Applications and Extensions of PTIME Description Logics with Functional Constraints*

David Toman and **Grant Weddell**
D. R. Cheriton School of Computer Science, University of Waterloo, Canada

Abstract

We review and extend earlier work on the logic $\mathcal{CFD}$, a description logic that allows terminological cycles with universal restrictions over functional roles. In particular, we consider the problem of reasoning about concept subsumption and the problem of computing certain answers for a family of attribute-connected conjunctive queries, showing that both problems are in PTIME. We then consider the effect on the complexity of these problems after adding a concept constructor that expresses concept union, or after adding a concept constructor for the bottom class. Finally, we show that adding both constructors makes both problems EXPTIME-complete.

1 Introduction

The need to provide better guarantees on performance has prompted a resurgence of interest in ontology languages for which various reasoning tasks have PTIME complexity. The latest version of the OWL web ontology standard [W3C, 2008] reflects this trend by including so-called *profiles*. Notably, two of these are based on the description logics (DLs) $\mathcal{EL}$++ [Baader *et al.*, 2005] and DL-Lite [Calvanese *et al.*, 2007], each of which has PTIME complexity for the fundamental task of deciding *concept subsumption*. In this paper, we consider a third DL with PTIME complexity for this problem, called $\mathcal{CFD}$ [Khizder *et al.*, 2000], that can also serve as the basis for an OWL profile.

$\mathcal{CFD}$ is designed for ontologies that require an ability to express universal restrictions over functional roles, that have terminological cycles and that also require more general ways of expressing identification constraints such as composite keys and functional dependencies. This combination of features makes $\mathcal{CFD}$ another option to DL-Lite for integrating ontological knowledge about formatted data sources such as those conforming to underlying relational database schemata, a very common circumstance for many web applications. To illustrate, one can say the following about a hypothetical human resources data source in $\mathcal{CFD}$:

$$\begin{aligned}
&\forall x : \mathrm{EMP}(x) \rightarrow \mathrm{BOSS}(Sup(x)),\\
&\forall x : \mathrm{EMP}(x) \rightarrow \forall y : \mathrm{EMP}(y) \rightarrow\\
&\quad (Ct(x) = Ct(y) \wedge Pn(x) = Pn(y)) \rightarrow x = y, \qquad (1)\\
&\forall x : \mathrm{BOSS}(x) \rightarrow \mathrm{EMP}(x) \text{ and}\\
&\forall x : \mathrm{BOSS}(x) \rightarrow \mathrm{DIRECTOR}(Sup(x)).
\end{aligned}$$

The statements assert that *an employee is supervised by a boss and has a unique phone number within a given city*, and also that *a boss is an employee that is more specifically supervised by a director*. In contrast, $\mathcal{EL}$++ is not able to capture (abstract) functionality and functional restrictions such as keys, the first two statements for example, and DL-Lite is not able to capture additional conditions on attributes or roles that might be satisfied by subclasses, as in the case of the last statement about the supervision of bosses. However, both $\mathcal{EL}$++ and DL-Lite are able to capture disjointness conditions which is not possible in $\mathcal{CFD}$, as we show. Also note that Calvanese et al. [Calvanese *et al.*, 2008] have recently extended DL-Lite with a path-based variety of identification constraints. Consequently, DL-Lite is now able to capture the second statement in (1).

We review and extend the earlier work on $\mathcal{CFD}$, first considering the problem of reasoning about concept subsumption with respect to a $\mathcal{CFD}$ TBox, and then considering the problem of computing certain answers for a large subclass of conjunctive queries with respect to a $\mathcal{CFD}$ knowledge base consisting of both a TBox and ABox. In both cases, we consider the effect on the complexity of these problems with respect to the addition of concept constructors for expressing concept union and the bottom (unsatisfiable) class. In particular, in Sections 2 and 3, we show the following for $\mathcal{CFD}$ (with a TBox that may contain terminological cycles).

1. The problem of determining if a particular concept subsumption, called a *posed question*, is logically implied by the TBox is in PTIME. Our proof is based on a reduction of the problem to a Horn formulation over a finite universe, a much simpler approach than the procedure outline in [Khizder *et al.*, 2000], and one that enables a more transparent consideration of extensions to $\mathcal{CFD}$ that follow.
2. This problem is coNP-complete if one adds a concept constructor that expresses concept union to $\mathcal{CFD}$.
3. The problem is PSPACE-complete if, alternatively, one adds a concept for the bottom class.

*Preliminary version appeared in [Toman and Weddell, 2008].

4. The problem is ultimately EXPTIME-complete if one adds concept constructors for both concept union and the bottom class.

We also consider extending the expressiveness of posed questions to allow primitive negation, disjunction, general negation and inverse features, and show the effect of these extensions on the complexity of the implication problem, in particular when measured only in the size of the posed question.

We then consider the problem of computing certain answers to a family of attribute-connected conjunctive queries over an arbitrary $\mathcal{CFD}$ knowledge base. Such queries are more general than those that can be folded into a single concept description in more typical DLs and include, for example, queries with existential restrictions corresponding to foreign-key joins. We show how *all* of the above complexity bounds for the concept subsumption problem for $\mathcal{CFD}$ and its extensions transfer to this second problem.

The remainder of this section introduces $\mathcal{CFD}$, in particular, the definitions related to the logical implication problem for inclusion dependencies that are needed for an understanding of the results in Sections 2 and 3. Additional definitions concerning ABox reasoning are left to the start of Section 4 in which we consider certain answer computation. We conclude in Section 5 and 6 with a discussion of related work and with summary comments.

1.1 The Description Logic $\mathcal{CFD}$

A formal definition of the syntax and semantics and of the logical implication problem of $\mathcal{CFD}$ follows. Note that $\mathcal{CFD}$ itself is based on *attributes* (also called *features*) instead of the more common case of *roles* (which are easily accommodated by reification [Toman and Weddell, 2005a]).

SYNTAX	SEMANTICS: "$(\cdot)^{\mathcal{I}}$"
$C ::= A$	
$\mid C_1 \sqcap C_2$	$(C_1)^{\mathcal{I}} \cap (C_2)^{\mathcal{I}}$
$D ::= C$	
$\mid D_1 \sqcap D_2$	$(D_1)^{\mathcal{I}} \cap (D_2)^{\mathcal{I}}$
$\mid \forall \mathsf{Pf}.C$	$\{x : (\mathsf{Pf})^{\mathcal{I}}(x) \in (C)^{\mathcal{I}}\}$
$\mid C : \mathsf{Pf}_1, ..., \mathsf{Pf}_k \rightarrow \mathsf{Pf}$	$\{x : \forall y \in (C)^{\mathcal{I}}.$
	$\bigwedge_{i=1}^{k} (\mathsf{Pf}_i)^{\mathcal{I}}(x) = (\mathsf{Pf}_i)^{\mathcal{I}}(y) \Rightarrow (\mathsf{Pf})^{\mathcal{I}}(x) = (\mathsf{Pf})^{\mathcal{I}}(y)\}$
$E ::= C$	
$\mid \bot$	$\emptyset$
$\mid E_1 \sqcap E_2$	$(E_1)^{\mathcal{I}} \cap (E_2)^{\mathcal{I}}$
$\mid \forall \mathsf{Pf}.E$	$\{x : (\mathsf{Pf})^{\mathcal{I}}(x) \in (E)^{\mathcal{I}}\}$
$\mid (\mathsf{Pf}_1 = \mathsf{Pf}_2)$	$\{x : (\mathsf{Pf}_1)^{\mathcal{I}}(x) = (\mathsf{Pf}_2)^{\mathcal{I}}(x)\}$

Figure 1: SYNTAX AND SEMANTICS OF $\mathcal{CFD}$.

Definition 1 (The logic $\mathcal{CFD}$) *Let* F *and* A *be disjoint sets of (names of) attributes and primitive concepts, respectively. A* path expression Pf *is a word in* F^* *with the usual convention that the empty word is denoted by* Id *and concatenation by "." .* Concept descriptions *are defined by the grammar on the left-hand-side of Figure 1. A concept produced by the "*$\mathsf{C} : \mathsf{Pf}_1, \ldots, \mathsf{Pf}_k \rightarrow \mathsf{Pf}$*" production of this grammar is called a* path functional dependency *(PFD). In addition, any occurrence of a PFD must adhere to one of the following two forms:*

$$\begin{array}{ll} 1. & C : \mathsf{Pf}_1, \ldots, \mathsf{Pf} \,.\, \mathsf{Pf}_i, \ldots, \mathsf{Pf}_k \rightarrow \mathsf{Pf} \;\; \textit{or} \\ 2. & C : \mathsf{Pf}_1, \ldots, \mathsf{Pf} \,.\, \mathsf{Pf}_i, \ldots, \mathsf{Pf}_k \rightarrow \mathsf{Pf} \,.f \end{array} \tag{2}$$

An inclusion dependency $\mathcal{C}$ *is an expression of the form* $C \sqsubseteq D$. *(Note the distinction of concept descriptions that can appear on the left- and right-hand sides of* $\mathcal{C}$*.) A* terminology (TBox) $\mathcal{T}$ *consists of a finite set of inclusion dependencies. A* posed question $\mathcal{Q}$ *is an expression of the form* $E_1 \sqsubseteq E_2$.

The semantics *of expressions is defined with respect to a structure* $(\triangle, \cdot^{\mathcal{I}})$, *where* $\triangle$ *is a domain of "objects" and* $(.)^{\mathcal{I}}$ *an interpretation function that fixes the interpretation of primitive concepts* A *to be subsets of* $\triangle$ *and attributes* f *to be total functions* $(f)^{\mathcal{I}} : \triangle \rightarrow \triangle$. *The interpretation is extended to path expressions by interpreting the empty word* (Id) *as the identity function* $\lambda x.x$, *concatenation as function composition, and to derived concept descriptions* C, D *and* E *as defined on the right-hand-side of Figure 1.*

An interpretation satisfies an inclusion dependency $C \sqsubseteq D$ *(resp.* a posed question $E_1 \sqsubseteq E_2$*) if* $(C)^{\mathcal{I}} \subseteq (D)^{\mathcal{I}}$ *(resp.* $(E_1)^{\mathcal{I}} \subseteq (E_2)^{\mathcal{I}}$*).*

The logical implication problem *asks if* $\mathcal{T} \models \mathcal{Q}$ *holds; that is, for a* posed question $\mathcal{Q}$*, if* $\mathcal{Q}$ *is satisfied by any interpretation that satisfies all inclusion dependencies in* $\mathcal{T}$.

The conditions imposed on PFDs in (2) distinguish, for example, PFDs of the form $C : f \rightarrow Id$ and $C : f \rightarrow g$ from PFDs of the form $C : f \rightarrow g.h$. The strategic advantage of these conditions is that chase procedures can employ a simple saturation strategy for PFDs that will "fire" them unconditionally, a process that would otherwise not terminate. Indeed, we rely on this in presenting a PTIME procedure for the implication problem in the next section.

Note that the conditions are still satisfied by PFDs that correspond to arbitrary keys or functional dependencies in the sense of the relational model. Indeed, we have found that the conditions do not appear to hinder any modeling utility of $\mathcal{CFD}$ for ontologies that occur in practice. Moreover, relaxing the boundary conditions, such as allowing PFDs of the form $C : f \rightarrow g.f$, already leads to undecidability [Toman and Weddell, 2006].

To illustrate, consider our introductory example relating to a hypothetical human resources data source. The statements in (1) can be captured in $\mathcal{CFD}$ as follows:

$$\begin{array}{l} \text{EMP} \sqsubseteq \forall Sup.\text{BOSS}, \\ \text{EMP} \sqsubseteq \text{EMP} : Ct, Pn \rightarrow Id, \\ \text{BOSS} \sqsubseteq \text{EMP and} \\ \text{BOSS} \sqsubseteq \forall Sup.\text{DIRECTOR}. \end{array} \tag{3}$$

In particular, the ith statement in (1) is captured by the ith inclusion dependency in (3) above.

There are two final things to note about the grammar. First, in presenting our PTIME procedure in the next section and again when considering the addition of the bottom class to $\mathcal{CFD}$ in Section 3, we presume each inclusion dependency in a TBox $\mathcal{T}$ has the form $\mathrm{A}_1 \sqcap \cdots \sqcap \mathrm{A}_k \sqsubseteq \mathrm{D}$ for A_i a primitive

concept. Second, although the grammar does not allow a PFD to appear directly in a posed question, the following lemma establishes that an indirect check for the logical implication of such concepts can be easily simulated.

Lemma 2 *Let $\mathcal{T}$ be a $\mathcal{CFD}$ terminology and $\mathrm{E}_1 \sqsubseteq \mathrm{E}_2 : \mathsf{Pf}_1, \ldots, \mathsf{Pf}_k \rightarrow \mathsf{Pf}$ be a posed question in an extension of $\mathcal{CFD}$ (that allows such a construct in posed questions). Then there is a posed question $\mathcal{Q}$ in $\mathcal{CFD}$ such that $\mathcal{T} \models \mathrm{E}_1 \sqsubseteq \mathrm{E}_2 : \mathsf{Pf}_1, \ldots, \mathsf{Pf}_k \rightarrow \mathsf{Pf}$ iff $\mathcal{T} \models \mathcal{Q}$.*

Proof (sketch): Define $\mathcal{Q}$ to be the subsumption

$$\forall f.\mathrm{E}_1 \sqcap \forall g.\mathrm{E}_2 \sqcap (\prod_{i=1}^{k}(f.\mathsf{Pf}_i = g.\mathsf{Pf}_i)) \sqsubseteq (f.\mathsf{Pf} = g.\mathsf{Pf})$$

where f and g are attributes not occurring in $\mathcal{T}$. □

2 The Polynomial Case

We first show that the logical implication problem for $\mathcal{CFD}$ is in PTIME. Our proof is based on encoding a given problem as a collection of Horn clauses. The reduction introduces terms that correspond to path expressions, and relies on the fact that the number of required terms is polynomial in the size of the problem itself.

Definition 3 (Expansion Rules) *Let $\mathcal{T}$ and $\mathcal{Q}$ be a $\mathcal{CFD}$ terminology and a posed question, respectively. We write $\mathsf{CON}(\mathcal{T}, \mathcal{Q})$ to denote the set of all subconcepts appearing in $\mathcal{T}$ and $\mathcal{Q}$, define $\mathsf{PF}(\mathcal{T}, \mathcal{Q})$ as*

$$\{\mathsf{Pf}\,.\,\mathsf{Pf}' \mid \mathsf{Pf} \text{ is a prefix of a path expression in } \mathcal{Q} \text{ and } \mathsf{Pf}' \text{ is a path expression in } \mathcal{T} \text{ or } Id\},$$

write C_C to denote unary predicates for $\mathrm{C} \in \mathsf{CON}(\mathcal{T}, \mathcal{Q})$, and introduce the binary predicate E, with all predicates ranging over the universe $\mathsf{PF}(\mathcal{T}, \mathcal{Q})$. The expansion rules *for a given terminology $\mathcal{T}$, denoted $\mathsf{R}(\mathcal{T})$, are defined in Figure 2.*

$\mathsf{E}(\mathsf{Pf}_1, \mathsf{Pf}_1)$
$\mathsf{E}(\mathsf{Pf}_1, \mathsf{Pf}_2) \rightarrow \mathsf{E}(\mathsf{Pf}_2, \mathsf{Pf}_1)$
$\mathsf{E}(\mathsf{Pf}_1, \mathsf{Pf}_2) \wedge \mathsf{E}(\mathsf{Pf}_2, \mathsf{Pf}_3) \rightarrow \mathsf{E}(\mathsf{Pf}_1, \mathsf{Pf}_3)$
$\mathsf{E}(\mathsf{Pf}_1, \mathsf{Pf}_2) \rightarrow \mathsf{E}(\mathsf{Pf}_1\,.\,\mathsf{Pf}, \mathsf{Pf}_2\,.\,\mathsf{Pf})$ for all $\{\mathsf{Pf}_1\,.\,\mathsf{Pf}, \mathsf{Pf}_2\,.\,\mathsf{Pf}\} \subseteq \mathsf{PF}(\mathcal{T}, \mathcal{Q})$
$\mathsf{E}(\mathsf{Pf}_1, \mathsf{Pf}_2) \wedge \mathsf{C}_\mathrm{C}(\mathsf{Pf}_1) \rightarrow \mathsf{C}_\mathrm{C}(\mathsf{Pf}_2)$

$\mathsf{C}_{\mathrm{C}_1 \sqcap \mathrm{C}_2}(\mathsf{Pf}) \rightarrow \mathsf{C}_{\mathrm{C}_1}(\mathsf{Pf})$ and $\mathsf{C}_{\mathrm{C}_1 \sqcap \mathrm{C}_2}(\mathsf{Pf}) \rightarrow \mathsf{C}_{\mathrm{C}_2}(\mathsf{Pf})$
$\mathsf{C}_{\forall\,\mathsf{Pf}'\,.\,\mathrm{C}}(\mathsf{Pf}) \rightarrow \mathsf{C}_\mathrm{C}(\mathsf{Pf}\,.\,\mathsf{Pf}')$ for $\mathsf{Pf}\,.\,\mathsf{Pf}' \in \mathsf{PF}(\mathcal{T}, \mathcal{Q})$
$\mathsf{C}_{(\mathsf{Pf}_1 = \mathsf{Pf}_2)}(\mathsf{Pf}) \rightarrow \mathsf{E}(\mathsf{Pf}\,.\,\mathsf{Pf}_1, \mathsf{Pf}\,.\,\mathsf{Pf}_2)$
$\mathsf{C}_{\mathrm{C}:\mathsf{Pf}_1,\ldots,\mathsf{Pf}_k \rightarrow \mathsf{Pf}_0}(\mathsf{Pf}) \wedge \mathsf{C}_\mathrm{C}(\mathsf{Pf}') \wedge (\bigwedge_{0<i\leq k} \mathsf{E}(\mathsf{Pf}\,.\,\mathsf{Pf}_i, \mathsf{Pf}'\,.\,\mathsf{Pf}_i)) \rightarrow \mathsf{E}(\mathsf{Pf}\,.\,\mathsf{Pf}_0, \mathsf{Pf}'\,.\,\mathsf{Pf}_0)$

$\mathsf{C}_{\mathrm{A}_1}(\mathsf{Pf}) \wedge \ldots \wedge \mathsf{C}_{\mathrm{A}_k}(\mathsf{Pf}) \rightarrow \mathsf{C}_\mathrm{D}(\mathsf{Pf})$ for all $(\mathrm{A}_1 \sqcap \ldots \sqcap \mathrm{A}_k \sqsubseteq \mathrm{D}) \in \mathcal{T}$

Figure 2: EXPANSION RULES.

A goal *for each concept* E *is a set of ground assertions defined as follows:*

$$\mathsf{G}_\mathrm{E} = \begin{cases} \{\mathsf{C}_\mathrm{A}(Id)\} & \text{for } \mathrm{E} = \mathrm{A} \\ \{\mathsf{C}_\bot(Id)\} & \text{for } \mathrm{E} = \bot \\ \{\mathsf{E}(\mathsf{Pf}_1, \mathsf{Pf}_2)\} & \text{for } \mathrm{E} = (\mathsf{Pf}_1 = \mathsf{Pf}_2) \\ \mathsf{G}_{\mathrm{E}_1} \cup \mathsf{G}_{\mathrm{E}_2} & \text{for } \mathrm{E} = \mathrm{E}_1 \sqcap \mathrm{E}_2 \\ \{\mathsf{E}(\mathsf{Pf}'\,\mathsf{Pf}_1, \mathsf{Pf}'\,.\,\mathsf{Pf}_2) \mid \mathsf{E}(\mathsf{Pf}_1, \mathsf{Pf}_2) \in \mathsf{G}_{\mathrm{E}'}\} \cup & \\ \quad \{\mathsf{C}_\mathrm{C}(\mathsf{Pf}'\,.\,\mathsf{Pf}) \mid \mathsf{C}_\mathrm{C}(\mathsf{Pf}) \in \mathsf{G}_{\mathrm{E}'}\} & \text{for } \mathrm{E} = \forall\,\mathsf{Pf}'\,.\mathrm{E}' \end{cases}$$

Given two concept descriptions E_1 *and* E_2*, we say that*

$$\mathsf{R}(\mathcal{T}) \cup \{\mathsf{C}_{\mathrm{E}_1}(Id)\} \models \mathsf{G}_{\mathrm{E}_2}$$

if $\mathsf{G}_{\mathrm{E}_2} \subseteq M$ for every ground model M of $\mathsf{R}(\mathcal{T})$ over $\mathsf{PF}(\mathcal{T}, \mathcal{Q})$ that contains $\mathsf{C}_{\mathrm{E}_1}(Id)$.

Intuitively, $\mathsf{PF}(\mathcal{T}, \mathcal{Q})$ represents a finite graph of objects, predicates $\mathsf{E}(\mathsf{Pf}_1, \mathsf{Pf}_2)$ express equality of the objects at the end of paths Pf_1 and Pf_2, and predicates $\mathsf{C}_{C'}(\mathsf{Pf})$ express that the object at the end of path Pf is in the interpretation of concept C'.

Our PTIME result for the $\mathcal{CFD}$ implication problem follows by a simple check for goals occurring in a ground model for expansion rules generated by a polynomial sized collection of path expressions.

Theorem 4 *Let $\mathcal{T}$ be a $\mathcal{CFD}$ terminology and $\mathcal{Q}$ a posed question of the form $\mathrm{E}_1 \sqsubseteq \mathrm{E}_2$. Then*

$$\mathcal{T} \models \mathcal{Q} \text{ iff } \mathsf{R}(\mathcal{T}) \cup \{\mathsf{C}_{\mathrm{E}_1}(Id)\} \models \mathsf{G}_{\mathrm{E}_2} \text{ or } \mathsf{R}(\mathcal{T}) \cup \{\mathsf{C}_{\mathrm{E}_1}(Id)\} \models \mathsf{C}_\bot(\mathsf{Pf}) \text{ for } \mathsf{Pf} \in \mathsf{PF}(\mathcal{T}, \mathcal{Q}).$$

Proof (sketch): The least model of $\mathsf{R}(\mathcal{T}) \cup \{\mathsf{C}_{\mathrm{E}_1}(Id)\}$, if one exists, can be extended to a model of $\mathcal{T}$. Since the original model is the least model, it suffices to determine whether E_2 subsumes E_1 in (the extension of) this model. □

Since the expansion rules are Horn clauses over a finite universe $\mathsf{PF}(\mathcal{T}, \mathcal{Q})$ of polynomial size, we have the following:

Corollary 5 *Let $\mathcal{T}$ and $\mathcal{Q}$ denote a terminology and posed question in $\mathcal{CFD}$. Then the implication problem $\mathcal{T} \models \mathcal{Q}$ is decidable in PTIME.*

Proof (sketch): The least model of $\mathsf{R}(\mathcal{T}) \cup \{\mathsf{C}_{\mathrm{E}_1}(Id)\}$ can be obtained by using a bottom-up construction of the least fix-point of the rules in time polynomial in $|\mathcal{T}| + |\mathcal{Q}|$. □

In practice, elements of this set can be constructed on demand by using additional Horn rules in such a way that only path expressions needed to confirm subsumption or non-subsumption are generated.

3 Intractable Extensions

We now consider the consequences of the various extensions to $\mathcal{CFD}$ outlined in our introductory comments. The extensions add options for the right-hand-sides of inclusion dependencies and are illustrated in Figure 3. We begin by considering the case for concept union. Subsection 3.2 then considers an alternative addition of a constructor for the bottom class, and Subsection 3.3 considers the addition of both constructors. Finally, in Subsection 3.4, we consider consequences of various extensions to a posed question.

SYNTAX	SEMANTICS: "$(\cdot)^{\mathcal{I}}$"	
$D ::= ...$		(any production of $\mathcal{CFD}$)
$\mid D_1 \sqcup D_2$	$(D_1)^{\mathcal{I}} \cup (D_2)^{\mathcal{I}}$	(in $\mathcal{CFD}^{\sqcup}$ and $\mathcal{CFD}^{\sqcup,\bot}$)
$\mid \bot$	$\emptyset$	(in $\mathcal{CFD}^{\bot}$ and $\mathcal{CFD}^{\sqcup,\bot}$)

Figure 3: EXTENSIONS FOR $\mathcal{CFD}$ TERMINOLOGIES.

3.1 The Coverage Case: $\mathcal{CFD}^{\sqcup}$

To handle disjunction in right-hand sides of inclusion dependencies, we extend the expansion rules with "or" rules as follows:

$$\mathsf{R}^{\sqcup}(\mathcal{T}) := \mathsf{R}(\mathcal{T}) \cup \{\mathsf{C}_{\mathrm{C}_1 \sqcup \mathrm{C}_2}(\mathsf{Pf}) \rightarrow \mathsf{C}_{\mathrm{C}_1}(\mathsf{Pf}) \text{ or } \mathsf{C}_{\mathrm{C}_2}(\mathsf{Pf})\}.$$

This rule is added to the expansion rules used for the polynomial case. The intuition behind using this rule is to nondeterministically find a model with an object that belongs to the left-hand side of the posed question but which does not belong to the right-hand side.

Theorem 6 *Let $\mathcal{T}$ be a $\mathcal{CFD}^{\sqcup}$ terminology and $\mathcal{Q}$ a posed question of the form $\mathrm{E}_1 \sqsubseteq \mathrm{E}_2$. Then*

$$\mathcal{T} \models \mathcal{Q} \text{ iff } \mathsf{G}_{\mathrm{E}_2} \subseteq M \text{ or } \mathsf{C}_{\bot}(\mathsf{Pf}) \in M \text{ for } \mathsf{Pf} \in \mathsf{PF}(\mathcal{T}, \mathcal{Q})$$

in each minimal model M of $\mathsf{R}^{\sqcup}(\mathcal{T}) \cup \{\mathsf{C}_{\mathrm{E}_1}(Id)\}$.

Proof (sketch): The proof is similar to the proof of Theorem 4 augmented with a non-deterministic application of the above rule. Thus non-subsumption can be detected using polynomially many non-deterministic guesses. □

Corollary 7 *Let $\mathcal{T}$ and $\mathcal{Q}$ denote a terminology and posed question in $\mathcal{CFD}^{\sqcup}$. Then the implication problem $\mathcal{T} \models \mathcal{Q}$ is decidable and coNP-complete.*

Proof (sketch): To obtain the upper bound, simply apply rules of the form "$\mathsf{C}_{\mathrm{C}_1 \sqcup \mathrm{C}_2}(\mathsf{Pf}) \rightarrow \mathsf{C}_{\mathrm{C}_1}(\mathsf{Pf})$ or $\mathsf{C}_{\mathrm{C}_2}(\mathsf{Pf})$" nondeterministically during the construction of a minimal (but not necessarily unique) model of $\mathsf{R}^{\sqcup}(\mathcal{T}) \cup \{\mathsf{C}_{\mathrm{E}_1}(Id)\}$ that yields an counterexample to $\mathcal{T} \models \mathcal{Q}$; a construction similar to Theorem 4. Non-implication can be detected in a polynomial number of guesses (is in NP). To obtain the lower bound, we reduce 2+2-SAT to non-implication $\mathcal{T} \not\models \mathcal{Q}$ in $\mathcal{CFD}^{\sqcup}$. □

3.2 The Disjointness Case: $\mathcal{CFD}^{\bot}$

We now show that the logical implication problem for $\mathcal{CFD}$ becomes PSPACE-complete if the bottom class is allowed in inclusion dependencies. This is also accomplished by appealing to our construction in Section 2, expanding in particular on the issue of "completing" the construction of counterexample interpretations.

Definition 8 (Satisfiable Atomic Type) *Let T denote a finite set of primitive concepts. An* atomic type *over T is the concept $\sqcap_{A \in T} A$. We say that T is satisfiable with respect to a $\mathcal{CFD}^{\bot}$ terminology $\mathcal{T}$ if there is an interpretation $\mathcal{I}_T$ such that $\mathcal{I}_T \models \mathcal{T}$ and $(\sqcap_{A \in T} A)^{\mathcal{I}} \neq \emptyset$.*

Lemma 9 *Let $\mathcal{T}$ be a $\mathcal{CFD}^{\bot}$ terminology and T a finite set of primitive concepts. The problem of determining if T is satisfiable with respect to $\mathcal{T}$ is decidable and in PSPACE.*

Note that it is easy to construct terminologies (using only $\sqcap$ and $\forall f.$) in which a contradiction is found only after exponentially many steps (of f). We use a reachability search algorithm to detect these situations. Since the graph of all atomic types T is exponential in the size of the terminology, the reachability of an unsatisfiable atomic type starting from T can be determined in PSPACE.

Theorem 10 *Let $\mathcal{T}$ be a $\mathcal{CFD}^{\bot}$ terminology and $\mathcal{Q}$ a posed question of the form $\mathrm{E}_1 \sqsubseteq \mathrm{E}_2$. Then $\mathcal{T} \models \mathcal{Q}$ iff either*

1. *$\mathsf{C}_{\bot}(\mathsf{Pf}) \in M$ for some $\mathsf{Pf} \in \mathsf{PF}(\mathcal{T}, \mathcal{Q})$, or*
2. *the set $\{\mathrm{A} \in \mathsf{CON}(\mathcal{T}, \mathcal{Q}) \mid \mathsf{C}_{\mathrm{A}}(\mathsf{Pf}) \in M, \mathrm{A} \text{ primitive}\}$ is unsatisfiable w.r.t. $\mathcal{T}$ for some $\mathsf{Pf} \in \mathsf{PF}(\mathcal{T}, \mathcal{Q})$, or*
3. *$\mathsf{G}_{\mathrm{E}_2} \subseteq M$*

holds in the minimal model M of $\mathsf{R}(\mathcal{T}) \cup \{\mathsf{C}_{\mathrm{E}_1}(Id)\}$.

Proof (sketch): The construction of a counterexample follows the construction in the proof of Theorem 4. However, the completion of the interpretation by infinite trees now depends on the existence of such trees and must be checked using Lemma 9. Note that this also relies on the fact that any interpretation $\mathcal{I}_T$ that exists for an atomic type T can be made into a tree interpretation. □

Corollary 11 *Let $\mathcal{T}$ and $\mathcal{Q}$ denote a terminology and posed question in $\mathcal{CFD}^{\bot}$. Then the implication problem $\mathcal{T} \models \mathcal{Q}$ is decidable and PSPACE-complete.*

Proof (sketch): To show hardness for PSPACE, we reduce the (deterministic) *Linear Bounded Automaton Acceptance* [Garey and Johnson, 1979; Karp, 1972] to the implication problem in $\mathcal{CFD}^{\bot}$. □

3.3 The Boolean Complete Case: $\mathcal{CFD}^{\sqcup,\bot}$

We now show that adding both disjointness and coverage constraints leads to EXPTIME completeness. In particular, we show how to simulate axioms of the form

$$\forall f_1.\mathrm{A}_1 \sqcap \forall f_2.\mathrm{A}_2 \sqsubseteq \forall f_3.\mathrm{A}_3. \tag{4}$$

Consider the following auxiliary inclusion dependencies relating to primitive concepts occurring in a logical implication problem for $\mathcal{CFD}$:

- Define T as a top-replacement concept and pairs of A_i and $\bar{\mathrm{A}}_i$ as disjoint partitions of the simulated top (and similarly for B and $\bar{\mathrm{B}}$):

$$\mathrm{A}_i \sqcap \bar{\mathrm{A}}_i \sqsubseteq \bot, \quad \mathrm{T} \sqsubseteq \mathrm{A}_i \sqcup \bar{\mathrm{A}}_i, \quad \mathrm{A}_i \sqsubseteq \mathrm{T}, \quad \bar{\mathrm{A}}_i \sqsubseteq \mathrm{T}.$$

- Propagate the top-replacement concept across all attributes:

$$\mathrm{T} \sqsubseteq \forall f.\mathrm{T} \text{ for all } f \in \mathsf{F}.$$

Then the pair of inclusion dependencies

$$\bar{\mathrm{B}} \sqsubseteq \forall f_1.\bar{\mathrm{A}}_1 \sqcup \forall f_2.\bar{\mathrm{A}}_2 \text{ and } \mathrm{B} \sqsubseteq \forall f_3.\mathrm{A}_3$$

simulates the dependency (4) above, and, as a consequence, the decision problem is EXPTIME-hard [Toman and Weddell, 2005a]. To obtain a matching upper bound, we note that $\mathcal{CFD}^{\sqcup,\bot}$ is a fragment of $\mathcal{DLFDE}^{-}$ that is EXPTIME-complete [Khizder *et al.*, 2007]. The latter logic is Boolean-complete and thus might be more convenient to use.

3.4 Extensions to the Posed Questions

In this final subsection, we consider several ways that the structure of *posed questions* for a $\mathcal{CFD}$ implication problem might be extended.

Primitive Inequalities and Negations: $\mathcal{CFD}_{\neq,(\neg)}$
Extending the grammar for the posed questions with primitive negations, i.e., negated primitive concepts, $\neg$A, and primitive inequalities, $(\mathsf{Pf}_1 \neq \mathsf{Pf}_2)$, still enjoys a PTIME complexity bound for the logical implication problem.

Theorem 12 *Let $\mathcal{T}$ be a $\mathcal{CFD}$ terminology and $\mathcal{Q} = \mathrm{E}_1 \sqsubseteq \mathrm{E}_2$ that allows primitive negations and/or inequalities. Then $\mathcal{T} \models \mathcal{Q}$ is decidable in PTIME.*

Proof (sketch): We replace $\mathcal{Q}$ with a (linearly-sized) set of questions that do not use negations or inequalities. The result then follows from Theorem 4. □

Similar PTIME results can be obtained for $\mathcal{CFD}^{\bot}$, i.e., when terminologies do not use disjunction and only when complexity is measured solely in $|\mathcal{Q}|$, i.e., under the *data complexity* assumption. For the extensions that allow disjunctions in the terminology, the data complexity increases to coNP:

Theorem 13 *Let $\mathcal{T}$ be a $\mathcal{CFD}^{\sqcup}$ terminology and $\mathcal{Q} = \mathrm{E}_1 \sqsubseteq \mathrm{E}_2$ that allows primitive negations and/or inequalities. Then $\mathcal{T} \models \mathcal{Q}$ is coNP complete.*

Proof (sketch): We encode a propositional DNF formula using the concept E_1, force a valuation using disjunction in $\mathcal{T}$, encode evaluation of the boolean formula (also in $\mathcal{T}$), and then test if every such valuation evaluates to true using E_2. □

Primitive Inequalities and Disjunction: $\mathcal{CFD}_{\neq,\sqcup}$
or Full Negation: $\mathcal{CFD}_{\neg}$
Adding disjunctions to posed questions with primitive negations and/or allowing general negations in posed questions leads to an increase in data complexity.

Theorem 14 *Let $\mathcal{T}$ be a $\mathcal{CFD}$ terminology and $\mathcal{Q} = \mathrm{E}_1 \sqsubseteq \mathrm{E}_2$ with disjunctions and inequalities. Then $\mathcal{T} \models \mathcal{Q}$ is decidable and coNP-complete.*

The coNP-completeness result can be lifted to the intractable extensions of $\mathcal{CFD}$ under the data complexity assumption.

Inverse Features: $\mathcal{CFD}_{inv}$
Last, we consider adding the *inverse feature* constructor, $\exists f^{-1}$.E, to the grammar for posed questions[1]. Intuitively, such an extension allows for enforcing and checking for existence of additional objects that are not reachable from the objects interpreting E_1 and E_2 in the posed questions.

To accommodate the new constructor, we extend the set $\mathsf{PF}(\mathcal{T}, \mathcal{Q})$ to accommodate the new paths that are induced by the new constructor; the extended set is defined as

$$\{\mathsf{Pf}(o) \mid \mathsf{Pf} \in \mathsf{PF}(\mathcal{T}, \mathcal{Q})\} \cup \{f_i.\,\mathsf{Pf}(a_i) \mid \mathsf{Pf} \in \mathsf{PF}(\mathcal{T}, \mathcal{Q}), \exists f_i{}^{-1}.\mathrm{E} \text{ appears in } E_1 \sqcup E_2\},$$

where o and a_i are distinct constants. Intuitively, we create a *new* constant a_i for each instance of the $\exists f^{-1}$.D constructor in the posed question. This set is still polynomial in the size of the original problem. Note also that the expansion rules naturally apply to this new set. Hence we can use similar techniques to solve the logical implication problem: we simply extend the definition of G_E to accommodate the new constructor by introducing the new constants a_i and by using equations to ensure their f_i attributes agree with the existing paths as required. However, note that allowing the $\exists f^{-1}$.D constructor in the E_2 concept of the posed question leads to an increase of complexity: to verify that E_2 is entailed by $\mathcal{T}$ and E_1, a *subgraph isomorphism* problem must be solved (one needs to match the new constants in E_2 to those that originated from E_1).

Theorem 15 *Let $\mathcal{T}$ be a $\mathcal{CFD}$ terminology and $\mathcal{Q} = \mathrm{E}_1 \sqsubseteq \mathrm{E}_2$ with inverse attributes. Then $\mathcal{T} \models \mathcal{Q}$ is decidable and NP-complete.*

The data complexity for $\mathcal{CFD}^{\bot}$ terminologies is also NP-complete (for the same reasons). For terminologies that allow the $\sqcup$ constructor, the complexity increases to Π_p^2 since matching must be tested in *all* minimal models of $\mathcal{T}$ and E_1.

The complexity results for $\mathcal{CFD}$ and its extensions are summarized in Figure 4: the superscripts denote the extensions allowed in the terminology $\mathcal{T}$ and the subscripts the extensions in the posed questions $\mathcal{Q}$.

4 Query Answering

In this section, we consider the problem of computing certain answers to a family of conjunctive queries over an arbitrary *$\mathcal{CFD}$ knowledge base* that augments a $\mathcal{CFD}$ TBox with an ABox.

Definition 16 (The *ABox consistency problem* for $\mathcal{CFD}$) *Let* N *be a set of (names of) individuals disjoint from* F *and* A. *An* ABox $\mathcal{A}$ *consists of a finite set of assertions of the form $C(a)$ or $f(a) = b$, where C is a concept description, $f \in \mathsf{F}$ and $\{a, b\} \subseteq \mathsf{N}$.*

An interpretation satisfies an ABox assertion $C(a)$ *(resp. $f(a) = b$) if $(a)^{\mathcal{I}} \in (C)^{\mathcal{I}}$ (resp. $(f)^{\mathcal{I}}((a)^{\mathcal{I}}) = (b)^{\mathcal{I}}$).*

The ABox consistency problem *asks if $\mathcal{T} \cup \mathcal{A}$ is consistent, that is, if there exists an interpretation that satisfies all inclusion dependencies in $\mathcal{T}$ and all assertions in $\mathcal{A}$.*

We write $\mathsf{Pf}(a) = b$ as shorthand for the equivalent set of primitive ABox assertions with "single use" intermediate individuals.

Proposition 17 *Let $\mathcal{T}$ be a $\mathcal{CFD}^{\sqcup,\bot}$ terminology. Then*

1. *for every ABox $\mathcal{A}$ there is a concept E such that $\mathcal{T} \cup \mathcal{A}$ is not consistent if and only if $\mathcal{T} \models E \sqsubseteq \bot$; and*
2. *for every equational concept E there is an ABox $\mathcal{A}$ such that $\mathcal{T} \models E \sqsubseteq \bot$ iff $\mathcal{T} \cup \mathcal{A}$ is not consistent.*

This proposition is a special case of a result for $\mathcal{DLFDE}^{-}$ [Khizder *et al.*, 2007]; we also use a similar representation of the ABox in the proof of Theorem 19 below. Note that, for $\mathcal{CFD}$ and $\mathcal{CFD}^{\sqcup}$, inconsistency cannot be derived. The expansion rules from Definition 3, however, still allow us to construct (a representation) of a minimal model(s) of the TBox and ABox that we shall use to show whether or not a particular tuple of ABox individuals is a certain answer to a query with respect to the given TBox and ABox.

[1] Adding inverse to terminologies leads to undecidability in the presence of functional dependencies [Toman and Weddell, 2005b].

$\mathcal{Q}/\mathcal{T}$	$\mathcal{CFD}$	$\mathcal{CFD}^{\sqcup}$	$\mathcal{CFD}^{\bot}$	$\mathcal{CFD}^{\sqcup,\bot}$
$\mathcal{CFD}$ or $\mathcal{CFD}_{\neq,(\neg)}$	in P / in P	coNP-c / coNP-c	PSPACE-c / in P	EXPTIME-c / coNP-c
$\mathcal{CFD}_{\neq,\sqcup}$ or $\mathcal{CFD}_{\neg}$	in P / coNP-c	coNP-c / coNP-c	PSPACE-c / coNP-c	EXPTIME-c / coNP-c
$\mathcal{CFD}_{inv}$	in P / NP-c	coNP-c / in Π_2^p	PSPACE-c / NP-c	EXPTIME-c / in Π_2^p

Figure 4: TBox / Posed Question Complexity Results for $\mathcal{CFD}$ and Extensions.

Conjunctive Queries.
The equational constructs in $\mathcal{CFD}$ posed questions allow us to characterize certain conjunctive queries as equational concepts. This provides a straightforward approach to *query answering over ABoxes*. Unfortunately, the equational construct is not powerful enough to capture all conjunctive queries.

Definition 18 *A conjunctive query Q is an expression*

$$q(x_{i_1},\ldots,x_{i_k}) \leftarrow \left(\bigwedge C_l(x_i)\right) \wedge \left(\bigwedge f_l(x_i) = x_j\right)$$

where C_l and f_l are $\mathcal{CFD}$ concepts and attributes, respectively. We call the variables $x_{i_1},\ldots,x_{i_k}$ distinguished.

A query graph *for a conjunctive query Q is a directed graph with variables x_i of Q playing the role of nodes that are connected by (directed) edges (x_i, x_j) whenever $f_i(x_i) = x_j$ is an atom in the query Q.*

We say that a conjunctive query Q is attribute-connected *(resp.* connected*) if every variable in the query graph of Q is reachable from a distinguished variable along a directed (resp. undirected) path.*

Although attribute-connected conjunctive queries disallow some forms of arbitrary conjunctive queries, they do admit *non tree*-shaped queries that cannot be *folded* into concept descriptions in common description logics.

We assume a standard definition of query answering under constraints utilizing the *certain answer* semantics, i.e., the ground instantiation of the head of the query is entailed by the underlying theory $\mathcal{T}$ extended with the query Q considered to be a universally quantified implication.

Theorem 19 *Let $\mathcal{T}$ be a $\mathcal{CFD}$ terminology, $\mathcal{A}$ an ABox, Q a attribute-connected conjunctive query, and θ a substitution $\langle a_1/x_{i_1},\ldots,a_k/x_{i_k}\rangle$ for $a_1,\ldots,a_k$ ABox objects. Then there are equational concepts $\mathrm{E}_{\mathcal{A}}$, E_{θ}, and E_Q such that*

$$\mathcal{T} \cup \mathcal{A} \models Q\theta \iff \mathcal{T} \models \mathrm{E}_{\mathcal{A}} \sqcap \mathrm{E}_{\theta} \sqsubseteq \mathrm{E}_Q.$$

In addition, $\mathrm{E}_{\mathcal{A}}$, E_{θ}, and E_Q can be constructed from $\mathcal{A}$ and Q in polynomial time.

Proof (sketch): Let $\mathcal{A}$ be an ABox expressing assertions about the set of individuals $\{a_1,\ldots,a_l\}$. We associate an equational concept $E_{\mathcal{A}}$ as follows:

$$\mathrm{E}_{\mathcal{A}} = \left(\bigsqcap_{\mathrm{C}(a_i)\in\mathcal{A}} \forall h_i.\mathrm{C}\right) \sqcap \left(\bigsqcap_{f(a_i)=a_j\in\mathcal{A}} (h_i.f = h_j)\right)$$

where the attributes $h_1,\ldots,h_l$ do not occur in $\mathcal{T}\cup\mathcal{A}$ nor in Q. Intuitively, these attributes represent the individual ABox objects.

For an attribute-connected query Q, let Pf_i^j be a path that leads from a distinguished query variable x_i to the variable x_j in the query graph of Q. At least one such path that starts from some distinguished variable x_i must exist for every variable x_j $(0 < j \leq n)$ in Q (for the distinguished variables, the path can be Id). We define the concept E_{θ} to capture the substitution $\theta = \langle a_1/x_{i_1},\ldots,a_k/x_{i_k}\rangle$ as follows:

$$\mathrm{E}_{\theta} = \bigsqcap_{0<j\leq n} (h_i.\,\mathsf{Pf}_i^j = x_j).$$

To determine whether $Q\theta$ is a certain answer to Q (under $\mathcal{T}\cup\mathcal{A}$) we create an equational concept E_Q as follows:

$$\mathrm{E}_Q = \left(\bigsqcap_{\mathrm{C}(x_i)\in Q} \forall x_i.\mathrm{C}\right) \sqcap \left(\bigsqcap_{f(x_i)=x_j\in Q} (x_i.f = x_j)\right)$$

It is now easy to show that the tuple $\langle (h_1)^{\mathcal{I}}(o),\ldots,(h_k)^{\mathcal{I}}(o)\rangle$ can serve as certain answer to Q whenever $o \in (\mathrm{E}_Q)^{\mathcal{I}}$ and vice versa. □

This result allows transferring complexity bounds derived in Sections 2 and 3 for the implication problem in $\mathcal{CFD}$ and its extensions to answering of path-connected conjunctive queries over an ABox under the certain answer semantics.

A similar approach can be used to determine subsumption for more general connected queries. However, the restriction to at least connected queries is essential to maintain the low complexity:

Theorem 20 *The problem of finding certain answers over $\mathcal{CFD}$ terminologies and ABoxes is PSPACE-complete even for queries of the form $\exists x.C(x)$ for C a primitive concept.*

Proof (sketch): Follows directly from the $\mathcal{CFD}^{\bot}$ case. □

The restriction is also needed to maintain decidability in the presence of inequalities in queries [Calvanese *et al.*, 1998].

5 Related Work

It has been shown that removing the boundary conditions imposed on right-hand-sides of PFDs in $\mathcal{CFD}$ makes its implication problem EXPTIME-complete [Khizder *et al.*, 2001], and leads to undecidability of both the implication problems and ABox consistency problems for $\mathcal{CFD}^{\sqcup,\bot}$ [Khizder *et al.*, 2007; Toman and Weddell, 2006]. It is also easy to see that allowing path agreements in terminologies makes the implication problem for $\mathcal{CFD}$ undecidable (by virtue of a straightforward reduction of the uniform word problem [Machtey and Young, 1978]). And it is interesting that: 1) the following two restricted cases have decidable decision problems:

- allowing arbitrary PFDs in terminologies, or
- allowing path agreements in the posed question;

but that 2) the combination of these two cases leads to undecidability [Khizder *et al.*, 2007].

PFDs were first introduced and studied in the context of object-oriented data models [Ito and Weddell, 1994; Weddell, 1989]. Subsequently, an FD concept constructor was proposed and incorporated in Classic [Borgida and Weddell, 1997], an early DL with a PTIME reasoning procedure, without changing the complexity of its implication problem. The

generalization of this constructor to unrestricted PFDs *alone* leads to EXPTIME completeness of the implication problem [Khizder *et al.*, 2001], a complexity that remains unchanged in the presence of additional concept constructors common in rich DLs such as roles, qualified number restrictions, and so on [Toman and Weddell, 2001; 2004]. PFDs have also been applied to a number of applications in object-oriented schema diagnosis and synthesis [Biskup and Polle, 2000; 2003], in query optimization [DeHaan *et al.*, 2003; Khizder *et al.*, 2000] and in the selection of indexing for a database [Stanchev and Weddell, 2003]. Finally, we note that $\mathcal{CFD}$ compliments $\mathcal{EL}$++ [Baader *et al.*, 2005] and DL-Lite [Calvanese *et al.*, 2007] as an additional DL-based option for PTIME OWL profiles.

6 Summary

We have reviewed and extended earlier work on $\mathcal{CFD}$, a dialect of description logic with universal restrictions over functional roles that allows terminological cycles in a given TBox. Notably, $\mathcal{CFD}$ includes a concept constructor that can capture a general form of functional constraint that can in turn be used to express, among other things, TBox constraints for capturing knowledge that relates to object identification. $\mathcal{CFD}$ complements existing dialects such as $\mathcal{EL}$++ and DL-Lite with PTIME reasoning procedures. In particular, we have shown that both the concept implication problem and the problem of computing certain answers for a $\mathcal{CFD}$ knowledge base for an important family of conjunctive queries have PTIME procedures. We have also considered the effect on the complexity of these problems following the addition of concept constructors for expressing concept union and the bottom class, showing for each possible extension that the complexity becomes progressively more intractable. Finally, we considered extending the expressiveness of posed questions to allow primitive negation, disjunction, general negation and inverse features, and have shown the effect of these extensions on the complexity of the associated implication problems when measured in the size of the posed question.

References

[Baader *et al.*, 2005] Franz Baader, Sebastian Brandt, and Carsten Lutz. Pushing the $\mathcal{EL}$ Envelope. In *Proc. of the 19th Int. Joint Conf. on Artificial Intelligence (IJCAI)*, pages 364–369, 2005.

[Biskup and Polle, 2000] Joachim Biskup and Torsten Polle. Decomposition of Database Classes under Path Functional Dependencies and Onto Constraints. In *Foundations of Information and Knowledge Systems*, pages 31–49, 2000.

[Biskup and Polle, 2003] Joachim Biskup and Torsten Polle. Adding inclusion dependencies to an object-oriented data model with uniqueness constraints. *Acta Informatica*, 39:391–449, 2003.

[Borgida and Weddell, 1997] Alexander Borgida and Grant Weddell. Adding Uniqueness Constraints to Description Logics (Preliminary Report). In *International Conference on Deductive and Object-Oriented Databases*, pages 85–102, 1997.

[Calvanese *et al.*, 1998] Diego Calvanese, Giuseppe De Giacomo, and Maurizio Lenzerini. On the Decidability of Query Containment under Constraints. In *ACM Symp. on Principles of Database Systems*, pages 149–158, 1998.

[Calvanese *et al.*, 2007] Diego Calvanese, Giuseppe de Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. Tractable Reasoning and Efficient Query Answering in Description Logics: The DL-Lite Family. *Journal of Automated Reasoning*, 39(3):385–429, 2007.

[Calvanese *et al.*, 2008] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. Path-Based Identification Constraints in Description Logics. In *Proc. of the 11th Int. Joint Conf. on Principles of Knowledge Representation and Reasoning (KR)*, pages 231–241, 2008.

[DeHaan *et al.*, 2003] David DeHaan, David Toman, and Grant Weddell. Rewriting Aggregate Queries using Description Logics. In *Description Logics 2003*, pages 103–112. CEUR-WS vol.81, 2003.

[Garey and Johnson, 1979] Michael R. Garey and David S. Johnson. *Computers and intractability*. Freeman, 1979.

[Ito and Weddell, 1994] Minoru Ito and Grant Weddell. Implication Problems for Functional Constraints on Databases Supporting Complex Objects. *Journal of Computer and System Sciences*, 49(3):726–768, 1994.

[Karp, 1972] Richard M. Karp. Reducibility among combinatorial problems. In R.E. Miller and J.W. Thatcher, editors, *Complexity of Computer Computations*. Plenum Press, 1972.

[Khizder *et al.*, 2000] Vitaliy L. Khizder, David Toman, and Grant Weddell. Reasoning about Duplicate Elimination with Description Logic. In *Rules and Objects in Databases (DOOD, part of CL'00)*, pages 1017–1032, 2000.

[Khizder *et al.*, 2001] Vitaliy L. Khizder, David Toman, and Grant Weddell. On Decidability and Complexity of Description Logics with Uniqueness Constraints. In *Int. Conf. on Database Theory ICDT'01*, pages 54–67, 2001.

[Khizder *et al.*, 2007] Vitaliy Khizder, David Toman, and Grant E. Weddell. Adding ABoxes to a Description Logic with Uniqueness Constraints via Path Agreements. In *Proc. International Workshop on Description Logics DL2007*, pages 339–346, 2007.

[Machtey and Young, 1978] Michael Machtey and Paul Young. *An Introduction to the General Theory of Algorithms*. North-Holland Amsterdam, 1978.

[Stanchev and Weddell, 2003] Lubomir Stanchev and Grant Weddell. Index Selection for Embedded Control Applications using Description Logics. In *Description Logics 2003*, pages 9–18. CEUR-WS vol.81, 2003.

[Toman and Weddell, 2001] David Toman and Grant Weddell. On Attributes, Roles, and Dependencies in Description Logics and the Ackermann Case of the Decision Problem. In *Description Logics 2001*, pages 76–85. CEUR-WS vol.49, 2001.

[Toman and Weddell, 2004] David Toman and Grant Weddell. Attribute Inversion in Description Logics with Path Functional Dependencies. In *Description Logics 2004*, pages 178–187. CEUR-WS vol.104, 2004.

[Toman and Weddell, 2005a] David Toman and Grant Weddell. On Reasoning about Structural Equality in XML: A Description Logic Approach. *Theoretical Computer Science*, 336(1):181–203, 2005.

[Toman and Weddell, 2005b] David Toman and Grant Weddell. On the Interaction between Inverse Features and Path-functional Dependencies in Description Logics. In *Proc. of the 19th Int. Joint Conf. on Artificial Intelligence (IJCAI)*, pages 603–608, 2005.

[Toman and Weddell, 2006] David Toman and Grant Weddell. On Keys and Functional Dependencies as First-Class Citizens in Description Logics. In *Proc. of Int. Joint Conf. on Automated Reasoning (IJCAR)*, pages 647–661, 2006.

[Toman and Weddell, 2008] David Toman and Grant E. Weddell. PTIME Description Logics with Functional Constraints and their Extensions. In *CEDAR'08*, pages 5–19, 2008.

[W3C, 2008] W3C. OWL2 Web Ontology Language: Profiles, 2008.

[Weddell, 1989] Grant Weddell. A Theory of Functional Dependencies for Object Oriented Data Models. In *International Conference on Deductive and Object-Oriented Databases*, pages 165–184, 1989.