairing for $\mathsf{ch}_{\sigma/\mathcal{K}}$ and σ. Then the interpretation $\mathcal{J}$ of σ (according to $\mathsf{ch}_{\sigma/\mathcal{K}}$, $\mathsf{pair}_{\mathsf{ch}_{\sigma/\mathcal{K}}}$, and $(\mathsf{succ}_{\mathrm{R}}^{\mathcal{K}})_{\mathrm{R}}$) is a model.*

Proof. Assertions from the ABox and positive inclusions from $\mathcal{T}$ are satisfied since the interpretation $\mathcal{J}$ is built from $\mathcal{C}_{\mathcal{K}}$. Indeed, suppose that $\mathrm{B} \sqsubseteq \mathrm{C} \in \mathcal{T}$ and $d \in \mathrm{B}^{\mathcal{J}}$. Then from the definition of $\mathcal{J}$, there exists $w \in \Delta^{\mathcal{C}_{\mathcal{K}}}$ such that $w \in \mathrm{B}^{\mathcal{C}_{\mathcal{K}}}$ and $d = \chi(w)$. Since $\mathcal{C}_{\mathcal{K}}$ satisfies $\mathrm{B} \sqsubseteq \mathrm{C}$, we have $w \in \mathrm{C}^{\mathcal{C}_{\mathcal{K}}}$. If $\mathrm{C} \in \mathsf{N_C}$, this immediately gives $d \in \mathrm{C}^{\mathcal{J}}$. If $\mathrm{C} = \exists \mathrm{R}$, there exists w' such that $(w, w') \in \mathrm{R}^{\mathcal{C}_{\mathcal{K}}}$, and hence we will have $(\chi(w), \chi(w')) \in R^{\mathcal{J}}$, which yields $w \in \mathrm{C}^{\mathcal{C}_{\mathcal{K}}}$.

Consider now a negative axiom of the form $\mathrm{B} \sqsubseteq \neg \mathrm{C}$. By contradiction assume there is an element d such that $d \in \mathrm{B}^{\mathcal{J}} \cap \mathrm{C}^{\mathcal{J}}$. In what follows, χ is the function used in the definition of $\mathcal{J}$.

1. If $\mathcal{K} \models \mathrm{B}(d)$ and $\mathcal{K} \models \mathrm{C}(d)$, then this contradicts $\mathcal{K}$ being satisfiable.

2. If $\mathcal{K} \models \mathrm{B}(d)$ and $\mathcal{K} \not\models \mathrm{C}(d)$, then $d = \chi(w\mathrm{T})$ with $\mathrm{T} \in \mathsf{gen}_{\mathcal{K}}$ and $\mathcal{T} \models \exists \mathrm{T}^- \sqsubseteq \mathrm{C}$. Indeed $\mathcal{K} \not\models \mathrm{C}(d)$ ensures d is not the image through χ of some individual certainly satisfiying C. Nevertheless, since $d \in \mathrm{C}^{\mathcal{J}}$, it must be that d is the image through χ of some anonymous element, say $w\mathrm{T}$, such that $w\mathrm{T} \in \mathrm{C}^{\mathcal{C}}_{\mathcal{K}}$. By definition of $\mathrm{C}^{\mathcal{C}}_{\mathcal{K}}$, it yields $\mathcal{T} \models \exists \mathrm{T}^- \sqsubseteq \mathrm{C}$.

   (a) If $d = \mathsf{succ}^{\mathcal{K}}_{\mathrm{T}}(\chi(w))$ with $\mathrm{T} \in \{\mathrm{S}, \mathrm{S}^-\}$, then $\mathrm{T}(\chi(w), d) \in \mathcal{A}$, contradicting $\mathcal{K} \not\models \mathrm{C}(d)$.

   (b) If $d = \mathsf{pair}^+_{\mathsf{ch}_{\sigma/\mathcal{K}}}(\chi(w))$ with $\mathrm{T} = \mathrm{S}$, then in particular $d \in \mathsf{crit}^-_{\mathsf{ch}_{\sigma/\mathcal{K}}}$.

   - If $d \in \mathcal{D}^-_{\mathcal{K}}$, then in particular $\mathcal{K} \models \exists \mathrm{S}^-(d)$, contradicting $\mathcal{K} \not\models \mathrm{C}(d)$.
   - If $d = \mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{T}_0)$ with $\mathrm{T}_0 \in \mathcal{D}^-_{\sigma}$, then in particular $\mathcal{T} \models \exists \mathrm{T}_0^- \sqsubseteq \exists \mathrm{S}^-$. Hence $\mathcal{T} \models \exists \mathrm{T}_0^- \sqsubseteq \neg \mathrm{B}$. Condition 1 in the definition of a choice of well-typed elements ensures $\sigma(\mathrm{T}_0) = (\theta_{\mathcal{K}}(d), i)$. Condition 1 in the definition of a strategy ensures $\mathrm{B} \notin \theta_{\mathcal{K}}(d)$, contradicting $\mathcal{K} \models \mathrm{B}(d)$.

   (c) If $d = \mathsf{pair}^-_{\mathsf{ch}_{\sigma/\mathcal{K}}}(\chi(w))$ with $\mathrm{T} = \mathrm{S}^-$. Same argument as in Case 2.b, based on $d \in \mathsf{crit}^+_{\mathsf{ch}_{\sigma/\mathcal{K}}}$.

   (d) If $d = \mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{T})$. Condition 1 from the definition of choice of well-typed elements ensures $\sigma(\mathrm{T}) = (\theta_{\mathcal{K}}(d), i)$. Condition 1 from the definition of a strategy ensures $\mathrm{B} \notin \theta_{\mathcal{K}}(d)$, contradicting $\mathcal{K} \models \mathrm{B}(d)$.

3. If $\mathcal{K} \not\models \mathrm{B}(d)$ and $\mathcal{K} \models \mathrm{C}(d)$. Symmetric to Case 2.

4. If $\mathcal{K} \not\models \mathrm{B}(d)$ and $\mathcal{K} \not\models \mathrm{C}(d)$, then $d = \chi(w\mathrm{R}) = \chi(w'\mathrm{T})$ with $\mathrm{R}, \mathrm{T} \in \mathsf{gen}_{\mathcal{K}}$ such that $\mathcal{T} \models \exists \mathrm{R}^- \sqsubseteq \mathrm{B}$ and $\mathcal{T} \models \exists \mathrm{T}^- \sqsubseteq \mathrm{C}$, due to the same reason than in Case 2, applied here to both concepts B and C.

   (a) If $d = \mathsf{succ}^{\mathcal{K}}_{\mathrm{R}}(\chi(w))$ with $\mathrm{R} \in \{\mathrm{S}, \mathrm{S}^-\}$, then $\mathrm{R}(\chi(w), d) \in \mathcal{A}$, contradicting $\mathcal{K} \not\models \mathrm{B}(d)$.

   (b) If $d = \mathsf{pair}^+_{\mathsf{ch}_{\sigma/\mathcal{K}}}(\chi(w))$ with $\mathrm{R} = \mathrm{S}$, then in particular $d \in \mathsf{crit}^-_{\mathsf{ch}_{\sigma/\mathcal{K}}}$.

   - If $d \in \mathcal{D}^-_{\mathcal{K}}$, then it contradicts $\mathcal{K} \not\models \mathrm{B}(d)$.
   - If $d = \mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{R}_0)$ with $\mathrm{R}_0 \in \mathcal{D}^-_{\sigma}$, then in particular $\mathcal{T} \models \exists \mathrm{R}_0^- \sqsubseteq \exists \mathrm{S}^-$.
     i. If $d = \mathsf{succ}^{\mathcal{K}}_{\mathrm{T}}(\chi(w'))$ with $\mathrm{T} \in \{\mathrm{S}, \mathrm{S}^-\}$, then it contradicts $\mathcal{K} \not\models \mathrm{C}(d)$.
     ii. If $d = \mathsf{pair}^+_{\mathsf{ch}_{\sigma/\mathcal{K}}}(\chi(w'))$ with $\mathrm{T} = \mathrm{S}$, then $\mathcal{T} \models \exists \mathrm{S}^- \sqsubseteq \neg \exists \mathrm{S}^-$. Contradiction.
     iii. If $d = \mathsf{pair}^-_{\mathsf{ch}_{\sigma/\mathcal{K}}}(\chi(w'))$ with $\mathrm{T} = \mathrm{S}^-$, in particular $d \in \mathsf{crit}^+_{\mathsf{ch}_{\sigma/\mathcal{K}}}$.
       - If $d \in \mathcal{D}^+_{\mathcal{K}}$ then it contradicts $\mathcal{K} \not\models \mathrm{C}(d)$.

- If $d = \mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{T}_0)$ with $\mathrm{T}_0 \in \mathcal{D}^+_\sigma$, then in particular $\mathcal{T} \models \exists \mathrm{T}_0^- \sqsubseteq \exists \mathrm{S}$. Condition 2 in the definition of the choice of well-typed elements ensures: $\sigma(\mathrm{R}_0) = \sigma(\mathrm{T}_0)$. Condition 2 in the definition of a strategy ensures: $\mathcal{T} \not\models \exists \mathrm{R}_0^- \sqsubseteq \neg\exists \mathrm{T}_0^-$, contradicting $\mathcal{T} \models \mathrm{B} \sqsubseteq \neg\mathrm{C}$.

iv. If $d = \mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{T})$. Condition 2 in the definition of the choice of well-typed elements ensures: $\sigma(\mathrm{R}_0) = \sigma(\mathrm{T})$. Condition 2 in the definition of a strategy ensures: $\mathcal{T} \not\models \exists \mathrm{R}_0^- \sqsubseteq \neg\exists \mathrm{T}^-$, contradicting $\mathcal{T} \models \mathrm{B} \sqsubseteq \neg\mathrm{C}$.

(c) If $d = \mathsf{pair}^-_{\mathsf{ch}_{\sigma/\mathcal{K}}}(\chi(w))$ with $\mathrm{R} = \mathrm{S}^-$. Analogous argument to Case 4.b.

(d) If $d = \mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{R})$.

i. If $d = \mathsf{succ}^{\mathcal{K}}_{\mathrm{T}}(\chi(w'))$ with $\mathrm{T} \in \{\mathrm{S}, \mathrm{S}^-\}$, then it contradicts $\mathcal{K} \not\models \mathrm{C}(d)$.

ii. If $d = \mathsf{pair}^+_{\mathsf{ch}_{\sigma/\mathcal{K}}}(\chi(w'))$ with $\mathrm{T} = \mathrm{S}$. Symmetric to Case 4.b.iv.

iii. If $d = \mathsf{pair}^-_{\mathsf{ch}_{\sigma/\mathcal{K}}}(\chi(w'))$ with $\mathrm{T} = \mathrm{S}^-$. Symmetric to Case 4.c.iv.

iv. If $d = \mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{T})$. Condition 2 in the definition of the choice of well-typed elements ensures: $\sigma(\mathrm{R}) = \sigma(\mathrm{T})$. Condition 2 in the definition of a strategy ensures: $\mathcal{T} \not\models \exists \mathrm{R}^- \sqsubseteq \neg\exists \mathrm{T}^-$, contradicting $\mathcal{T} \models \mathrm{B} \sqsubseteq \neg\mathrm{C}$. □

In order to prove the second point of Lemma 28, stating that the interpretation $\mathcal{J}$ of the strategy extracted from a model $\mathcal{I}$ has at most as many matches as the initial model $\mathcal{I}$, we need to understand which pairs appear in the role of interest S in an interpretation of our strategy. This is the purpose of the following result.

Lemma 31. *Let $\mathcal{A}$ be an ABox and $\mathcal{K} := (\mathcal{T}, \mathcal{A})$. Let $(\mathsf{succ}^{\mathcal{K}}_{\mathrm{R}})_{\mathrm{R}}$ be a certain successor preference. Let σ be a legal strategy for $\mathcal{K}$. Let $\mathsf{ch}_{\sigma/\mathcal{K}}$ be a choice of well-typed elements for σ over $\mathcal{K}$. Let $\mathsf{pair}_{\mathsf{ch}_{\sigma/\mathcal{K}}} := (\mathsf{pair}^+_{\mathsf{ch}_{\sigma/\mathcal{K}}}, \mathsf{pair}^-_{\mathsf{ch}_{\sigma/\mathcal{K}}})$ be a pairing for $\mathsf{ch}_{\sigma/\mathcal{K}}$. Denote by $\mathcal{J}$ the interpretation of σ (according to $\mathsf{ch}_{\sigma/\mathcal{K}}$, $\mathsf{pair}_{\mathsf{ch}_{\sigma/\mathcal{K}}}$, and*

$(\mathsf{succ}^{\mathcal{K}}_{\mathrm{R}})_{\mathrm{R}}$). *We have:*

$$
\begin{aligned}
\mathrm{S}^{\mathcal{J}} = \; & \{(\mathsf{a},\mathsf{b}) \mid \mathcal{K} \models \mathrm{S}(\mathsf{a},\mathsf{b})\} && \textit{Shape 1} \\
\cup \; & \left\{ (x,y) \;\middle|\; \begin{array}{l} (x,y) \in \mathsf{crit}^+_{\mathsf{ch}_{\sigma/\mathcal{K}}} \times \mathsf{crit}^-_{\mathsf{ch}_{\sigma/\mathcal{K}}} \\ \mathsf{pair}^+_{\mathsf{ch}_{\sigma/\mathcal{K}}}(x) = y \end{array} \right\} && \textit{Shape 2} \\
\cup \; & \left\{ (x, \mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{S})) \;\middle|\; x \in \mathsf{crit}^+_{\mathsf{ch}_{\sigma/\mathcal{K}}} \setminus \mathsf{dom}(\mathsf{pair}^+_{\mathsf{ch}_{\sigma/\mathcal{K}}}) \right\} && \textit{Shape 3}^+ \\
\cup \; & \left\{ (\mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{S}^-), y) \;\middle|\; y \in \mathsf{crit}^-_{\mathsf{ch}_{\sigma/\mathcal{K}}} \setminus \mathsf{dom}(\mathsf{pair}^-_{\mathsf{ch}_{\sigma/\mathcal{K}}}) \right\} && \textit{Shape 3}^- \\
\cup \; & \left\{ (\mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{S}), \mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{S})) \;\middle|\; \begin{array}{l} \left|\mathsf{crit}^+_{\mathsf{ch}_{\sigma/\mathcal{K}}}\right| > \left|\mathsf{crit}^-_{\mathsf{ch}_{\sigma/\mathcal{K}}}\right| \\ \mathcal{T} \models \exists \mathrm{S}^- \sqsubseteq \exists \mathrm{S} \\ \exists \mathrm{S} \notin \theta_{\mathcal{K}}(\mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{S})) \\ \mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{S}) \notin \mathsf{ch}_{\sigma/\mathcal{K}}(\mathcal{D}^+_\sigma) \end{array} \right\} && \textit{Shape 4}^+ \\
\cup \; & \left\{ (\mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{S}^-), \mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{S}^-)) \;\middle|\; \begin{array}{l} \left|\mathsf{crit}^-_{\mathsf{ch}_{\sigma/\mathcal{K}}}\right| > \left|\mathsf{crit}^+_{\mathsf{ch}_{\sigma/\mathcal{K}}}\right| \\ \mathcal{T} \models \exists \mathrm{S} \sqsubseteq \exists \mathrm{S}^- \\ \exists \mathrm{S}^- \notin \theta_{\mathcal{K}}(\mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{S}^-)) \\ \mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{S}^-) \notin \mathsf{ch}_{\sigma/\mathcal{K}}(\mathcal{D}^-_\sigma) \end{array} \right\} && \textit{Shape 4}^-
\end{aligned}
$$

Notice there can be no overlap between two distinct shapes and that shapes with opposite superscripts cannot coexist.

Proof sketch. The full proof can be found in the appendix and simply proceeds by case analysis based on the definition of the interpretation of the role S in the canonical model and on the function χ from Definition 62. □

We next need to understand how to relate the elements of $\mathcal{J}$ with the elements of the original model $\mathcal{I}$. In particular, we are interested in critical elements, which motivates the following definition.

Definition 64. *Let $\mathcal{A}$ be an ABox and $\mathcal{K} := (\mathcal{T}, \mathcal{A})$. Let $\mathsf{repr}_{\mathcal{K}}$ be function mapping each role $\mathrm{R} \in \mathsf{gen}_{\mathcal{K}}$ to an element with shape $w\mathrm{R}$ from $\Delta^{\mathcal{C}_{\mathcal{K}}}$. Let $\mathcal{I}$ be a model of $\mathcal{K}$ and $f : \mathcal{C}_{\mathcal{K}} \to \mathcal{I}$ be a homomorphism. Let σ be the strategy extracted from $\mathcal{I}$ (for f and $\mathsf{repr}_{\mathcal{K}}$). Let $\mathsf{ch}_{\sigma/\mathcal{K}}$ be a choice of well-typed elements for σ over $\mathcal{K}$. The* origins of critical elements *are given by:*

$$
\begin{aligned}
\mathsf{ori}^+ : \mathsf{crit}^+_{\mathsf{ch}_{\sigma/\mathcal{K}}} & \to \Delta^{\mathcal{I}} \\
x & \mapsto \begin{cases} x & \textit{if } x \in \mathcal{D}^+_{\mathcal{K}} \\ f(\mathsf{repr}_{\mathcal{K}}(\mathrm{R})) & \textit{if } x = \mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{R}) \textit{ with } \mathrm{R} \in \mathcal{D}^+_\sigma \end{cases} \\
\mathsf{ori}^- : \mathsf{crit}^-_{\mathsf{ch}_{\sigma/\mathcal{K}}} & \to \Delta^{\mathcal{I}} \\
y & \mapsto \begin{cases} y & \textit{if } y \in \mathcal{D}^-_{\mathcal{K}} \\ f(\mathsf{repr}_{\mathcal{K}}(\mathrm{T})) & \textit{if } y = \mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{T}) \textit{ with } \mathrm{T} \in \mathcal{D}^-_\sigma \end{cases}
\end{aligned}
$$

Notice the second point in Lemma 28 ensures that ori^+*, resp.* ori^-*, is well defined, that is, it does not depend on the choice of the role* R*, resp.* T.

Observe that this way of associating critical elements with elements of the original model is injective.

Lemma 32. *The functions* ori^+ *and* ori^- *as defined in Definition 64, are injective.*

Proof. Let $x, x' \in \mathsf{crit}^+_{\mathsf{ch}_{\sigma/\mathcal{K}}}$ such that $\mathsf{ori}^+(x) = \mathsf{ori}^+(x')$. We consider the four possible cases.

1. Suppose $x \in \mathcal{D}^+_{\mathcal{K}}$.
   (a) Suppose $x' \in \mathcal{D}^+_{\mathcal{K}}$.
   Trivial: $x = \mathsf{ori}^+(x) = \mathsf{ori}^+(x') = x'$.
   (b) Suppose $x' = \mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{R}')$ with $\mathrm{R}' \in \mathcal{D}^+_\sigma$.
   On the one hand, statement 1 from Lemma 28 ensures $\theta_{\mathcal{K}}(x') = \theta_{\mathcal{K}}(f(\mathsf{repr}_{\mathcal{K}}(\mathrm{R}')))$. Since by our assumptions we have $x = \mathsf{ori}^+(x) = \mathsf{ori}^+(x') = f(\mathsf{repr}_{\mathcal{K}}(\mathrm{R}')$, we get $\theta_{\mathcal{K}}(x) = \theta_{\mathcal{K}}(f(\mathsf{repr}_{\mathcal{K}}(\mathrm{R}'))$, hence $\theta_{\mathcal{K}}(x') = \theta_{\mathcal{K}}(x)$. Since $x \in \mathcal{D}^+_{\mathcal{K}}$, this means in particular that $\exists \mathrm{S} \in \theta_{\mathcal{K}}(x')$.
   On the other hand, $\mathsf{ch}_{\sigma/\mathcal{K}}$ must satisfy Condition 1 of the definition of choice of well-typed elements, so $\sigma(\mathrm{R}') = (\theta_{\mathcal{K}}(x'), i)$ for some i. However, from $\mathrm{R}' \in \mathcal{D}^+_\sigma$, we have that $\exists \mathrm{S} \notin \theta_{\mathcal{K}}(x')$, a contradiction.
2. Suppose $x = \mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{R})$ with $\mathrm{R} \in \mathcal{D}^+_\sigma$.
   (a) Suppose $x' \in \mathcal{D}^+_{\mathcal{K}}$.
   Symmetric to Case 1.b.
   (b) Suppose $x' = \mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{R}')$ with $\mathrm{R}' \in \mathcal{D}^+_\sigma$.
   Then $\theta_{\mathcal{K}}(f(\mathsf{repr}_{\mathcal{K}}(\mathrm{R})) = \theta_{\mathcal{K}}(f(\mathsf{repr}_{\mathcal{K}}(\mathrm{R}'))$. Statement 2 from Lemma 28 yields $\mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{R}) = \mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{R}')$, hence $x' = x$.

Therefore, ori^+ is injective. The argument for ori^- is symmetric. □

We are now ready to prove the second point of Lemma 29, which is formulated in full detail in the following statement.

Lemma 33. *Let* $\mathcal{A}$ *be an ABox and* $\mathcal{K} := (\mathcal{T}, \mathcal{A})$*. Let* $\mathsf{repr}_{\mathcal{K}}$ *be function mapping each role* $\mathrm{R} \in \mathsf{gen}_{\mathcal{K}}$ *to an element with shape* $w\mathrm{R}$ *from* $\Delta^{\mathcal{C}_{\mathcal{K}}}$*. Let* $\mathcal{I}$ *be a model of* $\mathcal{K}$ *and* $f : \mathcal{C}_{\mathcal{K}} \to \mathcal{I}$ *be a homomorphism. Let* σ *be the strategy extracted from* $\mathcal{I}$ *(for* f *and* $\mathsf{repr}_{\mathcal{K}}$*). Let* $\mathsf{ch}_{\sigma/\mathcal{K}}$ *be a choice of well-typed elements for* σ *over* $\mathcal{K}$*. Let* $\mathsf{pair}_{\mathsf{ch}_{\sigma/\mathcal{K}}}$ *be a pairing for* $\mathsf{ch}_{\sigma/\mathcal{K}}$*. Denote by* $\mathcal{J}$ *the model resulting from interpreting the strategy* σ *(according to* $\mathsf{ch}_{\sigma/\mathcal{K}}$*,* $\mathsf{pair}_{\mathsf{ch}_{\sigma/\mathcal{K}}}$*, and any certain successor preference). Then we have:*

$$q_S^{\mathcal{J}} \leq q_S^{\mathcal{I}}.$$

Proof. In the following, assume $\left|\mathsf{crit}^{+}_{\mathsf{ch}_{\sigma/\mathcal{K}}}\right| \geq \left|\mathsf{crit}^{-}_{\mathsf{ch}_{\sigma/\mathcal{K}}}\right|$ so that the only possible shapes for matches are 1, 2, 3^+ and 4^+. The case $\left|\mathsf{crit}^{-}_{\mathsf{ch}_{\sigma/\mathcal{K}}}\right| > \left|\mathsf{crit}^{+}_{\mathsf{ch}_{\sigma/\mathcal{K}}}\right|$ with possible shapes 1, 2, 3^- and 4^- is symmetrical.

Pick some successor preference $(\mathsf{succ}^{\mathcal{I}}_{\mathrm{R}})_{\mathrm{R}}$ for $\mathcal{I}$ (refer back to Definition 57). We associate with each match π of q_S in $\mathcal{J}$, seen as the pair $(\pi(z_1), \pi(z_2))$, a match $\rho(\pi)$ in $\mathcal{I}$ depending on the shape of π:

$$\rho(\pi) : \begin{cases} (\mathsf{a}, \mathsf{b}) & \text{if } \pi = (\mathsf{a}, \mathsf{b}) \text{ has Shape 1} \\ (\mathsf{ori}^+(x), \mathsf{succ}^{\mathcal{I}}_{\mathrm{S}}(\mathsf{ori}^+(x))) & \text{if } \pi = (x, y) \text{ has Shape 2 or } 3^+ \\ (f(\mathsf{repr}_{\mathcal{K}}(\mathrm{S})), \mathsf{succ}^{\mathcal{I}}_{\mathrm{S}}(f(\mathsf{repr}_{\mathcal{K}}(\mathrm{S})))) & \text{if } \pi = (\mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{S}), \mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{S})) \text{ has Shape } 4^+ \end{cases}$$

Notice that in all cases $\rho(\pi)$ is indeed a match in $\mathcal{I}$. This is obvious if π is of Shape 1. When π of Shape 2 or 3^+, $\mathsf{ori}^+(x)$ is an element of $\Delta^{\mathcal{I}}$ that possesses an S-successor, so $\mathsf{succ}^{\mathcal{I}}_{\mathrm{S}}(\mathsf{ori}^+(x))$ is well defined, and we have $\rho(\pi) = (\mathsf{ori}^+(x), \mathsf{succ}^{\mathcal{I}}_{\mathrm{S}}(\mathsf{ori}^+(x))) \in \mathrm{S}^{\mathcal{I}}$. Finally, if π is of Shape 4^+, this means $\mathcal{T} \models \exists \mathrm{S}^- \sqsubseteq \exists \mathrm{S}$, $\mathsf{succ}^{\mathcal{I}}_{\mathrm{S}}(f(\mathsf{repr}_{\mathcal{K}}(\mathrm{S})))$ is well defined, and $(f(\mathsf{repr}_{\mathcal{K}}(\mathrm{S})), \mathsf{succ}^{\mathcal{I}}_{\mathrm{S}}(f(\mathsf{repr}_{\mathcal{K}}(\mathrm{S})))) \in \mathrm{S}^{\mathcal{I}}$.

It now remains to verify ρ is indeed injective, a case analysis can be found in the appendix. □

5.4.2 Construction of the TC^0 circuits

We now sketch how to construct a family of TC^0 circuits (one for each size of ABox) to decide the role cardinality query q_{S} over a DL-Lite$_{\mathsf{core}}$ TBox $\mathcal{T}$. Each such circuit first computes the set $\mathsf{gen}_{\mathcal{K}}$ and the type of each ABox individual. Next, for each function $\varrho : \mathsf{gen}_{\mathcal{K}} \to \Theta_{\mathcal{T}} \times \{1, \dots, \left|\mathsf{sig}(\mathcal{T})^{\pm}_{\mathsf{R}}\right|\}$ satisfying Definition 54, the circuit decides whether ϱ is a legal strategy for $\mathcal{K}$ (i.e. if Definition 55 holds), and if so, computes the number of matches of q_{S} in interpretations induced by ϱ. Importantly, this can be done without actually building interpretations: in Lemma 35 below, we give an explicit formula for this number which can be computed with a TC^0 circuit. Moreover, the number of strategies depends only on $|\mathcal{T}|$, so is constant w.r.t. data complexity. Finally, the circuit computes the minimum value across strategies and compares it with the input number.

To avoid computing an actual model interpreting a strategy and then computing its number of matches, it is useful to observe that this value is easily decided in advance and, in particular, is independent of the choice of well-typed elements and of the pairing. This is expressed by the following lemma.

Lemma 34. *Let $\mathcal{A}$ be an ABox and $\mathcal{K} := (\mathcal{T}, \mathcal{A})$. Let σ be a legal strategy over $\mathcal{K}$. Every model interpreting the strategy σ provides the following number $\lambda_{\sigma/\mathcal{K}}$ of* *matches:*

$$\lambda_{\sigma/\mathcal{K}} \;:=\; |\{(\mathsf{a},\mathsf{b}) \mid \mathcal{K} \models \mathrm{S}(\mathsf{a},\mathsf{b})\}| \;+\; \max\left(\left|\mathcal{D}^{+}_{\mathcal{K}}\right| + |\sigma(\mathcal{D}^{+}_{\sigma})|, \left|\mathcal{D}^{-}_{\mathcal{K}}\right| + |\sigma(\mathcal{D}^{-}_{\sigma})|\right)$$

$$+1 \quad \textit{if} \quad \left| \begin{array}{l} \left|\mathcal{D}^{+}_{\mathcal{K}}\right| + |\sigma(\mathcal{D}^{+}_{\sigma})| > \left|\mathcal{D}^{-}_{\mathcal{K}}\right| + |\sigma(\mathcal{D}^{-}_{\sigma})| \\ \mathcal{T} \models \exists \mathrm{S}^{-} \sqsubseteq \exists \mathrm{S} \\ \exists \mathrm{S} \notin \mathsf{t} \textit{ if } \sigma(\mathrm{S}) = (\mathsf{t}, k) \\ \sigma(\mathrm{S}) \notin \sigma(\mathcal{D}^{+}_{\sigma}) \end{array} \right.$$

$$+1 \quad \textit{if} \quad \left| \begin{array}{l} \left|\mathcal{D}^{-}_{\mathcal{K}}\right| + |\sigma(\mathcal{D}^{-}_{\sigma})| > \left|\mathcal{D}^{+}_{\mathcal{K}}\right| + |\sigma(\mathcal{D}^{+}_{\sigma})| \\ \mathcal{T} \models \exists \mathrm{S} \sqsubseteq \exists \mathrm{S}^{-} \\ \exists \mathrm{S}^{-} \notin \mathsf{t} \textit{ if } \sigma(\mathrm{S}^{-}) = (\mathsf{t}, k) \\ \sigma(\mathrm{S}^{-}) \notin \sigma(\mathcal{D}^{-}_{\sigma}) \end{array} \right.$$

Before giving the proof of the preceding lemma, it will helpful to first establish the relationship holding between the sizes of the sets $\mathcal{D}^{+}_{\mathcal{K}}, \mathcal{D}^{+}_{\sigma}, \mathcal{D}^{-}_{\mathcal{K}}, \mathcal{D}^{-}_{\sigma}$ and the sets of critical elements.

Lemma 35. *Let $\mathsf{ch}_{\sigma/\mathcal{K}}$ be a choice of well-typed elements for a legal strategy σ over $\mathcal{K}$. Then the sets $\mathsf{crit}^{+}_{\mathsf{ch}_{\sigma/\mathcal{K}}}$ and $\mathsf{crit}^{-}_{\mathsf{ch}_{\sigma/\mathcal{K}}}$ satisfy the following:*

$$\left|\mathsf{crit}^{+}_{\mathsf{ch}_{\sigma/\mathcal{K}}}\right| \;=\; \left|\mathcal{D}^{+}_{\mathcal{K}}\right| \;+\; \left|\sigma(\mathcal{D}^{+}_{\sigma})\right| \qquad \left|\mathsf{crit}^{-}_{\mathsf{ch}_{\sigma/\mathcal{K}}}\right| \;=\; \left|\mathcal{D}^{-}_{\mathcal{K}}\right| \;+\; \left|\sigma(\mathcal{D}^{-}_{\sigma})\right|.$$

In particular, the sizes of $\mathsf{crit}^{+}_{\mathsf{ch}_{\sigma/\mathcal{K}}}$ and $\mathsf{crit}^{-}_{\mathsf{ch}_{\sigma/\mathcal{K}}}$ do not depend on $\mathsf{ch}_{\sigma/\mathcal{K}}$.

Proof. First we prove that $\mathcal{D}^{+}_{\mathcal{K}}$ and $\mathsf{ch}_{\sigma/\mathcal{K}}(\mathcal{D}^{+}_{\sigma})$ are disjoint. Notice that if $\mathsf{a} \in \mathcal{D}^{+}_{\mathcal{K}}$, then $\exists \mathrm{S} \in \theta_{\mathcal{K}}(\mathsf{a})$. Therefore, if ever $\mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{R}) = \mathsf{a}$, then by Condition 1 from the definition of a choice of well-typed elements: $\sigma(\mathrm{R}) = (\theta_{\mathcal{K}}(\mathsf{a}), k)$, which would contradict $\mathrm{R} \in \mathcal{D}^{+}_{\sigma}$. Hence $\mathcal{D}^{+}_{\mathcal{K}} \cap \mathsf{ch}_{\sigma/\mathcal{K}}(\mathcal{D}^{+}_{\sigma}) = \emptyset$. We conclude by applying Condition 2 from the definition of a choice of well-typed elements, which ensures that $\left|\mathsf{ch}_{\sigma/\mathcal{K}}(\mathcal{D}^{+}_{\sigma})\right| = |\sigma(\mathcal{D}^{+}_{\sigma})|$. The case of $\mathsf{crit}^{-}_{\mathsf{ch}_{\sigma/\mathcal{K}}}$ is symmetric. □

We now return to the proof of Lemma 34:

Proof of Lemma 34. Let $\mathcal{J}$ be the interpretation of σ obtained according to a choice of well-typed element $\mathsf{ch}_{\sigma/\mathcal{K}}$, a pairing $\mathsf{pair}_{\mathsf{ch}_{\sigma/\mathcal{K}}}$, and some certain successor preference. From Lemma 31, and recalling that distinct shapes are incompatible, we have:

$$
\begin{aligned}
\left|\mathrm{S}^{\mathcal{J}}\right| &= |\{(\mathsf{a},\mathsf{b}) \mid \mathcal{K} \models \mathrm{S}(\mathsf{a},\mathsf{b})\}| && \text{from Shape 1}\\
&\quad + \min\left(\left|\mathsf{crit}^{+}_{\mathsf{ch}_{\sigma/\mathcal{K}}}\right|, \left|\mathsf{crit}^{-}_{\mathsf{ch}_{\sigma/\mathcal{K}}}\right|\right) && \text{from Shape 2}\\
&\quad + \max\left(\left|\mathsf{crit}^{+}_{\mathsf{ch}_{\sigma/\mathcal{K}}}\right| - \left|\mathsf{crit}^{-}_{\mathsf{ch}_{\sigma/\mathcal{K}}}\right|, 0\right) && \text{from Shape } 3^{+}\\
&\quad + \max\left(\left|\mathsf{crit}^{-}_{\mathsf{ch}_{\sigma/\mathcal{K}}}\right| - \left|\mathsf{crit}^{+}_{\mathsf{ch}_{\sigma/\mathcal{K}}}\right|, 0\right) && \text{from Shape } 3^{-}\\
&\quad +1 && \text{if Shape } 4^{+} \text{ is active}\\
&\quad +1 && \text{if Shape } 4^{-} \text{ is active}\\
&= |\{(\mathsf{a},\mathsf{b}) \mid \mathcal{K} \models \mathrm{S}(\mathsf{a},\mathsf{b})\}| && \text{from Shape 1}\\
&\quad + \max\left(\left|\mathsf{crit}^{+}_{\mathsf{ch}_{\sigma/\mathcal{K}}}\right|, \left|\mathsf{crit}^{-}_{\mathsf{ch}_{\sigma/\mathcal{K}}}\right|\right) && \text{joint Shapes 2, } 2^{+} \text{ and } 3^{-}\\
&\quad +1 && \text{if Shape } 4^{+} \text{ is active}\\
&\quad +1 && \text{if Shape } 4^{-} \text{ is active}
\end{aligned}
$$

We can then apply Lemma 35 to express $\mathsf{crit}^{+}_{\mathsf{ch}_{\sigma/\mathcal{K}}}$ and $\mathsf{crit}^{-}_{\mathsf{ch}_{\sigma/\mathcal{K}}}$ in terms of the sets $\mathcal{D}^{+}_{\mathcal{K}}, \mathcal{D}^{+}_{\sigma}, \mathcal{D}^{-}_{\mathcal{K}}, \mathcal{D}^{-}_{\sigma}$. We also use Condition 1 from the definition of a choice of well-typed elements in order to replace $\exists \mathrm{S} \notin \theta_{\mathcal{K}}(\mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{S}))$ by $\exists \mathrm{S} \notin \mathsf{t}$ if $\sigma(\mathrm{S}) = (\mathsf{t}, k)$, and Condition 2 to replace $\mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{S}) \notin \mathsf{ch}_{\sigma/\mathcal{K}}(\mathcal{D}^{+}_{\sigma})$ by $\sigma(\mathrm{S}) \notin \sigma(\mathcal{D}^{+}_{\sigma})$ (and similarly for S^{-}). It can be verified that this indeed yields the desired number $\lambda_{\sigma/\mathcal{K}}$. □

We now describe in detail the family of TC^0 circuits to decide our problem. We create one circuit for each possible number ℓ of individual names. We can assume w.l.o.g. that the same set of individuals, denoted Ind_{ℓ}, is used for all of the ABoxes having ℓ individuals. In what follows, we introduce the different gates which are used for computing the various sets and values used in the construction and how they are connected to each other. Input gates are represented by ?, conjunctive gates by $\wedge$, disjunctive gate by $\vee$, negation gates by $\neg$ and a threshold gate with threshold k by $\mathtt{T}^{(k)}$. Each gate is identified and refereed to by a label *label* indicated by a subscript, e.g. $\vee_{label}$.

We start by the input gates which show how we represent an input $(\mathcal{A}^*, m^*)$ to the circuit that handles ℓ-individual ABoxes. It can be verified that for each of the gates we introduce decides the statement or property occurring in its label (with $\mathcal{A}^*$, resp. $\mathcal{K}^* = (\mathcal{T}, \mathcal{A}^*)$ substituted for $\mathcal{A}$, resp. $\mathcal{K}$).

Input gates

Each atomic role P appearing in $\mathcal{T}$ is represented by input gates $?_{\mathrm{P}(\mathsf{a},\mathsf{b})\in\mathcal{A}}$ for $\mathsf{a}, \mathsf{b} \in \mathsf{Ind}_{\ell}$. The gate $?_{\mathrm{P}(\mathsf{a},\mathsf{b})\in\mathcal{A}}$ is set to 1 iff $\mathrm{P}(\mathsf{a}, \mathsf{b}) \in \mathcal{A}^*$.

Each atomic concept A appearing in $\mathcal{T}$ is represented by input gates $?_{\mathrm{A}(\mathsf{a})\in\mathcal{A}}$ for $\mathsf{a} \in \mathsf{Ind}_\ell$. The gate $?_{\mathrm{A}(\mathsf{a})\in\mathcal{A}}$ is set to 1 iff $\mathrm{A}(\mathsf{a}) \in \mathcal{A}^*$.

The integer m^* is represented in binary by input gates $?_{b_k=1}$ for each $0 \leq k < \log_2(|\mathsf{Ind}(\mathcal{A}^*)| + |\mathcal{T}|)^{|q|})$. The gate $?_{b_k=1}$ is set to 1 iff the k^{th} bit of m^* is 1 (with 0^{th}-bit being the least significant bit).

Regarding the last point, we use the observation from Kostylev and Reutter [2015] that if m^* is a certain answer for q over $\mathcal{K}^*$, then m^* cannot exceed $(|\mathsf{Ind}(\mathcal{A}^*)| + |\mathcal{T}|)^{|q|} = (|\mathsf{Ind}_\ell| + |\mathcal{T}|)^{|q|}$. We denote K this upper bound. This is a direct consequence of the fact that every satisfiable DL-Lite$^{\mathcal{H}}_{\mathsf{core}}$ KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ has a model with at most $|\mathsf{Ind}(\mathcal{A})| + |\mathcal{T}|$ elements.

Gates computing available roles and entailed concepts for individuals

For each positive role R and each individual name $\mathsf{a} \in \mathsf{Ind}_\ell$, introduce a disjunctive gate $\vee_{\exists \mathsf{b}, \mathcal{K} \models \mathrm{R}(\mathsf{a},\mathsf{b})}$ taking as inputs:

- $?_{\mathrm{R}(\mathsf{a},\mathsf{b})\in\mathcal{A}}$ for each $\mathsf{b} \in \mathsf{Ind}(\mathcal{A})$, if $\mathrm{R} \in \mathsf{N_R}$.
- $?_{\mathrm{P}(\mathsf{b},\mathsf{a})\in\mathcal{A}}$ for each $\mathsf{b} \in \mathsf{Ind}(\mathcal{A})$, if $\mathrm{R} = \mathrm{P}^-$ with $\mathrm{P} \in \mathsf{N_R}$.

For each positive concept B and each individual name $\mathsf{a} \in \mathsf{Ind}_\ell$, introduce a disjunctive gate $\vee_{\mathcal{K} \models \mathrm{B}(\mathsf{a})}$ taking as inputs:

- $?_{\mathrm{A}(\mathsf{a})\in\mathcal{A}}$ for each atomic concept A such that $\mathcal{T} \models \mathrm{A} \sqsubseteq \mathrm{B}$.
- $\vee_{\exists \mathsf{b}, \mathcal{K} \models \mathrm{R}(\mathsf{a},\mathsf{b})}$ for all role $\mathrm{R} \in \mathsf{N}^{\pm}_{\mathsf{R}}$ such that $\mathcal{T} \models \exists \mathrm{R} \sqsubseteq \mathrm{B}$.

Computing types and counting number of occurring types

For each type $\mathfrak{t} \in \Theta_{\mathcal{T}}$ and each individual name $\mathsf{a} \in \mathsf{Ind}_\ell$, introduce a conjunctive gate $\wedge_{\theta_{\mathcal{K}}(\mathsf{a})=\mathfrak{t}}$ taking as inputs:

- $\vee_{\mathcal{K} \models \mathrm{B}(\mathsf{a})}$ for each positive concept B such that $\mathrm{B} \in \mathfrak{t}$.
- the negation of $\vee_{\mathcal{K} \models \mathrm{B}(\mathsf{a})}$ for each positive concept B such that $\mathrm{B} \notin \mathfrak{t}$.

For each type $\mathfrak{t} \in \Theta_{\mathcal{T}}$ and each $k \in \{0, \dots \left|\mathsf{sig}(\mathcal{T})^{\pm}_{\mathsf{R}}\right|\}$, introduce a threshold gate $\mathsf{T}^{(k)}_{\exists_{\geq k} \text{ ind. of type } \mathfrak{t}}$ taking as inputs: $\wedge_{\theta_{\mathcal{K}}(\mathsf{a})=\mathfrak{t}}$ for each individual name $\mathsf{a} \in \mathsf{Ind}_\ell$.

Remark: Notice here that k ranges up to $\left|\mathsf{sig}(\mathcal{T})^{\pm}_{\mathsf{R}}\right|$ as any strategy requires at most this many copies of a type (see availability condition from Definition 55). Notice also the label "$\exists_{\geq k}$ ind. of type $\mathfrak{t}$", which stands for $|\{\mathsf{a} \in \mathsf{Ind}_\ell \mid \theta_{\mathcal{K}}(\mathsf{a}) = \mathfrak{t}\}| \geq k$.

Identifying generated roles

For each individual name $\mathsf{a} \in \mathsf{Ind}_\ell$ and each positive role R, introduce a conjunctive gate $\wedge_{\mathsf{a}\mathrm{R}\in\Delta^{\mathcal{C}_\mathcal{K}}}$ taking as inputs: $\vee_{\mathcal{K}\models\exists\mathrm{R}(\mathsf{a})}$ and the negation of $\vee_{\exists\mathsf{b},\mathcal{K}\models\mathrm{R}(\mathsf{a},\mathsf{b})}$.

For each positive role R, introduce a disjunctive gate $\vee_{\mathrm{R}\in\mathsf{gen}_\mathcal{K}}$ taking as inputs: $\wedge_{\mathsf{a}\mathrm{T}\in\Delta^{\mathcal{C}_\mathcal{K}}}$ for each positive role T such that $\mathcal{T}$ ensures that if $\mathsf{a}\mathrm{T} \in \Delta^{\mathcal{C}_\mathcal{K}}$, then there exists a word w starting with T and ending by R s.t $\mathsf{a}w \in \Delta^{\mathcal{C}_\mathcal{K}}$.

Identifying demanding individuals (see Definition 58)

For each $\mathsf{a} \in \mathsf{Ind}_\ell$, introduce a conjunctive gate $\wedge_{\mathsf{a}\in\mathcal{D}^+_\mathcal{K}}$ taking as inputs: $\vee_{\mathcal{K}\models\exists\mathrm{S}(\mathsf{a})}$ and the negation of $\vee_{\exists\mathsf{b},\mathcal{K}\models\mathrm{S}(\mathsf{a},\mathsf{b})}$.

For each $\mathsf{a} \in \mathsf{Ind}_\ell$, introduce a conjunctive gate $\wedge_{\mathsf{a}\in\mathcal{D}^-_\mathcal{K}}$ taking as inputs: $\vee_{\mathcal{K}\models\exists\mathrm{S}^-(\mathsf{a})}$ and the negation of $\vee_{\exists\mathsf{b},\mathcal{K}\models\mathrm{S}^-(\mathsf{a},\mathsf{b})}$.

Deciding legality of each strategy σ (see Definitions 54 and 55)

Introduce a conjunctive gate $\wedge_{\text{coverage }\sigma}$ taking as inputs:

- $\vee_{\mathrm{R}\in\mathsf{gen}_\mathcal{K}}$ for each positive role $\mathrm{R} \in \mathsf{dom}(\sigma)$,
- the negation of $\vee_{\mathrm{R}\in\mathsf{gen}_\mathcal{K}}$ for each positive role $\mathrm{R} \notin \mathsf{dom}(\sigma)$.

Introduce a conjunctive gate $\wedge_{\text{availability }\sigma}$ taking as inputs: $\mathtt{T}^{(k)}_{\exists_{\geq k}\text{ ind. of type }\mathsf{t}}$ for each type t being required k times by σ.

Introduce a conjunctive gate $\wedge_{\text{legal }\sigma}$ taking as inputs: $\wedge_{\text{coverage }\sigma}$ and $\wedge_{\text{availability }\sigma}$.

Computing $\lambda_{\sigma/\mathcal{K}}$ for each strategy σ (see Lemma 34).

A threshold gate $\mathtt{T}^{(k)}_{m_\mathcal{A}+|\mathcal{D}^+_\mathcal{K}|+|\sigma(\mathcal{D}^+_\sigma)|\geq k}$ for each $k \in \{0, \dots, K\}$ with inputs:

- $?_{\mathrm{S}(\mathsf{a},\mathsf{b})\in\mathcal{A}}$ for each $(\mathsf{a}, \mathsf{b}) \in \mathsf{Ind}_\ell \times \mathsf{Ind}_\ell$,
- $\wedge_{\mathsf{a}\in\mathcal{D}^+_\mathcal{K}}$ for each $\mathsf{a} \in \mathcal{D}^+_\mathcal{K}$,
- $|\sigma(\mathcal{D}^+_\sigma)|$ copies of a true gate *true*.

A threshold gate $\mathtt{T}^{(k)}_{m_\mathcal{A}+|\mathcal{D}^-_\mathcal{K}|+|\sigma(\mathcal{D}^-_\sigma)|\geq k}$ for each $k \in \{0, \dots, K\}$ with inputs:

- $?_{\mathrm{S}(\mathsf{a},\mathsf{b})\in\mathcal{A}}$ for each $(\mathsf{a}, \mathsf{b}) \in \mathsf{Ind}_\ell \times \mathsf{Ind}_\ell$,
- $\wedge_{\mathsf{a}\in\mathcal{D}^-_\mathcal{K}}$ for each $\mathsf{a} \in \mathcal{D}^-_\mathcal{K}$,
- $|\sigma(\mathcal{D}^-_\sigma)|$ copies of a true gate *true*.

Introduce $\wedge_{|\mathcal{D}^+_\mathcal{K}|+|\sigma(\mathcal{D}^+_\sigma)|=|\mathcal{D}^-_\mathcal{K}|+|\sigma(\mathcal{D}^-_\sigma)|=k-m_\mathcal{A}}$ for each $k \in \{0, \dots, K\}$ with inputs:

- $\mathtt{T}^{(k)}_{m_\mathcal{A}+|\mathcal{D}^+_\mathcal{K}|+|\sigma(\mathcal{D}^+_\sigma)|\geq k}$ and the negation of $\mathtt{T}^{(k+1)}_{m_\mathcal{A}+|\mathcal{D}^+_\mathcal{K}|+|\sigma(\mathcal{D}^+_\sigma)|\geq k+1}$,
- $\mathtt{T}^{(k)}_{m_\mathcal{A}+|\mathcal{D}^-_\mathcal{K}|+|\sigma(\mathcal{D}^-_\sigma)|\geq k}$ and the negation of $\mathtt{T}^{(k+1)}_{m_\mathcal{A}+|\mathcal{D}^-_\mathcal{K}|+|\sigma(\mathcal{D}^-_\sigma)|\geq k+1}$,
- $\wedge_{\text{legal }\sigma}$.

Notice that for the latter and upcoming gates of this block, we omit "legal σ" from the labels for clarity.

Gate $\wedge_{|\mathcal{D}^+_\mathcal{K}|+|\sigma(\mathcal{D}^+_\sigma)|<|\mathcal{D}^-_\mathcal{K}|+|\sigma(\mathcal{D}^-_\sigma)|=k-m_\mathcal{A}}$ for each $k \in \{0, \ldots, K\}$ with inputs:

- the negation of $\mathtt{T}^{(k)}_{m_\mathcal{A}+|\mathcal{D}^+_\mathcal{K}|+|\sigma(\mathcal{D}^+_\sigma)|\geq k}$,
- $\mathtt{T}^{(k)}_{m_\mathcal{A}+|\mathcal{D}^-_\mathcal{K}|+|\sigma(\mathcal{D}^-_\sigma)|\geq k}$ and the negation of $\mathtt{T}^{(k+1)}_{m_\mathcal{A}+|\mathcal{D}^-_\mathcal{K}|+|\sigma(\mathcal{D}^-_\sigma)|\geq k+1}$,
- $\wedge_{\text{legal } \sigma}$.

Gate $\wedge_{|\mathcal{D}^-_\mathcal{K}|+|\sigma(\mathcal{D}^-_\sigma)|<|\mathcal{D}^+_\mathcal{K}|+|\sigma(\mathcal{D}^+_\sigma)|=k-m_\mathcal{A}}$ for each $k \in \{0, \ldots, K\}$ with inputs:

- $\mathtt{T}^{(k)}_{m_\mathcal{A}+|\mathcal{D}^+_\mathcal{K}|+|\sigma(\mathcal{D}^+_\sigma)|\geq k}$ and the negation of $\mathtt{T}^{(k+1)}_{m_\mathcal{A}+|\mathcal{D}^+_\mathcal{K}|+|\sigma(\mathcal{D}^+_\sigma)|\geq k+1}$,
- the negation of $\mathtt{T}^{(k)}_{m_\mathcal{A}+|\mathcal{D}^-_\mathcal{K}|+|\sigma(\mathcal{D}^-_\sigma)|\geq k}$,
- $\wedge_{\text{legal } \sigma}$.

For each $k \in \{0, \ldots, K\}$, introduce a disjunctive gate $\vee_{\lambda_{\sigma/\mathcal{K}}=k}$ taking as inputs:

- $\wedge_{|\mathcal{D}^+_\mathcal{K}|+|\sigma(\mathcal{D}^+_\sigma)|=|\mathcal{D}^-_\mathcal{K}|+|\sigma(\mathcal{D}^-_\sigma)|=k-m_\mathcal{A}}$,
- If $\mathcal{T} \models \exists \mathrm{S} \sqsubseteq \exists \mathrm{S}^-$, $\exists \mathrm{S}^- \notin \mathfrak{t}$ with $\sigma(\mathrm{S}^-) = (\mathfrak{t}, k)$ and $\sigma(\mathrm{S}^-) \notin \sigma(\mathcal{D}^-_\sigma)$, then gate $\wedge_{|\mathcal{D}^+_\mathcal{K}|+|\sigma(\mathcal{D}^+_\sigma)|<|\mathcal{D}^-_\mathcal{K}|+|\sigma(\mathcal{D}^-_\sigma)|=k-1-m_\mathcal{A}}$, otherwise gate $\wedge_{|\mathcal{D}^+_\mathcal{K}|+|\sigma(\mathcal{D}^+_\sigma)|<|\mathcal{D}^-_\mathcal{K}|+|\sigma(\mathcal{D}^-_\sigma)|=k-m_\mathcal{A}}$,
- If $\mathcal{T} \models \exists \mathrm{S}^- \sqsubseteq \exists \mathrm{S}$, $\exists \mathrm{S} \notin \mathfrak{t}$ with $\sigma(\mathrm{S}) = (\mathfrak{t}, k)$, and $\sigma(\mathrm{S}) \notin \sigma(\mathcal{D}^+_\sigma)$, then gate $\wedge_{|\mathcal{D}^-_\mathcal{K}|+|\sigma(\mathcal{D}^-_\sigma)|<|\mathcal{D}^+_\mathcal{K}|+|\sigma(\mathcal{D}^+_\sigma)|=k-1-m_\mathcal{A}}$, otherwise gate $\wedge_{|\mathcal{D}^+_\mathcal{K}|+|\sigma(\mathcal{D}^+_\sigma)|<|\mathcal{D}^-_\mathcal{K}|+|\sigma(\mathcal{D}^-_\sigma)|=k-m_\mathcal{A}}$.

We are now able to compute the minimal number of matches given by legal strategies.

Final comparison with the input integer (see Lemma 29).
For each $k \in \{0, \ldots, K\}$, introduce a disjunctive gate $\vee_{\min_{\text{legal } \sigma} \lambda_{\sigma/\mathcal{K}}<k}$ taking as inputs: $\wedge_{\lambda_{\sigma/\mathcal{K}}=k',\ \text{legal } \sigma}$ for each strategy σ and each $k' < k$.

For each $k \in \{0, \ldots, K\}$, introduce a conjunctive gate $\wedge_{m=k}$ taking as inputs:

- $?_{b_j=1}$ such that the j^{th} bit of the binary encoding of k is 1,
- the negation of $?_{b_j=1}$ such that the j^{th} bit of the binary encoding of k is 0.

For each $k \in \{0, \ldots, K\}$, introduce a conjunctive gate $\wedge_{\min_{\text{legal } \sigma} \lambda_{\sigma/\mathcal{K}}\geq m=k}$ taking as inputs: $\wedge_{m=k}$ and the negation of $\vee_{\min_{\text{legal } \sigma} \lambda_{\sigma/\mathcal{K}}<k}$.

Introduce an **output** disjunctive gate $\vee_{\min_{\text{legal } \sigma} \lambda_{\sigma/\mathcal{K}}\geq m}$ taking as inputs: $\wedge_{\min_{\text{legal } \sigma} \lambda_{\sigma/\mathcal{K}}\geq m=k}$ for each $k \in \{0, \ldots, K\}$.

To complete the proof, we observe that, since the TBox $\mathcal{T}$ is fixed, the number of gates is polynomial in the described family of circuits. Moreover, all circuits in the family have the same depth (13). Thus, the construction yields a TC^0 of circuits for deciding the role cardinality query q_S over the DL-Lite$_\mathsf{core}$ TBox $\mathcal{T}$ and establishes membership in TC^0.

5.4.3 Concept cardinality over DL-Lite$^\mathcal{H}_\mathsf{core}$ without role inclusions

We now turn to the case where $\mathcal{T}$ is a DL-Lite$^\mathcal{H}_\mathsf{core}$ TBox *without negative role inclusions* and q_C is the concept cardinality query: $\exists z\ \mathrm{C}(z)$.

Due to a simpler shape of the query, several notions simplify. In particular, distinguishing between positive and negative critical elements is no longer necessary and these notions can be unified as follows.

Definition 65. *Let σ be a strategy. Define* demanding roles $\mathcal{D}_\sigma$ *as:*

$$\mathcal{D}_\sigma \ := \ \left\{ \mathrm{R} \ \middle| \ \begin{array}{l} \mathrm{R} \in \mathsf{dom}(\sigma) \\ \mathcal{T} \models \exists \mathrm{R}^- \sqsubseteq \mathrm{C} \\ \mathrm{C} \notin \mathsf{t} \ \textit{if}\ \sigma(\mathrm{R}) = (\mathsf{t}, k) \end{array} \right\}$$

Let $\mathsf{ch}_{\sigma/\mathcal{K}}$ be a choice of well-typed elements for σ over $\mathcal{K}$. Define the set of critical elements *as:*

$$\mathsf{crit}_{\mathsf{ch}_{\sigma/\mathcal{K}}} = \mathsf{ch}_{\sigma/\mathcal{K}}(\mathcal{D}_\sigma)$$

Pairing is also no longer necessary, which means the interpretation of a strategy can be drastically simplified as follows.

Definition 66. *Let $\mathcal{A}$ be an ABox and $\mathcal{K} := (\mathcal{T}, \mathcal{A})$. Let σ be a legal strategy over $\mathcal{K}$. Let $\mathsf{ch}_{\sigma/\mathcal{K}}$ be a choice of well-typed elements for σ over $\mathcal{A}$. Consider the following mapping:*

$$\begin{array}{rrcl} \chi: & \Delta^{\mathcal{C}_\mathcal{K}} & \rightarrow & \mathsf{Ind}(\mathcal{A}) \ \cup \ \{\bot_i \mid i = 1, \dots \left|\mathsf{sig}(\mathcal{T})^\pm_\mathsf{R}\right|\} \\ & \mathsf{a} & \mapsto & \mathsf{a} \\ & w\mathrm{R} & \mapsto & \mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{R}) \end{array}$$

The interpretation $\mathcal{J}$ of σ w.r.t. $\mathsf{ch}_{\sigma/\mathcal{K}}$ is defined as the image of $\cdot^\mathcal{C}_\mathcal{K}$ through χ: its domain is $\Delta^\mathcal{J} = \chi(\Delta^{\mathcal{C}_\mathcal{K}})$, and its interpretation function is $\cdot^\mathcal{J} = \chi \circ \cdot^{\mathcal{C}_\mathcal{K}}$.

Under these updated definitions, notice the Lemma 29 still makes perfect sense, and we start by proving it, following closely the analogous proof for role cardinality queries.

Proof of Lemma 29 for concept cardinality queries

We first prove the first point of Lemma 29, stating that the interpretation of a strategy extracted from a model is also a model, in the following stronger form, not requiring the strategy to be extracted from a model in the first place.

Lemma 36. *Let $\mathcal{A}$ be an ABox and $\mathcal{K} := (\mathcal{T}, \mathcal{A})$ be a satisfiable KB. Let σ be a legal strategy over $\mathcal{K}$. Let $\mathsf{ch}_{\sigma/\mathcal{K}}$ be a choice of well-typed elements for σ over $\mathcal{A}$. The interpretation $\mathcal{J}$ of the strategy σ w.r.t. $\mathsf{ch}_{\sigma/\mathcal{K}}$ is a model.*

Proof. Assertions from the ABox and axioms without negation are satisfied since the interpretation $\mathcal{J}$ is built from $\mathcal{C}_{\mathcal{K}}$. Consider now a negative concept inclusion $\mathrm{B}_1 \sqsubseteq \neg\mathrm{B}_2$. Assume for a contradiction that there is an element d such that $d \in \mathrm{B}_1{}^{\mathcal{J}} \cap \mathrm{B}_2{}^{\mathcal{J}}$. There are four cases to consider:

1. If $\mathcal{K} \models \mathrm{B}_1(d)$ and $\mathcal{K} \models \mathrm{B}_2(d)$, then this contradicts $\mathcal{K}$ being satisfiable.
2. If $\mathcal{K} \models \mathrm{B}_1(d)$ and $\mathcal{K} \not\models \mathrm{B}_2(d)$, then $d = \chi(w\mathrm{R})$ with $\mathrm{R} \in \mathsf{gen}_{\mathcal{K}}$ and $\mathcal{T} \models \exists\mathrm{R}^- \sqsubseteq \mathrm{B}_2$. In particular, $d = \mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{R})$ and $\mathcal{T} \models \exists\mathrm{R}^- \sqsubseteq \neg\mathrm{B}_1$. Condition 1 from the definition of choice of well-typed elements ensures $\sigma(\mathrm{R}) = (\theta_{\mathcal{K}}(d), i)$ for some i. Condition 1 from the definition of a strategy implies that $\mathrm{B}_1 \notin \theta_{\mathcal{K}}(d)$, contradicting $\mathcal{K} \models \mathrm{B}_1(d)$.
3. If $\mathcal{K} \not\models \mathrm{B}_1(d)$ and $\mathcal{K} \models \mathrm{B}_2(d)$. Symmetric to Case 2.
4. If $\mathcal{K} \not\models \mathrm{B}_1(d)$ and $\mathcal{K} \not\models \mathrm{B}_2(d)$, then $d = \chi(w_1\mathrm{R}_1) = \chi(w_2\mathrm{R}_2)$ with $\mathrm{R}_1 \in \mathsf{gen}_{\mathcal{K}}$, $\mathrm{R}_2 \in \mathsf{gen}_{\mathcal{K}}$, $\mathcal{T} \models \exists\mathrm{R}_1^- \sqsubseteq \mathrm{B}_1$ and $\mathcal{T} \models \exists\mathrm{R}_2^- \sqsubseteq \mathrm{B}_2$. In particular, $d = \mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{R}_1) = \mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{R}_2)$. Condition 2 in the definition of the choice of well-typed elements ensures: $\sigma(\mathrm{R}_1) = \sigma(\mathrm{R}_2)$. Condition 2 in the definition of a strategy ensures: $\mathcal{T} \not\models \exists\mathrm{R}_1^- \sqsubseteq \neg\exists\mathrm{R}_2^-$, contradicting $\mathcal{T} \models \mathrm{B}_1 \sqsubseteq \neg\mathrm{B}_2$. □

In order to prove the second point of Lemma 28, stating an interpretation $\mathcal{J}$ of the strategy extracted from a model $\mathcal{I}$ has at most as matches as the original model $\mathcal{I}$, we need to better understand what kinds of matches of q_C can be found in $\mathcal{J}$. This is achieved by the following result which precisely characterizes $\mathrm{C}^{\mathcal{J}}$

Lemma 37. *Let $\mathcal{A}$ be an ABox, and $\mathcal{K} := (\mathcal{T}, \mathcal{A})$ be satisfiable KB. Let σ be a legal strategy over $\mathcal{K}$, and let $\mathsf{ch}_{\sigma/\mathcal{K}}$ be a choice of well-typed elements for σ over $\mathcal{K}$. Denote by $\mathcal{J}$ the interpretation of σ w.r.t. $\mathsf{ch}_{\sigma/\mathcal{K}}$. Then we have:*

$$\begin{aligned} \mathrm{C}^{\mathcal{J}} &= \{\mathsf{a} \mid \mathcal{K} \models \mathrm{C}(\mathsf{a})\} && \textit{(Shape 1)} \\ &\cup \left\{\mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{R}) \mid \mathrm{R} \in \mathcal{D}_\sigma\right\} && \textit{(Shape 2)} \end{aligned}$$

Furthermore, there is no overlap between these two distinct shapes.

Proof. The first inclusion ($\subseteq$) is rather straightforward. We therefore focus on proving the direction ($\supseteq$).

1. Let a be such that $\mathcal{K} \models \mathrm{C}(\mathsf{a})$, in particular $\mathsf{a} \in \mathsf{Ind}(\mathcal{A})$. By definition, $\chi(\mathsf{a}) = \mathsf{a}$, hence $\mathsf{a} \in \mathrm{C}^{\mathcal{J}}$.

2. Let $\mathrm{R} \in \mathcal{D}_\sigma$. By definition of $\mathsf{gen}_{\mathcal{K}}$, there exists $w\mathrm{R} \in \mathcal{C}_{\mathcal{K}}$. By definition of the interpretation of a strategy, $\chi(w\mathrm{R}) = \mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{R})$. Moreover, $\mathrm{R} \in \mathcal{D}_\sigma$ implies that $\mathcal{T} \models \exists \mathrm{R}^- \sqsubseteq \mathrm{C}$, which ensures $w\mathrm{R} \in \mathrm{C}_{\mathcal{K}}^{\mathcal{C}}$. Therefore $\mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{R}) \in \mathrm{C}^{\mathcal{J}}$. □

We can now prove the second point of Lemma 29, recalled in the following statement.

Lemma 38. *Let $\mathcal{A}$ be an ABox and $\mathcal{K} := (\mathcal{T}, \mathcal{A})$. Let $\mathcal{I}$ be a model of $\mathcal{K}$. Let σ be the strategy extracted from $\mathcal{I}$. Let $\mathsf{ch}_{\sigma/\mathcal{K}}$ be a choice of well-typed elements for σ over $\mathcal{K}$. Denote $\mathcal{J}$ the resulting interpretation of σ. We have:*

$$q^{\mathcal{J}} \leq q^{\mathcal{I}}.$$

Proof. Associate each match π of q in $\mathcal{J}$ to a match $\rho(\pi)$ in $\mathcal{I}$ depending of the shape of π:

$$\rho(\pi) : \begin{cases} z \;\mapsto\; \pi(z) & \text{if } \pi \text{ has Shape 1} \\ z \;\mapsto\; f(\mathsf{repr}_{\mathcal{K}}(\mathrm{R})) & \text{if } \pi \text{ has Shape 2 with } \pi(z) = \mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{R}) \end{cases}$$

Notice $\rho(\pi)$ is indeed a match in $\mathcal{I}$. We now prove that ρ is injective. Let $\pi_1, \pi_2 : q \to \mathcal{J}_{\sigma_{f \circ \mathsf{repr}_{\mathcal{K}}}}$ be two matches such that $\rho(\pi_1) = \rho(\pi_2)$. We consider all four cases:

1.
    1. $\pi_1(z_1) = \rho(\pi_1)(z_1) = \rho(\pi_2)(z_1) = \pi_2(z_1)$ and $\pi_1(z_2) = \rho(\pi_1)(z_2) = \rho(\pi_2)(z_2) = \pi_2(z_2)$.
    2. We have $\pi_2(z) = \mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{R})$ with $\mathrm{R} \in \mathcal{D}_\sigma$. Therefore $\mathrm{C} \notin \theta_{\mathcal{K}}(\mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{R}))$. Lemma 28 provides $\theta_{\mathcal{K}}(\mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{R})) = \theta_{\mathcal{K}}(\rho(\pi_2)(z))$. Recall $\rho(\pi_1) = \rho(\pi_2)$, hence $\mathrm{C} \notin \theta_{\mathcal{K}}(\pi_1(z))$. Contradiction with $\mathcal{K} \models \mathrm{C}(\pi_1(z))$.

2.
    1. Symmetric to Case 1.2.
    2. We have $\pi_1(z) = \mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{R}_1)$ with $\mathrm{R}_1 \in \mathcal{D}_\sigma$ and $\pi_2(z) = \mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{R}_2)$ with $\mathrm{R}_2 \in \mathcal{D}_\sigma$. Therefore $f(\mathsf{repr}_{\mathcal{K}}(\mathrm{R}_1)) = f(\mathsf{repr}_{\mathcal{K}}(\mathrm{R}_2))$. Lemma 28 provides $\pi_1(z) = \pi_2(z)$. □

Number of matches in the interpretation of a strategy.

We will again avoid having to produce interpretations of strategies by showing that we can directly determine the number of matches occurring in such models. This is the purpose of the following lemma.

Theorem 52. *Let $\mathcal{A}$ be an ABox, and $\mathcal{K} := (\mathcal{T}, \mathcal{A})$ be a satisfiable KB. Let σ be a legal strategy over $\mathcal{K}$. Any interpretation $\mathcal{J}$ of the strategy σ has the following number $\lambda_{\sigma/\mathcal{K}}$ of matches:*

$$\lambda_{\sigma/\mathcal{K}} \;=\; |\{\mathsf{a} \mid \mathcal{K} \models \mathrm{C}(\mathsf{a})\}| \;+\; |\sigma(\mathcal{D}_\sigma)|$$

Proof. The equation immediately follows from Lemma 37 and by noticing that $\left|\mathsf{crit}_{\mathsf{ch}_{\sigma/\mathcal{K}}}\right| = |\sigma(\mathcal{D}_\sigma)|$ due to second condition in the definition of a choice of well-typed elements. □

The family of circuits.

To complete the proof, we describe how to construct a family of TC^0 circuits that can be used to decide our problem. The construction is very similar to the one given for role cardinality queries, so we simply mention the updates required to adapt the family of circuits to concept cardinality queries.

- We need to introduce further gates in the second block to compute entailed role assertions.

- The circuits in the block "Deciding demanding individuals" are no longer required.

- Each block dedicated to a particular strategy simplifies as we no longer need to compare the size of positive vs negative critical elements: each strategy still comes with a specific number of additional matches $|\sigma(\mathcal{D}_\sigma)|$ due to demanding roles, again introduced through constant gates, and counting ABox matches needs to be slightly updated from the role setting to the concept one.

To match our TC^0 membership results, it is natural to investigate the TC^0-hardness of our problem in these situations. We show that as soon as the query predicate is satisfiable, then it is sufficient to obtain TC^0-hardness (for any DL-Lite$^{\mathcal{H}}_{\mathsf{core}}$ TBox), and it also necessary as the excluded situations can be decided within AC^0 (which we recall is the circuit complexity class obtained from TC^0 by disallowing threshold gates). We thus prove the following statement:

Theorem 53 (TC^0-hard / in AC^0)**.** *Let q be a cardinality query and $\mathcal{T}$ be a DL-Lite$^{\mathcal{H}}_{\mathsf{core}}$ TBox. If the query predicate is satisfiable w.r.t. $\mathcal{T}$, then answering q over $\mathcal{T}$ is TC^0-hard. Otherwise it is in AC^0.*

The argument for AC^0 membership is trivial: a cardinality query with an unsatisfiable predicate admits as certain answers precisely those intervals of the form $[0, M]$, since every model will contain 0 matches.

For both concept and role cardinality queries, we show TC^0-hardness by AC^0-reduction from the NumOnes problem, known to be TC^0-complete Aehlig et al. [2007]. The problem NumOnes is to decide, given as input an integer $k \geq 1$ (given in binary) and a binary string X, whether the number of 1-bits in X is at least k.

We note that we cannot reuse the TC^0-hardness proof given in Bienvenu et al. [2020], since that result used a rooted counting query coupled with an empty TBox. By contrast, we consider non-empty TBoxes which may include existential axioms, and our queries may match to unnamed elements.

Proof for concept cardinality queries. Let q_{C} be our concept cardinality query and assume C is satisfiable w.r.t. our TBox $\mathcal{T}$. Set $\mathcal{K}_{(\mathcal{T},q)} := (\mathcal{T}, \{\mathrm{C}(\mathsf{a})\})$. Our assumption ensures $\mathcal{K}_{(\mathcal{T},q)}$ is satisfiable hence its canonical interpretation (model) $\mathcal{C}_{\mathcal{K}_{(\mathcal{T},q)}}$ is indeed a model. Let (k, X) be an instance of NumOnes. Consider the following ABox:

$$\begin{aligned}\mathcal{A} = &\{\mathrm{A}(\mathsf{aux_1}) \mid \mathsf{a} \in \mathrm{A}^{\mathcal{C}_{\mathcal{K}_{(\mathcal{T},q)}}}\} \cup \{\mathrm{R}'(\mathsf{aux_1}, \mathsf{aux}_{\mathrm{R}}) \mid \mathsf{a}\mathrm{R} \in \Delta^{\mathcal{C}_{\mathcal{K}_{(\mathcal{T},q)}}}, \mathcal{T} \models \mathrm{R} \sqsubseteq \mathrm{R}'\}\\ &\cup \{\mathrm{A}(\mathsf{b}) \mid \text{bit } b \text{ of } X \text{ is equal to } 1, \mathsf{a} \in \mathrm{A}^{\mathcal{C}_{\mathcal{K}_{(\mathcal{T},q)}}}\}\\ &\cup \{\mathrm{R}'(\mathsf{b}, \mathsf{aux}_{\mathrm{R}}) \mid \text{bit } b \text{ of } X \text{ is equal to } 1, \mathsf{a}\mathrm{R} \in \Delta^{\mathcal{C}_{\mathcal{K}_{(\mathcal{T},q)}}}, \mathcal{T} \models \mathrm{R} \sqsubseteq \mathrm{R}'\}\\ &\cup \{\mathrm{A}(\mathsf{aux}_{\mathrm{R}}) \mid w\mathrm{R} \in \Delta^{\mathcal{C}_{\mathcal{K}_{(\mathcal{T},q)}}}, \mathcal{T} \models \exists \mathrm{R}^- \sqsubseteq \mathrm{A}\}\\ &\cup \{\mathrm{R}'(\mathsf{aux}_{\mathrm{T}}, \mathsf{aux}_{\mathrm{R}}) \mid w\mathrm{TR} \in \Delta^{\mathcal{C}_{\mathcal{K}_{(\mathcal{T},q)}}}, \mathcal{T} \models \mathrm{R} \sqsubseteq \mathrm{R}'\}\end{aligned}$$

Note that in particular that $\mathcal{A}$ will contain $\mathrm{C}(\mathsf{b})$ for every 1-bit b of X, as well as $\mathrm{C}(\mathsf{aux_1})$. The auxiliary individual $\mathsf{aux_1}$ mimics a 1-bit from X in order to appropriately handle the case in which X doesn't contain any such bit. As the notation suggests, auxiliary individuals $\mathsf{aux}_{\mathrm{R}}$ are intended to receive all needed outgoing roles R from other elements (so $\mathsf{aux}_{\mathrm{R}}$ is intended to satisfy the concept $\exists \mathrm{R}^-$). Note that by construction the interpretation based upon $\mathcal{A}$ already satisfies all of the TBox axioms. In particular, this means that there exists a model of $(\mathcal{T}, \mathcal{A})$ all of whose matches are already present in $\mathcal{A}$. We can thus focus on counting the matches explicitly given in $\mathcal{A}$.

Observe that the number m of matches of q_{C} *among the auxiliary elements* only depends on the OMQ $(q_{\mathrm{C}}, \mathcal{T})$. In particular notice that $\mathsf{aux_1}$ always provides a match, hence $m \geq 1$. It is straightforward to verify that $m + k$ is a certain answer for q over $\mathcal{K} := (\mathcal{T}, \mathcal{A})$ iff $(k, X) \in$ NumOnes. Moreover, the input $(\mathcal{A}, m + k)$ to our OMQA problem can be computed from (k, X) by an AC^0 circuit (recall that binary integer addition is known to be computable in AC^0). □

5.5 Role cardinality over DL-Lite$^{\mathcal{H}}_{\mathsf{pos}}$

In this section, we consider DL-Lite$^{\mathcal{H}}_{\mathsf{pos}}$ TBoxes. We show that coNP-hard OMQs exist and prove a complexity trichotomy which precisely delineates the tractability boundary. We begin by recalling the coNP-complete[1] situation exhibited in Theorem 50.

Example 29. *Answering the role cardinality query* q_{S} *over the DL-Lite*$^{\mathcal{H}}_{\mathsf{pos}}$ *TBox given by* $\mathcal{T} = \{\mathrm{B} \sqsubseteq \exists\mathrm{R}_1, \mathrm{R}_1 \sqsubseteq \mathrm{S}, \exists\mathrm{R}_1^- \sqsubseteq \exists\mathrm{R}_2, \mathrm{R}_2 \sqsubseteq \mathrm{S}\}$ *is* coNP*-complete. We consider the* NP*-complete* Set Cover *problem: given a set* $\mathcal{U}$*, set of subsets* $\mathcal{S} \subseteq 2^{\mathcal{U}}$ *whose union is* $\mathcal{U}$*, and number* k*, decide whether there exists a* k*-cover, i.e. a subset* $\mathcal{C}$ *of* $\mathcal{S}$ *with* $|\mathcal{C}| \leq k$ *whose union is* $\mathcal{U}$*. We prove that there exists a* k*-cover iff* $[\Sigma_{s\in\mathcal{S}}|s| + k + 1, +\infty]$ *is* not *a certain answer on the following ABox:* $\{\mathrm{B}(\mathsf{u}) \mid u \in \mathcal{U}\} \ \cup \ \{\mathrm{S}(\mathsf{u}, \mathsf{s}) \mid u \in s, s \in \mathcal{S}\}$.

The following definition abstracts the preceding example.

Definition 67. *A TBox* $\mathcal{T}$ *admits a* propagation of role W *by a concept* $\mathrm{B} \in \mathsf{sig}(\mathcal{T})^{\pm}_{\mathsf{C}}$ *and roles* $\mathrm{R}_1, \mathrm{R}_2$ *if* $\mathcal{T}$ *entails* $\{\mathrm{B} \sqsubseteq \exists\mathrm{R}_1, \mathrm{R}_1 \sqsubseteq \mathrm{W}, \exists\mathrm{R}_1^- \sqsubseteq \exists\mathrm{R}_2, \mathrm{R}_2 \sqsubseteq \mathrm{W}\}$.

A propagation of S (or S^-) is not sufficient to ensure coNP-hardness: the reduction sketched in Example 29 will fail in the presence of 'interferences', which can be of three types.

Definition 68. *A role* U interferes with the propagation of W by $\mathrm{B}, \mathrm{R}_1, \mathrm{R}_2$ *if it satisfies one of the following conditions:*

1. $\mathcal{T} \models \{\mathrm{B} \sqsubseteq \exists\mathrm{U}, \mathrm{U} \sqsubseteq \mathrm{W}, \mathrm{U} \sqsubseteq \mathrm{W}^-\}$;
2. $\mathcal{T} \models \{\exists\mathrm{W}^- \sqsubseteq \exists\mathrm{U}, \mathrm{U} \sqsubseteq \mathrm{W}\}$ *and either* $\mathcal{T} \models \mathrm{U} \sqsubseteq \mathrm{W}^-$ *or* $\mathcal{T} \not\models \mathrm{R}_2 \sqsubseteq \mathrm{W}^-$;
3. *if* $\mathrm{B} = \exists\mathrm{T}$ *and* $\mathrm{T} \sqsubseteq \mathrm{W}$*, then* $\mathcal{T} \models \{\exists\mathrm{T}^- \sqsubseteq \exists\mathrm{U}, \mathrm{U} \sqsubseteq \mathrm{W}\}$ *and either* $\mathcal{T} \models \mathrm{U} \sqsubseteq \mathrm{W}^-$ *or* $\mathcal{T} \not\models \mathrm{R}_2 \sqsubseteq \mathrm{W}^-$.

Remarkably, the existence of a propagation without any interfering role (which we call a *non-trivial propagation*) ensures coNP-hardness of answering the corresponding role cardinality query, while its absence ensure P-membership. We further distinguish two tractable cases, depending on the existence of a *non-trivial pairing*.

Definition 69. *A TBox* $\mathcal{T}$ *admits a* non-trivial pairing of S *if there exist* $\mathrm{B} \in \mathsf{sig}(\mathcal{T})^{\pm}_{\mathsf{C}}$ *and* $\mathrm{R} \in \mathsf{sig}(\mathcal{T})^{\pm}_{\mathsf{R}}$ *such that*

$$\mathcal{T} \models \mathrm{B} \sqsubseteq \exists\mathrm{R} \quad \mathcal{T} \models \mathrm{R} \sqsubseteq \mathrm{S} \quad \mathcal{T} \models \mathrm{R} \sqsubseteq \mathrm{S}^- \quad \mathcal{T} \not\models \mathrm{S} \sqsubseteq \mathrm{S}^-$$

and if $\mathrm{B} = \exists\mathrm{T}$*, then either* $\mathcal{T} \not\models \mathrm{T} \sqsubseteq \mathrm{S}$ *or* $\mathcal{T} \not\models \mathrm{T} \sqsubseteq \mathrm{S}^-$.

[1]A P upper bound for atomic counting queries in DL-Lite$^{\mathcal{H}}_{\mathsf{pos}}$ erroneously appears in Table 1 of Calvanese et al. [2020a], but was corrected in a later arXiv version [Calvanese et al., 2020b].

To formulate our trichotomy result, we recall that a *matching* in a graph $(\mathcal{V}, \mathcal{E})$ is a set of edges that are pairwise vertex-disjoint. The Perfect Matching problem asks whether there exists a matching such that every vertex is incident to one of its edges. Despite being the focus of intensive research, its exact complexity remains open: in P [Edmonds, 1965] and NL-hard [Chandra et al., 1984].

Theorem 54. *Let $\mathcal{T}$ be a DL-Lite$^{\mathcal{H}}_{\mathsf{pos}}$ TBox. Answering the role cardinality query $q_{\mathrm{S}} := \exists z_1 \exists z_2\ \mathrm{S}(z_1, z_2)$ over $\mathcal{T}$ is coNP-complete if $\mathcal{T}$ admits a non-trivial propagation of either S or S^-, is L-equivalent to the complement of* Perfect Matching *if it does not admit such a non-trivial propagation but admits a non-trivial pairing of S, and is in TC^0 otherwise.*

The rest of this section is devoted to the proof of the latter theorem. In Section 5.5.1, we generalize the reduction sketched in Example 29 to obtain coNP-hardness. If there is a non-trivial pairing (but no non-trivial propagation), we show in Section 5.5.2 that, up to trivial cases solvable in TC^0, the existence of a model with few matches is equivalent to the existence of a large matching between critical individuals. This yields L-equivalence with the Maximum Matching decision problem, which is L-equivalent to the better-known Perfect Matching problem [Rabin and Vazirani, 1989]. In Section 5.5.3 we conclude the proof with TC^0 membership, obtained by case analysis, where we exhibit for each case a model with an optimal (and easily computable) number of matches.

5.5.1 coNP-hardness in presence of propagation

We begin by proving coNP-hardness of answering the role cardinality q_{S} over a DL-Lite$^{\mathcal{H}}_{\mathsf{pos}}$ TBox $\mathcal{T}$ that admits a non-trivial propagation of either S or S^-. Recall that coNP membership is an immediate consequence of existing results on counting queries [Kostylev and Reutter, 2015].

Let us thus assume that $\mathcal{T}$ has a non-trivial propagation $\mathrm{B}, \mathrm{R}_1, \mathrm{R}_2$ of S (the case of a non-trivial propagation of S^- being symmetrical). We proceed by reduction from the Set Cover problem, and distinguish two cases based on the nature of the concept B.

Consider an instance $(\mathcal{U}, \mathcal{S}, k)$ of Set Cover: each element $u \in \mathcal{U}$ occurs in at least one subset of $\mathcal{S}$, we denote s_u by such a subset. We introduce an individual name u for each $u \in \mathcal{U}$, and an individual name s for each $s \in \mathcal{S}$. The individual introduced for the subset s_u is denoted $\mathsf{s_u}$. We further introduce auxiliary individuals a and b.

We now provide the reductions for the two cases.

Case 1: $\mathrm{B} \in \mathsf{N_C}$ or $\mathrm{B} = \exists \mathrm{T}$ with $\mathcal{T} \not\models \mathrm{T} \sqsubseteq \mathrm{S}$. We first describe the ABox. Elements from $\mathcal{U}$ are represented by facts $\{\mathrm{B}(\mathsf{u}) \mid u \in \mathcal{U}\}$ if $\mathrm{B} \in \mathsf{N_C}$, otherwise by facts $\{\mathrm{T}(\mathsf{u},\mathsf{a}) \mid u \in \mathcal{U}\}$ if $\mathrm{B} = \exists\mathrm{T}$. Subsets from $\mathcal{S}$ are represented by facts $\{\mathrm{S}(\mathsf{u},\mathsf{s}) \mid u \in s, s \in \mathcal{S}\}$. All the roles issuing from B and which are not subroles of S are also introduced as follows, pointing to auxiliary a:

$$\big\{\mathrm{U}(\mathsf{u},\mathsf{a}) \,\big|\, \mathrm{U} \in \mathsf{N}_\mathsf{R}^{\pm},\ u \in \mathcal{U},\ \mathcal{T} \models \mathrm{B} \sqsubseteq \exists\mathrm{U},\ \mathcal{T} \not\models \mathrm{U} \sqsubseteq \mathrm{S}\big\}$$

We proceed as well with the roles issuing from $\exists\mathrm{S}^-$, pointing either to b or to a.

$$\big\{\mathrm{U}(\mathsf{s},\mathsf{b}) \,\big|\, \mathrm{U} \in \mathsf{N}_\mathsf{R}^{\pm},\ s \in \mathcal{S},\ \mathcal{T} \models \exists\mathrm{S}^- \sqsubseteq \exists\mathrm{U},\ \mathcal{T} \models \mathrm{U} \sqsubseteq \mathrm{S}\big\}$$

and:

$$\big\{\mathrm{U}(\mathsf{s},\mathsf{a}) \,\big|\, \mathrm{U} \in \mathsf{N}_\mathsf{R}^{\pm},\ s \in \mathcal{S},\ \mathcal{T} \models \exists\mathrm{S}^- \sqsubseteq \exists\mathrm{U},\ \mathcal{T} \not\models \mathrm{U} \sqsubseteq \mathrm{S}\big\}$$

To complete our description of the ABox, we saturate a and b with facts:

$$\{\mathrm{U}(\mathsf{a},\mathsf{a}) \mid \mathrm{U} \in \mathsf{sig}(\mathcal{T})_\mathsf{R}\} \ \cup\ \{\mathrm{U}(\mathsf{b},\mathsf{b}) \mid \mathrm{U} \in \mathsf{sig}(\mathcal{T})_\mathsf{R}\}$$

Let $\mathcal{A}$ be the union of all preceding facts, and consider the KB $\mathcal{K} := (\mathcal{T},\mathcal{A})$. Notice that due to the many role assertions included in the ABox, all of the anonymous elements in $\mathcal{C}_\mathcal{K}$ are of the form $\mathsf{u}\mathrm{U}w$ with $u \in \mathcal{U}$, w some word, and $\mathcal{T} \models \mathrm{U} \sqsubseteq \mathrm{S}$ but $\mathcal{T} \not\models \mathrm{U}^- \sqsubseteq \mathrm{S}$ (because we are considering a non-trivial propagation, the role U cannot satisfy Condition 1 of Definition 68). Notice also that (the negation of) Condition 2 from the same definition further ensures that there is no ABox match (b,s), with $s \in \mathcal{S}$.

Let us denote by $m_\mathcal{A}$ be the number of matches for q_S present in the ABox $\mathcal{A}$. In particular, $m_\mathcal{A} \geq \sum_{s\in\mathcal{S}} |s| + 1$, due to the representation of the subsets and the saturation of a. We prove the following claim:

$$[m_\mathcal{A} + k + 1, +\infty] \text{ is a certain answer for } q_\mathrm{S} \text{ w.r.t. } \mathcal{K} \ \Leftrightarrow\ (\mathcal{U},\mathcal{S},k) \notin \textsc{Set Cover}.$$

($\Rightarrow$). Assume $(\mathcal{U},\mathcal{S},k) \in$ Set Cover. Take some k-cover $F \subseteq \mathcal{S}$ of $\mathcal{U}$. For each $u \in s$ with $s \in F$ and each positive role U such that $\mathcal{T} \models \mathrm{B} \sqsubseteq \exists\mathrm{U}$ and $\mathcal{T} \models \mathrm{U} \sqsubseteq \mathrm{S}$, enrich the ABox $\mathcal{A}$ with the assertion $\mathrm{U}(\mathsf{u},\mathsf{s})$. Saturate now the used subsets, that is, for each $s \in F$, add the assertions $\mathrm{U}(\mathsf{s},\mathsf{s})$ for all $\mathrm{U} \in \mathsf{sig}(\mathcal{T})_\mathsf{R}$.

Up to introducing the entailed concepts, the resulting interpretation $\mathcal{I}_F$ (based upon the described enriched ABox) is a model, as we introduced the missing roles for the elements, the used subsets are now saturated, and the non-used subsets were already given their needed roles.

In addition to the $m_\mathcal{A}$ ABox matches, each used subset provides one additional match since the assertion $\mathrm{S}(\mathsf{s},\mathsf{s})$ has been added. Recall Condition 1 from Definition 68 which ensures no match with shape $\mathrm{S}(\mathsf{s},\mathsf{u})$ is introduced, hence the roles added between the elements and subset individuals only reuse pre-existing matches. We thus obtain a model with exactly $m_\mathcal{A} + k$ matches, and thus a countermodel for $[m_\mathcal{A} + k + 1, +\infty]$ being a certain answer.

($\Leftarrow$). Assume $(\mathcal{U}, \mathcal{S}, k) \notin$ Set Cover. Consider a model $\mathcal{I}$ of $\mathcal{K}$ and a homomorphism $f : \mathcal{C}_\mathcal{K} \to \mathcal{I}$. For each $u \in \mathcal{U}$, we associate a subset $\rho(u) := s$ if $f(\mathsf{u}\mathrm{R}_1) = \mathsf{s}$ and $u \in s \in \mathcal{S}$, otherwise set $\rho(u) := s_u$. The image $\rho(\mathcal{U})$ is a covering of $\mathcal{U}$, hence $|\rho(\mathcal{U})| \geq k + 1$. By definition, for each $s \in \rho(\mathcal{U})$ there exists $u \in \mathcal{U}$ such that: either $f(\mathsf{u}\mathrm{R}_1) = \mathsf{s}$ with $u \in s \in \mathcal{S}$, or $f(\mathsf{u}\mathrm{R}_1) \neq \mathsf{s}'$ for all $u \in s' \in \mathcal{S}$.

In the first case, since $\mathcal{T} \models \{\mathrm{R}_1 \sqsubseteq \mathrm{S}, \exists\mathrm{R}_1^- \sqsubseteq \exists\mathrm{R}_2, \mathrm{R}_2 \sqsubseteq \mathrm{S}\}$ (due to the assumed non-trivial propagation), we focus on the pair $(f(\mathsf{u}\mathrm{R}_1), f(\mathsf{u}\mathrm{R}_1\mathrm{R}_2))$. If $(f(\mathsf{u}\mathrm{R}_1), f(\mathsf{u}\mathrm{R}_1\mathrm{R}_2))$ is not already an ABox match, then we have found an additional match. Otherwise $(f(\mathsf{u}\mathrm{R}_1), f(\mathsf{u}\mathrm{R}_1\mathrm{R}_2))$ is an ABox match (i.e. $\mathcal{K} \models \mathrm{S}(f(\mathsf{u}\mathrm{R}_1), f(\mathsf{u}\mathrm{R}_1\mathrm{R}_2))$. By construction of $\mathcal{A}$, this must be due to S propagating a subrole U of S (see 'Introduction of subroles of S for subsets' in the definition of the ABox), which means we have $f(\mathsf{u}\mathrm{R}_1\mathrm{R}_2) = \mathsf{b}$. Condition 2 from Definition 68 applied with U provides $\mathcal{T} \models \mathrm{R}_2^- \sqsubseteq \mathrm{S}$, hence $(f(\mathsf{u}\mathrm{R}_1\mathrm{R}_2), f(\mathsf{u}\mathrm{R}_1))$ is a new match (recall that (b, s) is not an ABox match!). In the second case, $(f(\mathsf{u}), f(\mathsf{u}\mathrm{R}_1))$ is a new match (in the case where $\mathrm{B} = \exists\mathrm{T}$, simply recall that $\mathcal{T} \not\models \mathrm{T} \sqsubseteq \mathrm{S}$). Therefore we can conclude that there are at least $m_\mathcal{A} + k + 1$ matches in $\mathcal{I}$.

Case 2: $\mathrm{B} = \exists\mathrm{T}$ **with** $\mathcal{T} \models \mathrm{T} \sqsubseteq \mathrm{S}$**.** We first describe the ABox. Elements from $\mathcal{U}$ and the subsets in which they occur are represented by facts $\{\mathrm{T}(\mathsf{u}, \mathsf{s}) \mid u \in s \in \mathcal{S}\}$. All the roles issuing from $\exists$T and which are not subroles of S are introduced as follows, pointing to auxiliary a:

$$\left\{\mathrm{U}(\mathsf{u}, \mathsf{a}) \,\middle|\, \mathrm{U} \in \mathsf{N}_\mathsf{R}^\pm,\ u \in \mathcal{U},\ \mathcal{T} \models \exists\mathrm{T} \sqsubseteq \exists\mathrm{U},\ \mathcal{T} \not\models \mathrm{U} \sqsubseteq \mathrm{S}\right\}$$

We proceed as well with the roles issuing from either $\exists\mathrm{T}^-$ or $\exists\mathrm{S}^-$, pointing either to b or to a:

$$\left\{\mathrm{U}(\mathsf{s}, \mathsf{b}) \,\middle|\, \mathrm{U} \in \mathsf{N}_\mathsf{R}^\pm,\ s \in \mathcal{S},\ \mathcal{T} \models \exists\mathrm{S}^- \sqsubseteq \exists\mathrm{U} \vee \mathcal{T} \models \exists\mathrm{T}^- \sqsubseteq \exists\mathrm{U}\ \mathcal{T} \models \mathrm{U} \sqsubseteq \mathrm{S}\right\}$$

and

$$\left\{\mathrm{U}(\mathsf{s}, \mathsf{a}) \,\middle|\, \mathrm{U} \in \mathsf{N}_\mathsf{R}^\pm,\ s \in \mathcal{S},\ \mathcal{T} \models \exists\mathrm{S}^- \sqsubseteq \exists\mathrm{U},\ \mathcal{T} \not\models \mathrm{U} \sqsubseteq \mathrm{S}\right\}$$

To complete our description of the ABox, we saturate a and b with facts:

$$\{\mathrm{U}(\mathsf{a}, \mathsf{a}) \mid \mathrm{U} \in \mathsf{sig}(\mathcal{T})_\mathsf{R}\} \ \cup\ \{\mathrm{U}(\mathsf{b}, \mathsf{b}) \mid \mathrm{U} \in \mathsf{sig}(\mathcal{T})_\mathsf{R}\}$$

We again let $\mathcal{A}$ be the union of all the preceding facts and set $\mathcal{K} := (\mathcal{T}, \mathcal{A})$ and observe that the anonymous elements in $\mathcal{C}_\mathcal{K}$ are all of the form $\mathsf{u}\mathrm{U}w$ with $u \in \mathcal{U}$, $\mathcal{T} \models \mathrm{U} \sqsubseteq \mathrm{S}$, and w a word. Notice again that *no* pair (b, s), with $s \in \mathcal{S}$, is an ABox match, which is due here to *both* Conditions 2 and 3 from Definition 68.

As before, we denote by $m_\mathcal{A}$ the number of matches for q_S in the ABox $\mathcal{A}$. In particular $m_\mathcal{A} \geq \sum_{s \in \mathcal{S}} |s| + 1$, due to the representation of the problem instance and the saturation of a. We establish the following claim:

$[m_\mathcal{A} + k + 1, +\infty]$ is a certain answer for q_S w.r.t. $\mathcal{K} \iff (\mathcal{U}, \mathcal{S}, k) \notin$ Set Cover.

($\Rightarrow$). The proof is essentially the same as for Case 1.

($\Leftarrow$). Assume $(\mathcal{U}, \mathcal{S}, k) \notin$ Set Cover. Consider a model $\mathcal{I}$ of $\mathcal{K}$ and a homomorphism $f : \mathcal{C}_\mathcal{K} \to \mathcal{I}$. For each $u \in \mathcal{U}$, we associate a subset $\rho(u) := s$ if $f(\mathsf{u}\mathrm{R}_1) = \mathsf{s}$ and $u \in s \in \mathcal{S}$, otherwise set $\rho(u) := s_u$. The image $\rho(\mathcal{U})$ is a covering of $\mathcal{U}$, hence $|\rho(\mathcal{U})| \geq k+1$. By definition, for each $s \in \rho(\mathcal{U})$, there exists $u \in \mathcal{U}$ such that: either $f(\mathsf{u}\mathrm{R}_1) = \mathsf{s}$ with $u \in s \in \mathcal{S}$, or $f(\mathsf{u}\mathrm{R}_1) \neq \mathsf{s}'$ for all $u \in s' \in \mathcal{S}$.

In the first case, since $\mathcal{T} \models \{\mathrm{R}_1 \sqsubseteq \mathrm{S}, \exists\mathrm{R}_1^- \sqsubseteq \exists\mathrm{R}_2, \mathrm{R}_2 \sqsubseteq \mathrm{S}\}$ (due to the assumed non-trivial propagation), we focus on the pair $(f(\mathsf{u}\mathrm{R}_1), f(\mathsf{u}\mathrm{R}_1\mathrm{R}_2))$. If $(f(\mathsf{u}\mathrm{R}_1), f(\mathsf{u}\mathrm{R}_1\mathrm{R}_2))$ is not already an ABox match, then we are done. Otherwise $(f(\mathsf{u}\mathrm{R}_1), f(\mathsf{u}\mathrm{R}_1\mathrm{R}_2))$ is an ABox match, then by construction of $\mathcal{A}$, it must be due to either S or T propagating a subrole U of S, in particular, we get $f(\mathsf{u}\mathrm{R}_1\mathrm{R}_2) = \mathsf{b}$. Condition 2 (resp. Condition 3) from Definition 68 applied with U provides $\mathcal{T} \models \mathrm{R}_2^- \sqsubseteq \mathrm{S}$, hence $(f(\mathsf{u}\mathrm{R}_1\mathrm{R}_2), f(\mathsf{u}\mathrm{R}_1))$ is a new match (recall (b, s) is not an ABox match!). In the second case, $(f(\mathsf{u}), f(\mathsf{u}\mathrm{R}_1))$ is a new match. Therefore there are at least $m_\mathcal{A} + k + 1$ matches in $\mathcal{I}$.

5.5.2 Equivalence with Perfect Matching

We now turn to the second part of Theorem 54, which characterizes the complexity of answering q_S over $\mathcal{T}$ in the case in which the DL-Lite$^\mathcal{H}_\mathsf{pos}$ TBox $\mathcal{T}$ admits a non-trivial pairing of S but does not have any non-trivial propagation of S or S$^-$. We start by proving a logspace reduction from the complement of Maximum Matching to our problem. The problem Maximum Matching asks whether, given a non-oriented graph $\mathcal{G}$ and an integer k, there exists a matching of $\mathcal{G}$ with size at least k. As explained before, the latter problem is known to be equivalent, up to logspace reductions, to the better known Perfect Matching problem [Rabin and Vazirani, 1989]. Thus, the reduction we give also proves a reduction from Perfect Matching to our problem.

Proof of the reduction from Maximum Matching. Consider a DL-Lite$^\mathcal{H}_\mathsf{pos}$ TBox $\mathcal{T}$ that admits a non-trivial pairing of S and does not admit any non-trivial propagation of S or S$^-$. Let B and R verify the pairing conditions, that is,

$$\mathcal{T} \models \mathrm{B} \sqsubseteq \exists\mathrm{R} \quad \mathcal{T} \models \mathrm{R} \sqsubseteq \mathrm{S} \quad \mathcal{T} \models \mathrm{R} \sqsubseteq \mathrm{S}^- \quad \mathcal{T} \not\models \mathrm{S} \sqsubseteq \mathrm{S}^-$$

and if $\mathrm{B} = \exists\mathrm{T}$, then either $\mathcal{T} \not\models \mathrm{T} \sqsubseteq \mathrm{S}$ or $\mathcal{T} \not\models \mathrm{T} \sqsubseteq \mathrm{S}^-$.

Consider an instance of Maximum Matching given by the undirected graph $\mathcal{G} := (\mathcal{V}, \mathcal{E})$ and integer k. Let $\leq_\mathcal{V}$ be any total order on the vertices of $\mathcal{G}$. We encode $\mathcal{G}$ using the following ABox $\mathcal{A}_\mathcal{G}$:

$$\mathcal{A}_\mathcal{G} := \left| \begin{array}{ll} \{\mathrm{B}(\mathsf{u}) \mid u \in \mathcal{V}\} & \text{if } \mathrm{B} \in \mathsf{N_C} \\ \{\mathrm{T}(\mathsf{u},\mathsf{a}) \mid u \in \mathcal{V}\} & \text{else, with } \mathrm{B} = \exists\mathrm{T} \end{array} \right. \qquad \text{(Representing vertices)}$$
$$\cup \{\mathrm{S}(\mathsf{u},\mathsf{v}) \mid \{u,v\} \in \mathcal{E}, u \leq_\mathcal{V} v\} \qquad \text{(Representing edges)}$$
$$\cup \{\mathrm{U}(\mathsf{a},\mathsf{a}) \mid \mathrm{U} \in \mathsf{sig}(\mathcal{T})_\mathsf{R}\} \qquad \text{(Saturating } \mathsf{a})$$

Let $\mathcal{K}_\mathcal{G}$ be the KB $(\mathcal{T}, \mathcal{A}_\mathcal{G})$. Let $m_\mathcal{A}$ be the number of matches in the ABox. Notice each edge $\{u,v\}$ gives one match in the ABox, through the added assertion $\mathrm{S}(\mathsf{u},\mathsf{v})$ with $u \leq_\mathcal{V} v$, and *exactly* one as $\mathcal{T} \not\models \mathrm{S}^- \sqsubseteq \mathrm{S}$. We claim that $[m_\mathcal{A} + |\mathcal{V}| - k + 1, +\infty]$ is a certain answer for q_S w.r.t. $\mathcal{K}_\mathcal{G}$ iff $(\mathcal{G}, k) \notin$ MAXIMUM MATCHING. Notice that both $\mathcal{A}_\mathcal{G}$ and the integer $m_\mathcal{A} + |\mathcal{V}| - k + 1$ are easily computable in logarithmic space from any reasonable representation of the instance $(\mathcal{G}, k)$, so we will get the desired within logspace reduction.

($\Leftarrow$). Assume $(\mathcal{G}, k) \notin$ MAXIMUM MATCHING. Consider a model $\mathcal{I}$ of $\mathcal{K}$ and a homomorphism $f : \mathcal{C}_\mathcal{K} \to \mathcal{I}$. Consider the following matching:

$$M_\mathcal{I} := \{\{u,v\} \in \mathcal{E} \mid f(\mathsf{u}\mathrm{R}) = \mathsf{v}, f(\mathsf{v}\mathrm{R}) = \mathsf{u}\}$$

As f is a function, it is clear that each vertex is incident to at most one of the edges from $M_\mathcal{I}$, ensuring $M_\mathcal{I}$ is a matching. In particular, it yields $|M_\mathcal{I}| < k$. Each edge from $M_\mathcal{I}$ provides exactly one additional match, since there was already exactly one match per edge, and the role R is a subrole of both S and S^-. Each vertex that is not incident to any edge in $M_\mathcal{I}$ provides at least one additional match: recall that since $\mathcal{T} \not\models \mathrm{T} \sqsubseteq \mathrm{S}$ or $\mathcal{T} \not\models \mathrm{T}^- \sqsubseteq \mathrm{S}$, either $(f(\mathsf{u}), f(\mathsf{u}\mathrm{R}))$ or $(f(\mathsf{u}\mathrm{R}), f(\mathsf{u}))$ is a new match.Therefore there are at least $m_\mathcal{A} + |M_\mathcal{I}| + |\mathcal{V}| - 2|M_\mathcal{I}| > m_\mathcal{A} + |\mathcal{V}| - k$ matches in $\mathcal{I}$.

($\Rightarrow$). Assume $(\mathcal{G}, k) \in$ MAXIMUM MATCHING. Consider a matching $M \subseteq \mathcal{E}$ with $|M| \geq k$. Consider the enriched ABox $\mathcal{A}_M$ such that for each $\{u,v\} \in M$ and each positive role $\mathrm{U} \in \mathsf{N}^\pm_\mathsf{R}$, we have $\mathrm{U}(\mathsf{u},\mathsf{v}) \in \mathcal{A}_M$. This yields exactly one additional match per edge in M, again because exactly one match per edge was already present. For each $u \in \mathcal{V}$ such that u is not incident to any edge in M, also add all the assertions $\mathrm{U}(\mathsf{u},\mathsf{u}) \in \mathcal{A}_M$. This yields exactly one new match per vertex not incident to any edge in M. Up to adding the entailed concepts wherever needed, this provides a model with at most: $m_\mathcal{A} + |\mathcal{E}| + |\mathcal{V}| - 2|\mathcal{E}| \leq m_\mathcal{A} + |\mathcal{V}| - k$ matches of q_S, being a counter model for $[m_\mathcal{A} + |\mathcal{V}| - k + 1, +\infty]$. □

We complete the proof of the second part of Theorem 2 by showing how answering q_S over $\mathcal{T}$ can be reduced, via logspace reductions, to the complement of Maximum Matching in the case in which $\mathcal{T}$ is a DL-Lite$^{\mathcal{H}}_{\mathsf{pos}}$ TBox without non-trivial propagation. Again, this yields a logspace reduction to the complement of Perfect Matching due to the previously cited logspace-equivalence between these two matching problems. We let q_{S} be our role cardinality and $\mathcal{T}$ a DL-Lite$^{\mathcal{H}}_{\mathsf{pos}}$ TBox $\mathcal{T}$ without non-trivial propagation and start with some general remarks.

Compared with the tractable settings of Section 5.4.1, with DL-Lite$^{\mathcal{H}}_{\mathsf{pos}}$ we no longer need to take care of negative concept inclusions, but we will now need to take into account role inclusions when handling role cardinality queries. In particular, role inclusions allow for a class $\mathcal{B}_{\mathcal{T}}$ of what we call *bipotent* roles, *i.e.,* subroles of both S and S$^-$ (formally: positive roles U such that $\mathcal{T} \models \mathrm{U} \sqsubseteq \mathrm{S}$ and $\mathcal{T} \models \mathrm{U} \sqsubseteq \mathrm{S}^-$). On the other hand, the class $\mathcal{N}_{\mathcal{T}}$ of positive roles not being a subrole of S nor a subrole of S$^-$ are called *nilpotent* (formally: positive roles U such that $\mathcal{T} \not\models \mathrm{U} \sqsubseteq \mathrm{S}$ and $\mathcal{T} \not\models \mathrm{U} \sqsubseteq \mathrm{S}^-$).

Recall that our previous notion of type aimed to characterize individuals based on their ability to receive some roles (is there a negative concept preventing my anonymous element to merge with this individual?) and to provide ABox matches on which to fold (is there an ABox match on which to fold matches propagated by a given anonymous element?). This typing notion needs to be modified for the setting we consider here. On the one hand, negative inclusions being disallowed, all individuals are able to receive all roles. On the other hand, we must now distinguish ABox matches on which we can fold bipotent roles from those on which we can only fold non-bipotent roles. We also extend our typing notion to nilpotent roles: their type being a characterization of the subroles they propagate.

Definition 70. *Let $\mathcal{A}$ be an ABox and $\mathcal{K} := (\mathcal{T}, \mathcal{A})$. The* type $\theta_{\mathcal{K}}(d)$ of an element $d \in \mathsf{Ind}(\mathcal{A})$ over $\mathcal{K}$ *is the set:*

$$\theta_{\mathcal{K}}(d) \;:=\; \left\{ \mathfrak{R} \;\middle|\; \begin{array}{l} \mathfrak{R} \in \{\{\mathrm{S}, \mathrm{S}^-\}, \{\mathrm{S}\}, \{\mathrm{S}^-\}\} \\ \exists e \in \Delta^{\mathcal{C}_{\mathcal{K}}} \; \forall \mathrm{R} \in \mathfrak{R},\; \mathcal{C}_{\mathcal{K}} \models \mathrm{R}(d, e) \end{array} \right\}.$$

The type $\theta_{\mathcal{K}}(R)$ of a nilpotent role $\mathrm{R} \in \mathcal{N}_{\mathcal{T}}$ over $\mathcal{K}$ *is the set:*

$$\theta_{\mathcal{K}}(R) \;:=\; \left\{ \mathfrak{U} \;\middle|\; \begin{array}{l} \mathfrak{U} \in \{\{\mathrm{S}, \mathrm{S}^-\}, \{\mathrm{S}\}, \{\mathrm{S}^-\}\} \\ \exists \mathrm{V} \in \mathsf{N}^{\pm}_{\mathsf{R}}, \forall \mathrm{U} \in \mathfrak{U},\; \mathcal{T} \models \exists \mathrm{R}^- \sqsubseteq \exists \mathrm{V} \;\wedge\; \mathcal{T} \models \mathrm{V} \sqsubseteq \mathrm{U} \end{array} \right\}.$$

The set $\Theta_{\mathcal{T}}$ of possible types is hence:

$$\{\{\{\mathrm{S}, \mathrm{S}^-\}, \{\mathrm{S}\}, \{\mathrm{S}^-\}\}, \{\{\mathrm{S}\}, \{\mathrm{S}^-\}\}, \{\{\mathrm{S}^-\}\}, \{\{\mathrm{S}\}\}, \emptyset\}.$$

Following the line of the TC^0 membership proofs for role cardinality queries, we are still interested in *demanding elements*. In particular, bipotent roles might create a new kind of such elements: *bidemanding elements*, which are defined as follows.

Definition 71. *Let $\mathcal{A}$ be an ABox and $\mathcal{K} := (\mathcal{T}, \mathcal{A})$. We consider* bidemanding individuals $\mathcal{D}^{\pm}_{\mathcal{K}}$ *and* bidemanding roles $\mathcal{D}^{\pm}_{\sigma}$ *as follows:*

$$\mathcal{D}^{\pm}_{\mathcal{K}} := \left\{ \mathsf{a} \;\middle|\; \begin{array}{l} \mathsf{a} \in \mathsf{Ind}(\mathcal{A}) \\ \{\mathrm{S}, \mathrm{S}^-\} \in \theta_{\mathcal{K}}(\mathsf{a}) \\ \forall \mathsf{b} \in \mathsf{Ind}(\mathcal{A}),\ (\mathcal{K} \not\models \mathrm{S}(\mathsf{a}, \mathsf{b})) \vee (\mathcal{K} \not\models \mathrm{S}^-(\mathsf{a}, \mathsf{b})) \end{array} \right\}$$

$$\mathcal{D}^{\pm}_{\sigma} := \left\{ \mathrm{R} \;\middle|\; \begin{array}{l} \mathrm{R} \in \mathsf{gen}_{\mathcal{K}} \\ \{\mathrm{S}, \mathrm{S}^-\} \in \theta_{\mathcal{K}}(R) \end{array} \right\}$$

Notice here the assumptions that bidemanding roles should be nilpotent and not only “non-bipotent”.We now redefine for our setting the notions of positive / negative demanding individuals.

Definition 72. *Let $\mathcal{A}$ be an ABox and $\mathcal{K} := (\mathcal{T}, \mathcal{A})$. Define* positive demanding individuals $\mathcal{D}^{+}_{\mathcal{K}}$, *resp.* negative demanding individuals $\mathcal{D}^{-}_{\mathcal{K}}$ *as:*

$$\mathcal{D}^{+}_{\mathcal{K}} := \left\{ \mathsf{a} \;\middle|\; \begin{array}{l} \mathsf{a} \in \mathsf{Ind}(\mathcal{A}) \setminus \mathcal{D}^{\pm}_{\mathcal{K}} \\ \{\mathrm{S}\} \in \theta_{\mathcal{K}}(\mathsf{a}) \\ \forall \mathsf{b} \in \mathsf{Ind}(\mathcal{A}),\ \mathcal{K} \not\models \mathrm{S}(\mathsf{a}, \mathsf{b}) \end{array} \right\}$$

$$\mathcal{D}^{-}_{\mathcal{K}} := \left\{ \mathsf{a} \;\middle|\; \begin{array}{l} \mathsf{a} \in \mathsf{Ind}(\mathcal{A}) \setminus \mathcal{D}^{\pm}_{\mathcal{K}} \\ \{\mathrm{S}^-\} \in \theta_{\mathcal{K}}(\mathsf{a}) \\ \forall \mathsf{b} \in \mathsf{Ind}(\mathcal{A}),\ \mathcal{K} \not\models \mathrm{S}^-(\mathsf{a}, \mathsf{b}) \end{array} \right\}$$

Strategies are no longer needed in our setting, as negative inclusions have been removed. Due to the adaptation of our notions of types, a choice of well-typed elements is redefined to now apply to types (of roles) instead of applying to the positive roles themselves. This is simply because the absence of negative concept inclusions allows us to apply the same choice to all nilpotent roles having the same type.

Definition 73. *Let $\mathcal{A}$ be an ABox and $\mathcal{K} := (\mathcal{T}, \mathcal{A})$. A choice of well-typed elements for $\mathcal{K}$ is a function $\mathsf{ch}_{\sigma/\mathcal{K}} : \Theta_{\mathcal{T}} \to \mathsf{Ind}(\mathcal{A})$ such that for each type $\mathsf{t} \in \Theta_{\mathcal{T}}$, if there exists a nilpotent generated role $\mathrm{R} \in \mathsf{gen}_{\mathcal{K}} \cap \mathcal{N}_{\mathcal{T}}$ such that $\theta_{\mathcal{K}}(\mathrm{R}) = \mathsf{t}$, then we have $\mathsf{t} \subseteq \theta_{\mathcal{K}}(\mathsf{ch}_{\sigma/\mathcal{K}}(\mathsf{t}))$.*

We now state our fundamental theorem, which proves that, if a choice of well-typed elements is available and in the absence of demanding individuals, then the canonical model can fully fold on the individuals without creating any additional match. This central property crucially relies on the absence of a non-trivial propagation schema.

Theorem 55. *Let $\mathcal{A}$ be an ABox and $\mathcal{K} := (\mathcal{T}, \mathcal{A})$. If there is a choice $\mathsf{ch}_{\sigma/\mathcal{K}}$ of well-typed elements over $\mathsf{Ind}(\mathcal{A})$ and if $\mathcal{K}$ admits no bidemanding individuals, then there exists a mapping $\chi : \Delta^{\mathcal{C}_{\mathcal{K}}} \to \mathsf{Ind}(\mathcal{A})$ s.t. the matches in the resulting model $\chi(\mathcal{C}_{\mathcal{K}})$ are exactly the ABox matches.*

Proof sketch. The proof proceeds by induction on $\mathcal{C}_{\mathcal{K}}$, exploiting the choice of well-typed elements to build an image for each of its elements. As it requires some additional technical definitions, the full proof is deferred to the appendix. □

With this key result in hand, we can now observe that, in the absence of bidemanding individuals, our problem is easy to decide: within TC^0. Indeed, without bidemanding individuals, the best way to combine positive and negative demanding individuals is still to pair them 1-to-1. Therefore, the optimal number of matches can easily be decided by counting such elements.

Lemma 39. *Let $\mathcal{A}$ be an ABox and $\mathcal{K} := (\mathcal{T}, \mathcal{A})$. If $\mathcal{K}$ admits no bidemanding individuals, then the minimal number of matches can be decided within* TC^0.

Proof. Assume $\mathcal{K}$ does not admit any bidemanding individuals. Set a classic pairing $\mathsf{pair}_{\mathsf{ch}_{\sigma/\mathcal{K}}} := (\mathsf{pair}^+_{\mathsf{ch}_{\sigma/\mathcal{K}}}, \mathsf{pair}^-_{\mathsf{ch}_{\sigma/\mathcal{K}}})$ for positive and negative demanding individuals. We distinguish several cases, but the proof idea is always the same: in each case we exhibit the optimal number of matches that can be easily computed from the types of individuals. We then prove it is minimal and exhibit a model with this precise number of matches using Theorem 55 on ABox $\mathcal{A}^*$ and some $\mathsf{ch}_{\mathcal{K}^*}$ that will be specified in each case:

$$\begin{array}{rcl}\mathcal{A}^* & := & \mathcal{A} \cup \{\mathrm{S}(\mathsf{x},\mathsf{y}) \mid \mathsf{pair}^+_{\mathsf{ch}_{\sigma/\mathcal{K}}}(x) = y\} \\ & & \cup \left\{\mathrm{S}(\mathsf{x},\mathsf{x}) \,\middle|\, \begin{array}{l} x \in \mathcal{D}^+_{\mathcal{K}} \\ x \notin \mathsf{dom}(\mathsf{pair}^+_{\mathsf{ch}_{\sigma/\mathcal{K}}}) \end{array}\right\} \\ & & \cup \left\{\mathrm{S}(\mathsf{x},\mathsf{x}) \,\middle|\, \begin{array}{l} x \in \mathcal{D}^-_{\mathcal{K}} \\ x \notin \mathsf{dom}(\mathsf{pair}^-_{\mathsf{ch}_{\sigma/\mathcal{K}}}) \end{array}\right\}\end{array}$$

The recurrent arguments to prove minimality are the following mappings, always defined and injective, $\mathcal{I}$ being a model of $\mathcal{K}$:

$$\begin{array}{rcl}\rho^+ : \mathcal{D}^+_{\mathcal{K}} & \to & \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \\ x & \mapsto & (x, \mathsf{succ}^{\mathcal{I}}_{\mathrm{S}}(x))\end{array} \qquad \begin{array}{rcl}\rho^- : \mathcal{D}^-_{\mathcal{K}} & \to & \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \\ x & \mapsto & (\mathsf{succ}^{\mathcal{I}}_{\mathrm{S}^-}(x), x)\end{array}$$

We denote $M := m_{\mathcal{A}} + \max(|\mathcal{D}^+_{\mathcal{K}}|, |\mathcal{D}^-_{\mathcal{K}}|)$.

1. If there exists an individual a such that $\{\mathrm{S}, \mathrm{S}^-\} \in \theta_{\mathcal{K}}(\mathsf{a})$. Optimum is M reached with $\mathcal{A}^*$. $\mathcal{A}^*$ does not admit demanding elements, and we choose, for all $\mathsf{t} \in \Theta_{\mathcal{T}}$, and $\mathsf{ch}_{\mathcal{K}^*}(\mathsf{t}) := \mathsf{a}$.

2. Else if $|\mathcal{D}^+_{\mathcal{K}}| > |\mathcal{D}^-_{\mathcal{K}}|$. Optimum is M reached with $\mathcal{A}^*$ not admitting demanding elements and setting $\forall \mathsf{t} \in \Theta_{\mathcal{T}}, \mathsf{ch}_{\mathcal{K}^*}(\mathsf{t}) \in \mathcal{D}^+_{\mathcal{K}} \setminus \mathsf{dom}(\mathsf{pair}^+_{\mathsf{ch}_{\sigma/\mathcal{K}}})$.

3. Else if $|\mathcal{D}^+_{\mathcal{K}}| < |\mathcal{D}^-_{\mathcal{K}}|$. Optimum is M reached with $\mathcal{A}^*$ not admitting demanding elements and setting $\forall \mathsf{t} \in \Theta_{\mathcal{T}}, \mathsf{ch}_{\mathcal{K}^*}(\mathsf{t}) \in \mathcal{D}^-_{\mathcal{K}} \setminus \mathsf{dom}(\mathsf{pair}^-_{\mathsf{ch}_{\sigma/\mathcal{K}}})$.

4. Else if there exists an individual $\mathsf{a} \in \mathcal{D}^+_\mathcal{K} \cap \mathcal{D}^-_\mathcal{K}$. Optimum is M reached with $\mathcal{A}^*$ not admitting demanding elements, assuming w.l.o.g $\mathsf{pair}^+_{\mathsf{ch}_{\sigma/\mathcal{K}}}(\mathsf{a}) = \mathsf{a}$, and setting $\forall \mathsf{t} \in \Theta_\mathcal{T}, \mathsf{ch}_{\mathcal{K}^*}(\mathsf{t}) := \mathsf{a}$.

5. Else if there exists $(\mathsf{a}, \mathsf{b}) \in \mathcal{D}^+_\mathcal{K} \times \mathcal{D}^-_\mathcal{K}$ such that $\mathcal{K} \models \mathrm{S}(\mathsf{b}, \mathsf{a})$. Optimum is M reached with $\mathcal{A}^*$ not admitting demanding elements, assuming w.l.o.g $\mathsf{pair}^+_{\mathsf{ch}_{\sigma/\mathcal{K}}}(\mathsf{a}) = \mathsf{b}$, and setting $\forall \mathsf{t} \in \Theta_\mathcal{T}, \mathsf{ch}_{\mathcal{K}^*}(\mathsf{t}) := \mathsf{a}$.

6. Else if there exists a role $\mathrm{R} \in \mathcal{N}_\mathcal{T} \cap \mathsf{gen}_\mathcal{K}$ such that $\{\mathrm{S}, \mathrm{S}^-\} \in \theta_\mathcal{K}(R)$. Let V be a bipotent role generated by R^-. Optimum is $M+1$ reached with $\mathcal{A}^* \cup \{\mathrm{S}(\bot, \bot)\}$ not admitting demanding elements and setting $\forall \mathsf{t} \in \Theta_\mathcal{T}, \mathsf{ch}_{\mathcal{K}^*}(\mathsf{t}) := \bot$. To ensure this number of matches is still a lower bound for the number of matches in any model $\mathcal{I}$, we need to specify where the extra match can be found in any model $\mathcal{I}$ of $\mathcal{K}$. Consider $f : \mathcal{C}_\mathcal{K} \to \mathcal{I}$ a homomorphism. Because of all the excluded previous cases, it can be verified that either $(f(\mathsf{repr}_\mathcal{K}(\mathrm{R})), \mathsf{succ}^\mathcal{I}_\mathrm{V}(f(\mathsf{repr}_\mathcal{K}(\mathrm{R}))))$ or $(\mathsf{succ}^\mathcal{I}_\mathrm{V}(f(\mathsf{repr}_\mathcal{K}(\mathrm{R}))), f(\mathsf{repr}_\mathcal{K}(\mathrm{R})))$ is an additional match in $\mathcal{I}$ (in particular, not already counted by one's favorite mapping ρ^+ or ρ^-).

7. Else if there exists an individual a such that $\{\mathrm{S}\}, \{\mathrm{S}^-\} \in \theta_\mathcal{K}(\mathsf{a})$. Optimum is M reached with $\mathcal{A}^*$ not admitting demanding elements and setting $\forall \mathsf{t} \in \Theta_\mathcal{T}, \mathsf{ch}_{\mathcal{K}^*}(\mathsf{t}) := \mathsf{a}$.

8. Else if there exists a role $\mathrm{R} \in \mathcal{N}_\mathcal{T} \cap \mathsf{gen}_\mathcal{K}$ and $\{\mathrm{S}\}, \{\mathrm{S}^-\} \in \theta_\mathcal{K}(\mathrm{R})$. Optimum is $M + 1$ reached with $\mathcal{A}^* \cup \{\mathrm{S}(\bot, \bot)\}$ not admitting demanding elements and $\forall \mathsf{t} \in \Theta_\mathcal{T}, \mathsf{ch}_{\mathcal{K}^*}(\mathsf{t}) := \bot$. Again, we need to specify where the extra match can be found in any model $\mathcal{I}$ of $\mathcal{K}$. Consider $f : \mathcal{C}_\mathcal{K} \to \mathcal{I}$ a homomorphism. It can be verified that either $(f(\mathsf{repr}_\mathcal{K}(\mathrm{R})), \mathsf{succ}^\mathcal{I}_\mathrm{S}(f(\mathsf{repr}_\mathcal{K}(\mathrm{R}))))$ or $(\mathsf{succ}^\mathcal{I}_{\mathrm{S}^-}(f(\mathsf{repr}_\mathcal{K}(\mathrm{R}))), f(\mathsf{repr}_\mathcal{K}(\mathrm{R})))$ is an additional match in $\mathcal{I}$ (in particular, not already counted by one's favorite mapping ρ^+ or ρ^-).

9. Else if there exists an individual a such that $\{\mathrm{S}\} \in \theta_\mathcal{K}(\mathsf{a})$. Optimum is M reached with $\mathcal{A}^*$ not admitting demanding elements and $\mathsf{ch}_{\sigma/\mathcal{K}}(\{\mathrm{S}\}) := \mathsf{a}$ and $\mathsf{ch}_{\sigma/\mathcal{K}}(\{\mathrm{S}^-\}) := \mathsf{b}$ with b the other endpoint (either certain or from pairing).

10. Else if there exists a role $\mathrm{R} \in \mathcal{N}_\mathcal{T} \cap \mathsf{gen}_\mathcal{K}$ and either $\{\mathrm{S}\} \in \theta_\mathcal{K}(\mathrm{R})$ or $\{\mathrm{S}^-\} \in \theta_\mathcal{K}(R)$. Optimum is $m_\mathcal{A} + 1$ reached with $\mathcal{A}^* \cup \{\mathrm{S}(\bot, \bot)\}$ not admitting demanding elements and setting $\forall \mathsf{t} \in \Theta_\mathcal{T}, \mathsf{ch}_{\mathcal{K}^*}(\mathsf{t}) := \bot$. Again, we need to specify where the extra match can be found in any model $\mathcal{I}$ of $\mathcal{K}$. Consider $f : \mathcal{C}_\mathcal{K} \to \mathcal{I}$ a homomorphism. It can be verified that either $(f(\mathsf{repr}_\mathcal{K}(\mathrm{R})), \mathsf{succ}^\mathcal{I}_\mathrm{S}(f(\mathsf{repr}_\mathcal{K}(\mathrm{R}))))$ or $(\mathsf{succ}^\mathcal{I}_{\mathrm{S}^-}(f(\mathsf{repr}_\mathcal{K}(\mathrm{R}))), f(\mathsf{repr}_\mathcal{K}(\mathrm{R})))$ is an additional match in $\mathcal{I}$ (in particular, not already counted by one's favorite mapping ρ^+ or ρ^-).

11. Otherwise. Optimum is $m_{\mathcal{A}}$ with $\mathcal{A}$ not admitting demanding elements and setting $\forall \mathsf{t} \in \Theta_{\mathcal{T}}, \mathsf{ch}_{\mathcal{K}^*}(\mathsf{t}) := \bot$.

To conclude the TC^0 membership proof, we describe the slight changes required to adapt the circuits already provided for the role cardinality queries:

- In the block "A closer look at roles and concepts over the input", one should extend the inputs to all subroles of R.
- The typing block should be adapted to fit the new typing notion (see Definition 70).
- All the blocks, each dedicated to a single strategy, can now be united as a single block computing $|\mathcal{D}^+_{\mathcal{K}}|$ and $|\mathcal{D}^-_{\mathcal{K}}|$.
- From this previous step, the typing block and the generated roles block, deciding if Situation 6, 8 or 10 occurs is easy, in which case one should add 1 to the final number of matches. □

We can now prove the desired reduction from our problem to Maximum Matching.

Proof of the reduction. Let $\mathcal{A}$ be an ABox and $\mathcal{K} := (\mathcal{T}, \mathcal{A})$.

If $\mathcal{K}$ does not admit bidemanding individuals, then Lemma 39 ensures we can actually compute the answer within TC^0, in particular within L, and create a trivial instance of Maximum Matching co-equivalent to it.

Otherwise, there are some bidemanding individuals. Consider then the following graph $\mathcal{G}_{\mathcal{K}}$:

$$
\begin{aligned}
\mathcal{V} &:= (\mathcal{D}^+_{\mathcal{K}} \times \{1\}) \cup (\mathcal{D}^-_{\mathcal{K}} \times \{-1\}) \cup \mathcal{D}^{\pm}_{\mathcal{K}} \\
\mathcal{E} &:= \{\{(x,1),(y,-1)\} \mid (x,y) \in \mathcal{D}^+_{\mathcal{K}} \times \mathcal{D}^-_{\mathcal{K}}\} \\
&\cup \left\{ \{x,(y,1)\} \;\middle|\; \begin{array}{l} (x,y) \in \mathcal{D}^{\pm}_{\mathcal{K}} \times \mathcal{D}^+_{\mathcal{K}} \\ \mathcal{K} \models \mathsf{S}(\mathsf{x},\mathsf{y}) \end{array} \right\} \\
&\cup \left\{ \{x,(y,-1)\} \;\middle|\; \begin{array}{l} (x,y) \in \mathcal{D}^{\pm}_{\mathcal{K}} \times \mathcal{D}^-_{\mathcal{K}} \\ \mathcal{K} \models \mathsf{S}^-(\mathsf{x},\mathsf{y}) \end{array} \right\} \\
&\cup \left\{ \{x,y\} \;\middle|\; \begin{array}{l} (x,y) \in \mathcal{D}^{\pm}_{\mathcal{K}} \times \mathcal{D}^{\pm}_{\mathcal{K}} \\ \mathcal{K} \models \mathsf{S}(\mathsf{x},\mathsf{y}) \end{array} \right\}
\end{aligned}
$$

We claim $[k, +\infty]$ is a certain answer for q_{S} w.r.t. $\mathcal{K}$ iff $(\mathcal{G}, m_{\mathcal{A}} + |\mathcal{V}| - k + 1) \notin$ Maximum Matching. Notice the graph $\mathcal{G}$ and the integer $m_{\mathcal{A}} + |\mathcal{V}| - k + 1$ are easily computable within L.

($\Leftarrow$). Assume $(\mathcal{G}, m_{\mathcal{A}} + |\mathcal{V}| - k + 1) \notin$ MAXIMUM MATCHING. Consider a model $\mathcal{I}$ of $\mathcal{K}$ and a homomorphism $f : \mathcal{C}_{\mathcal{K}} \to \mathcal{I}$. For each demanding individual x, we denote R_x a role causing this element to be demanding (that is, a bipotent role for bidemanding elements, a subrole of S for positive demanding elements, a subrole of S^- for negative demanding elements such that $x\mathrm{R}_x \in \Delta^{\mathcal{C}}_{\mathcal{K}}$). Consider the following matching, induced by $\mathcal{I}$:

$$\begin{aligned}
M_{\mathcal{I}} \quad := \quad & \{\{(x,1),(y,-1)\} \mid (x,y) \in \mathcal{D}^+_{\mathcal{K}} \times \mathcal{D}^-_{\mathcal{K}}, f(x\mathrm{R}_x) = y \wedge x = f(y\mathrm{R}_y)\} \\
\cup & \left\{ \{x,(y,1)\} \;\middle|\; \begin{matrix} (x,y) \in \mathcal{D}^{\pm}_{\mathcal{K}} \times \mathcal{D}^{+}_{\mathcal{K}} \\ \mathcal{K} \models \mathsf{S}(\mathsf{x},\mathsf{y}) \end{matrix}, f(x\mathrm{R}_x) = y \wedge x = f(y\mathrm{R}_y) \right\} \\
\cup & \left\{ \{x,(y,-1)\} \;\middle|\; \begin{matrix} (x,y) \in \mathcal{D}^{\pm}_{\mathcal{K}} \times \mathcal{D}^{-}_{\mathcal{K}} \\ \mathcal{K} \models \mathrm{S}^-(\mathsf{x},\mathsf{y}) \end{matrix}, f(x\mathrm{R}_x) = y \wedge x = f(y\mathrm{R}_y) \right\} \\
\cup & \left\{ \{x,y\} \;\middle|\; \begin{matrix} (x,y) \in \mathcal{D}^{\pm}_{\mathcal{K}} \times \mathcal{D}^{\pm}_{\mathcal{K}} \\ \mathcal{K} \models \mathsf{S}(\mathsf{x},\mathsf{y}) \end{matrix}, f(x\mathrm{R}_x) = y \wedge x = f(y\mathrm{R}_y) \right\}
\end{aligned}$$

Being a matching, $|M_{\mathcal{I}}| < m_{\mathcal{A}} + |\mathcal{V}| - k + 1$. Each edge from $M_{\mathcal{I}}$ provides exactly one additional match: either through the pairing of a positive with a negative, or through the pairing of a bidemanding with another demanding given that one match was already present in between. Each non-covered vertex provides one additional match, being $(x, \mathsf{succ}^{\mathcal{I}}_{\mathrm{R_x}}(x))$ for positive demanding uncovered elements, $(\mathsf{succ}^{\mathcal{I}}_{\mathrm{R_y}}(y), y)$ for negative demanding uncovered elements, and at least one of the latter two shapes for bidemanding elements. In addition with ABox matches, all these matches are distinct, hence there are at least $m_{\mathcal{A}} + |M_{\mathcal{I}}| + |\mathcal{V}| - 2|M_{\mathcal{I}}| = m_{\mathcal{A}} + |\mathcal{V}| - M_{\mathcal{I}} > k - 1$ matches in $\mathcal{I}$. That is at least k matches.

($\Rightarrow$). Assume $(\mathcal{G}, m_{\mathcal{A}} + |\mathcal{V}| - k + 1) \in$ MAXIMUM MATCHING. Consider a matching $M \subseteq \mathcal{E}$ with $|M| \geq m_{\mathcal{A}} + |\mathcal{V}| - k + 1$. Consider the enriched ABox $\mathcal{A}_M$:

$$\begin{aligned}
\mathcal{A}_M \quad := \quad & \mathcal{A} && \text{Shape 1} \\
& \cup \{\mathrm{S}(\mathsf{x},\mathsf{y}) \mid (x,y) \in \mathcal{D}^+_{\mathcal{K}} \times \mathcal{D}^-_{\mathcal{K}}, \{(x,1),(y,-1)\} \in M\} && \text{Shape 2} \\
& \cup \left\{ \mathrm{S}(\mathsf{x},\mathsf{y}) \;\middle|\; \begin{matrix} (x,y) \in \mathcal{D}^{\pm}_{\mathcal{K}} \times \mathcal{D}^{+}_{\mathcal{K}} \\ \mathcal{K} \models \mathrm{S}(\mathsf{x},\mathsf{y}) \end{matrix} \{x,(y,1)\} \in M \right\} && \text{Shape } 3^+ \\
& \cup \left\{ \mathrm{S}(\mathsf{x},\mathsf{y}) \;\middle|\; \begin{matrix} (x,y) \in \mathcal{D}^{\pm}_{\mathcal{K}} \times \mathcal{D}^{-}_{\mathcal{K}} \\ \mathcal{K} \models \mathrm{S}(\mathsf{y},\mathsf{x}) \end{matrix}, \{x,(y,-1)\} \in M \right\} && \text{Shape } 3^- \\
& \cup \left\{ \mathrm{S}(\mathsf{x},\mathsf{y}) \;\middle|\; \begin{matrix} (x,y) \in \mathcal{D}^{\pm}_{\mathcal{K}} \times \mathcal{D}^{\pm}_{\mathcal{K}} \\ \mathcal{K} \models \mathrm{S}(\mathsf{y},\mathsf{x}) \end{matrix}, \{x,y\} \in M \right\} && \text{Shape 4} \\
& \cup \{\mathrm{S}(\mathsf{x},\mathsf{x}) \mid x \in \mathcal{V}, x \text{ uncovered by } M\} && \text{Shape 5}
\end{aligned}$$

Notice $\mathcal{K}_M := (\mathcal{T}, \mathcal{A}_M)$ does not admit any demanding individuals. Since there exists at least one bidemanding individual a for $\mathcal{K}$, setting $\mathsf{ch}_{\sigma/\mathcal{K}}(*) := \mathsf{a}$ provides a well-typed choice of elements for both $\mathcal{K}$ and $\mathcal{K}_M$. Applying Theorem 55 provides a model of $\mathcal{K}_M$, hence of $\mathcal{K}$, in which the matches are exactly: $m_\mathcal{A}$ ABox matches (Shape 1), $|M|$ matches from shapes 2, 3^+, 3^-, and 4, and $|\mathcal{V}| - 2|M|$ for uncovered by M elements of $\mathcal{V}$ (Shape 5). Hence a total of exactly: $m_\mathcal{A} + |\mathcal{V}| - |M|$ matches. Recall $|M| \geq m_\mathcal{A} + |\mathcal{V}| - k + 1$, hence that is at most $k - 1$ matches, that is less than k, hence this model is a countermodel for k. □

5.5.3 TC^0 membership in the remaining cases

We now prove that if a DL-Lite$^{\mathcal{H}}_{\mathsf{pos}}$ TBox $\mathcal{T}$ does not admit a non-trivial propagation of S or S$^-$, and does not admit a non-trivial pairing, then answering the role cardinality q_S over $\mathcal{T}$ is in TC^0.

Notice that if $\mathcal{T}$ satisfies $\mathcal{T} \not\models \mathrm{S} \sqsubseteq \mathrm{S}^-$, then for any ABox $\mathcal{A}$, $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ does not admit bidemanding individuals. Indeed, the existence of a bidemanding individual a implies the existence of B and R such that $\mathrm{B}(\mathsf{a}) \in \mathcal{A}$, $\mathcal{T} \models \mathrm{B} \sqsubseteq \exists \mathrm{R}$, $\mathcal{T} \models \mathrm{R} \sqsubseteq \mathrm{S}$ and $\mathcal{T} \models \mathrm{R} \sqsubseteq \mathrm{S}^-$. If B is a concept name, this is non-trivial pairing. If $\mathrm{B} = \exists \mathrm{T}$, then to prevent a non-trivial pair, $\mathcal{T} \models \mathrm{T} \sqsubseteq \mathrm{S}$ and $\mathcal{T} \models \mathrm{T} \sqsubseteq \mathrm{S}^-$, which would prevent a from being bidemanding. In that case, Lemma 39 holds and solves the problem.

Otherwise $\mathcal{T} \models \mathrm{S} \sqsubseteq \mathrm{S}^-$, in which case the only possible demanding individuals are bidemanding individuals (which disallows Shapes 2, 3^+ and 3^- from the proof just above) not touching any pre-existing match as $\mathcal{T} \models \mathrm{S} \sqsubseteq \mathrm{S}^-$ (which also disallows Shape 4). In particular the easiest way to minimize the number of matches is simply by introducing a self-S-loop on each bidemanding individual, and the optimal number of matches is therefore $m_\mathcal{A} + \left|\mathcal{D}^\pm_\mathcal{K}\right|$ in general except if $m_\mathcal{A} = \left|\mathcal{D}^\pm_\mathcal{K}\right| = 0$ and there exists a generated bipotent role R, in which case it is exactly 1. This is easily shown through the following injective mapping, providing at least $\left|\mathcal{D}^\pm_\mathcal{K}\right|$ non-ABox matches in any model $\mathcal{I}$:

$$\begin{array}{rcl} \rho^\pm : \mathcal{D}^\pm_\mathcal{K} & \to & \Delta^\mathcal{I} \times \Delta^\mathcal{I} \\ x & \mapsto & (x, \mathsf{succ}^\mathcal{I}_\mathrm{S}(x)) \end{array}$$

Furthermore, in the exception stated above the single match is found in any model $\mathcal{I}$ by considering where the representative $\mathsf{repr}_\mathcal{K}(\mathrm{R}) = w\mathrm{R} \in \Delta^{\mathcal{C}_\mathcal{K}}$ maps in $\mathcal{I}$ through a homomorphism $f : \mathcal{C}_\mathcal{K} \to \mathcal{I}$. It gives a match $(f(w), f(w\mathrm{R}))$, or alternatively $(f(w\mathrm{R}), f(w))$ as R is bipotent (but if the model $\mathcal{I}$ is optimal enough, these two are the same match!). Notice that again, with slight adaptations of the circuits, this is still easily computable within TC^0, the threshold gates being here essential to count the number of bidemanding individuals.

5.5.4 Towards DL-Lite$^{\mathcal{H}}_{\mathsf{core}}$

We now turn to DL-Lite$^{\mathcal{H}}_{\mathsf{core}}$ and exhibit new situations that are not captured by the preceding complexity classification.

First, we observe that negative concept and role inclusions introduce two new sources of coNP-hardness.

Theorem 56. *Answering the role cardinality query* q_{S} *over the DL-Lite*$^{\mathcal{H}}_{\mathsf{core}}$ *TBox* $\mathcal{T} = \{\mathrm{B} \sqsubseteq \exists\mathrm{U},\ \mathrm{U} \sqsubseteq \mathrm{S},\ \mathrm{C} \sqsubseteq \exists\mathrm{V},\ \mathrm{V} \sqsubseteq \mathrm{S},\ \exists\mathrm{U}^- \sqsubseteq \neg\exists\mathrm{V}^-\ \}$ *is* coNP*-complete.*

Proof. Consider the ABox:

$$\mathcal{A} = \{\mathrm{B}(\mathsf{u}) \mid u \in \mathcal{U}\} \cup \{\mathrm{S}(\mathsf{u}, \mathsf{s}^*) \mid u \in s \in \mathcal{S}\} \cup \{\mathrm{C}(\mathsf{s}) \mid s \in \mathcal{S}\} \cup \{\mathrm{S}(\mathsf{s}, \mathsf{s}^*) \mid s \in \mathcal{S}\}$$

and set $\mathcal{K} = (\mathcal{T}, \mathcal{A})$. Notice there are $|\mathcal{S}| + \sum_{s\in\mathcal{S}} |s|$ ABox matches. We claim that $[|\mathcal{S}| + \sum_{s\in\mathcal{S}} |s| + k + 1, +\infty]$ is a certain answer of q_{S} w.r.t. $\mathcal{K}$ iff $(\mathcal{U}, \mathcal{S}, k) \notin$ Set Cover.

($\Rightarrow$). Assume $(\mathcal{U}, \mathcal{S}, k) \in$ Set Cover. Consider a covering $F \subseteq \mathcal{S}$ of $\mathcal{U}$ with $|F| \leq k$. Consider the interpretation obtained from $\mathcal{K}$ in which we add, for each $u \in s \in F$ the fact $\mathrm{U}(\mathsf{u}, \mathsf{s}^*)$ and $\mathrm{V}(\mathsf{s}, \mathsf{s})$, which provide k additional matches from $\mathrm{S}(\mathsf{s}, \mathsf{s})$. For the remaining $s \in \mathcal{S}$, we can add the fact $\mathrm{V}(\mathsf{s}, \mathsf{s}^*)$, which does not provide an additional match. We obtain a model $\mathcal{I}_F$, with exactly $|\mathcal{S}| + \sum_{s\in\mathcal{S}} |s| + k$ matches, being a countermodel.

($\Leftarrow$). Assume $(\mathcal{U}, \mathcal{S}, k) \notin$ Set Cover. Consider a model $\mathcal{I}$ of $\mathcal{K}$ and a homomorphism $f : \mathcal{C}_{\mathcal{K}} \to \mathcal{I}$. For each $u \in \mathcal{U}$, we associate a subset $\rho(u) = s$ if $f(\mathsf{u}\mathrm{U}) = \mathsf{s}^*$ and $u \in s \in \mathcal{S}$, otherwise set $\rho(u) = s_u$, where s_u is an arbitrary set containing u. The image $\rho(\mathcal{U})$ is a covering of $\mathcal{U}$, hence $|\rho(\mathcal{U})| \geq k + 1$. By definition, for each $s \in \rho(\mathcal{U})$ there exists $u \in \mathcal{S}$ such that: either $f(\mathsf{u}\mathrm{U}) = \mathsf{s}^*$, or $f(\mathsf{u}\mathrm{U}) \neq \hat{\mathsf{s}}^*$ for all $\hat{s}$ such that $u \in \hat{s} \in \mathcal{S}$. In the first case, $(\mathsf{s}, f(\mathsf{s}\mathrm{V}))$ must be a new match as $f(\mathsf{s}\mathrm{V})$ cannot be s^*. In the second case $(\mathsf{u}, f(\mathsf{u}\mathrm{U}))$ is a new match. Therefore there are at least $|\mathcal{S}| + \sum_{s\in\mathcal{S}} |s| + k + 1$ matches in $\mathcal{I}$. □

Theorem 57. *Answering the role cardinality query* q_{S} *over the DL-Lite*$^{\mathcal{H}}_{\mathsf{core}}$ *ontology* $\mathcal{T} = \{\ \mathrm{B} \sqsubseteq \exists\mathrm{U},\ \mathrm{U} \sqsubseteq \mathrm{S},\ \exists\mathrm{U}^- \sqsubseteq \exists\mathrm{V},\ \mathrm{V} \sqsubseteq \mathrm{S}^-,\ \mathrm{V} \sqsubseteq \neg\mathrm{W}\ \}$ *is* coNP*-complete.*

Proof. Consider the ABox:

$$\mathcal{A} = \{\mathrm{B}(\mathsf{u}) \mid u \in \mathcal{U}\} \cup \{\mathrm{S}(\mathsf{u}, \mathsf{s}) \mid u \in s \in \mathcal{S}\} \cup \{\mathrm{W}(\mathsf{s}, \mathsf{u}) \mid u \in s \in \mathcal{S}\}$$

and set $\mathcal{K} = (\mathcal{T}, \mathcal{A})$. Notice there are $\sum_{s\in\mathcal{S}} |s|$ ABox matches. We claim that $[\sum_{s\in\mathcal{S}} |s| + k + 1, +\infty]$ is a certain answer of q_{S} w.r.t. $\mathcal{K}$ iff $(\mathcal{U}, \mathcal{S}, k) \notin$ Set Cover.

($\Rightarrow$). Assume $(\mathcal{U}, \mathcal{S}, k) \in$ SET COVER. Consider a covering $F \subseteq \mathcal{S}$ of $\mathcal{U}$ with $|F| \leq k$. Consider the interpretation obtained from $\mathcal{K}$ in which we add, for each $u \in s \in F$ the fact $\mathrm{U}(\mathsf{u}, \mathsf{s})$ and $\mathrm{V}(\mathsf{s}, \mathsf{s})$, which provide k additional matches from $\mathrm{S}(\mathsf{s}, \mathsf{s})$. We obtain a model $\mathcal{I}_F$, with exactly $\sum_{s \in \mathcal{S}} |s| + k$ matches, being a countermodel.

($\Leftarrow$). Assume $(\mathcal{U}, \mathcal{S}, k) \notin$ SET COVER. Consider a model $\mathcal{I}$ of $\mathcal{K}$ and a homomorphism $f : \mathcal{C}_{\mathcal{K}} \to \mathcal{I}$. For each $u \in \mathcal{U}$, we associate a subset $\rho(u) = s$ if $f(\mathsf{u}\mathrm{U}) = \mathsf{s}$ and $u \in s \in \mathcal{S}$, otherwise set $\rho(u) = s_u$, where s_u is an arbitrary set containing u. The image $\rho(\mathcal{U})$ is a covering of $\mathcal{U}$, hence $|\rho(\mathcal{U})| \geq k + 1$. By definition, for each $s \in \rho(\mathcal{U})$ there exists $u \in \mathcal{S}$ such that: either $f(\mathsf{u}\mathrm{U}) = \mathsf{s}$, or $f(\mathsf{u}\mathrm{U}) \neq \hat{\mathsf{s}}^*$ for all $\hat{s}$ such that $u \in \hat{s} \in \mathcal{S}$. In the first case, $(f(\mathsf{u}\mathrm{UV}), f(\mathsf{u}\mathrm{U}))$ must be a new match as $f(\mathsf{u}\mathrm{UV})$ cannot be any v with $v \in \mathcal{U}$ (roles W prevent it!). In the second case $(\mathsf{u}, f(\mathsf{u}\mathrm{U}))$ is a new match. Therefore there are at least $\sum_{s \in \mathcal{S}} |s| + k + 1$ matches in $\mathcal{I}$. □

Moreover, we further show that L-complete OMQs exist. The next result employs a role cardinality query, but a similar result is further obtained using a concept cardinality query (Theorem 59). For the two L lower bounds, we proceed by reduction from the UNDIRECTED FOREST ACCESSIBILITY (UFA) problem, known to be L-complete [Cook and McKenzie, 1987]. The UFA problem is to decide, given an undirected acyclic graph $(\mathcal{V}, \mathcal{E})$ with two components, a source vertex $s \in \mathcal{V}$ and a target vertex $t \in \mathcal{V}$, whether t is reachable from s.

Theorem 58. *Answering the role cardinality query q_{S} over the DL-Lite$^{\mathcal{H}}_{\mathsf{core}}$ TBox $\mathcal{T} = \{\ \mathrm{B} \sqsubseteq \exists \mathrm{R},\ \mathrm{R} \sqsubseteq \mathrm{S},\ \mathrm{R} \sqsubseteq \neg \mathrm{R}^-\ \}$ is L-complete.*

Proof. We start with L membership. Let us first describe how to compute, given an ABox $\mathcal{A}$, the minimal number of matches of q_{S}. Intuitively, whenever an outgoing $\mathrm{R}(\mathsf{v}, \mathsf{v}')$ is required (by the presence of $\mathrm{B}(\mathsf{v})$) but not already provided in the ABox, one aims at adding $\mathrm{R}(\mathsf{v}, \mathsf{v}')$ in such a way that $\mathrm{S}(\mathsf{v}, \mathsf{v}')$ is already present in the ABox. This is always possible, except for two cases: *(i)* there are no outgoing S from v, or *(ii)* all the $\mathrm{S}(\mathsf{v}, \mathsf{v}')$ are such that $\mathrm{B}(\mathsf{v}')$ holds and $\mathrm{S}(\mathsf{v}', \mathsf{v})$ holds as well.

In case *(i)*, a new atom of the shape $\mathrm{S}(\mathsf{v}, \mathsf{v}')$ has to be added, creating a new match. In the second case, since $\mathrm{R} \sqsubseteq \neg \mathrm{R}^-$, one could create an inconsistency if the choice were to be done in a local fashion. Let us study how to perform optimally these choices.

We call *exit point* an individual v that satisfies one of the three following conditions:

- $\mathrm{B}(\mathsf{v}) \notin \mathcal{A}$;
- $\exists \mathsf{v}'\ \mathrm{R}(\mathsf{v}, \mathsf{v}') \in \mathcal{A}$;

- $\exists \mathsf{v}'\ \mathrm{S}(\mathsf{v},\mathsf{v}') \in \mathcal{A}$ and either $\mathrm{B}(\mathsf{v}') \notin \mathcal{A}$ or $\mathcal{K} \not\models \mathrm{S}(\mathsf{v}',\mathsf{v})$.

Intuitively, an exit point either already satisfies the concept inclusion $\mathrm{B} \sqsubseteq \exists \mathrm{R}$ (the first two conditions) or can satisfy it in a globally optimal way by adding $\mathrm{R}(\mathsf{v},\mathsf{v}')$ (in the third case, if a model minimizing the number of matches contains $\mathrm{R}(\mathsf{v}',\mathsf{v})$ and $\mathrm{S}(\mathsf{v}',\mathsf{v})$, one can get another minimal mode by adding $\mathrm{S}(\mathsf{v}',\mathsf{v}^*)$ and $\mathrm{S}(\mathsf{v}',\mathsf{v}^*)$), where v^* is a fresh element).

Let us thus consider the tradeoff graph of $\mathcal{A}$ having as vertices the individuals of $\mathcal{A}$ and an edge between u and v if it holds that $\mathrm{S}(\mathsf{u},\mathsf{v}), \mathrm{S}(\mathsf{v},\mathsf{u}), \mathrm{B}(\mathsf{u}), \mathrm{B}(\mathsf{v}) \in \mathcal{A}$, and $\mathrm{R}(\mathsf{u},\mathsf{v}), \mathrm{R}(\mathsf{v},\mathsf{u}) \notin \mathcal{A}$. This graph may contain several connected components, which can be of several types:

a. the connected component contains a cycle: there exists a consistent way to add R atoms wherever necessary in such a way that all the new R atoms fold on S atoms present in $\mathcal{A}$;

b. the connected component contains an exit point: similarly, add R atoms wherever necessary in such a way that all the new R atoms fold on S atoms present in $\mathcal{A}$;

c. the connected component is a tree and does not contain an exit point: an atom $\mathrm{R}(\mathsf{v}, x)$ for which $\mathrm{S}(\mathsf{v}, x) \notin \mathcal{A}$ has to be added. v can be chose arbitrarily among the vertices of the connected component, and x can be chosen to be a fresh element.

Thus, the minimal number of matches is the number of pairs (v,v') such that either $\mathrm{R}(\mathsf{v},\mathsf{v}')$ or $\mathrm{S}(\mathsf{v},\mathsf{v}')$ holds, plus the number of connected components of type c. in the previous case distinction. Algorithm 2 computes this minimum number of matches, and compare it to the number provided in input. Let us notice that checking for the existence of a cycle in a connected component can be done by making calls to an oracle for reachability in undirected graphs.

Algorithm 2 runs in logarithmic space, as undirected reachability is decidable in L, and L is low for itself. This proves membership to L.

For the lower bound, let us reduce UFA to our problem. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an undirected acyclic graph with two components and let $s, t \in \mathcal{V}$ be two vertices. Consider the following ABox:

$$\mathcal{A} = \{\mathrm{B}(\mathsf{u}) \mid u \in \mathcal{V}\} \cup \{\mathrm{S}(\mathsf{u},\mathsf{v}) \mid \{u,v\} \in \mathcal{E}\} \cup \{\mathrm{S}(\mathsf{s},\mathsf{v}^*), \mathrm{S}(\mathsf{t},\mathsf{v}^*)\} \cup \{\mathrm{R}(\mathsf{s},\mathsf{v}^*), \mathrm{R}(\mathsf{t},\mathsf{v}^*)\},$$

where v^* is a fresh individual. Note that we have thus made both s and t exit points, and they are the only such individuals. Let us notice that $\mathcal{A}$ is first-order definable from $\mathcal{G}$. We thus focus on the following claim:

$$((\mathcal{V},\mathcal{E}), s, t) \in \mathrm{UFA} \Leftrightarrow [2|\mathcal{E}| + 3, +\infty] \text{ is a certain answer for } q_{\mathrm{S}} \text{ w.r.t. } (\mathcal{T}, \mathcal{A})$$

Data: An ABox $\mathcal{A}$, an integer n
Result: Yes if and only if $[n, +\infty]$ is a certain answer for q_S w.r.t. $(\mathcal{T}, \mathcal{A})$
$m \leftarrow |\{(\mathsf{v}, \mathsf{v}') \mid \mathrm{S}(\mathsf{v}, \mathsf{v}') \in \mathcal{A} \vee \mathrm{R}(\mathsf{v}, \mathsf{v}') \in \mathcal{A}\}|$;
$\mathcal{G} \leftarrow$ tradeoff graph of $\mathcal{A}$
for $i \leftarrow 1$ **to** n **do**
  if *no* $\mathsf{v_j}$ *with* $j < i$ *is reachable from* $\mathsf{v_i}$ *in* $\mathcal{G}$ **then**
    if *no exit point is reachable from* $\mathsf{v_i}$ *in* $\mathcal{G}$ **then**
      if *the connected component of* $\mathsf{v_i}$ *in* $\mathcal{G}$ *does not contain a cycle* **then**
        $m \leftarrow m + 1$;
      end
    end
  end
  return *Yes if* $n \leq m$*, no otherwise*
end

Algorithm 2: An algorithm for checking whether $[n, +\infty]$ is a certain answer for q_S w.r.t. $(\mathcal{T}, \mathcal{A})$

Let us first notice that in any model, there are $2|\mathcal{E}| + 2$ matches of q_S, as there are that many matches from q_S in $\mathcal{A}$.

Let us consider the case where s is not reachable from t. As $\mathcal{G}$ has exactly two connected components, for any vertex v (distinct from both s and t), there exists a unique vertex among $\{s, t\}$ that is reachable from v and a unique S-edge $\mathrm{S}(\mathsf{v}, \mathsf{f}(\mathsf{v}))$ outgoing from v on the shortest path to s or t (depending on which connected component v belongs). Let us consider the interpretation $\mathcal{I} = \mathcal{A} \cup \{\mathrm{R}(\mathsf{v}, \mathsf{f}(\mathsf{v}))| \, v \in \mathcal{V} \setminus \{s, t\}\}$. $\mathcal{I}$ is a model of $\mathcal{T}$: for any v such that $\mathrm{B}(\mathsf{v})$ holds, there is an atom $\mathrm{R}(\mathsf{v}, \mathsf{v}')$. Moreover, if v is on the shortest path from v' to s (resp. to t), then v' cannot be on the shortest path from v to s (resp. to t), hence $\mathrm{R}^{\mathcal{I}} \cap (\mathrm{R}^{-})^{\mathcal{I}} = \emptyset$. $\mathcal{I}$ is thus a model of $\mathcal{A}$ and $\mathcal{T}$ in which there are exactly $2|\mathcal{E}| + 2$ matches of q_S, proving that if $((\mathcal{V}, \mathcal{E}), s, t) \notin$ UFA, then $[2|\mathcal{E}| + 3, +\infty]$ is not a certain answer of q_S w.r.t. $(\mathcal{T}, \mathcal{A})$.

Let us now consider the case where s is reachable from t. We already know that in any model of $\mathcal{A}$ and $\mathcal{T}$, there are $2|\mathcal{E}| + 2$ matches of q_S. We prove there must be another match of q_S. We show that there must be some $\mathrm{R}(\mathsf{v}, \mathsf{v}')$ in any model such that $\mathrm{S}(\mathsf{v}, \mathsf{v}') \notin \mathcal{A}$. Let v be in the connected component that contains neither s nor t. Let us consider a maximal (possibly infinite) sequence $\mathsf{v_1}, \mathsf{v_2}, \ldots, \mathsf{v_n}$ with $\mathsf{v_1} = \mathsf{v}$ and such that for any i, $\mathrm{R}(\mathsf{v_i}, \mathsf{v_{i+1}})$ belongs to $\mathcal{I}$. As there are no cycle in $\mathcal{G}$ and that $\mathrm{R} \sqsubseteq \neg\mathrm{R}^{-}$, there exists i such that $\mathrm{S}(\mathsf{v_i}, \mathsf{v_{i+1}}) \notin \mathcal{A}$, which provides a new match for q_S, which concludes the proof. □

As previously mentioned, a similar statement is obtained in the case of concept cardinality queries.

Theorem 59. *Answering the concept cardinality query* q_{C} *over the DL-Lite*$^{\mathcal{H}}_{\mathsf{core}}$ *TBox* $\mathcal{T} = \{\ \mathrm{B} \sqsubseteq \exists\mathrm{R},\ \exists\mathrm{R}^- \sqsubseteq \mathrm{C},\ \mathrm{R} \sqsubseteq \neg\mathrm{R}^-,\ \mathrm{R} \sqsubseteq \neg\mathrm{T}\ \}$ *is* L*-complete.*

Proof. We start by proving L membership. Let us first notice that the minimum number of matches can only be one of the two following values:

- $n = |\{\mathsf{v} \mid C(\mathsf{v}) \in \mathcal{A} \vee \exists \mathsf{v}'\ R(\mathsf{v}', \mathsf{v}) \in \mathcal{A}\}|$, which is the number of matches in the ABox on which concept inclusions have been applied;
- $n+1$, which can be obtained by introducing a fresh element α, and adding $\mathrm{R}(\mathsf{v}, \alpha)$ for any v in $\mathsf{Ind}(\mathcal{A})$, as well $\mathrm{C}(\alpha)$.

Let us consider a model $\mathcal{I}$ having n matches. Let f be a homomorphism from $\mathcal{C}_{\mathcal{K}}$ to $\mathcal{I}$. Let $\mathsf{v} \in \mathsf{Ind}(\mathcal{A})$ such that $\mathsf{v}\mathrm{R} \in \Delta^{\mathcal{C}}_{\mathcal{K}}$. Then:

- $f(\mathsf{v}\mathrm{R}) \in \mathsf{Ind}(\mathcal{A})$ (otherwise, a new match would be created);
- $\mathrm{C}(f(\mathsf{v}\mathrm{R})) \in \mathcal{A}$ or there is v' s.t. $\mathrm{R}(\mathsf{v}', \mathsf{v}) \in \mathcal{A}$ (otherwise, a new match would be created);
- $\mathrm{T}(\mathsf{v}, f(\mathsf{v}\mathrm{R})) \notin \mathcal{A}$ (otherwise $\mathcal{I}$ would not be a model)
- $\mathrm{R}(f(\mathsf{v}\mathrm{R}), \mathsf{v}) \notin \mathcal{A}$ (otherwise $\mathcal{I}$ would not be a model)
- if $f(\mathsf{v}\mathrm{R})\mathrm{R} \in \Delta^{\mathcal{C}}_{\mathcal{K}}$, then $f(f(\mathsf{v}\mathrm{R})\mathrm{R}) \neq \mathsf{v}$ (otherwise $\mathcal{I}$ would not be a model).

All the conditions except the last one can be checked for each individual separately. We call *exit point* an individual v for which either $\mathsf{v}R \notin \Delta^{\mathcal{C}}_{\mathcal{K}}$ or there exists v' such that by setting $f(\mathsf{v}R) = \mathsf{v}'$, the first four conditions are satisfied, and the fifth one is satisfied by vacuity, *i.e.*, $\mathsf{v}'\mathrm{R} \notin \Delta^{\mathcal{C}}_{\mathcal{K}}$.

Let us define the tradeoff graph $\mathcal{G}$ of $\mathcal{A}$ having as vertices the individuals of $\mathcal{A}$ and an edge $\{\mathsf{v}, \mathsf{v}'\}$ if and only if:

$$\{\mathrm{B}(\mathsf{v}), \mathrm{B}(\mathsf{v}'), \mathrm{C}(\mathsf{v}), \mathrm{C}(\mathsf{v}')\} \in \mathcal{A} \text{ and } \{\mathrm{T}(\mathsf{v}, \mathsf{v}'), \mathrm{T}(\mathsf{v}', \mathsf{v}), \mathrm{R}(\mathsf{v}, \mathsf{v}'), \mathrm{R}(\mathsf{v}', \mathsf{v})\} \cap \mathcal{A} = \emptyset.$$

This is called a tradeoff graph because if $\{\mathsf{v}, \mathsf{v}'\}$ is an edge, then we could either set $f(\mathsf{v}\mathrm{R}) = \mathsf{v}'$ or $f(\mathsf{v}'\mathrm{R}) = \mathsf{v}$ without creating new matches, but not both, as this would violate the negative role inclusion $\mathrm{R} \sqsubseteq \neg\mathrm{R}^-$.

We claim that there exists a model with exactly n matches if and only if in every connected component of G there is either an exit point or a cycle. Indeed, notice that if $\{\mathsf{v}, \mathsf{v}'\}$ is an edge of the tradeoff graph, then adding an atom $\mathrm{R}(\mathsf{v}, \mathsf{v}')$ does not increase the number of matches of q_{C}. If there is an exit point v^* in a connected component, there is a way to add an atom $\mathrm{R}(\mathsf{v}^*, \hat{\mathsf{v}})$ without adding a match and with $\hat{\mathsf{v}}$ not being in the same connected component as v^* (by definition of the tradeoff graph). Then, by a breadth first traversal of the connected component, one can add R atoms as required. Similarly, when there is a cycle, one starts by such a cycle, and add other atoms in a breadth first fashion.

Conversely, if there exists a model with n matches, then $f(\mathsf{v}\mathrm{R}) \in \mathsf{Ind}(\mathcal{A})$ for any v such that $\mathsf{a}\mathrm{R}$ is defined. Let $\mathsf{v}_1, \dots, \mathsf{v}_n, \dots$ be a sequence such that $f(\mathsf{v}_i\mathrm{R}) = \mathsf{v}_{i+1}$ whenever $\mathsf{v}_i\mathrm{R} \in \Delta_{\mathcal{K}}^{\mathcal{C}}$, and such that v_i is the last element of the sequence otherwise. If $f(\mathsf{v}_i\mathrm{R})$ is not an exit point, then there is an edge $\{\mathsf{v}_i, \mathsf{v}_{i+1}\}$ in the tradeoff graph. If the sequence if finite, then the one before the last is an exit point. Otherwise, there must be a cycle in the connected component containing v_1.

Algorithm 3 checks this condition. As it amounts to several reachability checks in an undirected graph, this algorithm can be made to run in L.

Data: An ABox $\mathcal{A}$, an integer n
Result: Yes if and only if $[n, +\infty]$ is a certain answer of q_{C} w.r.t. $(\mathcal{T}, \mathcal{A})$
$m \leftarrow |\{(\mathsf{v}) \mid \mathrm{R}(\mathsf{v}', \mathsf{v}) \in \mathcal{A} \vee \mathrm{C}(\mathsf{v}) \in \mathcal{A}\}|$;
$r \leftarrow m$;
$\mathcal{G} \leftarrow$ tradeoff graph of $\mathcal{A}$ **for** $i \leftarrow 1$ **to** n **do**
  if *no* v_j *with* $j < i$ *is reachable from* v_i *in* $\mathcal{G}$ **then**
    if *no exit point is reachable from* v_i *in* $\mathcal{G}$ **then**
      if *the connected component of* v_i *in* $\mathcal{G}$ *does not contain a cycle* **then**
        $r \leftarrow m + 1$; `// r can take only two values`
      end
    end
  end
  return *Yes if* $n \leq r$, *no otherwise*
end

Algorithm 3: An algorithm for checking whether $[n, +\infty]$ is a certain answer of q_{C} w.r.t $(\mathcal{T}, \mathcal{A})$

To prove L-hardness, we again proceed by reduction from UFA. Consider the following ABox:

$$\begin{aligned}\mathcal{A} \;=\; & \{\mathrm{B}(\mathsf{u}), \mathrm{C}(\mathsf{u}) \mid u \in \mathcal{V}\} \\ & \cup\{\mathrm{T}(\mathsf{u}, \mathsf{v}) \mid \{u, v\} \notin \mathcal{E}\} \\ & \cup\{\mathrm{T}(\mathsf{u}, \mathsf{v}^*), \mid u \in \mathcal{V} \setminus \{s, t\}\} \\ & \cup\{\mathrm{R}(\mathsf{s}, \mathsf{v}^*), \mathrm{R}(\mathsf{t}, \mathsf{v}^*)\},\end{aligned}$$

There are $|\mathcal{V}| + 1$ matches of q_{C} in $\mathcal{A}$. We prove that:

$$((\mathcal{V}, \mathcal{E}), s, t) \in \mathrm{UFA} \Longleftrightarrow [|\mathcal{V}| + 2, +\infty] \text{ is a certain answer of } q_{\mathrm{C}} \text{ w.r.t. } (\mathcal{T}, \mathcal{A}).$$

Let us consider the case where s is not reachable from t. As $(\mathcal{V}, \mathcal{E})$ has exactly two connected components, for any vertex v (distinct from both s and t), there exists a unique vertex among $\{s, t\}$ that is reachable from v and a unique vertex $f(v)$ that is on the shortest path from v to s (or t). Let us consider the interpretation $\mathcal{I} = \mathcal{A} \cup \{\mathrm{R}(\mathsf{v}, \mathsf{f}(\mathsf{v})) \mid v \in \mathcal{V} \setminus \{s, t\}\}$. $\mathcal{I}$ is a model of $\mathcal{T}$: for any v such that $\mathrm{B}(\mathsf{v})$ holds, there is an atom $\mathrm{R}(\mathsf{v}, \mathsf{v}')$. Moreover, if v is on the shortest path from v' to s, then v' cannot be on the shortest path from v to s, hence $\mathrm{R}^{\mathcal{I}} \cap (\mathrm{R}^{-})^{\mathcal{I}} = \emptyset$. Moreover, $\{v, f(v)\} \in \mathcal{E}$, hence $(v, f(v)) \notin \mathrm{T}^{\mathcal{I}}$. $\mathcal{I}$ is thus a model of $\mathcal{A}$ and $\mathcal{T}$ in which there are exactly $|\mathcal{V}| + 1$ matches of q_{C}, proving that if $((\mathcal{V}, \mathcal{E}), s, t) \notin \mathrm{UFA}$, then $[|\mathcal{V}| + 2, +\infty] \notin q_{\mathrm{C}}^{(\mathcal{T}, \mathcal{A})}$.

Let us now consider the case where s is reachable from t. We already know that in any model of $\mathcal{A}$ and $\mathcal{T}$, there are at least $|\mathcal{V}| + 1$ matches of q_{C}. As there are no cycle in the connected component not containing s and t, in any model of $(\mathcal{A}, \mathcal{T})$ there must be an individual v having an outgoing edge $\mathrm{R}(\mathsf{v}, \mathsf{v}')$ with $\{\mathsf{v}, \mathsf{v}'\} \notin \mathcal{E}$. As $\mathrm{T}(\mathsf{v}, \mathsf{u})$ holds for any u such that $\{\mathsf{v}, \mathsf{u}\} \notin \mathcal{E}$, as well as for $\mathsf{u} = \mathsf{v}^*$, v' provides a novel match for q_{C}, concluding the proof. □

Our results imply that, under standard complexity-theoretic assumptions, at least four different complexities are possible for cardinality queries coupled with DL-Lite$^{\mathcal{H}}_{\mathsf{core}}$ ontologies.

6 Conclusion

Summary of the contributions

We explored the complexity of answering *counting conjunctive queries* over $\mathcal{ALCHI}$ ontologies, as part of the more general ontology-mediated query answering framework. This problem is structured around three main components: some *data* representing ground facts, an *ontology* representing domain knowledge, and a *query*, typically taking the form of an existentially quantified conjunction of atoms. A *knowledge base* is the combination of the data and the ontology, and a *model* of this combination is a way to extend the data so that the extension satisfies all the requirements from the ontology. In a given model, we are interested in how many ways we can satisfy the query, that is the number of so-called *(counting) matches*, which provides an answer to our query that might vary from model to model. The semantics we defined for CCQ answering over KB asks for bounds on these numbers when considering every possible model of the KB of interest, and calls such bounds *certain answers*.

This framework generalizes existing semantics for counting queries in OMQA, and subsumes the classical problem of conjunctive query answering. On the description logics side, we investigated ontologies expressed in $\mathcal{ALCHI}$ and in its sublogics, significantly extending the scope of previous explorations of counting queries in OMQA that were limited to fragments of the DL-Lite family. For such ontologies, we have seen that only the lower bound in a certain answer is non-trivial, and hence focused on deciding whether an input integer m is such that the query is satisfied at least m times in every model of the KB. We measured the complexity of this decision problem with respect to the standard *combined complexity*, considering everything as part of the input, but also with respect to *data complexity* for which the ontology and the CCQ are fixed.

Our main contribution is a complete landscape of the complexity of CCQ answering over $\mathcal{ALCHI}$ KBs, notably closing the cases left open in the literature. For the general case, we proved that the problem is 2EXP-complete for most sublogics of $\mathcal{ALCHI}$, but that it drops to coNEXP-complete for DL-Lite$_{\mathsf{core}}$. In term of data complexity, we showed the problem is coNP-complete for all considered DLs. The developed techniques rely on careful manipulations of models of interest, that both preserve the number of matches for the CCQ of interest and unfold the inner regularities of the model. Our constructions proved themselves robust as they also allowed us to close an open question in the related setting of OMQA with *closed predicates*, in which some designated predicates are interpreted under the closed-world assumption. We exhibited a coNEXP procedure to decide whether a DL-Lite$_{\mathsf{core}}$ KB with closed predicates is satisfiable, matching an existing coNEXP lower bound.

In an effort to identify subcases with better complexity, we first considered the impact of restricting to *rooted* CCQs. Rootedness is indeed a syntactic restriction that has been shown to lower the complexity of several OMQA settings. It turned out however that the most straightforward adaptation of this restriction to CCQs does not lead to better computational properties. This motivated us to focus on the more restricted, yet still natural, class of *exhaustive* rooted CCQs. For this latter class, we used variations of the constructions developed in the general case to obtain four different improvements depending on the considered DL, ranging from PP-completeness to coNEXP-completeness. Interestingly, the coNEXP-hardness result strongly relies on the presence of inverse roles in the ontologies, a feature that is already known to increase the complexity of answering rooted (plain) conjunctive queries. In terms of data complexity, we exhibited *tractable cases* for DL-Lite$_{\mathsf{core}}$ ontologies. This positive result relies upon showing that the canonical model minimizes the number of matches for any exhaustive rooted CCQ.

We continued our hunt for well-behaved subcases of our problem by considering another restriction on the query language, unrelated to rootedness, namely, atomicity. The class of CCQs consisting of a single atom, which we termed *cardinality queries*, comes in two flavors depending on whether this atom concerns a unary or a binary predicate. Several connections with the semantics of closed predicates naturally were exhibited and exploited to determine the combined complexity of cardinality query answering. We proved that while this problem is coNP-complete for the considered dialects of the DL-Lite family, it remains EXP-complete for $\mathcal{EL}$ and several of its extensions. When the ontologies are expressive enough to enforce that all of their models are exponentially large, then the complexity rises to coNEXP-completeness, which is surprisingly high for what appears to be a very simple setting. However, the situation is more favorable if we consider data complexity, as we were able to identify tractable cases for ontologies formulated in the DL-Lite family. Quite interestingly, these tractability results do not rely on the existence of an optimal canonical model, but rather on the existence of a family of models among which an optimal model can always be found. Finally, we refined our study of data complexity to the level the ontology-mediated queries. More precisely, we managed to fully characterize the complexity of answering OMQs consisting of a cardinality query paired with a DL-Lite$^{\mathcal{H}}_{\mathsf{pos}}$ ontology, and we provided simple criteria to distinguish between the three possible complexities.

Perspectives

Going forward, the most natural challenge is to develop practical algorithms for the tractable cases in data complexity. Rewriting techniques have already been explored for the case of exhaustive rooted CCQs over DL-Lite$_{\mathsf{core}}$ ontologies [Calvanese et al., 2020a,c], but not for the well-behaved classes of OMQs based upon cardinality queries that we identified. For these latter cases, our results give a rather precise insight into the underlying coNP-complete problem, with respect to combined complexity, and an implementation relying on Sat solvers to handle this part seems possible. More generally, despite the variety of coNP-complete situations we obtained, most of the procedures we describe are not easily reducible to Sat, and it would be desirable to develop more refined coNP procedures to this end, notably in the case of $\mathcal{EL}$ ontologies, for which no tractable class of CCQs is known yet. Advances in these direction could lead to efficient counting query rewriting algorithms, typically rewriting the CCQ of interest to a SQL query, based on existing ontology reasoners such that Ontop[1]. Once rewritten, the resulting query could further be evaluated over usual relational databases and provide more complete answers by taking the knowledge from the ontology into account.

A more theoretical challenge would be to extend our techniques and results to more expressive ontologies, and in particular to DLs involving more counting-oriented features. A first step in this direction would be to consider functionality axioms, for example, with DL-Lite$_{\mathcal{F}}$ ontologies, before moving to even more general forms of cardinality constraints, e.g. $\mathcal{ALCQ}$. We also drew several connections in this thesis with OMQA with closed predicates, but we haven't yet considered the problem of answering CCQs over KBs equipped with such closed predicates, which are known to already increase the complexity of answering usual conjunctive queries [Ngo et al., 2016]. Interestingly, the upper bound in the presented notion of certain answers is no longer trivial in the presence of the latter settings and hence becomes a relevant question. We believe that if we want to determine such upper bounds in the case of cardinality queries, then we may be able to take advantage of results and techniques from recent work on bounded predicates [Lukumbuzya and Šimkus, 2021].

[1] `https://ontop-vkg.org/`

As the techniques we developed to answer CCQs over $\mathcal{ALCHI}$ KBs are inspired by techniques from the realm of existential rules, it would be relevant to study to what extent our results can transfer to this setting. Indeed, our semantics of counting queries allows to take into account anonymous elements induced by the ontology. Those elements being of particular interest with existential rules, the semantics of CCQs naturally makes sense in this latter setting and it remains to understand the complexity of the associated problem. One could for example start with the linear or frontier-1 fragments of existential rules, to tackle the higher arity of predicates in controlled cases, before moving to less restricted rules.

More generally, our techniques to obtain models minimizing the number of matches of a query may appear helpful in other settings in which a minimization is required. We have already seen that interlacings easily adapt to query answering over closed predicates, but one could also turn to other forms of reasoning such as circumscription. Circumscription is a non-monotonic logic framework introduced in the 80's, which aims to capture the common sense assumption that things behave in an expected way unless there are specified reasons to think otherwise [McCarthy, 1980]. It has already been studied as an extension of various description logics such as $\mathcal{ALC}$, for which circumscription restricts the notion of models to those minimizing a given set of predicates [Bonatti et al., 2006].

It would also be relevant to explore variations of the considered problems. We have partially investigated the optimal variant of our decision problem, asking whether the input integer provides the tightest certain answer, but many questions remain open. One can also focus on the functional variant of this problem, asking for this tightest answer (or an approximation of this value) to be given as output. Indeed, if computing good approximations of this minimum number of matches were shown to be tractable, it would provide a nice counterpart to the high complexities obtained within this thesis. Exhibited connections with known hard to approximate problems tend however in the opposite direction. Regarding variations of the problem, the question of whether a unary encoding of the input integer lowers the complexity also remains open for several cases, notably for the combined complexity of exhaustive rooted CCQ answering in any of the investigated DLs.

Finally, counting queries are a special case of *aggregate queries* which use numeric operators (e.g. sum, max, average). Despite being widely used for data analysis, aggregate queries have been little explored in the OMQA setting. The case of the counting function is arguably different in nature from the other aggregate functions, as it is independent from the type of the counted elements. However, we believe the proposed semantics could serve as a relevant starting point for exploring other aggregate functions in OMQA, provided that the considered ontologies are equipped with some datatype features. Identifying ontology languages with such features is arguably already challenging. Indeed, attempts to equip Datalog, which does not involve anonymous elements, with some arithmetic operations on integers, *e.g.* with its extension $\text{Datalog}_{\mathbb{Z}}$, easily lead to undecidability. Similarly, extending the well-known DL $\mathcal{ALC}$ with such features makes basic reasoning tasks such as satisfiability and subsumption undecidable [Baader and Sattler, 2003]. Some solutions may be found in recent developments identifying fragments of $\text{Datalog}_{\mathbb{Z}}$ that regain decidability [Cuenca Grau et al., 2020; Kaminski et al., 2021], or in attempts to introduce aggregate in Answer Set Programming (ASP) [Faber et al., 2011; Ferraris, 2011]. The situation also seems more favorable when equipping lightweight description logics with aggregate features, for example as proposed in Artale et al. [2012], in Savkovic and Calvanese [2012] or in Hernich et al. [2017], which extend DL-Lite in this direction and identify cases in which conjunctive query answering enjoys tractable data complexity.

Bibliography

Aehlig, K., Cook, S. A., and Nguyen, P. (2007). Relativizing small complexity classes and their theories. In *Proceedings of the 21st International Workshop on Computer Science Logic (CSL)*, pages 374–388.

Afrati, F. and Kolaitis, P. (2008). Answering aggregate queries in data exchange. In *Proceedings of the 27th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS)*, pages 129–138.

Arenas, M., Bertossi, L., Chomicki, J., He, X., Raghavan, V., and Spinrad, J. (2003). Scalar aggregation in inconsistent databases. *Theoretical Computer Science (TCS)*, 296(3):405–434.

Artale, A., Calvanese, D., Kontchakov, R., and Zakharyaschev, M. (2009). The DL-Lite family and relations. *Journal of Artificial Intelligence Research (JAIR)*, 36(1):1–69.

Artale, A., Ryzhikov, V., and Kontchakov, R. (2012). DL-Lite with attributes and datatypes. In *Proceedings of the 20th European Conference on Artificial Intelligence (ECAI)*, pages 61–66.

Baader, F., Brandt, S., and Lutz, C. (2005). Pushing the $\mathcal{EL}$ envelope. In *Proceedings of the 19th International Joint Conference on Artificial intelligence (IJCAI)*, pages 364–369.

Baader, F., Calvanese, D., McGuinness, D. L., Nardi, D., and Patel-Schneider, P. F. (2003). *The Description Logic Handbook: Theory, Implementation and Applications.* Cambridge University Press.

Baader, F., Horrocks, I., Lutz, C., and Sattler, U. (2017). *An Introduction to Description Logic*. Cambridge University Press.

Baader, F., Küsters, R., and Molitor, R. (1999). Computing least common subsumers in description logics with existential restrictions. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 96–103.

Baader, F., Lutz, C., and Brandt, S. (2008). Pushing the $\mathcal{EL}$ envelope further. In *Proceedings of the 5th Workshop on OWL: Experiences and Directions (OWLED)*.

Baader, F. and Sattler, U. (2003). Description logics with aggregates and concrete domains. *Journal of Information Systems*, 28(8):979–1004.

Bailey, D. D., Dalmau, V., and Kolaitis, P. G. (2007). Phase transitions of PP-complete satisfiability problems. *Discrete Applied Mathematics*, 155(12):1627–1639.

Berkholz, C., Gerhardt, F., and Schweikardt, N. (2020). Constant delay enumeration for conjunctive queries: A tutorial. *ACM SIGLOG News*, 7(1):4–33.

Bienvenu, M., Calvanese, D., Ortiz, M., and Šimkus, M. (2014a). Nested regular path queries in description logics. In *Proceedings of the 14th International Conference on Principles of Knowledge Representation and Reasoning (KR)*, pages 218–227.

Bienvenu, M., Cate, B. T., Lutz, C., and Wolter, F. (2014b). Ontology-based data access: A study through disjunctive datalog, CSP, and MMSNP. *ACM Transactions on Database Systems (TODS)*, 39(4):1–44.

Bienvenu, M., Manière, Q., and Thomazo, M. (2020). Answering counting queries over DL-Lite ontologies. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1608–1614.

Bienvenu, M., Manière, Q., and Thomazo, M. (2021a). Cardinality queries over DL-Lite ontologies. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1801–1807.

Bienvenu, M., Manière, Q., and Thomazo, M. (2021b). Counting queries over $\mathcal{ELHI}_\bot$ ontologies. In *Proceedings of the 34th International Workshop on Description Logics (DL)*.

Bienvenu, M., Manière, Q., and Thomazo, M. (2022). Counting queries over $\mathcal{ALCHI}$ ontologies. In *Proceedings of the 19th International Conference on Principles of Knowledge Representation and Reasoning (KR)*, pages 53–62.

Bienvenu, M. and Ortiz, M. (2015). Ontology-mediated query answering with data-tractable description logics. In *Tutorial Lectures of the 11th Reasoning Web International Summer School (RW)*, pages 218–307.

Bienvenu, M., Ortiz, M., and Šimkus, M. (2015). Regular path queries in lightweight description logics: Complexity and algorithms. *Journal of Artificial Intelligence Research (JAIR)*, 53:315–374.

Bienvenu, M., Ortiz, M., Šimkus, M., and Xiao, G. (2013). Tractable queries for lightweight description logics. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 768–774.

Bonatti, P. A., Lutz, C., and Wolter, F. (2006). Description logics with circumscription. In *Proceedings of the 10th International Conference on Principles of Knowledge Representation and Reasoning (KR)*, pages 400–410.

Brachman, R. J. and Schmolze, J. G. (1985). An overview of the KL-ONE knowledge representation system. *Journal of Cognitive Science (CSJ)*, 9(2):171–216.

Brewka, G., Eiter, T., and Truszczyński, M. (2011). Answer set programming at a glance. *Communications of the ACM*, 54(12):92–103.

Cabalar, P., Fandinno, J., Schaub, T., and Wanko, P. (2020). An ASP semantics for constraints involving conditional aggregates. In *Proceedings of the 24th European Conference on Artificial Intelligence (ECAI)*, pages 664–671.

Calvanese, D., Corman, J., Lanti, D., and Razniewski, S. (2020a). Counting query answers over a DL-Lite knowledge base. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1658–1666.

Calvanese, D., Corman, J., Lanti, D., and Razniewski, S. (2020b). Counting query answers over a DL-Lite knowledge base (extended version). *arXiv:2005.05886v3*.

Calvanese, D., Corman, J., Lanti, D., and Razniewski, S. (2020c). Rewriting count queries over DL-Lite TBoxes with number restrictions. In *Proceedings of the 33rd International Workshop on Description Logics (DL)*.

Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., and Rosati, R. (2005). Dl-lite: Tractable description logics for ontologies. In *Proceedings of the 20th National Conference on Artificial Intelligence (AAAI)*, pages 602–607.

Calvanese, D., De Giacomo, G., and Lenzerini, M. (1998). On the decidability of query containment under constraints. In *Proceedings of the 17th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS)*, pages 149–158.

Calvanese, D., Eiter, T., and Ortiz, M. (2007a). Answering regular path queries in expressive description logics: An automata-theoretic approach. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence (AAAI)*, volume 7, pages 391–396.

Calvanese, D., Giacomo, G. D., Lembo, D., Lenzerini, M., and Rosati, R. (2006). Data complexity of query answering in description logics. In *Proceedings of the 10th International Conference on Principles of Knowledge Representation and Reasoning (KR)*, pages 260–270.

Calvanese, D., Giacomo, G. D., Lembo, D., Lenzerini, M., and Rosati, R. (2007b). Tractable reasoning and efficient query answering in description logics: The DL-Lite family. *Journal of Automated Reasoning (JAR)*, 39(3):385–429.

Calvanese, D., Kharlamov, E., Nutt, W., and Thorne, C. (2008). Aggregate queries over ontologies. In *Proceedings of the 2nd International Workshop on Ontologies and Information Systems for the Semantic Web (ONISW)*, pages 97–104.

Carral, D., Dragoste, I., and Krötzsch, M. (2018). The combined approach to query answering in Horn-$\mathcal{ALCHOIQ}$. In *Proceedings of the 16th International Conference on Principles of Knowledge Representation and Reasoning (KR)*, pages 339–348.

Ceri, S., Gottlob, G., and Tanca, L. (1990). *Logic Programming and Databases*. Springer-Verlag.

Chandra, A. K., Stockmeyer, L. J., and Vishkin, U. (1984). Constant depth reducibility. *SIAM Journal on Computing*, 13(2):423–439.

Consens, M. P. and Mendelzon, A. O. (1993). Low-complexity aggregation in GraphLog and Datalog. *Theoretical Computer Science (TCS)*, 116(1):95–116.

Cook, S. A. and McKenzie, P. (1987). Problems complete for deterministic logarithmic space. *Journal of Algorithms*, 8(3):385–394.

Cuenca Grau, B., Horrocks, I., Kaminski, M., Kostylev, E. V., and Motik, B. (2020). Limit datalog: A declarative query language for data analysis. *ACM SIGMOD Record*, 48(4):6–17.

Dell'Armi, T., Faber, W., Ielpa, G., Leone, N., and Pfeifer, G. (2003). Aggregate functions in DLV. In *Proceedings of the 2nd International Workshop on Answer Set Programming, Advances in Theory and Implementation (ASP)*.

Deutsch, A., Francis, N., Green, A., Hare, K., Li, B., Libkin, L., Lindaaker, T., Marsault, V., Martens, W., Michels, J., Murlak, F., Plantikow, S., Selmer, P., Voigt, H., van Rest, O., Vrgoč, D., Wu, M., and Zemke, F. (2021). Graph pattern matching in GQL and SQL/PGQ. *arXiv:2112.06217*.

Donini, F. M. and Massacci, F. (2000). ExpTime tableaux for $\mathcal{ALC}$. *Journal of Artificial Intelligence (AIJ)*, 124(1):87–138.

Edmonds, J. (1965). Paths, trees and flowers. *Canadian Journal of Mathematics*, 17:449–467.

Eiter, T., Gottlob, G., Ortiz, M., and Šimkus, M. (2008). Query answering in the description logic Horn-$\mathcal{SHIQ}$. In *Proceedings of the 11th European Conference on Logics in Artificial Intelligence (JELIA)*, pages 166–179.

Eiter, T., Lutz, C., Ortiz, M., and Šimkus, M. (2009). Query answering in description logics: the knots approach. In *Proceedings of the 16th International Workshop on Logic, Language, Information and Computation (WoLLIC)*, pages 26–36.

Eiter, T., Ortiz, M., Simkus, M., Tran, T., and Xiao, G. (2012a). Query rewriting for Horn-$\mathcal{SHIQ}$ plus rules. In Hoffmann, J. and Selman, B., editors, *Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI)*.

Eiter, T., Ortiz, M., and Šimkus, M. (2012b). Conjunctive query answering in the description logic $\mathcal{SH}$ using knots. *Journal of Computer and System Sciences (JCSS)*, 78(1):47–85.

Faber, W., Pfeifer, G., and Leone, N. (2011). Semantics and complexity of recursive aggregates in answer set programming. *Journal of Artificial Intelligence (AIJ)*, 175(1):278–298.

Feier, C., Lutz, C., and Przybylko, M. (2021). Answer counting under guarded TGDs. In *Proceedings of the 24th International Conference on Database Theory (ICDT)*, pages 11:1–11:22.

Ferraris, P. (2011). Logic programs with propositional connectives and aggregates. *ACM Transactions on Computational Logic (TOCL)*, 12(4):1–40.

Franconi, E., Ibáñez-García, Y. A., and Seylan, I. (2011). Query answering with DBoxes is hard. In *Proceedings of the 7th Workshop on Methods for Modalities (M4M)*, volume 278, pages 71–84.

Garey, M., Johnson, D., and Stockmeyer, L. (1976). Some simplified NP-complete graph problems. *Journal of Theoretical Computer Science (TCS)*, 1(3):237–267.

Gelfond, M. and Lifschitz, V. (1991). Classical negation in logic programs and disjunctive databases. *New Generation Computing*, 9:365–385.

Glimm, B., Horrocks, I., Lutz, C., and Sattler, U. (2008). Conjunctive query answering for the description logic $\mathcal{SHIQ}$. *Journal of Artificial Intelligence Research (JAIR)*, 31(1):157–204.

Gonthier, G. (2008). Formal proof – The four-color theorem. *Notices of the American Mathematical Society*, 55(11):1382–1393.

Grädel, E., Kolaitis, P. G., and Vardi, M. Y. (1997). On the decision problem for two-variable first-order logic. *Bulletin of Symbolic Logic*, 3:3–53.

Gutiérrez-Basulto, V., Ibáñez-García, Y. A., Kontchakov, R., and Kostylev, E. V. (2015). Queries with negation and inequalities over lightweight ontologies. *Journal of Web Semantics (JWS)*, 35:184–202.

Gutiérrez-Basulto, V., Ibáñez-García, Y., and Kontchakov, R. (2012). An update on query answering with restricted forms of negation. In *Proceedings of the 6th International Conference on Web Reasoning and Rule Systems (RR)*, pages 75–89.

Hernich, A., Lemos, J., and Wolter, F. (2017). Query answering in DL-Lite with datatypes: A non-uniform approach. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI)*, pages 1142–1148.

Hitzler, P., Krötzsch, M., and Rudolph, S. (2009). *Foundations of Semantic Web Technologies*. Chapman & Hall/CRC.

Horrocks, I., Kutz, O., and Sattler, U. (2006). The even more irresistible $\mathcal{SROIQ}$. In *Proceedings of the 10th International Conference on Principles of Knowledge Representation and Reasoning (KR)*, pages 57–67.

Horrocks, I., Patel-Schneider, P. F., and Harmelen, F. V. (2003). From $\mathcal{SHIQ}$ and RDF to OWL: The making of a web ontology language. *Journal of Web Semantics (JWS)*, 1(1):7–26.

Horrocks, I. and Tessaris, S. (2000). A conjunctive query language for description logic aboxes. In *Proceedings of the 17th National Conference on Artificial Intelligence (AAAI)*, pages 399–404.

Hustadt, U., Motik, B., and Sattler, U. (2005). Data complexity of reasoning in very expressive description logics. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 466–471.

Hustadt, U., Motik, B., and Sattler, U. (2007). Reasoning in description logics by a reduction to disjunctive datalog. *Journal of Automated Reasoning (JAR)*, 39:351–384.

Immerman, N. (1986). Relational queries computable in polynomial time. *Information and Control*, 68(1):86–104.

Immerman, N. (1999). *Descriptive Complexity.* Springer Graduate Texts in Computer Science.

Kaminski, M., Kostylev, E. V., and Cuenca Grau, B. (2016). Semantics and expressive power of subqueries and aggregates in SPARQL 1.1. In *Proceedings of the 25th International Conference on World Wide Web (WWW)*, pages 227–238.

Kaminski, M., Kostylev, E. V., Cuenca Grau, B., Motik, B., and Horrocks, I. (2021). The complexity and expressive power of limit datalog. *Journal of the ACM*, 69(1):1–83.

Kikot, S., Kontchakov, R., and Zakharyaschev, M. (2012). Conjunctive query answering with OWL 2 QL. In *Proceedings of the 13th International Conference on Principles of Knowledge Representation and Reasoning (KR)*, pages 275–285.

Klug, A. (1982). Equivalence of relational algebra and relational calculus query languages having aggregate functions. *Journal of the ACM*, 29(3):699–717.

Kontchakov, R., Lutz, C., Toman, D., Wolter, F., and Zakharyaschev, M. (2011). The combined approach to ontology-based data access. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2656–2661.

Kostylev, E. V. and Reutter, J. L. (2015). Complexity of answering counting aggregate queries over DL-Lite. *Journal of Web Semantics (JWS)*, 33:94–111.

Krisnadhi, A. and Lutz, C. (2007). Data complexity in the $\mathcal{EL}$ family of description logics. In *Proceedings of the 14th international conference on Logic for programming, artificial intelligence and reasoning (LAPR)*, pages 333–347.

Krötzsch, M. and Rudolph, S. (2007). Conjunctive queries for $\mathcal{EL}$ with composition of roles. In *Proceedings of the 20th International Workshop on Description Logics (DL).*

Krötzsch, M., Rudolph, S., and Hitzler, P. (2013). Complexities of Horn description logics. *ACM Transactions on Computational Logic (TOCL)*, 14(1):1–36.

Lechtenbörger, J., Shu, H., and Vossen, G. (2002). Aggregate queries over conditional tables. *Journal of Intelligent Information Systems (JIIS)*, 19(3):343–362.

Levy, A. Y. and Rousset, M.-C. (1998). Combining Horn rules and description logics in CARIN. *Journal of Artificial Intelligence (AIJ)*, 104(1):165–209.

Libkin, L. (2003). Expressive power of SQL. *Journal of Theoretical Computer Science (TCS)*, 296(3):379–404.

Liu, L. and Truszczynski, M. (2006). Properties and applications of programs with monotone and convex constraints. *Journal of Artificial Intelligence Research (JAIR)*, 27(1):299–334.

Lukumbuzya, S. and Šimkus, M. (2021). Bounded predicates in description logics with counting. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1966–1972.

Lutz, C. (2008). The complexity of conjunctive query answering in expressive description logics. In *Proceedings of the 4th International Joint Conference on Automated Reasoning (IJCAR)*, pages 179–193.

Lutz, C. and Przybylko, M. (2022). Efficiently enumerating answers to ontology-mediated queries. In *Proceedings of the 41st ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems (PODS)*, pages 277–289.

Lutz, C. and Sabellek, L. (2017). Ontology-mediated querying with the description logic $\mathcal{EL}$: Trichotomy and linear Datalog rewritability. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1181–1187.

Lutz, C., Seylan, I., and Wolter, F. (2012). Mixing open and closed world assumption in ontology-based data access: Non-uniform data complexity. In *Proceedings of the 25th International Workshop on Description Logics (DL)*, pages 268–278.

Lutz, C., Seylan, I., and Wolter, F. (2013). Ontology-based data access with closed predicates is inherently intractable (sometimes). In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1024–1030.

Lutz, C., Toman, D., and Wolter, F. (2009). Conjunctive query answering in the description logic $\mathcal{EL}$ using a relational database system. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2070–2075.

Lutz, C. and Wolter, F. (2012). Non-uniform data complexity of query answering in description logics. In *Proceedings of the 13th International Conference on Principles of Knowledge Representation and Reasoning (KR)*.

McCarthy, J. (1980). Circumscription – a form of non-monotonic reasoning. *Journal of Artificial intelligence (AIJ)*, 13(1-2):27–39.

Motik, B. (2006). *Reasoning in description logics using resolution and deductive databases*. PhD thesis, Karlsruhe Institute of Technology, Germany.

Nenov, Y., Piro, R., Motik, B., Horrocks, I., Wu, Z., and Banerjee, J. (2015). RDFox: A highly-scalable RDF store. In *Proceedings of the 14th International Semantic Web Conference (ISWC)*, volume 9367, pages 3–20.

Ngo, N., Ortiz, M., and Šimkus, M. (2016). Closed predicates in description logics: results on combined complexity. In *Proceedings of the 15th International Conference on Principles of Knowledge Representation and Reasoning (KR)*, pages 237–246.

Nikolaou, C., Kostylev, E. V., Konstantinidis, G., Kaminski, M., Cuenca Grau, B., and Horrocks, I. (2019). Foundations of ontology-based data access under bag semantics. *Journal of Artificial Intelligence (AIJ)*, pages 91–132.

Ortiz, M., Calvanese, D., and Eiter, T. (2008). Data complexity of query answering in expressive description logics via tableaux. *Journal of Automated Reasoning (JAR)*, 41(1):61–98.

Ortiz, M., Rudolph, S., and Šimkus, M. (2011). Query answering in the Horn fragments of the description logics $\mathcal{SHOIQ}$ and $\mathcal{SROIQ}$. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI)*.

Papadimitriou, C. H. and Yannakakis, M. (1986). A note on succinct representations of graphs. *Information and Control*, 71(3):181–185.

Poggi, A., Lembo, D., Calvanese, D., De Giacomo, G., Lenzerini, M., and Rosati, R. (2008). Linking data to ontologies. *Journal on Data Semantics (JoDS)*, 10:133–173.

Rabin, M. O. and Vazirani, V. V. (1989). Maximum matchings in general graphs through randomization. *Journal of Algorithms*, 10(4):557–567.

Rosati, R. (2007). On conjunctive query answering in $\mathcal{EL}$. In *Proceedings of the 20th International Workshop on Description Logics (DL)*.

Rudolph, S., Krötzsch, M., and Hitzler, P. (2012). Type-elimination-based reasoning for the description logic $\mathcal{SHIQ}b_s$ using decision diagrams and disjunctive datalog. *Logical Methods in Computer Science (LMCS)*, 8(1).

Savitch, W. J. (1970). Relationships between nondeterministic and deterministic tape complexities. *Journal of Computer and System Sciences (JCSS)*, 4(2):177–192.

Savkovic, O. and Calvanese, D. (2012). Introducing datatypes in DL-Lite. In *Proceedings of the 20th European Conference on Artificial Intelligence (ECAI)*, pages 720–725.

Schild, K. (1991). A correspondence theory for terminological logics: Preliminary report. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 466–471.

Sipser, M. (1996). *Introduction to the Theory of Computation*. PWS Publishing Company.

Spackman, K. (2000). Managing clinical terminology hierarchies using algorithmic calculation of subsumption: Experience with SNOMED-RT. *Journal of the American Medical Informatics Association*.

Stefanoni, G., Motik, B., Krötzsch, M., and Rudolph, S. (2014). The complexity of answering conjunctive and navigational queries over OWL 2 EL knowledge bases. *Journal of Artificial Intelligence Research (JAIR)*, 51(1):645–705.

Thomazo, M. (2013). *Conjunctive query answering under existential rules – Decidability, complexity, and algorithms*. PhD thesis, Montpellier 2 University, France.

Thomazo, M., Baget, J., Mugnier, M., and Rudolph, S. (2012). A generic querying algorithm for greedy sets of existential rules. In *Proceedings of the 13th International Conference on Principles of Knowledge Representation and Reasoning (KR)*, pages 96–106.

Tobies, S. (2001). *Complexity results and practical algorithms for logics in knowledge representation*. PhD thesis, RWTH Aachen University, Germany.

Ullman, J. D. (1988). *Principles of Database and Knowledge-Base Systems, Vol. I*. Computer Science Press, Inc.

Vardi, M. Y. (1982). The complexity of relational query languages. In *Proceedings of the 14th annual ACM symposium on Theory of computing (STOC)*, pages 137–146.

Vardi, M. Y. (1995). On the complexity of bounded-variable queries. In *Proceedings of the 14th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS)*, pages 266–276.

Xiao, G., Calvanese, D., Kontchakov, R., Lembo, D., Poggi, A., Rosati, R., and Zakharyaschev, M. (2018). Ontology-based data access: a survey. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 5511–5519.

Index

$\mathcal{A}$

Additional proof material

A.1 Proofs for Section 3.3 (Answering CCQs over $\mathcal{ALCHI}$ ontologies)

Lemma 9. *For all* $w\cdot(\mathbb{P},h) \in \mathcal{P}$, $d, e \in \Delta^{\mathfrak{I}^{\mathbb{P}}}$, *and* $\mathrm{P} \in \mathsf{N_R}$*: if* $(\lambda_{w\cdot(\mathbb{P},h)}(d), \lambda_{w\cdot(\mathbb{P},h)}(e)) \in \mathrm{P}^{\mathcal{I}}$*, then* $\mathfrak{I}^{\mathbb{P}}$ *remains* $\mathcal{T}$*-satisfiable if we add* (d,e) *to* $\mathrm{P}^{\mathfrak{I}^{\mathbb{P}}}$*.*

Proof. Let $w_1 \cdot (\mathbb{P}_1, h_1) \in \mathcal{P}$ and $d_1, e_1 \in \Delta^{\mathfrak{I}^{\mathbb{P}_1}}$ two elements. Let $\mathrm{P} \in \mathsf{N_R}$ be a role name. Assume $(\lambda_{w_1\cdot(\mathbb{P}_1,h_1)}(d_1), \lambda_{w_1\cdot(\mathbb{P}_1,h_1)}(e_1)) \in \mathrm{P}^{\mathcal{I}}$. By definition of $\mathrm{P}^{\mathcal{I}}$, there exist $w_2 \cdot (\mathbb{P}_2, h_2) \in \mathcal{P}$ and $(d_2, e_2) \in \mathrm{P}^{\mathfrak{I}^{\mathbb{P}_2}}$ with $\lambda_{w_2\cdot(\mathbb{P}_2,h_2)}(d_2) = \lambda_{w_1\cdot(\mathbb{P}_1,h_1)}(d_1)$ and $\lambda_{w_2\cdot(\mathbb{P}_2,h_2)}(e_2) = \lambda_{w_1\cdot(\mathbb{P}_1,h_1)}(e_1)$. We further refer to these two equalities as $(*_d)$ and $(*_e)$. We distinguish 5 main cases.

1. $(d_1 \in \Delta^{\mathcal{I}^*}$ or $d_2 \in \Delta^{\mathcal{I}^*})$ and $(e_1 \in \Delta^{\mathcal{I}^*}$ or $e_2 \in \Delta^{\mathcal{I}^*})$.
   $(*_d)$ yields $d_1 = d_2$ and $(*_e)$ yields $e_1 = e_2$. Interpretation $\mathfrak{I}^{\mathbb{P}_2}$ preserves $\mathcal{I}^*$, hence $(d_2, e_2) \in \mathrm{P}^{\mathcal{I}^*}$. Interpretation $\mathfrak{I}^{\mathbb{P}_1}$ preserves $\mathcal{I}^*$, hence $(d_1, e_1) \in \mathrm{P}^{\mathfrak{I}^{\mathbb{P}_1}}$. It then suffices to recall that $\mathfrak{I}^{\mathbb{P}_1}$ is $\mathcal{T}$-satisfiable.

   In the remaining cases, we assume that $e_1, e_2 \notin \Delta^{\mathcal{I}^*}$ *or* $d_1, d_2 \notin \Delta^{\mathcal{I}^*}$*, which ensures* $\mathbb{P}_1 \neq \mathbb{P}^*$ *and* $\mathbb{P}_2 \neq \mathbb{P}^*$*. In particular,* $\mathfrak{fr}^{\mathbb{P}_1}$*,* $\mathfrak{gen}^{\mathbb{P}_1}$*,* $\mathfrak{fr}^{\mathbb{P}_2}$ *and* $\mathfrak{gen}^{\mathbb{P}_2}$ *are singletons. Furthermore, the conditions on roles for a non-initial pattern (Condition 4) ensures* $d_2 \neq e_2$ *(recall we assume* $(d_2, e_2) \in \mathrm{P}^{\mathfrak{I}^{\mathbb{P}_2}}$*).*

2. $(d_1 \in \Delta^{\mathcal{I}^*}$ or $d_2 \in \Delta^{\mathcal{I}^*})$ and $(e_1, e_2 \notin \Delta^{\mathcal{I}^*})$.
   $(*_d)$ yields $d_1 = d_2$, we distinguish 4 remaining subcases.

   (a) $e_1 \in \mathfrak{gen}^{\mathbb{P}_1}$ and $e_2 \in \mathfrak{gen}^{\mathbb{P}_2}$.
   We have $\lambda_{w_1\cdot(\mathbb{P}_1,h_1)}(e_1) = w_1 \cdot (\mathbb{P}_1, h_1)$ and $\lambda_{w_2\cdot(\mathbb{P}_2,h_2)}(e_2) = w_2 \cdot (\mathbb{P}_2, h_2)$. Hence $(*_e)$ yields in particular $\mathbb{P}_1 = \mathbb{P}_2$. Recall that $\mathfrak{gen}^{\mathbb{P}_1}$ is a singleton,

so $e_1 = e_2$. Therefore $\mathfrak{I}^{\mathbb{P}_1}$ already contains the fact $\mathrm{P}(d_1, e_1)$. Recalling that $\mathfrak{I}^{\mathbb{P}_1}$ is satisfiable concludes this case.

(b) $e_1 \in \mathfrak{fr}^{\mathbb{P}_1}$ and $e_2 \in \mathfrak{gen}^{\mathbb{P}_2}$.
We have $\lambda_{w_1\cdot(\mathbb{P}_1,h_1)}(e_1) = w_1$ and $\lambda_{w_2\cdot(\mathbb{P}_2,h_2)}(e_2) = w_2 \cdot (\mathbb{P}_2, h_2)$. Hence $(*_e)$ yields $w_1 = w_2 \cdot (\mathbb{P}_2, h_2)$. In particular $w_2 \cdot (\mathbb{P}_2, h_2) \cdot (\mathbb{P}_1, h_1) \in \mathcal{P}$, therefore $\mathbb{P}_1 = \mathsf{ch}^{h_1}_{\mathbb{P}_2,e_2}$ and $e_1 = e_2$. Notice e_1, that is also e_2, satisfies the same concepts in $\mathfrak{I}^{\mathbb{P}_1}$ and in $\mathfrak{I}^{\mathbb{P}_2}$ (Lemma 8 applies to e_1 seen in $\mathfrak{I}^{\mathbb{P}_1}$ and e_1 seen in $\mathfrak{I}^{\mathbb{P}_2}$), and same for d_1, that is also d_2. Therefore, the $\mathcal{T}$-satisfiability of $\mathfrak{I}^{\mathbb{P}_2}$ ensures that adding fact $\mathrm{P}(d_1, e_1)$ to $\mathfrak{I}^{\mathbb{P}_1}$ does not violate any negative *concept* inclusion from $\mathcal{T}$. We make a case analysis to show the same is true for negative role inclusions:

- First suppose $\mathfrak{gen}^{\mathbb{P}_1} = \{d_1\}$. Since $(d_1, e_1) \in \mathrm{P}^{\mathfrak{I}^{\mathbb{P}_2}}$ and $e_1 \in \mathfrak{gen}^{\mathbb{P}_2}$, then we must have $\mathfrak{fr}^{\mathbb{P}_2} = \{d_1\}$ (Condition 4). We can hence apply Condition 5 from the definition of the link given by $\mathbb{P}_1 = \mathsf{ch}^{h_1}_{\mathbb{P}_2,e_2}$, ensuring that $\mathfrak{I}^{\mathbb{P}_1} \cup \mathfrak{I}^{\mathbb{P}_2}$, which contains $\mathfrak{I}^{\mathbb{P}_1}$ and fact $\mathrm{P}(d_1, e_1)$, is $\mathcal{T}$-satisfiable.
- If $\mathfrak{gen}^{\mathbb{P}_1} \neq \{d_1\}$, then there are no roles between d_1 and e_1 in $\mathfrak{I}^{\mathbb{P}_1}$ (Condition 4), hence no negative role inclusion is violated by adding fact $\mathrm{P}(d_1, e_1)$ in $\mathfrak{I}^{\mathbb{P}_1}$.

(c) $e_1 \in \mathfrak{gen}^{\mathbb{P}_1}$ and $e_2 \in \mathfrak{fr}^{\mathbb{P}_2}$.
Same arguments as for Case 2.b but with $\mathbb{P}_2 = \mathsf{ch}^{h_2}_{\mathbb{P}_1,e_1}$.

(d) $e_1 \in \mathfrak{fr}^{\mathbb{P}_1}$ and $e_2 \in \mathfrak{fr}^{\mathbb{P}_2}$.
We have $\lambda_{w_1\cdot(\mathbb{P}_1,h_1)}(e_1) = w_1$ and $\lambda_{w_2\cdot(\mathbb{P}_2,h_2)}(e_2) = w_2$. Hence $(*_e)$ yields the existence of $w \cdot (\mathbb{Q}, h)$ such that $w_1 = w_2 = w \cdot (\mathbb{Q}, h)$. In particular $w \cdot (\mathbb{Q}, h) \cdot (\mathbb{P}_1, h_1) \in \mathcal{P}$, hence $\mathbb{P}_1 = \mathsf{ch}^{h_1}_{\mathbb{Q},e_1}$. Similarly we obtain $\mathbb{P}_2 = \mathsf{ch}^{h_2}_{\mathbb{Q},e_2}$. As $e_1, e_2 \notin \Delta^*$, the pattern $\mathbb{Q}$ must be different from $\mathbb{P}^*$, hence its generated term is unique, which gives $e_1 = e_2$. Notice e_1, that is also e_2, satisfies the same concepts in $\mathfrak{I}^{\mathbb{P}_1}$ and in $\mathfrak{I}^{\mathbb{P}_2}$ (Lemma 8 applies to e_1 seen in $\mathfrak{I}^{\mathbb{P}_1}$ and e_1 seen in $\mathfrak{I}^{\mathbb{P}_2}$), and same for d_1, that is also d_2. Therefore, the $\mathcal{T}$-satisfiability of $\mathfrak{I}^{\mathbb{P}_2}$ ensures adding fact $\mathrm{P}(d_1, e_1)$ in $\mathfrak{I}^{\mathbb{P}_1}$ does not violate any negative *concept* inclusion from $\mathcal{T}$. It remains to treat the case of negative *role* inclusions. Notice that due to Condition 4 of links, and the facts that $(d_2, e_2) \in \mathrm{P}^{\mathfrak{I}^{\mathbb{P}_2}}$, $e_2 \notin \Delta^{\mathcal{I}^*}$, and $d_2 \in \Delta^{\mathcal{I}^*}$, we must have $\mathfrak{gen}^{\mathbb{P}_2} = \{d_2\} \subseteq \Delta^{\mathcal{I}^*}$. It follows then from Condition 6 that $\mathsf{next}_{\mathbb{Q}}(h_2) = d_2$. We consider two cases:

- If $\mathfrak{gen}^{\mathbb{P}_1} = \{d_1\}$, we obtain similarly $\mathsf{next}_{\mathbb{Q}}(h_1) = d_1$. Denoting $h_1 := \mathrm{R}_1.\mathrm{B}_1$ and $h_2 := \mathrm{R}_2.\mathrm{B}_2$, we obtain, by definition of a prediction, that R_1 and R_2 are non-contradictory. Due to Condition 4 (on the link between $\mathbb{Q}$ and $\mathbb{P}_2$), we have $\mathcal{T} \models \mathrm{R}_2 \sqsubseteq \mathrm{P}$. Therefore $\mathrm{P}(d_1, e_1)$

is non-contradictory with $\mathrm{R}_1(d_1, e_1)$ and hence with $\mathfrak{I}^{\mathbb{P}_1}$ as all roles between d_1 and e_1 in $\mathfrak{I}^{\mathbb{P}_1}$ are consequences of $\mathrm{R}_1(d_1, e_1)$ (Condition 4 on the link given by $\mathbb{P}_1 = \mathsf{ch}^{h_1}_{\mathbb{Q},e_1}$).

- If $\mathfrak{gen}^{\mathbb{P}_1} \neq \{d_1\}$, then there are no roles between d_1 and e_1 (Condition 4), hence no negative role inclusion is violated by adding fact $\mathrm{P}(d_1, e_1)$ in $\mathfrak{I}^{\mathbb{P}_1}$.

3. $(d_1, d_2 \notin \Delta^{\mathcal{I}^*})$ and $(e_1 \in \Delta^{\mathcal{I}^*}$ or $e_2 \in \Delta^{\mathcal{I}^*})$.
   This case is symmetric to Case 2.

4. $d_1, d_2, e_1, e_2 \notin \Delta^{\mathcal{I}^*}$.
   If $(d_1 \in \mathfrak{gen}^{\mathbb{P}_1}$ and $d_2 \in \mathfrak{gen}^{\mathbb{P}_2})$ or $(e_1 \in \mathfrak{gen}^{\mathbb{P}_1}$ and $e_2 \in \mathfrak{gen}^{\mathbb{P}_2})$, then $(*_d)$ (resp $(*_e)$) yields $\mathbb{P}_1 = \mathbb{P}_2$ and we are easily done. Recalling from the note at the end of Case 1 that we may assume that $d_2 \neq e_2$, we are left with 4 subcases, each immediately leading to a contradiction.

   (a) $d_2 \in \mathfrak{gen}^{\mathbb{P}_2}$ (thus $d_1 \in \mathfrak{fr}^{\mathbb{P}_1}$ and $e_2 \in \mathfrak{fr}^{\mathbb{P}_2}$) and $e_1 \in \mathfrak{gen}^{\mathbb{P}_1}$. $(*_d)$ yields $w_1 = w_2 \cdot (\mathbb{P}_2, h_2)$ and $(*_e)$ yields $w_2 = w_1 \cdot (\mathbb{P}_1, h_1)$, contradiction.
   (b) $d_2 \in \mathfrak{gen}^{\mathbb{P}_2}$ (thus $d_1 \in \mathfrak{fr}^{\mathbb{P}_1}$ and $e_2 \in \mathfrak{fr}^{\mathbb{P}_2}$) and $e_1 \in \mathfrak{fr}^{\mathbb{P}_1}$. $(*_d)$ yields $w_1 = w_2 \cdot (\mathbb{P}_2, h_2)$ and $(*_e)$ yields $w_2 = w_1$, contradiction.
   (c) $d_2 \in \mathfrak{fr}^{\mathbb{P}_2}$ (thus $e_2 \in \mathfrak{gen}^{\mathbb{P}_2}$, thus $e_1 \in \mathfrak{fr}^{\mathbb{P}_1}$) and $d_1 \in \mathfrak{gen}^{\mathbb{P}_1}$. $(*_d)$ yields $w_2 = w_1 \cdot (\mathbb{P}_1, h_1)$ and $(*_e)$ yields $w_1 = w_2 \cdot (\mathbb{P}_2, h_2)$, contradiction.
   (d) $d_2 \in \mathfrak{fr}^{\mathbb{P}_2}$ (thus $e_2 \in \mathfrak{gen}^{\mathbb{P}_2}$, thus $e_1 \in \mathfrak{fr}^{\mathbb{P}_1}$) and $d_1 \in \mathfrak{fr}^{\mathbb{P}_1}$. $(*_d)$ yields $w_2 = w_1$ and $(*_e)$ yields $w_1 = w_2 \cdot (\mathbb{P}_2, h_2)$, contradiction. □

Lemma 11. *If $\pi : r \to \mathcal{I}$ is a match of $r \subseteq q$, then for all $w \cdot (\mathbb{P}, h) \in \mathcal{P}$, we have $(r, \pi') \in \mathfrak{M}^{\mathbb{P}}$ where $\pi' := (\lambda_{w\cdot(\mathbb{P},h)})^{-1} \circ \pi_{|\Delta}$ with $\Delta := \pi^{-1}(\lambda_{w\cdot(\mathbb{P},h)}(\Delta^{\mathfrak{I}^{\mathbb{P}}}))$.*

Proof. Considering a breadth-first total order $\leqslant$ on $\mathcal{P}$, and given $W \in \mathcal{P}$, define $\mathcal{I}_W$ as follows:

$$\mathcal{I}_W = \bigcup_{w\cdot(\mathbb{P},h)\leqslant W} \lambda_{w\cdot(\mathbb{P},h)}(\mathfrak{I}^{\mathbb{P}}).$$

We prove by induction on $W \in \mathcal{P}$ that for all $r \subseteq q$, all matches $\pi : r \to \mathcal{I}_W$ and for all $w \cdot (\mathbb{P}, h) \leqslant W$, we have $(r, \pi') \in \mathfrak{M}^{\mathbb{P}}$ where $\pi' := (\lambda_{w\cdot(\mathbb{P},h)})^{-1} \circ \pi_{|\Delta}$ with $\Delta := \pi^{-1}(\lambda_{w\cdot(\mathbb{P},h)}(\Delta^{\mathfrak{I}^{\mathbb{P}}}))$.

- Assume $W = (\mathbb{P}^*, \emptyset)$, we have $\mathcal{I}_W = \mathcal{I}^*$. Consider $r \subseteq q$ and a match $\pi : r \to \mathcal{I}_W$. The only $w \leqslant W$ is $w = W = (\mathbb{P}^*, \emptyset)$. Recalling $\lambda_{(\mathbb{P}^*,\emptyset)} = \mathsf{Id}$, we have $\pi' = \pi$. Therefore (r, π) belongs to the induced specification of $\mathcal{I}^*$. Since $\mathfrak{M}^*$ is coherent, it contains in particular (r, π'), which concludes the base case.

- Assume $W \in \mathcal{P}$ with $(\mathbb{P}^*, \emptyset) < W$ and the statement holds for all $w_0 < W$ (Induction hypothesis 1). Consider $r \subseteq q$ and a match $\pi : r \to \mathcal{I}_W$. Consider $w \cdot (\mathbb{P}, h) \leqslant W$. Denote d the distance from W to $w \cdot (\mathbb{P}, h)$ in the tree $\mathcal{P}$, that is the number of links required to move from W to $w \cdot (\mathbb{P}, h)$. We prove by induction on d that $(r, \pi') \in \mathfrak{M}^{\mathbb{P}}$ where $\pi' := (\lambda_{w \cdot (\mathbb{P},h)})^{-1} \circ \pi_{|\Delta}$ and with $\Delta := \pi^{-1}(\lambda_{w \cdot (\mathbb{P},h)}(\Delta^{\mathfrak{I}^{\mathbb{P}}}))$.
  - When $d = 0$, we have $W = w \cdot (\mathbb{P}, h)$. Let W' the predecessor of W w.r.t. $\leqslant$. We partition r into r_1 the atoms α from r such that π is a match for α in $\lambda_W(\mathfrak{I}^{\mathbb{P}})$ and r_2 the other atoms, which are hence necessarily mapped by π into $\mathcal{I}_{W'}$. We denote by $\pi_1 := \pi_{|\mathsf{var}(r_1)}$ and $\pi_2 := \pi_{|\mathsf{var}(r_2)}$ the corresponding restrictions of π.

    First note that since $\mathfrak{M}^{\mathbb{P}}$ is coherent, it contains the pair (r_1, π'_1) where $\pi'_1 := (\lambda_{w \cdot (\mathbb{P},h)})^{-1} \circ (\pi_1)_{|\Delta^1}$ with $\Delta^1 := (\pi_1)^{-1}(\lambda_{w \cdot (\mathbb{P},h)}(\Delta^{\mathfrak{I}^{\mathbb{P}}}))$.

    Letting $w = w' \cdot (\mathbb{Q}, h')$, we next note that applying the Induction Hypothesis 1 on W' with w (which is indeed $\leqslant W'$) and r_2 and π_2, gives us $(r_2, \pi'_2) \in \mathfrak{M}^{\mathbb{Q}}$ where $\pi'_2 := (\lambda_{w' \cdot (\mathbb{Q},h')})^{-1} \circ (\pi_2)_{|\Delta^2}$ with $\Delta^2 := (\pi_2)^{-1}(\lambda_{w' \cdot (\mathbb{Q},h')}(\Delta^{\mathfrak{I}^{\mathbb{Q}}}))$.

    Since $w' \cdot (\mathbb{Q}, h') \cdot (\mathbb{P}, h) \in \mathcal{P}$, we can consider $\mathbb{P} = \mathsf{ch}^h_{\mathbb{Q},e}$, where e denotes the frontier of $\mathbb{P}$. Condition 3 in the definition of a link therefore ensures $(r_2, (\pi'_2)_{|\Delta^{\mathcal{I}^*} \cup \{e\}}) \in \mathfrak{M}^{\mathbb{P}}$. We'd like to form the union of this latter pair with (r_1, π'_1).

    Consider $v \in \mathsf{var}(r_1) \cap \mathsf{var}(r_2)$. Since r_1 contains only atoms that are mapped on $\lambda_W(\mathfrak{I}^{\mathbb{P}})$ by π, the variable v is thus mapped either to an element of Δ^*, to w or to $w \cdot (\mathbb{P}, h)$. The latter is excluded as r_2 only contains atoms that are mapped in $\mathcal{I}_{W'}$ but $w \cdot (\mathbb{P}, h) \notin \Delta^{\mathcal{I}_{W'}}$ since $\leqslant$ is breath-first and $W' < W = w \cdot (\mathbb{P}, h)$. If $\pi(v) \in \Delta^*$, then it is clear that π'_1 and $(\pi'_2)_{|\Delta^{\mathcal{I}^*} \cup \{e\}}$ are defined and equal on v. Otherwise $\pi(v) = w$, which yields that $\lambda_W(e) = w$ and $\lambda_w(e) = w$. The first ensures π'_1 is defined on v and equal to w, while the second ensures the same for $(\pi'_2)_{|\Delta^{\mathcal{I}^*} \cup \{e\}}$. As this holds for each variable in $v \in \mathsf{var}(r_1) \cap \mathsf{var}(r_2)$, and that $\mathfrak{M}^{\mathbb{P}}$ is coherent we have $(r_1 \cup r_2, \pi'_1 \cup (\pi'_2)_{|\Delta^{\mathcal{I}^*} \cup \{e\}}) \in \mathfrak{M}^{\mathbb{P}}$, which is the desired pair.
  - Assume now the property holds for all w at distance $d \geq 0$ from W (Induction Hypothesis 2). Let $w_{d+1} \leqslant W$ be exactly at distance $d + 1$ from W. In particular, notice that $w_{d+1} < W$. There exists a link between w_{d+1} and some $w_d \leqslant W$ at distance exactly d from W. We distinguish two cases:
    - $w_{d+1} = w_d \cdot (\mathbb{P}, h)$. We exhibit another suitable partition of r. Denote w^+_{d+1} the elements $w' \cdot (\mathbb{Q}, h') \in \mathcal{P}$ such that w_{d+1} is a prefix

of $w' \cdot (\mathbb{Q}, h')$ and $w' \cdot (\mathbb{Q}, h') \leqslant W$. Define r_{d+1} as the atoms α from r such that π is a match for α in some $\lambda_{w' \cdot (\mathbb{Q},h')}(\mathfrak{I}^{\mathbb{Q}})$ with $w' \cdot (\mathbb{Q}, h') \in w_{d+1}^{+}$. Let r_d consists of the remaining atoms, which are hence mapped on elements that *cannot* admit w_{d+1} as a prefix. Denote by π_{d+1} and π_d the corresponding restrictions of π.
We first note that $W \notin w_{d+1}^{+}$, as it would contradict w_d being closer to W than w_{d+1}. Therefore π_{d+1} maps r_{d+1} in $\mathcal{I}_{W'}$ and we can apply Induction Hypothesis 1 with w_{d+1}, r_{d+1} and π_{d+1}, which provides $(r_{d+1}, \pi'_{d+1}) \in \mathfrak{M}^{\mathbb{P}}$ where $\pi'_{d+1} := (\lambda_{w_d \cdot (\mathbb{P},h)})^{-1} \circ (\pi_{d+1})_{|\Delta^{d+1}}$ with $\Delta^{d+1} := (\pi_{d+1})^{-1}(\lambda_{w_d \cdot (\mathbb{P},h)}(\Delta^{\mathfrak{I}^{\mathbb{P}}}))$.
Letting $w_d = w_0 \cdot (\mathbb{P}_d, h_d)$, we next note that Induction Hypothesis 2 applied on w_d, r_d and π_d provides $(r_d, \pi'_d) \in \mathbb{P}_d$ where $\pi'_d := (\lambda_{w_0 \cdot (\mathbb{P}_d,h_d)})^{-1} \circ (\pi_d)_{|\Delta^d}$ with $\Delta^d := (\pi_d)^{-1}(\lambda_{w_0 \cdot (\mathbb{P}_d,h_d)}(\Delta^{\mathfrak{I}^{\mathbb{P}_d}}))$. The link between w_{d+1} and w_d then ensures that $(r_d, (\pi'_d)_{|\Delta^{\mathcal{I}^*} \cup \{e\}}) \in \mathbb{P}$ where e denotes the frontier term of $\mathbb{P}$.
Consider $v \in \mathsf{var}(r_{d+1}) \cap \mathsf{var}(r_d)$. Since r_{d+1} contains only atoms that are mapped on $\lambda_{w' \cdot (\mathbb{Q},h')}(\mathbb{Q})$ by π for some $w' \cdot (\mathbb{Q}, h') \in w_{d+1}^{+}$, the variable v is thus mapped either to an element of Δ^*, to w_d or to elements $w' \cdot (\mathbb{Q}, h')$ admitting w_{d+1} as a prefix. But since r_d contains only terms that can not map on elements admitting w_{d+1} as a prefix, only Δ^* or w_d remain possible. Noticing $\lambda_{w_{d+1}}(e) = \lambda_{w_d}(e) = w_d$ if ever $\pi(v) = w_d$ allows to conclude as in the Case $d = 0$.

- $w_d = w_{d+1} \cdot (\mathbb{P}_d, h_d)$. We exhibit another suitable partition of r. Denote w_d^{+} the elements $w' \cdot (\mathbb{Q}, h') \in \mathcal{P}$ such that w_d is a prefix of $w' \cdot (\mathbb{Q}, h')$ and $w' \cdot (\mathbb{Q}, h') \leqslant W$. Define r_d as the atoms α from r such that π is a match for α in some $\lambda_{w' \cdot (\mathbb{Q},h')}(\mathfrak{I}^{\mathbb{Q}})$ with $w' \cdot (\mathbb{Q}, h') \in w_d^{+}$. Let r_{d+1} consists of the remaining atoms, which are hence mapped on elements that *cannot* admit w_d as a prefix. Denote by π_d and π_{d+1} the corresponding restrictions of π.
  We first note that $W \in w_d^{+}$, as w_d is closer to W than w_{d+1}. Therefore π_{d+1} maps r_{d+1} in $\mathcal{I}_{W'}$ and we can apply Induction Hypothesis 1 with w_{d+1}, r_{d+1} and π_{d+1}, which provides $(r_{d+1}, \pi'_{d+1}) \in \mathfrak{M}^{\mathbb{P}}$ where $\pi'_{d+1} := (\lambda_{w_d \cdot (\mathbb{P},h)})^{-1} \circ (\pi_{d+1})_{|\Delta^{d+1}}$ with $\Delta^{d+1} := (\pi_{d+1})^{-1}(\lambda_{w_d \cdot (\mathbb{P},h)}(\Delta^{\mathfrak{I}^{\mathbb{P}}}))$.
  We next note that Induction Hypothesis 2 applied on w_d, r_d and π_d provides $(r_d, \pi'_d) \in \mathbb{P}_d$ where $\pi'_d := (\lambda_{w_0 \cdot (\mathbb{P}_d,h_d)})^{-1} \circ (\pi_d)_{|\Delta^d}$ with $\Delta^d := (\pi_d)^{-1}(\lambda_{w_0 \cdot (\mathbb{P}_d,h_d)}(\Delta^{\mathfrak{I}^{\mathbb{P}_d}}))$. The link between w_{d+1} and w_d then ensures that $(r_d, (\pi'_d)_{|\Delta^{\mathcal{I}^*} \cup \{e\}}) \in \mathbb{P}$ where e denotes the frontier term of $\mathbb{P}_d$.
  Consider $v \in \mathsf{var}(r_{d+1}) \cap \mathsf{var}(r_d)$. Since r_d contains only atoms that are mapped on $\lambda_{w' \cdot (\mathbb{Q},h')}(\mathbb{Q})$ by π for some $w' \cdot (\mathbb{Q}, h') \in w_d^{+}$, the

variable v is thus mapped either to an element of Δ^*, to w_{d+1} or to elements $w' \cdot (\mathbb{Q}, h')$ admitting w_d as a prefix. But since r_d contains only terms that can not map on elements admitting w_d as a prefix, only Δ^* or w_{d+1} remain possible. Noticing $\lambda_{w_{d+1}}(e) = \lambda_{w_d}(e) = w_{d+1}$ if ever $\pi(v) = w_{d+1}$ allows to conclude as in the previous cases. □

A.2 Proofs for Section 3.4 (Countermodels with bounded size)

Theorem 9. *For all $c \in \Delta^{\mathcal{I}^\flat}$ and all $n \leq |q|$, the following mapping:*

$$\rho_{n,c}(\overline{e}) : \mathcal{N}_n^{\mathcal{J},\overline{\Delta^*}}(\overline{c}) \to \mathcal{N}_n^{\mathcal{I}^\flat,\Delta^*}(c) \qquad \overline{e} \mapsto \begin{cases} \rho_{n-1,c}(\overline{e}) & \text{if } \overline{e} \in \mathcal{N}_{n-1}^{\mathcal{J},\overline{\Delta^*}}(\overline{c}) \\ e & \text{if } \overline{e} \in \overline{\Delta^*} \\ r_{\frac{n-\delta_{\overline{c}}(\overline{e})}{2},c} \cdot w^e_{\frac{n+\delta_{\overline{c}}(\overline{e})}{2},e} & \text{otherwise} \end{cases}$$

is a homomorphism satisfying $\rho_{n,c}(\overline{e}) \sim_{|q|+1-n} e$ and $\rho_{n,c}^{-1}(\overline{\Delta^*}) \subseteq \overline{\Delta^*}$.

Proof. Let $c \in \Delta^{\mathcal{I}^\flat}$. We proceed by induction on $n \leq |q|$ and prove along a technical statement. Property $\rho_{n,c}(\overline{e}) \sim_{|q|+1-n} e$ will already ensure $w^{\rho_{n,c}(\overline{e})}_{|q|+1-n,\rho_{n,c}(\overline{e})} = w^e_{|q|+1-n,e}$; we reinforce this latter fact as follows. If $e \in \mathcal{N}_n^{\mathcal{J},\overline{\Delta^*}}(\overline{c}) \setminus \mathcal{N}_{n-1}^{\mathcal{J},\overline{\Delta^*}}(\overline{c})$, then:

$$w^{\rho_{n,c}(\overline{e})}_{|q|+1-\frac{n-\delta_{\overline{c}}(\overline{e})}{2},\rho_{n,c}(\overline{e})} = w^e_{|q|+1-\frac{n-\delta_{\overline{c}}(\overline{e})}{2},e} \qquad (*)$$

It is indeed a stronger statement since $-n \leq \delta_{\overline{c}}(\overline{e}) \leq n$ leads to $0 \leq \frac{n-\delta_{\overline{c}}(\overline{e})}{2} \leq n$, hence $|q| + 1 - n \leq |q| + 1 - \frac{n-\delta_{\overline{c}}(\overline{e})}{2}$. Property $*$ therefore provides a more precise information about the suffix of $\rho_{n,c}e$.

Base case: $n = 0$. Let $\overline{e} \in \mathcal{N}_0^{\mathcal{J},\overline{\Delta^*}}(\overline{c})$, hence $\overline{e} = \overline{c}$. If $\overline{c} \in \overline{\Delta^*}$, then $\rho_{0,c}e = e = c$. Otherwise we have $\delta_{\overline{c}}(\overline{e}) = 0$, hence $\rho_{0,c}e = r_{0,c} \cdot w^c_{0,c} = c$. In both cases $\rho_{0,c}e = c$, and it is straightforward that all the desired properties hold. In particular, agreeing that $\mathcal{N}_{-1}^{\mathcal{J},\overline{\Delta^*}}(\overline{c})$ can reasonably be set to $\emptyset$, our technical statement holds.

Induction case. Assume the statement holds for $0 \leq n-1 < |q|$. Let $\overline{e} \in \mathcal{N}_n^{\mathcal{J},\overline{\Delta^*}}(\overline{c})$. If $\overline{e} \in \mathcal{N}_{n-1}^{\mathcal{J},\overline{\Delta^*}}(\overline{c})$, then the induction hypothesis applies directly on $\overline{e}$ and provides (stronger versions of) the desired properties. Otherwise, we have by definition of neighbourhoods an element $\overline{d} \in \mathcal{N}_{n-1}^{\mathcal{J},\overline{\Delta^*}}(\overline{c})$, not belonging to $\overline{\Delta^*}$ nor to $\mathcal{N}_{n-2}^{\mathcal{J},\overline{\Delta^*}}(\overline{c})$, and a role $\mathrm{P} \in \mathsf{N}_\mathsf{R}^\pm$ such that $(\overline{d}, \overline{e}) \in \mathrm{P}^\mathcal{J}$. We apply the induction hypothesis on $\overline{d}$, which gives $\rho_{n-1,c}(\overline{d}) = r_{\frac{n-1-\delta_{\overline{c}}(\overline{d})}{2},d} \cdot w^{d}_{\frac{n-1+\delta_{\overline{c}}(\overline{d})}{2},d}$ since $\overline{d} \notin \overline{\Delta^*}$. We further distinguish between $\overline{e} \in \Delta^*$ and $\overline{e} \notin \Delta^*$, the latter subcase yielding two subcases by applying Lemma 14 and distinguishing between Cases edge$^+$and edge$^-$. We have therefore three cases to treat.

$\overline{e} \in \overline{\Delta^*}$**.** We have $\rho_{n,c}(\overline{e}) = e$ and the only non-trivial property to prove is that $e \in \mathcal{N}^{\mathcal{I}^\flat,\Delta^*}_{n}(c)$. Recall the induction hypothesis ensures in particular $\rho_{n-1,c}(\overline{d}) \sim_1 d$. Lemma 15 applies and ensures $(\rho_{n-1,c}(\overline{d}), e) \in \mathrm{P}^{\mathcal{I}^\flat}$, which provides the desired property.

edge$^+$. Case edge$^+$ensures $|e| = |d| + 1 \mod 2|q| + 3$, hence $\delta_{\overline{c}}(\overline{e}) = \delta_{\overline{c}}(\overline{d}) + 1$, and $w^{e}_{|q|+1,e} = w^{d}_{|q|+1-1,d} \cdot \mathrm{R.B}$. Therefore, our element $\rho_{n,c}(\overline{e})$ of interest simplifies as:

$$\begin{aligned}
\rho_{n,c}(\overline{e}) &= r_{\frac{n-\delta_{\overline{c}}(\overline{e})}{2},c} \cdot w^{e}_{\frac{n+\delta_{\overline{c}}(\overline{e})}{2},e} \\
&= r_{\frac{n-(\delta_{\overline{c}}(\overline{d})+1)}{2},c} \cdot w^{e}_{\frac{n+(\delta_{\overline{c}}(\overline{d})+1)}{2},e} \\
&= r_{\frac{(n-1)-\delta_{\overline{c}}(\overline{d})}{2},c} \cdot w^{e}_{\frac{(n-1)+\delta_{\overline{c}}(\overline{d})}{2}+1,e} \\
&= r_{\frac{(n-1)-\delta_{\overline{c}}(\overline{d})}{2},c} \cdot w^{d}_{\frac{(n-1)+\delta_{\overline{c}}(\overline{d})}{2},d} \cdot \mathrm{R.B} \\
&= \rho_{n-1,c}(\overline{d}) \cdot \mathrm{R.B},
\end{aligned}$$

which is well-defined and satisfies $\rho_{n,c}(\overline{e}) \sim_{|q|+1-n} e$ from Lemma 14. Recalling that the induction hypothesis gives $\rho_{n-1,c}(\overline{d}) \in \mathcal{N}^{\mathcal{I}^\flat,\Delta^*}_{n-1}(c)$, it follows that $\rho_{n,c}(\overline{e}) \in \mathcal{N}^{\mathcal{I}^\flat,\Delta^*}_{n}(c)$. Furthermore, notice that $\overline{e}$ and $\overline{d}$ satisfy all conditions of our additional statement. Since in Case edge$^+$we have $\mathcal{T} \models \mathrm{R} \sqsubseteq \mathrm{P}$, reusing $\rho_{n,c}(\overline{e}) = \rho_{n-1,c}(\overline{d}) \cdot \mathrm{R.B}$ immediately yields $(\rho_{n-1,c}(\overline{d}), \rho_{n,c}(\overline{e})) \in \mathrm{P}^{\mathcal{I}^\flat}$.

Checking that Property $*$ holds is now a technicality, and recall that since $d \in \mathcal{N}^{\mathcal{J},\overline{\Delta^*}}_{n-1}(\overline{c}) \setminus \mathcal{N}^{\mathcal{J},\overline{\Delta^*}}_{n-2}(\overline{c})$, we can apply it to d by induction hypothesis. We hence have:

$$\begin{aligned}
w^{\rho_{n,c}(\overline{e})}_{|q|+1-\frac{n-\delta_{\overline{c}}(\overline{e})}{2},\rho_{n,c}(\overline{e})} &= w^{\rho_{n-1,c}(\overline{d})}_{|q|+1-\frac{n-\delta_{\overline{c}}(\overline{e})}{2}-1,\rho_{n-1,c}(\overline{d})} \cdot \mathrm{R.B} \\
&= w^{\rho_{n-1,c}(\overline{d})}_{|q|+1-\frac{(n-1)+1-(\delta_{\overline{c}}(\overline{d})+1)}{2}-1,\rho_{n-1,c}(\overline{d})} \cdot \mathrm{R.B} \\
&= w^{\rho_{n-1,c}(\overline{d})}_{|q|+1-\frac{(n-1)-\delta_{\overline{c}}(\overline{d})}{2}-1,\rho_{n-1,c}(\overline{d})} \cdot \mathrm{R.B} \\
&= w^{d}_{|q|+1-\frac{(n-1)-\delta_{\overline{c}}(\overline{d})}{2}-1,d} \cdot \mathrm{R.B} \\
&= w^{e}_{|q|+1-\frac{n-\delta_{\overline{c}}(\overline{e})}{2},e}.
\end{aligned}$$

edge$^-$. Case edge$^-$ensures $|e| = |d| - 1 \mod 2|q| + 3$, hence $\delta_{\overline{c}}(\overline{e}) = \delta_{\overline{c}}(\overline{d}) - 1$, and $w^{d}_{|q|+1,d} = w^{e}_{|q|+1-1,e} \cdot \mathrm{R.B}$. By induction hypothesis, element $\rho_{n-1,c}(\overline{d}) = r_{\frac{(n-1)-\delta_{\overline{c}}(\overline{d})}{2},d} \cdot w^{d}_{\frac{(n-1)+\delta_{\overline{c}}(\overline{d})}{2},d}$ is well-defined. Notice Property $*$ on d (which, again can be applied as $d \in \mathcal{N}^{\mathcal{J},\overline{\Delta^*}}_{n-1}(\overline{c}) \setminus \mathcal{N}^{\mathcal{J},\overline{\Delta^*}}_{n-2}(\overline{c})$) gives more precise information on the suffix of $\rho_{n-1,c}(\overline{d})$ than the definition of $\rho_{n-1,c}(\overline{d})$, because $n \leq |q|+1$ leads to $\frac{(n-1)+\delta_{\overline{c}}(\overline{d})}{2} + 1 \leq |q| + 1 - \frac{(n-1)-\delta_{\overline{c}}(\overline{d})}{2}$. Therefore, $w^d_{\frac{(n-1)+\delta_{\overline{c}}(\overline{d})}{2}+1,d}$ is itself a suffix of $w^d_{|q|+1-\frac{(n-1)-\delta_{\overline{c}}(\overline{d})}{2},d}$, which equals $w^{\rho_{n-1,c}(\overline{d})}_{|q|+1-\frac{(n-1)-\delta_{\overline{c}}(\overline{d})}{2},\rho_{n-1,c}(\overline{d})}$. Hence we obtain:

$$\begin{aligned}
\rho_{n-1,c}(\overline{d}) &= r_{\frac{(n-1)-\delta_{\overline{c}}(\overline{d})}{2}+1,d} \cdot w^d_{\frac{(n-1)+\delta_{\overline{c}}(\overline{d})}{2}+1,d} \\
&= r_{\frac{n-\delta_{\overline{c}}(\overline{e})}{2},d} \cdot w^d_{\frac{n+\delta_{\overline{c}}(\overline{e})}{2}+1,d} \\
&= r_{\frac{n-\delta_{\overline{c}}(\overline{e})}{2},d} \cdot w^e_{\frac{n+\delta_{\overline{c}}(\overline{e})}{2},e} \cdot \mathrm{R.B} \\
&= \rho_{n,c}(\overline{e}) \cdot \mathrm{R.B}
\end{aligned}$$

Lemma 14 now ensures $\rho_{n,c}(\overline{e}) \sim_{|q|+1-n} e$ (and could already ensure we can find this suffix of $\rho_{n,c}(\overline{d})$! However, we had to check that the formula still works here, in particular that the suffix of $\rho_{n-1,c}(\overline{d})$ matches long enough the suffix of d) .

Recalling that the induction hypothesis gives $\rho_{n-1,c}(\overline{d}) \in \mathcal{N}^{\mathcal{I}^\flat,\Delta^*}_{n-1}(c)$, it follows that $\rho_{n,c}(\overline{e}) \in \mathcal{N}^{\mathcal{I}^\flat,\Delta^*}_{n}(c)$. Furthermore, notice that $\overline{e}$ and $\overline{d}$ satisfy all conditions of our additional statement. Since in Case edge$^-$we have $\mathcal{T} \models \mathrm{R}^- \sqsubseteq \mathrm{P}$, reusing $\rho_{n-1,c}(\overline{d}) = \rho_{n,c}(\overline{e}) \cdot \mathrm{R.B}$ immediately yields $(\rho_{n-1,c}(\overline{d}), \rho_{n,c}(\overline{e})) \in \mathrm{P}^{\mathcal{I}^\flat}$.

Again, we check Property $*$ holds:

$$\begin{aligned}
w^{\rho_{n,c}(\overline{e})}_{|q|+1-\frac{n-\delta_{\overline{c}}(\overline{e})}{2},\rho_{n,c}(\overline{e})} \cdot \mathrm{R.B} &= w^{\rho_{n-1,c}(\overline{d})}_{|q|+1-\frac{n-\delta_{\overline{c}}(\overline{e})}{2}+1,\rho_{n-1,c}(\overline{d})} \\
&= w^{\rho_{n-1,c}(\overline{d})}_{|q|+1-\frac{(n-1)+1-(\delta_{\overline{c}}(\overline{d})-1)}{2}+1,\rho_{n-1,c}(\overline{d})} \\
&= w^{\rho_{n-1,c}(\overline{d})}_{|q|+1-\frac{(n-1)-\delta_{\overline{c}}(\overline{d})}{2},\rho_{n-1,c}(\overline{d})} \\
&= w^{d}_{|q|+1-\frac{(n-1)-\delta_{\overline{c}}(\overline{d})}{2},d} \\
&= w^{e}_{|q|+1-\frac{n-\delta_{\overline{c}}(\overline{e})}{2},e} \cdot \mathrm{R.B}
\end{aligned}$$

We now verify that $\rho_{n,c}$ is a homomorphism.

- Let $\overline{u} \in \mathrm{A}^{\mathcal{J}} \cap \mathcal{N}^{\mathcal{J},\overline{\Delta^*}}_n(\overline{c})$. By definition of $\mathrm{A}^{\mathcal{J}}$, we have $e \in \mathrm{A}^{\mathcal{I}^\flat}$. Since $n \leq |q|$ we have $\rho_{n,c}(\overline{u}) \sim_1 e$, hence applying Remark 13 we obtain $\rho_{n,c}(\overline{u}) \in \mathrm{A}^{\mathcal{I}^\flat}$.
- Let $(\overline{u}, \overline{v}) \in \mathrm{R}^{\mathcal{J}} \cap (\mathcal{N}^{\mathcal{J},\overline{\Delta^*}}_n(\overline{c}) \times \mathcal{N}^{\mathcal{J},\overline{\Delta^*}}_n(\overline{c}))$. If $\overline{u} \in \overline{\Delta^*}$ or $\overline{v} \in \overline{\Delta^*}$, then Lemma 15 applies on $\rho_{n,c}(\overline{u})$ or on $\rho_{n,c}(\overline{v})$ (recall $\rho_{n,c}(\overline{u}) \sim_1 u$ and $\rho_{n,c}(\overline{v}) \sim_1 v$) and gives $(\rho_{n,c}(\overline{u}), \rho_{n,c}(\overline{v})) \in \mathrm{R}^{\mathcal{J}}$. Otherwise $\overline{u} \notin \overline{\Delta^*}$ and $\overline{v} \notin \overline{\Delta^*}$. Let n_1, n_2 be the minimum integers such that $\overline{u} \in \mathcal{N}^{\mathcal{J},\overline{\Delta^*}}_{n_1}(\overline{c})$ and $\overline{v} \in \mathcal{N}^{\mathcal{J},\overline{\Delta^*}}_{n_2}(\overline{c})$. Since $(\overline{u}, \overline{v}) \in \mathrm{R}^{\mathcal{J}}$, we have $n_1 - n_2 \in \{-1, 0, 1\}$. Definitions of $\delta_{\overline{c}}(\overline{u})$ and $\delta_{\overline{c}}(\overline{v})$ lead to $|u| - |v| = \delta_{\overline{c}}(\overline{u}) - \delta_{\overline{c}}(\overline{v}) \mod 2|q|+3$. Lemma 14 gives $|u| = |v| \pm 1 \mod 2|q|+3$. Recall $\delta_{\overline{c}}(\overline{u}), \delta_{\overline{d}}(\overline{v}) \in [-|q|, |q|]$, hence $-2|q|-1 \leq \delta_{\overline{c}}(\overline{u}) - \delta_{\overline{c}}(\overline{u}) \mp$

$1 \leq 2|q|+1$. Since $\delta_{\overline{c}}(\overline{u}) - \delta_{\overline{d}}(\overline{v}) \mp 1 = 0 \mod 2|q|+3$ and $2|q|+1 < 2|q|+3$, we must have $\delta_{\overline{c}}(\overline{u}) - \delta_{\overline{c}}(\overline{v}) = \pm 1$. Joint to Remark 15, it excludes the case $n_1 - n_2 = 0$. We are hence left with $n_1 = n_2 \pm 1$. Applying our additional property with $k := \max(n_1, n_2)$ gives $(\rho_{n,c}(\overline{u}), \rho_{n,c}(\overline{v})) \in \mathrm{R}^{\mathcal{I}^\flat}$.

Finally, $\rho_{n,c}^{-1}(\Delta^*) \subseteq \overline{\Delta^*}$ is a straightforward consequence of $\rho_{n,c}(\overline{u}) \sim_1 u$ (and again, recall elements from Δ^* are alone in their equivalent class!). □

A.3 Proofs for Section 5.4 (Tractable cases in data complexity)

Lemma 31. *Let $\mathcal{A}$ be an ABox and $\mathcal{K} := (\mathcal{T}, \mathcal{A})$. Let $(\mathsf{succ}^{\mathcal{K}}_{\mathrm{R}})_{\mathrm{R}}$ be a certain successor preference. Let σ be a legal strategy for $\mathcal{K}$. Let $\mathsf{ch}_{\sigma/\mathcal{K}}$ be a choice of well-typed elements for σ over $\mathcal{K}$. Let $\mathsf{pair}_{\mathsf{ch}_{\sigma/\mathcal{K}}} := (\mathsf{pair}^+_{\mathsf{ch}_{\sigma/\mathcal{K}}}, \mathsf{pair}^-_{\mathsf{ch}_{\sigma/\mathcal{K}}})$ be a pairing for $\mathsf{ch}_{\sigma/\mathcal{K}}$. Denote by $\mathcal{J}$ the interpretation of σ (according to $\mathsf{ch}_{\sigma/\mathcal{K}}$, $\mathsf{pair}_{\mathsf{ch}_{\sigma/\mathcal{K}}}$, and $(\mathsf{succ}^{\mathcal{K}}_{\mathrm{R}})_{\mathrm{R}}$). We have:*

$$
\begin{aligned}
\mathrm{S}^{\mathcal{J}} = {} & \{(\mathsf{a}, \mathsf{b}) \mid \mathcal{K} \models \mathrm{S}(\mathsf{a}, \mathsf{b})\} && \textit{Shape 1} \\
& \cup \left\{ (x, y) \;\middle|\; \begin{array}{l} (x, y) \in \mathsf{crit}^+_{\mathsf{ch}_{\sigma/\mathcal{K}}} \times \mathsf{crit}^-_{\mathsf{ch}_{\sigma/\mathcal{K}}} \\ \mathsf{pair}^+_{\mathsf{ch}_{\sigma/\mathcal{K}}}(x) = y \end{array} \right\} && \textit{Shape 2} \\
& \cup \left\{ (x, \mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{S})) \;\middle|\; x \in \mathsf{crit}^+_{\mathsf{ch}_{\sigma/\mathcal{K}}} \setminus \mathsf{dom}(\mathsf{pair}^+_{\mathsf{ch}_{\sigma/\mathcal{K}}}) \right\} && \textit{Shape 3}^+ \\
& \cup \left\{ (\mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{S}^-), y) \;\middle|\; y \in \mathsf{crit}^-_{\mathsf{ch}_{\sigma/\mathcal{K}}} \setminus \mathsf{dom}(\mathsf{pair}^-_{\mathsf{ch}_{\sigma/\mathcal{K}}}) \right\} && \textit{Shape 3}^- \\
& \cup \left\{ (\mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{S}), \mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{S})) \;\middle|\; \begin{array}{l} \left|\mathsf{crit}^+_{\mathsf{ch}_{\sigma/\mathcal{K}}}\right| > \left|\mathsf{crit}^-_{\mathsf{ch}_{\sigma/\mathcal{K}}}\right| \\ \mathcal{T} \models \exists \mathrm{S}^- \sqsubseteq \exists \mathrm{S} \\ \exists \mathrm{S} \notin \theta_{\mathcal{K}}(\mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{S})) \\ \mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{S}) \notin \mathsf{ch}_{\sigma/\mathcal{K}}(\mathcal{D}^+_\sigma) \end{array} \right\} && \textit{Shape 4}^+ \\
& \cup \left\{ (\mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{S}^-), \mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{S}^-)) \;\middle|\; \begin{array}{l} \left|\mathsf{crit}^-_{\mathsf{ch}_{\sigma/\mathcal{K}}}\right| > \left|\mathsf{crit}^+_{\mathsf{ch}_{\sigma/\mathcal{K}}}\right| \\ \mathcal{T} \models \exists \mathrm{S} \sqsubseteq \exists \mathrm{S}^- \\ \exists \mathrm{S}^- \notin \theta_{\mathcal{K}}(\mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{S}^-)) \\ \mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{S}^-) \notin \mathsf{ch}_{\sigma/\mathcal{K}}(\mathcal{D}^-_\sigma) \end{array} \right\} && \textit{Shape 4}^-
\end{aligned}
$$

Proof. The first inclusion ($\subseteq$) is straightforward.

($\supseteq$) We consider each of the shapes in turn.

1. Let (a, b) such that $\mathcal{K} \models \mathrm{S}(\mathsf{a}, \mathsf{b})$.
   Therefore $(\mathsf{a}, \mathsf{b}) \in \mathcal{C}_{\mathcal{K}}$. By definition: $\chi(\mathsf{a}) = \mathsf{a}$ and $\chi(\mathsf{b}) = \mathsf{b}$, hence $(\mathsf{a}, \mathsf{b}) \in \mathrm{S}^{\mathcal{J}}$.

2. Let (x, y) such that $(x, y) \in \mathsf{crit}^{+}_{\mathsf{ch}_{\sigma/\mathcal{K}}} \times \mathsf{crit}^{-}_{\mathsf{ch}_{\sigma/\mathcal{K}}}$ and $\mathsf{pair}^{+}_{\mathsf{ch}_{\sigma/\mathcal{K}}}(x) = y$.
   Distinguish two cases based on $x \in \mathsf{crit}^{+}_{\mathsf{ch}_{\sigma/\mathcal{K}}}$:

   - If $x \in \mathcal{D}^{+}_{\mathcal{K}}$. By definition, we must have $x \in \mathsf{Ind}(\mathcal{A})$, so $\chi(x) = x$. Moreover, $x\mathrm{S} \in \mathcal{C}_{\mathcal{K}}$, hence $\mathsf{succ}^{\mathcal{K}}_{\mathrm{S}}(x)$ is not defined. Together with $x \in \mathsf{dom}(\mathsf{pair}^{+}_{\mathsf{ch}_{\sigma/\mathcal{K}}})$, this gives $\chi(x\mathrm{S}) = \mathsf{pair}^{+}_{\mathsf{ch}_{\sigma/\mathcal{K}}}(x)$. Since $(x, x\mathrm{S}) \in \mathrm{S}^{\mathcal{C}}_{\mathcal{K}}$, we have $(x, \mathsf{pair}^{+}_{\mathsf{ch}_{\sigma/\mathcal{K}}}(x)) \in \mathrm{S}^{\mathcal{J}}$.
   - If $x = \mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{R})$ with $\mathrm{R} \in \mathcal{D}^{+}_{\sigma}$. By definition of $\mathsf{gen}_{\mathcal{K}}$, there exists $w\mathrm{R} \in \mathcal{C}_{\mathcal{K}}$. Since $\mathrm{R} \notin \{\mathrm{S}, \mathrm{S}^{-}\}$, we have $\chi(w\mathrm{R}) = \mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{R})$. From $\mathrm{R} \in \mathcal{D}^{+}_{\sigma}$, we know that $\mathcal{T} \models \exists \mathrm{R}^{-} \sqsubseteq \exists \mathrm{S}$, which ensures $w\mathrm{RS} \in \Delta^{\mathcal{C}_{\mathcal{K}}}$. The definition of $\mathcal{D}^{+}_{\sigma}$ further tells us that $\exists \mathrm{S} \notin \theta_{\mathcal{K}}(\mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{R}))$. As $\chi(w\mathrm{R}) = x \in \mathsf{dom}(\mathsf{pair}^{+}_{\mathsf{ch}_{\sigma/\mathcal{K}}})$, we must have $\chi(w\mathrm{RS}) = \mathsf{pair}^{+}_{\mathsf{ch}_{\sigma/\mathcal{K}}}(x)$. Finally $(w\mathrm{R}, w\mathrm{RS}) \in \mathrm{S}^{\mathcal{C}}_{\mathcal{K}}$ ensures $(x, \mathsf{pair}^{+}_{\mathsf{ch}_{\sigma/\mathcal{K}}}(x)) \in \mathrm{S}^{\mathcal{J}}$.

3^{+}. Let $(x, \mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{S}))$ such that $x \in \mathsf{crit}^{+}_{\mathsf{ch}_{\sigma/\mathcal{K}}} \setminus \mathsf{dom}(\mathsf{pair}^{+}_{\mathsf{ch}_{\sigma/\mathcal{K}}})$.
Distinguish two cases based on $x \in \mathsf{crit}^{+}_{\mathsf{ch}_{\sigma/\mathcal{K}}}$:

   - If $x \in \mathcal{D}^{+}_{\mathcal{K}}$. By definition $x \in \mathsf{Ind}(\mathcal{A})$, so $\chi(x) = x$. Moreover, $x\mathrm{S} \in \mathcal{C}_{\mathcal{K}}$, hence $\mathsf{succ}^{\mathcal{K}}_{\mathrm{S}}(x)$ is not defined. Combined with $x \notin \mathsf{dom}(\mathsf{pair}^{+}_{\mathsf{ch}_{\sigma/\mathcal{K}}})$, we obtain $\chi(x\mathrm{S}) = \mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{S})$. Since $(x, x\mathrm{S}) \in \mathrm{S}^{\mathcal{C}}_{\mathcal{K}}$, we have $(x, \mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{S})) \in \mathrm{S}^{\mathcal{J}}$.
   - If $x = \mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{R})$ with $\mathrm{R} \in \mathcal{D}^{+}_{\sigma}$. By definition of $\mathsf{gen}_{\mathcal{K}}$, there exists $w\mathrm{R} \in \Delta^{\mathcal{C}_{\mathcal{K}}}$. Since $\mathrm{R} \notin \{\mathrm{S}, \mathrm{S}^{-}\}$, it gives $\chi(w\mathrm{R}) = \mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{R})$. The hypothesis $\mathcal{T} \models \exists \mathrm{R}^{-} \sqsubseteq \exists \mathrm{S}$ ensures $w\mathrm{RS} \in \Delta^{\mathcal{C}_{\mathcal{K}}}$. Since $\exists \mathrm{S} \notin \theta_{\mathcal{K}}(\mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{R}))$ and $\chi(w\mathrm{R}) = x \notin \mathsf{dom}(\mathsf{pair}^{+}_{\mathsf{ch}_{\sigma/\mathcal{K}}})$, it gives $\chi(w\mathrm{RS}) = \mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{S})$. Finally $(w\mathrm{R}, w\mathrm{RS}) \in \mathrm{S}^{\mathcal{C}}_{\mathcal{K}}$ ensures $(x, \mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{S})) \in \mathrm{S}^{\mathcal{J}}$.

3^{-}. Symmetric to Case 3^{+}.

4^{+}. Let $(\mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{S}), \mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{S}))$ with $\left|\mathsf{crit}^{+}_{\mathsf{ch}_{\sigma/\mathcal{K}}}\right| > \left|\mathsf{crit}^{-}_{\mathsf{ch}_{\sigma/\mathcal{K}}}\right|$, $\mathcal{T} \models \exists \mathrm{S}^{-} \sqsubseteq \exists \mathrm{S}$, $\exists \mathrm{S} \notin \theta_{\mathcal{K}}(\mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{S}))$ and $\mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{S}) \notin \mathsf{ch}_{\sigma/\mathcal{K}}(\mathcal{D}^{+}_{\sigma})$. Because of $\left|\mathsf{crit}^{+}_{\mathsf{ch}_{\sigma/\mathcal{K}}}\right| > \left|\mathsf{crit}^{-}_{\mathsf{ch}_{\sigma/\mathcal{K}}}\right|$, we know that there exists some $x \in \mathsf{crit}^{+}_{\mathsf{ch}_{\sigma/\mathcal{K}}} \setminus \mathsf{dom}(\mathsf{pair}^{+}_{\mathsf{ch}_{\sigma/\mathcal{K}}})$. Distinguish two cases based on $x \in \mathsf{crit}^{+}_{\mathsf{ch}_{\sigma/\mathcal{K}}}$:

- If $x \in \mathcal{D}^+_\mathcal{K}$. By definition, $x\mathrm{S} \in \mathcal{C}_\mathcal{K}$, hence $\mathsf{succ}^\mathcal{K}_\mathrm{S}(x)$ is not defined. Moreover, we chose $x \in \mathsf{crit}^+_{\mathsf{ch}_{\sigma/\mathcal{K}}} \setminus \mathsf{dom}(\mathsf{pair}^+_{\mathsf{ch}_{\sigma/\mathcal{K}}})$, so $x \notin \mathsf{dom}(\mathsf{pair}^+_{\mathsf{ch}_{\sigma/\mathcal{K}}})$. It follows that $\chi(x\mathrm{S}) = \mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{S})$. Since $\mathcal{T} \models \exists\mathrm{S}^- \sqsubseteq \exists\mathrm{S}$, we have $x\mathrm{SS} \in \Delta^{\mathcal{C}_\mathcal{K}}$. Combined with our assumptions $\exists\mathrm{S} \notin \theta_\mathcal{K}(\mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{S}))$ and $\mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{S}) \notin \mathsf{crit}^+_{\mathsf{ch}_{\sigma/\mathcal{K}}}$, we obtain $\chi(x\mathrm{SS}) = \mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{S})$. Finally from $(x\mathrm{S}, x\mathrm{SS}) \in \mathrm{S}^\mathcal{C}_\mathcal{K}$, we can infer $(\mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{S}), \mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{S})) \in \mathrm{S}^\mathcal{J}$.
- If $x = \mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{R})$ with $\mathrm{R} \in \mathcal{D}^+_\sigma$. By definition of $\mathsf{gen}_\mathcal{K}$, there exists $w\mathrm{R} \in \Delta^{\mathcal{C}_\mathcal{K}}$. From $\mathrm{R} \in \mathcal{D}^+_\sigma$, we have $\mathrm{R} \notin \{\mathrm{S}, \mathrm{S}^-\}$, which gives $\chi(w\mathrm{R}) = \mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{R})$. Moreover, we also have that $\mathcal{T} \models \exists\mathrm{R}^- \sqsubseteq \exists\mathrm{S}$, which ensures $w\mathrm{RS} \in \Delta^{\mathcal{C}_\mathcal{K}}$. Since $\exists\mathrm{S} \notin \theta_\mathcal{K}(\mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{R}))$ and $\chi(w\mathrm{R}) = x \notin \mathsf{dom}(\mathsf{pair}^+_{\mathsf{ch}_{\sigma/\mathcal{K}}})$, we have $\chi(w\mathrm{RS}) = \mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{S})$. Furthermore, we assumed $\exists\mathrm{S} \notin \theta_\mathcal{K}(\mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{S}))$ and $\mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{S}) \notin \mathsf{ch}_{\sigma/\mathcal{K}}(\mathcal{D}^+_\sigma)$, ensuring in particular $\mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{S}) \notin \mathsf{crit}^+_{\mathsf{ch}_{\sigma/\mathcal{K}}}$. Hence $\chi(w\mathrm{RSS}) = \mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{S})$. We conclude by using $(w\mathrm{RS}, w\mathrm{RSS}) \in \mathrm{S}^\mathcal{C}_\mathcal{K}$ to infer $(\mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{S}), \mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{S})) \in \mathrm{S}^\mathcal{J}$.

4^-. Symmetric to Case 4^+. □

End of the proof of Lemma 33 Now we prove that ρ is injective. Consider two matches π_1, π_2 of q_S in $\mathcal{J}$ such that $\rho(\pi_1) = \rho(\pi_2)$. We will use $\pi_1[1], \pi_1[2]$ to refer to the first and second arguments of π_1, and similarly for π_2. We consider all nine cases, showing in each case that either the situation cannot occur or that $\pi_1 = \pi_2$:

1\. 1. When π_1, π_2 are both of Shape 1, we have $\pi_1 = \rho(\pi_1) = \rho(\pi_2) = \pi_2$.

2, 3^+. $\pi_1 = (\mathsf{a}, \mathsf{b})$ is of Shape 1, so $\pi_1 = \rho(\pi_1)$, while $\pi_2 = (x, y)$ is of Shape 2 or 3^+, which implies that $x \in \mathsf{crit}^+_{\mathsf{ch}_{\sigma/\mathcal{K}}}$.

- If $x \in \mathcal{D}^+_\mathcal{K}$, then $\rho(\pi_2)[1] = \mathsf{ori}^+(x) = x$. It follows that $\rho(\pi_1)[1] = \mathsf{a} = x$. But $\mathsf{a} \notin \mathcal{D}^+_\mathcal{K}$ since $\mathrm{S}(\mathsf{a}, \mathsf{b}) \in \mathcal{A}$, which is a contradiction.
- If $x = \mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{R})$ with $\mathrm{R} \in \mathcal{D}^+_\sigma$, then in particular $\exists\mathrm{S} \notin \theta_\mathcal{K}(\mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{R}))$. Lemma 28 tells us that $\theta_\mathcal{K}(\mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{R})) = \theta_\mathcal{K}(f(\mathsf{repr}_\mathcal{K}(\mathrm{R})))$. We also have $\mathsf{ori}^+(x) = f(\mathsf{repr}_\mathcal{K}(\mathrm{R}))$, so $\rho(\pi_2)[1] = f(\mathsf{repr}_\mathcal{K}(\mathrm{R}))$. From $\rho(\pi_1) = \rho(\pi_2)$ we get $\mathsf{a} = f(\mathsf{repr}_\mathcal{K}(\mathrm{R}))$. Putting this together, we get $\exists\mathrm{S} \notin \theta_\mathcal{K}(\mathsf{a})$, which contradicts $\mathrm{S}(\mathsf{a}, \mathsf{b}) \in \mathcal{A}$.

4^+. In particular $\exists\mathrm{S} \notin \theta_\mathcal{K}(\mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{S}))$ and $\rho(\pi_2)[1] = f(\mathsf{repr}_\mathcal{K}(\mathrm{S}))$. Lemma 28 provides $\theta_\mathcal{K}(\mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{S})) = \theta_\mathcal{K}(\rho(\pi_2)[1])$. Recall $\pi_1 = \rho(\pi_1) = (\mathsf{a}, \mathsf{b})$ and $\rho(\pi_1) = \rho(\pi_2)$, hence $\exists\mathrm{S} \notin \theta_\mathcal{K}(\pi_1[1])$. Contradiction with $\mathrm{S}(\mathsf{a}, \mathsf{b}) \in \mathcal{A}$.

2, 3^+. 1. Symmetric to Case 1.(2, 3^+).

2, 3^+. As both π_1 and π_2 are of Shapes 2 / 3^+, we have $\mathsf{ori}^+(\pi_1[1]) = \mathsf{ori}^+(\pi_2[1])$. We can apply Lemma 32 to obtain $\pi_1[1] = \pi_2[1]$. By examining the conditions of Shapes 2 and 3^+, we can see that π_1 and π_2 must have the same shape, and moreover, their second arguments must coincide, yielding $\pi_1 = \pi_2$.

4^+. As $\pi_1 = (x, y)$ is of Shape 2 / 3^+, we have $x = \pi_1[1] \in \mathsf{crit}^+_{\mathsf{ch}_{\sigma/\mathcal{K}}}$. As $\pi_2 = (\mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{S}), \mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{S}))$ is of Shape 4, we have $\mathcal{T} \models \exists \mathrm{S}^- \sqsubseteq \exists \mathrm{S}$, $\exists \mathrm{S} \notin \theta_{\mathcal{K}}(\mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{S}))$, $\mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{S}) \notin \mathsf{ch}_{\sigma/\mathcal{K}}(\mathcal{D}^+_\sigma)$, and the following $\rho(\pi_2) = (f(\mathsf{repr}_{\mathcal{K}}(\mathrm{S})), \mathsf{succ}^{\mathcal{I}}_{\mathrm{S}}(f(\mathsf{repr}_{\mathcal{K}}(\mathrm{S}))))$.

- If $x \in \mathcal{D}^+_{\mathcal{K}}$, then $\rho(\pi_1)[1] = \mathsf{ori}^+(x) = x \in \mathsf{Ind}(\mathcal{A})$. From $\rho(\pi_1) = \rho(\pi_2)$ and above, we get $x = \rho(\pi_1)[1] = \rho(\pi_2)[1] = f(\mathsf{repr}_{\mathcal{K}}(\mathrm{S}))$. By statement 1 of Lemma 2, we have $\theta_{\mathcal{K}}(\mathsf{ch}_{\sigma_{f \circ \mathsf{repr}_{\mathcal{K}}}/\mathcal{K}}(\mathrm{S})) = \theta_{\mathcal{K}}(f(\mathsf{repr}_{\mathcal{K}}(\mathrm{S})))$, yielding $\theta_{\mathcal{K}}(x) = \theta_{\mathcal{K}}(\mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{S}))$. Recall that $x \in \mathcal{D}^+_{\mathcal{K}}$ ensures in particular $\exists \mathrm{S} \in \theta_{\mathcal{K}}(x)$, it contradicts the assumption $\exists \mathrm{S} \notin \theta_{\mathcal{K}}(\mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{S}))$.
- If $x = \mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{R})$ with $\mathrm{R} \in \mathcal{D}^+_\sigma$, then $\rho(\pi_1)[1] = f(\mathsf{repr}_{\mathcal{K}}(\mathrm{R}))$. As $\rho(\pi_1) = \rho(\pi_2)$ and $\rho(\pi_2)[1] = f(\mathsf{repr}_{\mathcal{K}}(\mathrm{S}))$, we have $f(\mathsf{repr}_{\mathcal{K}}(\mathrm{R})) = f(\mathsf{repr}_{\mathcal{K}}(\mathrm{S}))$. The second statement of Lemma 28 gives us $\mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{R}) = \mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{S})$. Since $\mathrm{R} \in \mathcal{D}^+_\sigma$, we get a contradiction with $\mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{S}) \notin \mathsf{ch}_{\sigma/\mathcal{K}}(\mathcal{D}^+_\sigma)$.

4^+. 1. Symmetric to Case 1.4^+.

2, 3^+. Symmetric to Case $(2, 3^+).4^+$.

4^+. By definition $\pi_1 = \pi_2 = (\mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{S}), \mathsf{ch}_{\sigma/\mathcal{K}}(\mathrm{S}))$.

A.4 Proofs for Section 5.5 (Role cardinality over DL-Lite$^{\mathcal{H}}_{\mathsf{pos}}$)

Before starting with the proper proof, we need some additional definitions.

First, why do we start from a demanding individual free KB? We want to take advantage of the absence of non-trivial propagation, in particular of violation of its Condition 2 (see Definition 68), which is involving roles generated by $\exists \mathrm{S}^-$ (resp $\exists \mathrm{S}$). Therefore, we somehow need these generated roles to be here as soon as possible: we need their causes, that are S-assertions, in our initial ABox.

Speaking about *causes*, take a look at Condition 3 from Definition 68. Here is a handy definition to take advantage of the cases in which this latter condition is broken.

Definition 74. *Let* $w\mathrm{R}$ *be an anonymous element of* $\Delta^{\mathcal{C}_{\mathcal{K}}}$*. A* cause *of* $w\mathrm{R}$ *is a positive concept such that: if* $w \in \mathsf{Ind}(\mathcal{A})$*, then* $\mathsf{cause}(w\mathrm{R})$ *is either an atomic* *concept* B *such that* $\mathcal{K} \models \mathrm{B}(w)$ *and* $\mathcal{T} \models \mathrm{B} \sqsubseteq \exists\mathrm{R}$*, or a positive concept* $\exists\mathrm{T}$ *such that there exists some* b *with* $\mathcal{K} \models \mathrm{T}(w, \mathsf{b})$ *and* $\mathcal{T} \models \exists\mathrm{T} \sqsubseteq \exists\mathrm{R}$*. Otherwise* $w = w_0\mathrm{T}$*, then* $\mathsf{cause}(w\mathrm{R}) := \exists\mathrm{T}^-$*.*

Following this line, here is a definition capturing the role provided by a violation of Condition 1 (again from Definition 68).

Definition 75. *For an element* $w \in \Delta^{\mathcal{C}_\mathcal{K}}$*, if there exists a positive role* U *such that* $w\mathrm{U} \in \Delta^{\mathcal{C}_\mathcal{K}}$*,* $\mathcal{T} \models \mathrm{U} \sqsubseteq \mathrm{S}$ *and* $\mathcal{T} \models \mathrm{U} \sqsubseteq \mathrm{S}^-$*, then we pick such a role* U *and say it is the* leader *of the element* w*, denoted* $\mathsf{leader}(w)$*.*

Our construction proceed by induction on $\mathcal{C}_\mathcal{K}$, so here is the order we pick.

Definition 76. *We pick an order* $\leq$ *on* $\Delta^{\mathcal{C}_\mathcal{K}}$ *such that:* $\leq$ *is breadth-first and for all* $w \in \mathcal{C}_\mathcal{K}$*, if* $\mathsf{leader}(w)$ *is defined, then* $\forall \mathrm{R}, w\mathrm{R} \in \Delta^{\mathcal{C}_\mathcal{K}} \;\Rightarrow\; w \cdot \mathsf{leader}(w) \leq w\mathrm{R}$*.*

We are now all setup for the main construction. Here is some intuition before this two-page long definition. Recall we explore the canonical model, especially anonymous elements being words ending by a particular positive role. Whenever we encounter a nilpotent role, we send it on its choice, because if ever it propagates some non-nilpotent roles, then the choice of well-typed elements ensures there are some further pre-existing matches on which to fold. Otherwise (and that is a big otherwise), if we previously encountered a bipotent role or a bidirectionnal match (that is a pair of element (a, b) such that both (a, b) and (b, a) are pre-existing matches), then it is costless to reuse it (protip: that's what the "flag" is for!). Otherwise, we look for such a bidirectionnal match around which would solve all further problems. If none, then the role you are encountering surely isn't bipotent: a nilpotent role propagating a bipotent role could not have let you end up on an element without a bidirectionnal match around (it would contradict the definition of a choice of well-typed element!), and non-nilpotent nor bipotent roles propagating a bipotent role could not have either (it would violate the absence of non-trivial propagation!). Therefore, at this point, the role you are encountering is either a subrole of S or of S^-, but not both. In both cases, you are ensured to find a pre-existing match on which to fold (otherwise it would again violate either the choice of well-typed elements of the absence of non-trivial propagation).

Here is the more formal approach. Various properties are carried along the construction. Property 1 ensures nilpotent roles behave as expected. Properties 2^+ and 2^- ensures we stay within the ABox matches. Property 3 ensures the flag is used as expected. Property 4 ensures violations of Conditions 1 and 2 are being used. Property 5^+ and 5^- ensure violations of Condition 3 are being used.

Proof. By induction on $(\Delta^{\mathcal{C}_\mathcal{K}}, \leq)$, we build two mappings $\mathsf{flag} : \Delta^{\mathcal{C}_\mathcal{K}} \to \{0, 1\}$ and $\chi : \Delta^{\mathcal{C}_\mathcal{K}} \to \mathsf{Ind}(\mathcal{A})$ and ensure alongside that any element $e \in \Delta^{\mathcal{C}_\mathcal{K}}$ satisfies the following properties:

1. If $e = w\mathrm{R} \in \Delta^{\mathcal{C}_\mathcal{K}}$ with R nilpotent, then $\chi(w\mathrm{R}) = \mathsf{ch}_{\sigma/\mathcal{K}}(\theta_\mathcal{K}(\mathrm{R}))$.

2^+. If $e = w\mathrm{R} \in \Delta^{\mathcal{C}_\mathcal{K}}$ and $\mathcal{T} \models \mathrm{R} \sqsubseteq \mathrm{S}$, then $\mathcal{K} \models \mathrm{S}(\chi(w), \chi(w\mathrm{R}))$.

2^-. If $e = w\mathrm{R} \in \Delta^{\mathcal{C}_\mathcal{K}}$ and $\mathcal{T} \models \mathrm{R} \sqsubseteq \mathrm{S}^-$, then $\mathcal{K} \models \mathrm{S}(\chi(w\mathrm{R}), \chi(w))$.

3. If $e = w\mathrm{R} \in \Delta^{\mathcal{C}_\mathcal{K}}$ and $\mathsf{flag}(w\mathrm{R})$, then $\mathcal{K} \models \mathrm{S}(\chi(w), \chi(w\mathrm{R}))$ and $\mathcal{K} \models \mathrm{S}(\chi(w\mathrm{R}), \chi(w))$.

4. If $e = w\mathrm{R} \in \Delta^{\mathcal{C}_\mathcal{K}}$ with R non-nilpotent and $\mathsf{leader}(w)$ is defined, then $\mathsf{flag}(w\mathrm{R})$ and $\chi(w\mathrm{R}) = \chi(w \cdot \mathsf{leader}(w))$.

5^+. If $e = w\mathrm{R}_1 \in \Delta^{\mathcal{C}_\mathcal{K}}$ with $\mathcal{T} \models \mathrm{R}_1 \sqsubseteq \mathrm{S}$ and $\neg\mathsf{flag}(w\mathrm{R}_1)$ and $\mathsf{cause}(w\mathrm{R}_1) = \exists\mathrm{T}$ with $\mathcal{T} \models \mathrm{T} \sqsubseteq \mathrm{S}$ and such that there exists $w\mathrm{R}_1\mathrm{R}_2 \in \Delta^{\mathcal{C}_\mathcal{K}}$ with $\mathcal{T} \models \mathrm{R}_2 \sqsubseteq \mathrm{S}$, then $\mathcal{K} \models \mathrm{T}(\chi(w), \chi(w\mathrm{R}_1))$ or $w = w'\mathrm{T}^-$ and $\chi(w'\mathrm{T}^-\mathrm{R}_1) = \chi(w')$.

5^-. If $e = w\mathrm{R}_1 \in \Delta^{\mathcal{C}_\mathcal{K}}$ with $\mathcal{T} \models \mathrm{R}_1 \sqsubseteq \mathrm{S}^-$ and $\neg\mathsf{flag}(w\mathrm{R}_1)$ and $\mathsf{cause}(w\mathrm{R}_1) = \exists\mathrm{T}$ with $\mathcal{T} \models \mathrm{T} \sqsubseteq \mathrm{S}^-$ and such that there exists $w\mathrm{R}_1\mathrm{R}_2 \in \Delta^{\mathcal{C}_\mathcal{K}}$ with $\mathcal{T} \models \mathrm{R}_2 \sqsubseteq \mathrm{S}^-$, then $\mathcal{K} \models \mathrm{T}(\chi(w), \chi(w\mathrm{R}_1))$ or $w = w'\mathrm{T}^-$ and $\chi(w'\mathrm{T}^-\mathrm{R}_1) = \chi(w')$.

Initialization: Individuals. For all $\mathsf{a} \in \mathsf{Ind}(\mathcal{A})$, we set $\chi(\mathsf{a}) := \mathsf{a}$ and $\mathsf{flag}(\mathsf{a}) := 0$. All properties are trivially satisfied on individuals.

Induction: Anonymous elements. Let $w\mathrm{R} \in \Delta^{\mathcal{C}_\mathcal{K}}$. Assume all properties hold for $e < w\mathrm{R}$.

- If R is nilpotent, then we set $\chi(w\mathrm{R}) := \mathsf{ch}_{\sigma/\mathcal{K}}(\theta_\mathcal{K}(\mathrm{R}))$ and $\mathsf{flag}(w\mathrm{R}) := 0$. Property 1 is satisfied and all other properties trivially hold.
- Else if $\mathsf{leader}(w)$ is defined and $w \cdot \mathsf{leader}(w) < w\mathrm{R}$, then $\chi(w \cdot \mathsf{leader}(w))$ is already defined, and we set $\chi(w\mathrm{R}) := \chi(w \cdot \mathsf{leader}(w))$ and $\mathsf{flag}(w\mathrm{R}) := 1$. By induction hypothesis, properties hold for $w \cdot \mathsf{leader}(w)$ and all transfer to $w\mathrm{R}$, but Property 4. By definition however, Property 4 also holds for $w\mathrm{R}$.
- Else if $\mathsf{flag}(w)$, then w must have shape $w = w_0\mathrm{R}_0$ (recall the initialization sets the flag of all individuals to 0). We set $\chi(w\mathrm{R}) := \chi(w_0)$ and $\mathsf{flag}(w\mathrm{R}) := 1$. Property 1, 5^+ and 5^- trivially hold for $w\mathrm{R}$. By induction hypothesis on $w_0\mathrm{R}_0$, Property 3 for $w_0\mathrm{R}_0$ ensures that Properties 2^+, 2^- and 3 continue to hold for $w\mathrm{R}$. Property 4 for $w\mathrm{R}$ is only relevant if $\mathsf{leader}(w) = \mathrm{R}$, in which case it trivially holds.
- Else if there exists an individual name b such that $\mathcal{K} \models \mathrm{S}(\chi(w), \mathsf{b}) \wedge \mathrm{S}(\mathsf{b}, \chi(w))$, then we set $\chi(w\mathrm{R}) := \mathsf{b}$ and $\mathsf{flag}(w\mathrm{R}) := 1$. Therefore Property 2^+, 2^-, 3 hold for $w\mathrm{R}$. Property 1, 5^+ and 5^- trivially hold for $w\mathrm{R}$. Again, Property 4 for $w\mathrm{R}$ is only relevant if $\mathsf{leader}(w) = \mathrm{R}$, in which case it trivially holds.

- Else if R is bipotent, that is $\mathcal{T} \models \mathrm{R} \sqsubseteq \mathrm{S}$ and $\mathcal{T} \models \mathrm{R} \sqsubseteq \mathrm{S}^-$. We distinguish several subcases, each leading to a contradiction.

  - If $w \in \mathsf{Ind}(\mathcal{A})$, then $w \in \mathcal{D}^{\pm}_{\mathcal{K}}$. Contradicts the absence of bidemanding individuals.
  - If $w = w_0\mathrm{R}_0$ with R_0 nilpotent. By induction assumption and Property 1 and the definition of $\mathsf{ch}_{\sigma/\mathcal{K}}$, $\theta_{\mathcal{K}}(\chi(w)) = \{\{\mathrm{S}, \mathrm{S}'\}, \{\mathrm{S}\}, \{\mathrm{S}'\}\}$. This is a contradiction with not being in the previous case.
  - Otherwise $w = w_0\mathrm{R}_0$, then $\mathsf{leader}(w_0)$ is not defined as Property 4 from induction hypothesis on $w_0\mathrm{R}_0$ would then contradicts $\mathsf{flag}(w)$ being false. In particular R_0 cannot be bipotent. Hence either $\mathcal{T} \models \mathrm{R}_0 \sqsubseteq \mathrm{S}$ or $\mathcal{T} \models \mathrm{R}_0 \sqsubseteq \mathrm{S}^-$, but not both. Both cases being symmetrical, we now focus on $\mathcal{T} \models \mathrm{R}_0 \sqsubseteq \mathrm{S}$. For the triple $(\mathsf{cause}(w_0), \mathrm{R}_0, \mathrm{R})$, we have a propagation of S. As there are no non-trivial propagation of S, there must be an interference (Definition 68). Note that there cannot be an interference of the first type. Indeed, if U were such an interference, then $\mathsf{flag}(w)$ would be set, which we excluded in a previous case. Hence an interference should be of one of the other types:
    - If it is of type 2, then we have a bipotent U generated by $\exists \mathrm{S}^-$. Property 2^+ from induction hypothesis gives $\mathcal{K} \models \mathrm{S}(\chi(w_0), \chi(w))$. Hence $\mathcal{K} \models \exists z\ \mathrm{U}(\chi(w), z)$. As U is bipotent and $\chi(w)$ cannot be a bidemanding element, there exists an individual b such that $\mathcal{K} \models \mathrm{S}(\chi(w), \mathsf{b})$ and $\mathcal{K} \models \mathrm{S}^-(\chi(w), \mathsf{b})$, which we excluded in a previous case.
    - If it is of type 3, then $\mathsf{cause}(w_0\mathrm{R}_0) = \exists \mathrm{T}$ with $\exists \mathrm{T}^-$ generating a bipotent role. Property 5^+ by induction hypothesis on $w_0\mathrm{R}_0$ provides either $\mathcal{K} \models \mathrm{T}(\chi(w_0), \chi(w_0\mathrm{R}_0))$ or $w_0 = w_0'\mathrm{T}^-$ and $\chi(w) = \chi(w_0'\mathrm{T}^-)$. If $\mathcal{K} \models \mathrm{T}(\chi(w_0), \chi(w_0\mathrm{R}_0))$, since $\chi(w)$ cannot be a bidemanding element, there exists an individual b such that $\mathcal{K} \models \mathrm{S}(\chi(w), \mathsf{b})$ and $\mathcal{K} \models \mathrm{S}^-(\chi(w), \mathsf{b})$, which we excluded in a previous case. Otherwise $w_0 = w_0'\mathrm{T}^-$ and $\chi(w) = \chi(w_0'\mathrm{T}^-)$. Properties 2^+ and 2^- from induction hypothesis on w ensure $\mathcal{K} \models \mathrm{S}(\chi(w), \mathsf{b})$ and $\mathrm{S}^-(\chi(w_0), \chi(w))$, which leads to the same excluded case.

- Otherwise either $\mathcal{T} \models \mathrm{R} \sqsubseteq \mathrm{S}$ or $\mathcal{T} \models \mathrm{R} \sqsubseteq \mathrm{S}^-$ (but not both, as we already dealt with bipotent R). These two cases are symmetrical, we focus on $\mathcal{T} \models \mathrm{R} \sqsubseteq \mathrm{S}$. We investigate the various possibilities for w and $\mathsf{cause}(w\mathrm{R})$:

  - If $\mathsf{cause}(w\mathrm{R}) = \exists \mathrm{T}$ with $\mathcal{T} \models \mathrm{T} \sqsubseteq \mathrm{S}$ and $w = w_0\mathrm{T}^-$, then we set $\chi(w\mathrm{R}) := \chi(w_0)$ and $\mathsf{flag}(w\mathrm{R}) := 0$. In particular, Property 5^+ is satisfied. Property 2^+ from induction hypothesis on w gives Property 2^+ for $w\mathrm{R}$. Other properties trivially hold.

- Else if $\mathsf{cause}(w\mathrm{R}) = \exists\mathrm{T}$ with $\mathcal{T} \models \mathrm{T} \sqsubseteq \mathrm{S}$ and $w \in \mathsf{Ind}(\mathcal{A})$, then there exists $\mathsf{b} \in \mathsf{Ind}(\mathcal{A})$ such that $\mathcal{K} \models \mathrm{T}(w, \mathsf{b})$, and we set $\chi(w\mathrm{R}) := \mathsf{b}$ and $\mathsf{flag}(w\mathrm{R}) := 0$. In particular, Property 2^+ and Property 5^+ are satisfied. Other properties trivially hold.
- Else if $\mathsf{cause}(w\mathrm{R}) = \exists\mathrm{T}$ with $\mathcal{T} \models \mathrm{T} \sqsubseteq \mathrm{S}^-$ and $w = w_0\mathrm{T}^-$, from all the preceding tests, $(\mathsf{cause}(w_0), \mathrm{T}^-, \mathrm{R})$ provides a propagation of S. As there are no non-trivial propagation, there must be an interference. It cannot be of the first type (otherwise $\mathsf{flag}(w)$ would be set), hence it must be of type 2 or 3:
  - If it is of type 2, then we have a role U generated by $\exists\mathrm{S}^-$ and with $\mathcal{T} \models \mathrm{U} \sqsubseteq \mathrm{S}$. Property 2^+ from induction hypothesis gives $\mathcal{K} \models \mathrm{S}(\chi(w_0), \chi(w))$. Since $\mathcal{K}$ does not contain any positive demanding individuals, there exists an individual b such that $\mathcal{K} \models \mathrm{U}(\chi(w), \mathsf{b})$, and we set $\chi(w\mathrm{R}) := \mathsf{b}$ and $\mathsf{flag}(w\mathrm{R}) := 0$.
  - If it is of type 3, then we set $\chi(w\mathrm{R}) := \chi(w_0)$ and $\mathsf{flag}(w\mathrm{R}) = 0$. Applying Property 5^+ by induction hypothesis on $w_0\mathrm{R}_0$ provides the desired properties.
- Else if $w \in \mathsf{Ind}(\mathcal{A})$, then, since there are no demanding individuals, there exists $\mathsf{b} \in \mathsf{Ind}(\mathcal{A})$ such that $\mathcal{K} \models \mathrm{S}(\chi(w), \mathsf{b})$, then set $\chi(w\mathrm{R}) = \mathsf{b}$ and $\mathsf{flag}(\chi(w\mathrm{R}) = 0$. In particular, Properties 2^+ and 5^+ hold.
- Otherwise $\mathsf{cause}(w\mathrm{R}) = \exists\mathrm{T}$ with T nilpotent, then by Property 1 of induction hypothesis applied on w we have $\chi(w) = \mathsf{ch}_{\sigma/\mathcal{K}}(\theta_{\mathcal{K}}(\mathrm{T}))$. By definition of the choice of well-typed elements, there exists $\mathsf{b} \in \mathsf{Ind}(\mathcal{A})$ such that $\mathcal{K} \models \mathrm{S}(\chi(w), \mathsf{b})$, and we set $\chi(w\mathrm{R}) = \mathsf{b}$ and $\mathsf{flag}(\chi(w\mathrm{R}) = 0$. □

B

Four flavors of interlacings

This annex aims to facilitate the understanding of the four variations of the interlacings by recalling the central Definitions 19 and 20 and Theorem 4 (we encourage the reader to keep a printed version of this annex close at hand).

We recall that these definitions assume given a model $\mathcal{I}$ of an $\mathcal{ALCHI}$ KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, and that Ω denotes the set of all R.A such that $\exists$R.A is the RHS of an axiom in $\mathcal{T}$. Furthermore, it assumes that, for every $\mathrm{R.A} \in \Omega$, we have chosen a function $\mathsf{succ}^{\mathcal{I}}_{\mathrm{R.A}}$ that maps every element $e \in (\exists \mathrm{R.A})^{\mathcal{I}}$ to an element $e' \in \Delta^{\mathcal{I}}$ such that $(e, e') \in \mathrm{R}^{\mathcal{I}}$ and $e' \in \mathrm{A}^{\mathcal{I}}$.

Definition 77. *Over the set* $\mathsf{Ind}(\mathcal{A}) \cdot \Omega^*$*, inductively build the following mapping:*

$$\begin{array}{rcl} f : \mathsf{Ind}(\mathcal{A}) \cdot \Omega^* & \to & \Delta^{\mathcal{I}} \cup \{\uparrow\} \\ \mathsf{a} & \mapsto & \mathsf{a} \\ w \cdot \mathrm{R.A} & \mapsto & \begin{cases} \uparrow & \textit{if } f(w) = \uparrow \textit{ or } f(w) \notin (\exists \mathrm{R.A})^{\mathcal{I}} \\ \mathsf{succ}^{\mathcal{I}}_{\mathrm{R.A}}(f(w)) & \textit{otherwise} \end{cases} \end{array}$$

where $\uparrow$ *is a fresh symbol witnessing the absence of a proper image for an element of* $\mathsf{Ind}(\mathcal{A}) \cdot \Omega^*$*. The* existential extraction *of* $\mathcal{I}$ *is* $\Delta^\circ := \{w \mid w \in \mathsf{Ind}(\mathcal{A}) \cdot \Omega^*, f(w) \neq \uparrow\}$*. Slightly abusing the notation, the mapping* $f_{|\Delta^\circ} : \Delta^\circ \to \Delta^{\mathcal{I}}$ *is also denoted* f *for readability.*

Definition 78. *The* f'-interlacing $\mathcal{I}'$ *of* $\mathcal{I}$ *is the interpretation whose domain is* $\Delta^{\mathcal{I}'} := f'(\Delta^\circ)$ *and which interprets concept and role names as follows:*

$$\begin{array}{rcll} \mathrm{A}^{\mathcal{I}'} & := & \{f'(u) \mid u \in \Delta^\circ, f(u) \in \mathrm{A}^{\mathcal{I}}\} & \\ \mathrm{P}^{\mathcal{I}'} & := & \{(\mathsf{a}, \mathsf{b}) \mid \mathsf{a}, \mathsf{b} \in \mathsf{Ind}(\mathcal{A}) \wedge \mathcal{K} \models \mathrm{P}(\mathsf{a}, \mathsf{b})\} & (\nabla_0) \\ & \cup & \{(f'(u), f'(u \cdot \mathrm{R.B})) \mid u, u \cdot \mathrm{R.B} \in \Delta^\circ \wedge \mathcal{T} \models \mathrm{R} \sqsubseteq \mathrm{P}\} & (\nabla_+) \\ & \cup & \{(f'(u \cdot \mathrm{R.B}), f'(u)) \mid u, u \cdot \mathrm{R.B} \in \Delta^\circ \wedge \mathcal{T} \models \mathrm{R}^- \sqsubseteq \mathrm{P}\} & (\nabla_-) \end{array}$$

Theorem 60. *If $f' : \Delta^\circ \to E$ is pseudo-injective, then $\mathcal{I}'$ is a model of $\mathcal{K}$ and the following mapping is a homomorphism from $\mathcal{I}'$ to $\mathcal{I}$:*

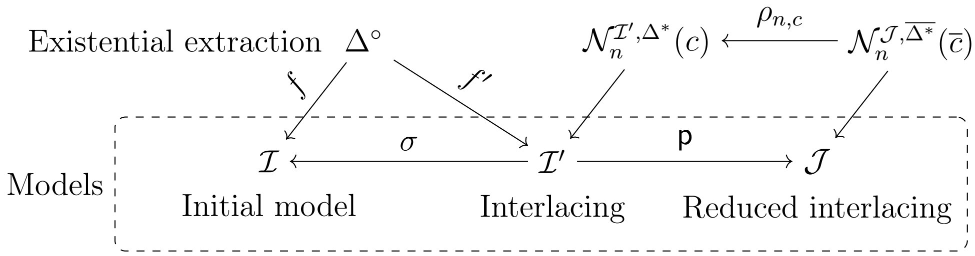
$$\begin{array}{rccc} \sigma : & \Delta^{\mathcal{I}'} & \to & \Delta^{\mathcal{I}} \\ & f'(u) & \mapsto & f(u) \end{array}$$

Notice that f' being pseudo-injective ensures σ is indeed well-defined.

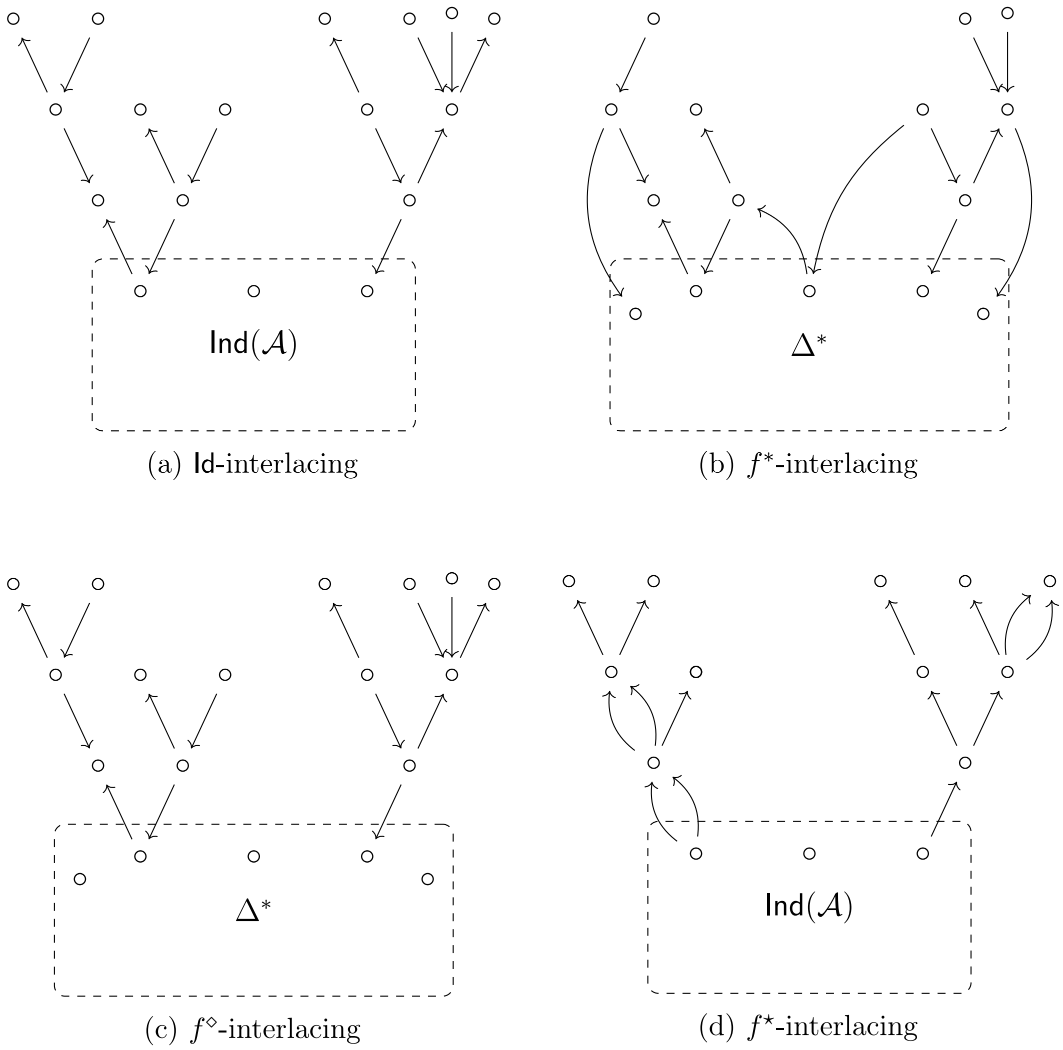
We also recall the domain $\Delta^* \subseteq \Delta^{\mathcal{I}}$ is defined as:

$$\Delta^* := \mathsf{Ind}(\mathcal{A}) \cup \bigcup_{\substack{\pi : q \to \mathcal{I} \\ \text{match}}} \pi(\mathbf{z}).$$

The following figure, borrowed from Section 3.4.1, summarizes the relations between the above constructions.



Finally, the four flavors of interlacings are obtained by setting f' to one of the functions Id, f^*, $f^\diamond$ or $f^\star$, respectively defined in Remark 8 and Definitions 23, 42 and 44 The intuitive shape of each obtained interlacing are depicted in Figure B.1.



(a) Id-interlacing (b) f^*-interlacing (c) $f^\diamond$-interlacing (d) $f^\star$-interlacing

Figure B.1: Intuition of the underlying structure for each type of interlacing.

Requêtes de comptage pour l'accès aux données en présence d'ontologies

Résumé : La réponse à des requêtes en présence d'ontologies est une approche prometteuse pour l'intégration et l'accès aux données qui a été activement étudiée ces quinze dernières années. La grande majorité des travaux dans ce domaine se concentre sur les requêtes conjonctives, alors que des requêtes plus expressives, qui offrent des fonctionnalités de comptage ou d'autres formes d'agrégation demeurent largement inexplorées. Dans cette thèse, nous introduisons une forme unifiée de requêtes de comptage, nous la relions à celles déjà existantes, et étudions la complexité du problème consistant à répondre à ces requêtes en présence d'ontologies exprimées dans la logique de description $\mathcal{ALCHI}$ ou l'une de ses sous-logiques. Dans la mesure où la complexité de ce problème dans le cas général est inaccessible en pratique et parfois très élevée sur de telles ontologies, nous considérons également deux restrictions sur ces requêtes: l'enracinement et l'atomicité, pour lesquelles nous établissons de meilleurs résultats en terme de complexité.
Mots-clés : Accès aux données en présence d'ontologie, Logiques de description, Requêtes de comptage, Complexité du raisonnement

---

Counting queries in ontology-based data access

Abstract: Ontology-mediated query answering (OMQA) is a promising approach to data access and integration that has been actively studied in the knowledge representation and database communities for more than a decade. The vast majority of work on OMQA focuses on conjunctive queries, whereas more expressive queries that feature counting or other forms of aggregation remain largely unexplored. In this thesis, we introduce a general form of counting conjunctive query (CCQ), relate it to previous proposals, and study the complexity of answering such queries in the presence of ontologies expressed in the description logic $\mathcal{ALCHI}$ or its sublogics. As the general case of CCQ answering is intractable and often of high complexity over such ontologies, we consider two practically relevant restrictions, namely rooted CCQs and Boolean atomic CCQs, for which we establish improved complexity bounds.
Keywords: Ontology-mediated query answering, Description Logics, Counting query, Complexity of reasoning

---

Laboratoire Bordelais de Recherche en informatique (LaBRI)
UMR 5800, Université de Bordeaux, 33000 Bordeaux, France.