Ontology Module Extraction via Datalog Reasoning*

Ana Armas Romero, Mark Kaminski, Bernardo Cuenca Grau, and Ian Horrocks

Department of Computer Science, University of Oxford, UK

Abstract. We propose a novel approach in which module extraction is reduced to a reasoning problem in datalog. Our approach generalises existing approaches and can be tailored to preserve only specific kinds of entailments, which allows us to extract significantly smaller modules. An evaluation on widely-used ontologies shows very encouraging results.

1 Introduction

Module extraction is the task of computing, given an ontology $\mathcal{T}$ and a signature of interest Σ, a (preferably small) subset $\mathcal{M}$ of $\mathcal{T}$ (a module) that preserves all relevant entailments in $\mathcal{T}$ over the set of symbols Σ. Such an $\mathcal{M}$ is indistinguishable from $\mathcal{T}$ w.r.t. Σ, and $\mathcal{T}$ can be safely replaced with $\mathcal{M}$ in applications of $\mathcal{T}$ that use only the symbols in Σ.

Module extraction has received a lot of attention in recent years [27, 6, 25, 17, 10, 9, 23], and modules have found a wide range of applications, including ontology reuse [6, 13], matching [12], debugging [29, 19] and classification [1, 30, 4].

Preservation of relevant entailments is formalised via *inseparability relations*. The strongest notion is *model* inseparability, which requires that it must be possible to turn any model of $\mathcal{M}$ into a model of $\mathcal{T}$ by (re-)interpreting only the symbols outside Σ; such an $\mathcal{M}$ preserves all second-order entailments of $\mathcal{T}$ w.r.t. Σ [15]. A weaker and more flexible notion is *deductive* inseparability, which requires only that $\mathcal{T}$ and $\mathcal{M}$ entail the same Σ-formulas *in a given query language*. Unfortunately, the decision problems associated with module extraction are invariably of high complexity, and often undecidable. For model inseparability, checking whether $\mathcal{M}$ is a Σ-module in $\mathcal{T}$ is undecidable even if $\mathcal{T}$ is restricted to be in the description logic (DL) $\mathcal{EL}$, for which standard reasoning is tractable. For deductive inseparability, the problem is typically decidable for lightweight DLs and "reasonable" query languages, albeit of high worst-case complexity; e.g., the problem is already ExpTime-hard for $\mathcal{EL}$ if we consider concept inclusions as the query language [20]. Practical algorithms that ensure minimality of the extracted modules are known only for acyclic $\mathcal{ELI}$ [15] and DL-Lite [17].

Practical module extraction techniques are typically based on sound approximations: they ensure that the extracted fragment $\mathcal{M}$ is a module (i.e., inseparable

* This paper summarises the results of our work published in [2, 3].

from $\mathcal{T}$ w.r.t. Σ), but they give no minimality guarantee. The most popular such techniques are based on a family of polynomially checkable conditions called syntactic locality [5, 6, 24]; in particular, $\bot$-locality and $\top\bot^*$-locality. Each locality-based module $\mathcal{M}$ enjoys a number of desirable properties for applications: *(i)* it is model inseparable from $\mathcal{T}$; *(ii)* it is *depleting*, i.e., $\mathcal{T} \setminus \mathcal{M}$ is inseparable from the empty ontology w.r.t. Σ; *(iii)* it contains all justifications (a.k.a. explanations) in $\mathcal{T}$ of every Σ-formula entailed by $\mathcal{T}$; and *(iv)* last but not least, it can be computed efficiently, even for very expressive ontology languages.

Locality-based techniques are easy to implement, and surprisingly effective in practice. However, the extracted modules can be rather large, which limits their usefulness in some applications [8]. One way to address this is to develop techniques that produce smaller modules while still preserving properties *(i)*–*(iii)*. Efforts in this direction have confirmed that locality-based modules can be far from optimal in practice [10]; however, these techniques only apply to rather restricted languages and utilise algorithms with high worst-case complexity.

Another approach to computing smaller modules is to weaken properties *(i)*–*(iii)*, which are stronger than required for many applications. In particular, model inseparability (property *(i)*) is a very strong condition, and deductive inseparability would usually suffice, with the query language determining which kinds of consequence are preserved; in modular classification, for example, only atomic concept inclusions need to be preserved. However, practical module extraction techniques for expressive ontology languages yield modules that satisfy all three properties, and hence are potentially much larger than they need to be.

In this paper, we propose a technique that reduces module extraction to a reasoning problem in datalog. The connection between module extraction and datalog was first observed in [28], where it was shown that locality $\bot$-module extraction for $\mathcal{EL}$ ontologies could be reduced to propositional datalog reasoning. Our approach takes this connection much farther by generalising both locality-based and reachability-based [23] modules for expressive ontology languages in an elegant way. A key distinguishing feature of our technique is that it can extract deductively inseparable modules, with the query language tailored to the requirements of the application at hand, which allows us to relax Property *(i)* and extract significantly smaller modules. In all cases our modules preserve the nice features of locality: they are widely applicable (even beyond DLs), they can be efficiently computed, they are depleting (Property *(ii)*) and they preserve all justifications of relevant entailments (Property *(iii)*). We have implemented our approach using the RDFox datalog engine [22]. Our evaluation shows that module size consistently decreases as we consider weaker inseparability relations, which could significantly improve the usefulness of modules in applications.

2 Preliminaries

Ontologies and Queries We use standard first-order logic and assume familiarity with description logics, ontology languages and theorem proving. A signature Σ is a set of predicates and $\mathsf{Sig}(F)$ denotes the signature of a set of

formulas F. To capture a wide range of KR languages, we formalise ontology axioms as *rules*: function-free sentences of the form $\forall \mathbf{x}.[\varphi(\mathbf{x}) \rightarrow \exists \mathbf{y}.[\bigvee_{i=1}^{n} \psi_i(\mathbf{x}, \mathbf{y})]]$, where φ, ψ_i are conjunctions of distinct atoms. Formula φ is the rule *body* and $\exists \mathbf{y}.[\bigvee_{i=1}^{n} \psi_i(\mathbf{x}, \mathbf{y})]$ is the *head*. Universal quantification is omitted for brevity. Rules are required to be safe (all variables in the head occur in the body). A TBox $\mathcal{T}$ is a finite set of rules; TBoxes mentioning equality ($\approx$) are extended with its standard axiomatisation. A fact γ is a function-free ground atom. An ABox $\mathcal{A}$ is a finite set of facts. A *positive existential query (PEQ)* is a formula $q(\mathbf{x}) = \exists \mathbf{y}.\varphi(\mathbf{x}, \mathbf{y})$, where φ is built from function-free atoms using only $\wedge$ and $\vee$.

Datalog A rule is *datalog* if its head has exactly one atom (which could be $\bot$, the nullary falsehood predicate) and all variables are universally quantified. A *datalog program* $\mathcal{P}$ is a set of datalog rules. Given $\mathcal{P}$ and an ABox $\mathcal{A}$, their *materialisation* is the set of facts entailed by $\mathcal{P} \cup \mathcal{A}$, which can be computed by means of forward-chaining. A fact γ is a *consequence* of a datalog rule $r = \bigwedge_{i=1}^{n} \gamma_i' \rightarrow \delta$ and facts $\gamma_1, \ldots, \gamma_n$ if $\gamma = \delta\sigma$ with σ a most-general unifier (MGU) of γ_i, γ_i' for each $1 \leq i \leq n$. A (forward-chaining) *proof* of γ in $\mathcal{P} \cup \mathcal{A}$ is a pair $\rho = (T, \lambda)$ where T is a tree, λ is a mapping from nodes in T to facts such that for each node v the following holds: 1. $\lambda(v) = \gamma$ if v is the root of T; 2. if v is a leaf then $\lambda(v) \in \mathcal{A}$ or $(\rightarrow \lambda(v)) \in \mathcal{P}$; and 3. if v has children $w_1, \ldots, w_n$ then $\lambda(v)$ is a consequence of r and $\lambda(w_1), \ldots, \lambda(w_n)$. Forward-chaining is sound (if γ has a proof in $\mathcal{P} \cup \mathcal{A}$ then it is entailed by $\mathcal{P} \cup \mathcal{A}$) and complete (if γ is entailed by $\mathcal{P} \cup \mathcal{A}$ then either γ has a proof in $\mathcal{P} \cup \mathcal{A}$ or so does $\bot$). Finally, the *support* of γ is the set of rules occurring in some proof of γ in $\mathcal{P} \cup \mathcal{A}$.

Inseparability Relations & Modules We next recapitulate the most common inseparability relations studied in the literature. TBoxes $\mathcal{T}$ and $\mathcal{T}'$ are

- Σ-*model inseparable* ($\mathcal{T} \equiv_{\Sigma}^{\mathsf{m}} \mathcal{T}'$), if for every model $\mathcal{I}$ of $\mathcal{T}$ there is a model $\mathcal{J}$ of $\mathcal{T}'$ with the same domain s.t. $\mathsf{A}^{\mathcal{I}} = \mathsf{A}^{\mathcal{J}}$ for each $\mathsf{A} \in \Sigma$, and vice versa.
- Σ-*query inseparable* ($\mathcal{T} \equiv_{\Sigma}^{\mathsf{q}} \mathcal{T}'$) if for every Boolean PEQ q and Σ-ABox $\mathcal{A}$ we have $\mathcal{T} \cup \mathcal{A} \models q$ iff $\mathcal{T}' \cup \mathcal{A} \models q$.
- Σ-*fact inseparable* ($\mathcal{T} \equiv_{\Sigma}^{\mathsf{f}} \mathcal{T}'$) if for every fact γ and ABox $\mathcal{A}$ over Σ we have $\mathcal{T} \cup \mathcal{A} \models \gamma$ iff $\mathcal{T}' \cup \mathcal{A} \models \gamma$.
- Σ-*implication inseparable* ($\mathcal{T} \equiv_{\Sigma}^{\mathsf{i}} \mathcal{T}'$) if for each φ of the form $\mathsf{A}(\mathbf{x}) \rightarrow \mathsf{B}(\mathbf{x})$ with $\mathsf{A}, \mathsf{B} \in \Sigma$, $\mathcal{T} \models \varphi$ iff $\mathcal{T}' \models \varphi$.

These relations are naturally ordered from strongest to weakest: $\equiv_{\Sigma}^{\mathsf{m}} \subsetneq \equiv_{\Sigma}^{\mathsf{q}} \subsetneq \equiv_{\Sigma}^{\mathsf{f}} \subsetneq \equiv_{\Sigma}^{\mathsf{i}}$ for each non-trivial Σ.

Given an inseparability relation $\equiv$ for Σ, a subset $\mathcal{M} \subseteq \mathcal{T}$ is a $\equiv$-*module of* $\mathcal{T}$ if $\mathcal{T} \equiv \mathcal{M}$. Furthermore, $\mathcal{M}$ is *minimal* if no $\mathcal{M}' \subsetneq \mathcal{M}$ is a $\equiv$-module of $\mathcal{T}$.

3 Module Extraction via Datalog Reasoning

In this section, we present our approach to module extraction by reduction into a reasoning problem in datalog. Our approach builds on recent techniques that exploit datalog engines for ontology reasoning [16, 26, 31]. In what follows, we fix an arbitrary TBox $\mathcal{T}$ and signature $\Sigma \subseteq \mathsf{Sig}(\mathcal{T})$. Unless otherwise stated,

(r_1)	$\mathsf{A}(x) \rightarrow \exists y_1.[\mathsf{R}(x, y_1) \wedge \mathsf{B}(y_1)]$	$\mathsf{A} \sqsubseteq \exists \mathsf{R}.\mathsf{B}$
(r_2)	$\mathsf{A}(x) \rightarrow \exists y_2.[\mathsf{R}(x, y_2) \wedge \mathsf{C}(y_2)]$	$\mathsf{A} \sqsubseteq \exists \mathsf{R}.\mathsf{C}$
(r_3)	$\mathsf{B}(x) \wedge \mathsf{C}(x) \rightarrow \mathsf{D}(x)$	$\mathsf{B} \sqcap \mathsf{C} \sqsubseteq \mathsf{D}$
(r_4)	$\mathsf{D}(x) \rightarrow \exists y_3.[\mathsf{S}(x, y_3) \wedge \mathsf{E}(y_3)]$	$\mathsf{D} \sqsubseteq \exists \mathsf{S}.\mathsf{E}$
(r_5)	$\mathsf{D}(x) \wedge \mathsf{S}(x, y) \rightarrow \mathsf{F}(y)$	$\mathsf{D} \sqsubseteq \forall \mathsf{S}.\mathsf{F}$
(r_6)	$\mathsf{S}(x, y) \wedge \mathsf{E}(y) \wedge \mathsf{F}(y) \rightarrow \mathsf{G}(x)$	$\exists \mathsf{S}.(\mathsf{E} \sqcap \mathsf{F}) \sqsubseteq \mathsf{G}$
(r_7)	$\mathsf{G}(x) \wedge \mathsf{H}(x) \rightarrow \bot$	$\mathsf{G} \sqcap \mathsf{H} \sqsubseteq \bot$

Fig. 1. Example TBox $\mathcal{T}^{ex}$ with DL translation

our definitions and theorems are parameterised by such $\mathcal{T}$ and Σ. We assume w.l.o.g. that rules in $\mathcal{T}$ share no existentially quantified variables. For simplicity, we also assume that $\mathcal{T}$ contains no constants (all results extend to constants [3]).

3.1 Overview and Main Intuitions

Our overall strategy to extract a module $\mathcal{M}$ of $\mathcal{T}$ for an inseparability relation $\equiv^z_\Sigma$, with $z \in \{\mathsf{m}, \mathsf{q}, \mathsf{f}, \mathsf{i}\}$, can be summarised by the following steps:

1. Pick a substitution θ mapping all existentially quantified variables in $\mathcal{T}$ to constants, and transform $\mathcal{T}$ into a datalog program $\mathcal{P}$ by *(i)* Skolemising all rules in $\mathcal{T}$ using θ and *(ii)* turning disjunctions into conjunctions while splitting them into different rules, thus replacing each function-free disjunctive rule of the form $\varphi(\mathbf{x}) \rightarrow \bigvee_{i=1}^{n} \psi_i(\mathbf{x})$ with datalog rules $\varphi(\mathbf{x}) \rightarrow \psi_1(\mathbf{x}), \ldots, \varphi(\mathbf{x}) \rightarrow \psi_n(\mathbf{x})$.
2. Pick a Σ-ABox $\mathcal{A}_0$ and materialise $\mathcal{P} \cup \mathcal{A}_0$.
3. Pick a set $\mathcal{A}_r$ of "relevant facts" in the materialisation and compute the supporting rules in $\mathcal{P}$ for each such fact.
4. The module $\mathcal{M}$ consists of all rules in $\mathcal{T}$ that yield a supporting rule in $\mathcal{P}$.

Thus, $\mathcal{M}$ is fully determined by the substitution θ and the ABoxes $\mathcal{A}_0$, $\mathcal{A}_r$.

The main intuition behind our module extraction approach is that we can pick θ, $\mathcal{A}_0$ and $\mathcal{A}_r$ (and hence $\mathcal{M}$) such that each proof ρ of a Σ-consequence φ of $\mathcal{T}$ to be preserved can be embedded in a forward chaining proof ρ' in $\mathcal{P} \cup \mathcal{A}_0$ of a relevant fact in $\mathcal{A}_r$. Such an embedding satisfies the key property that, for each rule r involved in ρ, at least one corresponding datalog rule in $\mathcal{P}$ is involved in ρ'. In this way we ensure that $\mathcal{M}$ contains the necessary rules to entail φ. This approach, however, does not ensure minimality of $\mathcal{M}$: since $\mathcal{P}$ is a strengthening of $\mathcal{T}$ there may be proofs of a relevant fact in $\mathcal{P} \cup \mathcal{A}_0$ that do not correspond to a Σ-consequence of $\mathcal{T}$, which may lead to unnecessary rules in $\mathcal{M}$.

To illustrate how our strategy might work in practice, consider $\mathcal{T}^{ex}$ in Fig. 1, $\Sigma = \{\mathsf{B}, \mathsf{C}, \mathsf{D}, \mathsf{G}\}$, and that we want a module $\mathcal{M}$ that is Σ-implication inseparable from $\mathcal{T}^{ex}$. This is a simple case since $\varphi = \mathsf{D}(x) \rightarrow \mathsf{G}(x)$ is the only non-trivial Σ-implication entailed by $\mathcal{T}^{ex}$; thus, for $\mathcal{M}$ to be a module we only need $\mathcal{M} \models \varphi$. Note that proving $\mathcal{T}^{ex} \models \varphi$ amounts to proving $\mathcal{T}^{ex} \cup \{\mathsf{D}(a)\} \models \mathsf{G}(a)$ (with a a fresh constant). Figure 2(a) depicts a hyper-resolution tree ρ showing how $\mathsf{G}(a)$ can be derived from the clauses corresponding to r_4–r_6 and $\mathsf{D}(a)$, with rule r_4

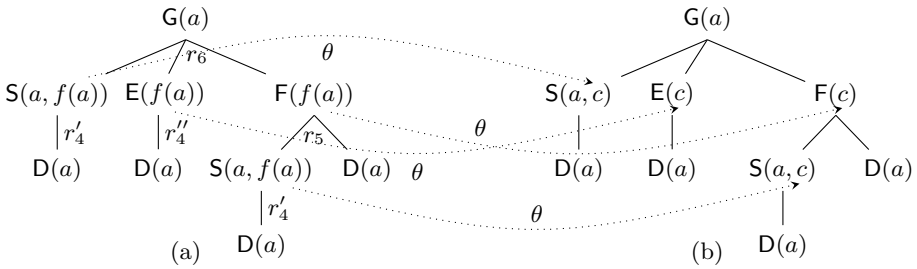

Fig. 2. Proofs of $\mathsf{G}(a)$ from $\mathsf{D}(a)$ in (a) $\mathcal{T}^{ex}$ and (b) the corresponding datalog program

transformed into clauses $r'_4 = \mathsf{D}(x) \rightarrow \mathsf{S}(x, f(x_3))$ and $r''_4 = \mathsf{D}(x) \rightarrow \mathsf{E}(f(x_3))$. Hence $\mathcal{M} = \{r_4\text{–}r_6\}$ is a Σ-implication inseparable module of $\mathcal{T}^{ex}$, and as $\mathsf{G}(a)$ cannot be derived from any subset of $\{r_4\text{–}r_6\}$, $\mathcal{M}$ is also minimal.

We pick $\mathcal{A}_0$ to contain the initial fact $\mathsf{D}(a)$, $\mathcal{A}_r$ to contain the fact to be proved $\mathsf{G}(a)$, and we make θ map variable y_3 in r_4 to a fresh constant c, in which case rule r_4 corresponds to the rules $\mathsf{D}(x) \rightarrow \mathsf{S}(x, c)$ and $\mathsf{D}(x) \rightarrow \mathsf{E}(c)$ in $\mathcal{P}$.

Figure 2(b) depicts a forward chaining proof ρ' of $\mathsf{G}(a)$ in $\mathcal{P} \cup \{\mathsf{D}(a)\}$. As shown, ρ can be embedded in ρ' via θ by mapping functional terms over f to the fresh constant c. This way, the rules involved in ρ are mapped to the datalog rules involved in ρ' via θ. Hence, we will extract the (minimal) module $\mathcal{M} = \{r_4\text{–}r_6\}$.

3.2 The Notion of Module Setting

The substitution θ and the ABoxes $\mathcal{A}_0$ and $\mathcal{A}_r$, which determine the extracted module, can be chosen in different ways to ensure the preservation of different kinds of Σ-consequences. The following notion of a module setting captures in a declarative way the main elements of our approach.

Definition 3.1. *A* module setting *for $\mathcal{T}$ and Σ is a tuple $\chi = \langle \theta, \mathcal{A}_0, \mathcal{A}_r \rangle$ with θ a substitution from existentially quantified variables in $\mathcal{T}$ to constants, $\mathcal{A}_0$ a Σ-ABox, $\mathcal{A}_r$ a $(\mathsf{Sig}(\mathcal{T}) \cup \{\bot\})$-ABox, and s.t. no constant in χ occurs in $\mathcal{T}$.*

The program *of χ is the smallest datalog program $\mathcal{P}^\chi$ containing, for each $r = \varphi(\mathbf{x}) \rightarrow \exists \mathbf{y}.[\bigvee_{i=1}^{n} \psi_i(\mathbf{x}, \mathbf{y})]$ in $\mathcal{T}$, the rule $\varphi \rightarrow \bot$ if $n = 0$ and all rules $\varphi \rightarrow \gamma\theta$ for each $1 \leq i \leq n$ and each atom γ in ψ_i. The* support *of χ is the set of rules $r \in \mathcal{P}^\chi$ that support a fact from $\mathcal{A}_r$ in $\mathcal{P}^\chi \cup \mathcal{A}_0$. The* module $\mathcal{M}^\chi$ *of χ is the set of rules in $\mathcal{T}$ that have a corresponding datalog rule in the support of χ.*

3.3 Modules for each Inseparability Relation

We next consider each inseparability relation $\equiv^z_\Sigma$, where $z \in \{\mathsf{m}, \mathsf{q}, \mathsf{f}, \mathsf{i}\}$, and formulate a specific setting χ_z which provably yields a $\equiv^z_\Sigma$-module of $\mathcal{T}$.

Implication Inseparability The example in Sect. 3.1 suggests a natural setting $\chi_\mathsf{i} = \langle \theta, \mathcal{A}_0, \mathcal{A}_r \rangle$ that guarantees implication inseparability. As in our example, we pick θ to be as "general" as possible by Skolemising each existentially

quantified variable to a fresh constant. For A and B predicates of the same arity n, proving that $\mathcal{T}$ entails a Σ-implication $\varphi = \mathsf{A}(x_1, \ldots, x_n) \rightarrow \mathsf{B}(x_1, \ldots, x_n)$, amounts to showing that $\mathcal{T} \cup \{\mathsf{A}(a_1, \ldots, a_n)\} \models \mathsf{B}(a_1, \ldots, a_n)$ for fresh constants $a_1, \ldots, a_n$. Thus, following the ideas of our example, we initialise $\mathcal{A}_0$ with a fact $\mathsf{A}(c^1_{\mathsf{A}}, \ldots, c^n_{\mathsf{A}})$ for each n-ary predicate $\mathsf{A} \in \Sigma$, and $\mathcal{A}_r$ with a fact $\mathsf{B}(c^1_{\mathsf{A}}, \ldots, c^n_{\mathsf{A}})$ for each pair of n-ary predicates $\{\mathsf{B}, \mathsf{A}\} \subseteq \Sigma$ with $\mathsf{B} \neq \mathsf{A}$.

Definition 3.2. *For each existentially quantified variable y_j in $\mathcal{T}$, let c_{y_j} be a fresh constant. Furthermore, for each $\mathsf{A} \in \Sigma$ of arity n, let $c^1_{\mathsf{A}}, \ldots, c^n_{\mathsf{A}}$ be also fresh constants. The setting $\chi_{\mathsf{i}} = \langle \theta^{\mathsf{i}}, \mathcal{A}^{\mathsf{i}}_0, \mathcal{A}^{\mathsf{i}}_r \rangle$ is defined as follows:*

- $\theta^{\mathsf{i}} = \{\, y_j \mapsto c_{y_j} \mid y_j$ *existentially quantified in* $\mathcal{T} \,\}$,
- $\mathcal{A}^{\mathsf{i}}_0 = \{\mathsf{A}(c^1_{\mathsf{A}}, \ldots, c^n_{\mathsf{A}}) \mid \mathsf{A}$ *n-ary predicate in* $\Sigma\}$, *and*
- $\mathcal{A}^{\mathsf{i}}_r = \{\mathsf{B}(c^1_{\mathsf{A}}, \ldots, c^n_{\mathsf{A}}) \mid \mathsf{A} \neq \mathsf{B}$ *n-ary predicates in* $\Sigma\} \cup \{\bot\}$.

The setting χ_{i} is reminiscent of the datalog encodings typically used to check whether a concept A is subsumed by concept B w.r.t. a "lightweight" ontology $\mathcal{T}$ [18, 26]. There, variables in rules are Skolemised as fresh constants to produce a datalog program $\mathcal{P}$ and it is then checked whether $\mathcal{P} \cup \{A(a)\} \models B(a)$.

Theorem 3.3. $\mathcal{M}^{\chi_{\mathsf{i}}} \equiv^{\mathsf{i}}_{\Sigma} \mathcal{T}$.

Fact Inseparability The setting χ_{i} in Def. 3.2 cannot be used to ensure fact inseparability. Consider again $\mathcal{T}^{ex}$ and $\Sigma = \{\mathsf{B}, \mathsf{C}, \mathsf{D}, \mathsf{G}\}$, for which $\mathcal{M}^{\chi_{\mathsf{i}}} = \{r_4, r_5, r_6\}$. For $\mathcal{A} = \{\mathsf{B}(a), \mathsf{C}(a)\}$ we have $\mathcal{T}^{ex} \cup \mathcal{A} \models \mathsf{G}(a)$ but $\mathcal{M}^{\chi_{\mathsf{i}}} \cup \mathcal{A} \not\models \mathsf{G}(a)$, and hence $\mathcal{M}^{\chi_{\mathsf{i}}}$ is not fact inseparable from $\mathcal{T}^{ex}$.

More generally, $\mathcal{M}^{\chi_{\mathsf{i}}}$ will only preserve Σ-fact entailments $\mathcal{T} \cup \mathcal{A} \models \gamma$ where $\mathcal{A}$ is a singleton. However, for a module to be fact inseparable from $\mathcal{T}$ it must preserve all Σ-facts when coupled with *any* Σ-ABox. We achieve this by choosing $\mathcal{A}_0$ to be the *critical ABox* for Σ, which consists of all facts that can be constructed using Σ and a single fresh constant [21]. Since every Σ-ABox can be homomorphically mapped into the critical Σ-ABox, we can show that all proofs of a Σ-fact in $\mathcal{T} \cup \mathcal{A}$ are embeddable into a proof of a relevant fact in $\mathcal{P}^{\chi} \cup \mathcal{A}_0$.

Definition 3.4. *Let constants c_{y_i} be as in Def. 3.2, and let $*$ be a fresh constant. The setting $\chi_{\mathsf{f}} = \langle \theta^{\mathsf{f}}, \mathcal{A}^{\mathsf{f}}_0, \mathcal{A}^{\mathsf{f}}_r \rangle$ is defined as follows: (i) $\theta^{\mathsf{f}} = \theta^{\mathsf{i}}$, (ii) $\mathcal{A}^{\mathsf{f}}_0 = \{\, \mathsf{A}(*, \ldots, *) \mid \mathsf{A} \in \Sigma \,\}$, and (iii) $\mathcal{A}^{\mathsf{f}}_r = \mathcal{A}^{\mathsf{f}}_0 \cup \{\bot\}$.*

The datalog programs for χ_{i} and χ_{f} coincide; hence, the only difference between the two settings is in the definition of their corresponding ABoxes. In our example, both $\mathcal{A}^{\mathsf{f}}_0$ and $\mathcal{A}^{\mathsf{f}}_r$ contain facts $\mathsf{B}(*)$, $\mathsf{C}(*)$, $\mathsf{D}(*)$, and $\mathsf{G}(*)$. Clearly, $\mathcal{P}^{\chi_{\mathsf{f}}} \cup \mathcal{A}_0 \models \mathsf{G}(*)$ and the proof additionally involves rule r_3. Thus $\mathcal{M}^{\chi_{\mathsf{f}}} = \{r_3, r_4, r_5, r_6\}$.

Theorem 3.5. $\mathcal{M}^{\chi_{\mathsf{f}}} \equiv^{\mathsf{f}}_{\Sigma} \mathcal{T}$.

Query Inseparability Positive existential queries constitute a much richer query language than facts as they allow for existentially quantified variables. Thus, the query inseparability requirement invariably leads to larger modules.

For instance, let $\mathcal{T} = \mathcal{T}^{ex}$ and $\Sigma = \{\mathsf{A}, \mathsf{B}\}$. Given the Σ-ABox $\mathcal{A} = \{\mathsf{A}(a)\}$ and Σ-query $q = \exists y.\mathsf{B}(y)$ we have $\mathcal{T}^{ex} \cup \mathcal{A} \models q$ (due to rule r_1). The module $\mathcal{M}^{\chi_\mathsf{f}}$ is, however, empty. Indeed, the materialisation of $\mathcal{P}^{\chi_\mathsf{f}} \cup \{\mathsf{A}(*)\}$ consists of the additional facts $\mathsf{R}(*, c_{y_1})$ and $\mathsf{B}(c_{y_1})$ and hence contains no relevant fact mentioning only $*$. Thus, $\mathcal{M}^{\chi_\mathsf{f}} \cup \mathcal{A} \not\models q$ and $\mathcal{M}^{\chi_\mathsf{f}}$ is not query inseparable from $\mathcal{T}^{ex}$.

Our example suggests that, although the critical ABox is constrained enough to embed every Σ-ABox, we need to consider additional relevant facts to capture all proofs of a Σ-query. In particular, rule r_1 implies that B contains an instance whenever A does: a dependency that is then checked by q. This can be captured by considering fact $\mathsf{B}(c_{y_1})$ as relevant, in which case r_1 would be in the module.

More generally, we consider a module setting χ that differs from χ_f only in that all Σ-facts (and not just those over $*$) are considered as relevant.

Definition 3.6. *Let constants c_{y_i} and $*$ be as in Def. 3.4. The setting $\chi_\mathsf{q} = \langle \theta^\mathsf{q}, \mathcal{A}_0^\mathsf{q}, \mathcal{A}_r^\mathsf{q} \rangle$ is as follows: (i) $\theta^\mathsf{q} = \theta^\mathsf{f}$, (ii) $\mathcal{A}_0^\mathsf{q} = \mathcal{A}_0^\mathsf{f}$, and (iii) $\mathcal{A}_r^\mathsf{q}$ consists of $\bot$ and all Σ-facts $\mathsf{A}(a_1, \ldots, a_n)$ with each a_j either a constant c_{y_i} or $*$.*

Theorem 3.7. *$\mathcal{M}^{\chi_\mathsf{q}} \equiv_\Sigma^\mathsf{q} \mathcal{T}$.*

Model Inseparability The modules generated by χ_q may not be model inseparable from $\mathcal{T}$. To see this, let $\mathcal{T} = \mathcal{T}^{ex}$ and $\Sigma = \{\mathsf{A}, \mathsf{D}, \mathsf{R}\}$, in which case $\mathcal{M}^{\chi_\mathsf{q}} = \{r_1, r_2\}$. The interpretation $\mathcal{I}$ where $\Delta^\mathcal{I} = \{a, b\}$, $\mathsf{A}^\mathcal{I} = \{a\}$, $\mathsf{B}^\mathcal{I} = \mathsf{C}^\mathcal{I} = \{b\}$, $\mathsf{D}^\mathcal{I} = \emptyset$ and $R^\mathcal{I} = \{(a, b)\}$ is a model of $\mathcal{M}^{\chi_\mathsf{q}}$. However, $\mathcal{I}$ cannot be extended to a model of r_3 (and hence of $\mathcal{T}$) without reinterpreting A, R or D.

To achieve model inseparability, it suffices to ensure that each model of the module can be extended to a model of $\mathcal{T}$ in a uniform way. Thus, $\mathcal{M} = \{r_1, r_2, r_3\}$ is a $\equiv_\Sigma^\mathsf{m}$-module of $\mathcal{T}^{ex}$ since all its models can be extended by interpreting E, F and G as the domain, H as empty, and S as the Cartesian product of the domain. We can capture this idea in our framework by means of the following setting.

Definition 3.8. *The setting $\chi_\mathsf{m} = \langle \theta^\mathsf{m}, \mathcal{A}_0^\mathsf{m}, \mathcal{A}_r^\mathsf{m} \rangle$ is as follows: θ^m maps each existentially quantified y_j to the fresh constant $*$, $\mathcal{A}_0^\mathsf{m} = \mathcal{A}_0^\mathsf{f}$, and $\mathcal{A}_r^\mathsf{m} = \mathcal{A}_0^\mathsf{m} \cup \{\bot\}$.*

In our example, $\mathcal{P}^{\chi_\mathsf{m}} \cup \mathcal{A}_0^\mathsf{m}$ entails the relevant facts $\mathsf{A}(*)$, $\mathsf{R}(*, *)$ and $\mathsf{D}(*)$, and hence $\mathcal{M}^{\chi_\mathsf{m}} = \{r_1, r_2, r_3\}$.

Theorem 3.9. *$\mathcal{M}^{\chi_\mathsf{m}} \equiv_\Sigma^\mathsf{m} \mathcal{T}$.*

3.4 Modules for Ontology Classification

Module extraction has been exploited for optimising ontology classification [1, 30, 4]. In this case, it is not only required that modules are implication inseparable from $\mathcal{T}$, but also that they preserve all implications $\mathsf{A}(\mathbf{x}) \to \mathsf{B}(\mathbf{x})$ with $\mathsf{A} \in \Sigma$ but $\mathsf{B} \notin \Sigma$. This requirement can be captured as given next.

Definition 3.10. *TBoxes $\mathcal{T}$ and $\mathcal{T}'$ are* Σ-classification inseparable *($\mathcal{T} \equiv_\Sigma^\mathsf{c} \mathcal{T}'$) if for each φ of the form $\mathsf{A}(\mathbf{x}) \to \psi$ where $\mathsf{A} \in \Sigma$ and either $\psi = \bot$ or $\psi = \mathsf{B}(\mathbf{x})$ for $\mathsf{B} \in \mathsf{Sig}(\mathcal{T} \cup \mathcal{T}')$, we have $\mathcal{T} \models \varphi$ iff $\mathcal{T}' \models \varphi$.*

Classification inseparability is a stronger requirement than implication inseparability. For $\mathcal{T} = \{\mathsf{A}(x) \to \mathsf{B}(x)\}$ and $\Sigma = \{\mathsf{A}\}$, $\mathcal{M} = \emptyset$ is implication inseparable from $\mathcal{T}$, whereas classification inseparability requires that $\mathcal{M} = \mathcal{T}$.

Modular reasoners such as MORe [1] and Chainsaw [30] rely on locality $\bot$-modules, which satisfy this requirement. Each model of a $\bot$-module $\mathcal{M}$ can be extended to a model of $\mathcal{T}$ by interpreting all additional predicates as empty, which is not possible if $\mathsf{A} \in \Sigma$ and $\mathcal{T}$ entails $\mathsf{A}(x) \to \mathsf{B}(x)$ but $\mathcal{M}$ does not. We can cast $\bot$-modules in our framework with the following setting, which extends χ_{m} in Def. 3.8 by also considering as relevant facts involving predicates not in Σ.

Definition 3.11. *The setting* $\chi_{\mathsf{b}} = \langle \theta^{\mathsf{b}}, \mathcal{A}_0^{\mathsf{b}}, \mathcal{A}_r^{\mathsf{b}} \rangle$ *is as follows:* $\theta^{\mathsf{b}} = \theta^{\mathsf{m}}$, $\mathcal{A}_0^{\mathsf{b}} = \mathcal{A}_0^{\mathsf{m}}$, *and* $\mathcal{A}_r$ *consists of* $\bot$ *and all facts* $\mathsf{A}(*, \ldots, *)$ *where* $\mathsf{A} \in \mathsf{Sig}(\mathcal{T})$.

The use of $\bot$-modules is, however, stricter than is needed for ontology classification. For instance, if we consider $\mathcal{T} = \mathcal{T}^{ex}$ and $\Sigma = \{\mathsf{A}\}$ we have that $\mathcal{M}^{\chi_{\mathsf{b}}}$ contains all rules r_1–r_6, but since A does not have any subsumers in $\mathcal{T}^{ex}$ the empty TBox is already classification inseparable from $\mathcal{T}^{ex}$.

The following module setting extends χ_{i} in Def. 3.2 to ensure classification inseparability. As in the case of χ_{b} in Def. 3.11, the only required modification is to also consider as relevant facts involving predicates outside Σ.

Definition 3.12. *Setting* $\chi_{\mathsf{c}} = (\theta^{\mathsf{c}}, \mathcal{A}_0^{\mathsf{c}}, \mathcal{A}_r^{\mathsf{c}})$ *is as follows:* $\theta^{\mathsf{c}} = \theta^{\mathsf{i}}$, $\mathcal{A}_0^{\mathsf{c}} = \mathcal{A}_0^{\mathsf{i}}$, *and* $\mathcal{A}_r^{\mathsf{c}}$ *consists of* $\bot$ *and all facts* $\mathsf{B}(c_{\mathsf{A}}^1, \ldots, c_{\mathsf{A}}^n)$ *s.t.* $\mathsf{A} \neq \mathsf{B}$ *are* n*-ary predicates,* $A \in \Sigma$ *and* $\mathsf{B} \in \mathsf{Sig}(\mathcal{T})$.

Indeed, given $\mathcal{T} = \mathcal{T}^{ex}$ and $\Sigma = \{\mathsf{A}\}$, the module for χ_{c} is empty, as desired.

Theorem 3.13. $\mathcal{M}^{\chi_{\mathsf{c}}} \equiv_{\Sigma}^{\mathsf{c}} \mathcal{T}$.

3.5 Additional Properties of Modules

Although the essential property of a module $\mathcal{M}$ is that it captures all relevant Σ-consequences of $\mathcal{T}$, in some applications modules are expected to satisfy additional requirements. For ontology reuse, it is sometimes desirable that a module $\mathcal{M}$ does not "leave any relevant information behind" in the sense that $\mathcal{T} \backslash \mathcal{M}$ does not entail any relevant Σ-consequence—a property known as *depletingness* [17].

Definition 3.14. *Let* $\equiv_{\Sigma}^{z}$ *be an inseparability relation. A* $\equiv_{\Sigma}^{z}$*-module* $\mathcal{M}$ *of* $\mathcal{T}$ *is* depleting *if* $\mathcal{T} \setminus \mathcal{M} \equiv_{\Sigma}^{z} \emptyset$.

Note that not all modules are depleting: for some relevant Σ-entailment φ we may have $\mathcal{M} \models \varphi$ (as required by the definition of module), but also $(\mathcal{T} \backslash \mathcal{M}) \models \varphi$, in which case $\mathcal{M}$ is not depleting. The following theorem establishes that all modules defined in Sect. 3.3 are depleting.

Theorem 3.15. $\mathcal{M}^{\chi_z}$ *is depleting for each* $z \in \{\mathsf{m}, \mathsf{q}, \mathsf{f}, \mathsf{i}, \mathsf{c}\}$.

Another application of modules is to optimise the computation of justifications: minimal subsets of a TBox that suffice to entail a given formula [14, 29].

Definition 3.16. *Let $\mathcal{T} \models \varphi$. A* justification for φ in $\mathcal{T}$ *is a minimal subset $\mathcal{T}' \subseteq \mathcal{T}$ such that $\mathcal{T}' \models \varphi$.*

Justifications are displayed in ontology development platforms as explanations of why an entailment holds, and tools typically compute all of them. Extracting justifications is a computationally intensive task, and locality-based modules have been used to reduce the size of the problem: if $\mathcal{T}'$ is a justification of φ in $\mathcal{T}$, then $\mathcal{T}'$ is contained in a locality module of $\mathcal{T}$ for $\Sigma = \mathsf{Sig}(\varphi)$. Our modules are also justification-preserving, and we can adjust our modules depending on what kind of first-order sentence φ is.

Theorem 3.17. *Let $\mathcal{T}'$ be a justification for a first-order sentence φ in $\mathcal{T}$ and let $\mathsf{Sig}(\varphi) \subseteq \Sigma$. Then, $\mathcal{T}' \subseteq \mathcal{M}^{\chi_\mathsf{m}}$. Additionally, the following properties hold: (i) if φ is a rule, then $\mathcal{T}' \subseteq \mathcal{M}^{\chi_\mathsf{q}}$; (ii) if φ is datalog, then $\mathcal{T}' \subseteq \mathcal{M}^{\chi_\mathsf{f}}$; and (iii) if φ is of the form $\mathsf{A}(\mathbf{x}) \to \mathsf{B}(\mathbf{x})$ or $A(\mathbf{x}) \to \bot$, then $\mathcal{T}' \subseteq \mathcal{M}^{\chi_\mathsf{i}}$; finally, if φ satisfies $A \in \Sigma, B \in \mathsf{Sig}(\mathcal{T})$, then $\mathcal{T}' \subseteq \mathcal{M}^{\chi_\mathsf{c}}$.*

3.6 Complexity of Module Extraction

We conclude this section by showing that our modules can be efficiently computed in most practically relevant cases.

Theorem 3.18. *Let m be a non-negative integer and L a class of TBoxes s.t. each rule in a TBox from L has at most m distinct universally quantified variables. The following problem is tractable: given $z \in \{\mathsf{q}, \mathsf{f}, \mathsf{i}, \mathsf{c}\}$, $\mathcal{T} \in L$, and $r \in \mathcal{T}$, decide whether $r \in \mathcal{M}^{\chi_z}$. The problem is tractable for arbitrary L if $z = \mathsf{m}$.*

We now provide a proof sketch for this result. Checking whether a datalog program $\mathcal{P}$ and an ABox $\mathcal{A}$ entail a fact is feasible in $\mathcal{O}(|\mathcal{P}| \cdot n^v)$, with n the number of constants in $\mathcal{P} \cup \mathcal{A}$ and v the maximum number of variables in a rule from $\mathcal{P}$ [7]. Thus, although datalog reasoning is exponential in the size of v (and hence of $\mathcal{P}$), it is tractable if v is bounded by a constant.

Given arbitrary $\mathcal{T}$ and Σ, and for $z \in \{\mathsf{m}, \mathsf{q}, \mathsf{f}, \mathsf{i}, \mathsf{c}\}$, the datalog program $\mathcal{P}^{\chi_z}$ can be computed in linear time in the size of $|\mathcal{T}|$. The number of constants n in χ_z (and hence in $\mathcal{P}^{\chi_z} \cup \mathcal{A}_0^z$) is linearly bounded in $|\mathcal{T}|$, whereas the maximum number of variables v coincides with the maximum number of universally quantified variables in a rule from $\mathcal{T}$. As shown in [31], computing the support of a fact in a datalog program is no harder than fact entailment, and thus module extraction in our approach is feasible in $\mathcal{O}(|\mathcal{T}| \cdot n^v)$, i.e., tractable for ontology languages where rules have a bounded number of variables (as is the case for most DLs). Finally, for $z = \mathsf{m}$ the setting χ_m involves a single constant $*$ and module extraction boils down to propositional datalog reasoning (a tractable problem regardless of $\mathcal{T}$).

3.7 Module Containment and Optimality

Intuitively, the more expressive the language for which preservation of consequences is required the larger modules need to be. The following proposition shows that our modules are consistent with this intuition.

Proposition 3.19. $\mathcal{M}^{\chi_{\mathsf{i}}} \subseteq \mathcal{M}^{\chi_{\mathsf{f}}} \subseteq \mathcal{M}^{\chi_{\mathsf{q}}} \subseteq \mathcal{M}^{\chi_{\mathsf{m}}} \subseteq \mathcal{M}^{\chi_{\mathsf{b}}}$; $\mathcal{M}^{\chi_{\mathsf{i}}} \subseteq \mathcal{M}^{\chi_{\mathsf{c}}} \subseteq \mathcal{M}^{\chi_{\mathsf{b}}}$.

Finally, we ask ourselves whether each χ_z with $z \in \{\mathsf{q}, \mathsf{f}, \mathsf{i}, \mathsf{m}, \mathsf{c}\}$ is optimal for its inseparability relation in the sense that there is no setting producing smaller modules. To make such statements precise we need to consider families of module settings, i.e., functions that assign a module setting to each pair of $\mathcal{T}$ and Σ.

Definition 3.20. *A* setting family *is a function Ψ that maps a TBox $\mathcal{T}$ and signature Σ to a module setting for $\mathcal{T}$ and Σ. Family Ψ is* uniform *if (i) for every Σ and pair $\mathcal{T}, \mathcal{T}'$ with the same number of existentially quantified variables, $\Psi(\mathcal{T}, \Sigma)$ and $\Psi(\mathcal{T}', \Sigma)$ are isomorphic; (ii) for every $\mathcal{T}$ and $\Sigma \subseteq \Sigma'$, $\Psi(\mathcal{T}, \Sigma)$ is a restriction of $\Psi(\mathcal{T}, \Sigma')$ to Σ. Let $z \in \{\mathsf{i}, \mathsf{f}, \mathsf{q}, \mathsf{m}, \mathsf{c}\}$; then Ψ is* z-admissible *if, for each $\mathcal{T}$ and Σ, $\mathcal{M}^{\Psi(\mathcal{T},\Sigma)}$ is a $\equiv^z_\Sigma$-module of $\mathcal{T}$. Finally, Ψ is* z-optimal *if $\mathcal{M}^{\Psi(\mathcal{T},\Sigma)} \subseteq \mathcal{M}^{\Psi'(\mathcal{T},\Sigma)}$ for every $\mathcal{T}$, Σ and every uniform Ψ' that is z-admissible.*

Uniformity ensures that settings do not depend on the specific shape of rules in $\mathcal{T}$, but rather only on Σ and the number of existentially quantified variables in $\mathcal{T}$. In turn, admissibility ensures that each setting yields a module. The (uniform and admissible) family Ψ^z for each setting χ^z in Sects. 3.3 and 3.4 is defined in the obvious way: for each $\mathcal{T}$ and Σ, $\Psi^z(\mathcal{T}, \Sigma)$ is the setting χ^z for $\mathcal{T}$ and Σ.

Theorem 3.21. *Ψ^z is z-optimal for $z \in \{\mathsf{i}, \mathsf{m}, \mathsf{c}\}$.*

In contrast, Ψ^{q} and Ψ^{f} are not optimal. To see this, let $\mathcal{T} = \{\mathsf{A}(x) \rightarrow \mathsf{B}(x), \mathsf{B}(x) \rightarrow \mathsf{A}(x)\}$ and $\Sigma = \{\mathsf{A}\}$. The empty TBox is fact inseparable from $\mathcal{T}$ since the only Σ-consequence of $\mathcal{T}$ is the tautology $\mathsf{A}(x) \rightarrow \mathsf{A}(x)$. However, $\mathcal{M}^{\chi_{\mathsf{f}}} = \mathcal{T}$ since fact $\mathsf{A}(a)$ is in $\mathcal{A}^{\mathsf{f}}_r$ and its support is included in the module. One can provide a family of settings that distinguishes tautological from non-tautological inferences; however, this family yields settings of exponential size in $|\mathcal{T}|$, which is undesirable in practice.

4 Evaluation

We have implemented a prototype system for module extraction, called PrisM, that uses RDFox [22] for datalog materialisation and exploits some code from PAGOdA [31] for computing the support of an entailed fact. We have evaluated PrisM on a set of ontologies identified by Glimm et al. [11] as non-trivial for reasoning. We have normalised all ontologies to make axioms equivalent to rules.[1]

We compared the size of our modules with the locality-based modules computed using the OWL API. We have followed the experimental methodology from [8] where two kinds of signatures are considered: *genuine signatures* corresponding to the signature of individual axioms, and *random signatures*. Unlike in [8], we extracted random signatures using a randomised graph sampling algorithm provided by RDFox. The advantage of this approach is that the resulting

[1] The ontologies used in our experiments are available for download at `https://krr-nas.cs.ox.ac.uk/2015/jair/PrisM/testOntologies.zip`.

	Galen-no-Fit $\mathcal{ALEH}$		Fly-anat.-Xp $\mathcal{ALERI}^+$		Fma-lite $\mathcal{ALEH}^+$		Gazetteer $\mathcal{ALE}^+$		Molecule-role $\mathcal{ALE}^+$		Snomed $\mathcal{SH}$		Nci-12.04e $\mathcal{SH}(D)$	
rules	66191		42107		168828		382158		153020		191891		193453	
	gen	rnd	gen	rnd	gen	rnd	gen	rnd	gen	rnd	gen	rnd	gen	rnd
$\bot$	14253	27771	22348	23139	47192	49345	214820	215886	143399	143448	433	11766	1140	16820
χ_{c}	9799	17879	112	595	12	402	<1	38	6	28	426	11342	390	7974
$\top\bot^*$	13749	27184	221	982	20	1658	9	1050	2	16	427	11762	1138	16817
χ_{m}	13686	27175	217	973	12	1450	8	1049	1	14	426	11651	1138	16817
χ_{q}	9448	18315	107	757	12	1450	8	1049	1	14	426	11644	385	8969
χ_{f}	5962	18225	80	664	1	74	<1	16	<1	5	426	11342	371	8415
χ_{i}	3279	17646	12	333	1	74	<1	16	<1	5	397	11342	120	6228
$\lvert\Sigma\rvert$	3	107	3	28	2	154	3	312	3	19	3	202	3	326

Table 1. Results for genuine and random signatures Σ

signatures are "semantically connected", which is likely to be the case in practical applications. For each type of signature and ontology, we took a sample of 400 runs and averaged module sizes. Random signatures were obtained from samples of size 1/1000 the size of the original ontology. We present results for the 7 largest ontologies from [11] in terms of signature (this selection is representative for the behaviour on the smaller ontologies; see [3]). All experiments were performed on a server with 2 Intel Xeon E5-2670 processors and 90GB of allocated RAM, running RDFox on 16 threads.

Table 1 summarises our results. We compared $\bot$-modules with the modules for χ_{c} (Sect. 3.4) and $\top\bot^*$-modules with those for χ_{m}, χ_{q}, χ_{f}, and χ_{i} (Sect. 3.3). We can see that module size consistently decreases as we consider weaker inseparability relations. In some cases, the modules for χ_{c} are over 3 orders of magnitude smaller than $\bot$-modules. The difference between $\top\bot^*$-modules and χ_{i} modules can also be considerable, reaching 2 orders of magnitude. In fact, χ_{c}, χ_{f}, and χ_{i} modules are sometimes empty, meaning that no two predicates in Σ are in an implication relationship (which can happen for large ontologies and small Σ). Also note that our modules for model inseparability slightly improve on $\top\bot^*$-modules. Finally, recall that our modules may not be minimal for their inseparability relation. Since techniques for extracting minimal modules are available only for model inseparability, and for restricted languages, we could not assess how close our modules are to minimal ones.

Computation times were generally higher for χ_{c} than for the other settings due to the larger signature involved in computing modules for χ_{c}. For random signatures, average times (over all settings) were 54s for GALEN, 2s for FLY, 5s for FMA, 13s for Gazetteer, 7s for Molecule, 32s for SNOMED, and 44s for NCI, which is considerably higher than for locality-based modules, but still acceptable for some applications. For genuine signatures, the times were accordingly lower.

It should be noted that PrisM currently relies on RDFox, which is not optimised for module extraction and is used in a black-box fashion. We conjecture that the performance of our approach can be considerably improved by dedicated systems. Another interesting task would be integrating our techniques in existing modular reasoners as well as in systems for justification extraction. Finally, the issue of optimality of our approach requires further investigation.

Acknowledgements

This work was supported by the Royal Society, the EPSRC projects MaSI^3, Score!, DBOnto, and ED3, and by the EU FP7 project OPTIQUE.

References

1. Armas Romero, A., Cuenca Grau, B., Horrocks, I.: MORe: Modular combination of OWL reasoners for ontology classification. In: ISWC. pp. 1–16 (2012)
2. Armas Romero, A., Kaminski, M., Cuenca Grau, B., Horrocks, I.: Ontology module extraction via datalog reasoning. In: AAAI. pp. 1410–1416 (2015)
3. Armas Romero, A., Kaminski, M., Cuenca Grau, B., Horrocks, I.: Module extraction in expressive ontology languages via datalog reasoning. J. Artif. Intell. Res. (2016), to appear
4. Cuenca Grau, B., Halaschek-Wiener, C., Kazakov, Y., Suntisrivaraporn, B.: Incremental classification of description logics ontologies. J. Autom. Reason. 44(4), 337–369 (2010)
5. Cuenca Grau, B., Horrocks, I., Kazakov, Y., Sattler, U.: Just the right amount: Extracting modules from ontologies. In: WWW. pp. 717–726 (2007)
6. Cuenca Grau, B., Horrocks, I., Kazakov, Y., Sattler, U.: Modular reuse of ontologies: Theory and practice. J. Artif. Intell. Res. 31, 273–318 (2008)
7. Dantsin, E., Eiter, T., Gottlob, G., Voronkov, A.: Complexity and expressive power of logic programming. ACM Comput. Surv. 33(3), 374–425 (2001)
8. Del Vescovo, C., Klinov, P., Parsia, B., Sattler, U., Schneider, T., Tsarkov, D.: Empirical study of logic-based modules: Cheap is cheerful. In: DL. pp. 144–155 (2013)
9. Del Vescovo, C., Parsia, B., Sattler, U., Schneider, T.: The modular structure of an ontology: Atomic decomposition. In: IJCAI. pp. 2232–2237 (2011)
10. Gatens, W., Konev, B., Wolter, F.: Lower and upper approximations for depleting modules of description logic ontologies. In: ECAI. pp. 345–350 (2014)
11. Glimm, B., Horrocks, I., Motik, B., Stoilos, G., Wang, Z.: HermiT: An OWL 2 reasoner. J. Autom. Reason. 53(3), 245–269 (2014)
12. Jiménez-Ruiz, E., Cuenca Grau, B.: LogMap: Logic-based and scalable ontology matching. In: ISWC. pp. 273–288 (2011)
13. Jiménez-Ruiz, E., Cuenca Grau, B., Sattler, U., Schneider, T., Berlanga Llavori, R.: Safe and economic re-use of ontologies: A logic-based methodology and tool support. In: ESWC. pp. 185–199 (2008)
14. Kalyanpur, A., Parsia, B., Horridge, M., Sirin, E.: Finding all justifications of OWL DL entailments. In: ISWC. pp. 267–280 (2007)
15. Konev, B., Lutz, C., Walther, D., Wolter, F.: Model-theoretic inseparability and modularity of description logic ontologies. Artif. Intell. 203, 66–103 (2013)
16. Kontchakov, R., Lutz, C., Toman, D., Wolter, F., Zakharyaschev, M.: The combined approach to ontology-based data access. In: IJCAI. pp. 2656–2661 (2011)
17. Kontchakov, R., Wolter, F., Zakharyaschev, M.: Logic-based ontology comparison and module extraction, with an application to DL-Lite. Artif. Intell. 174(15), 1093–1141 (2010)
18. Krötzsch, M., Rudolph, S., Hitzler, P.: ELP: Tractable rules for OWL 2. In: ISWC. pp. 649–664 (2008)

19. Ludwig, M.: Just: a tool for computing justifications w.r.t. $\mathcal{ELH}$ ontologies. In: ORE. pp. 1–7 (2014)
20. Lutz, C., Wolter, F.: Deciding inseparability and conservative extensions in the description logic EL. J. Symb. Comput. 45(2), 194–228 (2010)
21. Marnette, B.: Generalized schema-mappings: From termination to tractability. In: PODS. pp. 13–22 (2009)
22. Motik, B., Nenov, Y., Piro, R., Horrocks, I., Olteanu, D.: Parallel materialisation of datalog programs in centralised, main-memory RDF systems. In: AAAI. pp. 129–137 (2014)
23. Nortje, R., Britz, K., Meyer, T.: Reachability modules for the description logic $\mathcal{SRIQ}$. In: LPAR. pp. 636–652 (2013)
24. Sattler, U., Schneider, T., Zakharyaschev, M.: Which kind of module should I extract? In: DL (2009)
25. Seidenberg, J., Rector, A.L.: Web ontology segmentation: Analysis, classification and use. In: WWW. pp. 13–22 (2006)
26. Stefanoni, G., Motik, B., Horrocks, I.: Introducing nominals to the combined query answering approaches for $\mathcal{EL}$. In: AAAI. pp. 1177–1183 (2013)
27. Stuckenschmidt, H., Parent, C., Spaccapietra, S. (eds.): Modular Ontologies: Concepts, Theories and Techniques for Knowledge Modularization, LNCS, vol. 5445. Springer (2009)
28. Suntisrivaraporn, B.: Module extraction and incremental classification: A pragmatic approach for ontologies. In: ESWC. pp. 230–244 (2008)
29. Suntisrivaraporn, B., Qi, G., Ji, Q., Haase, P.: A modularization-based approach to finding all justifications for OWL DL entailments. In: ASWC. pp. 1–15 (2008)
30. Tsarkov, D., Palmisano, I.: Chainsaw: A metareasoner for large ontologies. In: ORE (2012)
31. Zhou, Y., Nenov, Y., Cuenca Grau, B., Horrocks, I.: Pay-as-you-go OWL query answering using a triple store. In: AAAI. pp. 1142–1148 (2014)