Ontology Module Extraction via Datalog Reasoning

Ana Armas Romero and **Mark Kaminski** and **Bernardo Cuenca Grau** and **Ian Horrocks**
Department of Computer Science, University of Oxford, UK

Abstract

Module extraction—the task of computing a (preferably small) fragment $\mathcal{M}$ of an ontology $\mathcal{T}$ that preserves entailments over a signature Σ—has found many applications in recent years. Extracting modules of minimal size is, however, computationally hard, and often algorithmically infeasible. Thus, practical techniques are based on approximations, where $\mathcal{M}$ provably captures the relevant entailments, but is not guaranteed to be minimal. Existing approximations, however, ensure that $\mathcal{M}$ preserves all second-order entailments of $\mathcal{T}$ w.r.t. Σ, which is stronger than is required in many applications, and may lead to large modules in practice. In this paper we propose a novel approach in which module extraction is reduced to a reasoning problem in datalog. Our approach not only generalises existing approximations in an elegant way, but it can also be tailored to preserve only specific kinds of entailments, which allows us to extract significantly smaller modules. An evaluation on widely-used ontologies has shown 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 great deal of attention in recent years (Stuckenschmidt, Parent, and Spaccapietra 2009; Cuenca Grau et al. 2008; Seidenberg and Rector 2006; Kontchakov, Wolter, and Zakharyaschev 2010; Gatens, Konev, and Wolter 2014; Del Vescovo et al. 2011; Nortje, Britz, and Meyer 2013), and modules have found a wide range of applications, including ontology reuse (Cuenca Grau et al. 2008; Jiménez-Ruiz et al. 2008), matching (Jiménez-Ruiz and Cuenca Grau 2011), debugging (Suntisrivaraporn et al. 2008; Ludwig 2014) and classification (Armas Romero, Cuenca Grau, and Horrocks 2012; Tsarkov and Palmisano 2012; Cuenca Grau et al. 2010).

The 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 (Konev et al. 2013). 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 (Lutz and Wolter 2010). Practical algorithms that ensure minimality of the extracted modules are known only for acyclic $\mathcal{ELI}$ (Konev et al. 2013) and DL-Lite (Kontchakov, Wolter, and Zakharyaschev 2010).

Practical module extraction techniques are typically based on sound approximations: they ensure that the extracted fragment $\mathcal{M}$ is a module (i.e., inseparable 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 (Cuenca Grau et al. 2007; 2008; Sattler, Schneider, and Zakharyaschev 2009); 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*, in the sense that $\mathcal{T} \setminus \mathcal{M}$ 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. Their main drawback is that the extracted modules can be rather large, which limits their usefulness in some applications (Del Vescovo et al. 2013). One way to address this issue is to develop techniques that more closely approximate minimal modules while still preserving properties *(i)–(iii)*. Efforts in this direction have confirmed that locality-based modules can be far from optimal in practice (Gatens, Konev, and Wolter 2014); however, these techniques apply only to rather restricted ontol-

Copyright © 2015, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

ogy 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 is required in 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, all practical module extraction techniques that are applicable to expressive ontology languages yield modules satisfying all three properties, and hence 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 (Suntisrivaraporn 2008), 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 (Nortje, Britz, and Meyer 2013) 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 (Motik et al. 2014). Our proof of concept evaluation shows that module size consistently decreases as we consider weaker inseparability relations, which could significantly improve the usefulness of modules in applications.

All our proofs are deferred to an extended version of the paper available online at `arXiv:1411.5313`.

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. It is assumed that the nullary falsehood predicate $\bot$ belongs to every Σ. 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) and we assume w.l.o.g. that $\top$ (resp. $\bot$) does not occur in rule heads (resp. in rule bodies). 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 at most one atom 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, and from edges in T to rules in $\mathcal{P}$, such that for each node v the following holds: 1. $\lambda(v) = \gamma$ if v is the root of T; 2. $\lambda(v) \in \mathcal{A}$ if v is a leaf; and 3. if v has children $w_1, \ldots, w_n$ then each edge from v to w_i is labelled by the same rule r and $\lambda(v)$ is a consequence of r and $\lambda(w_1), \ldots, \lambda(w_n)$. Forward-chaining is sound and complete: a fact γ is in the materialisation of $\mathcal{P} \cup \mathcal{A}$ iff it has a proof in $\mathcal{P} \cup \mathcal{A}$. 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. We say that 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 $\mathcal{T}$ (resp. of $\mathcal{T}'$) there exists a model $\mathcal{J}$ of $\mathcal{T}'$ (resp. of $\mathcal{T}$) with the same domain s.t. $\mathsf{A}^{\mathcal{I}} = \mathsf{A}^{\mathcal{J}}$ for each $\mathsf{A} \in \Sigma$.
- 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 (Kontchakov et al. 2011; Stefanoni, Motik, and Horrocks 2013; Zhou et al. 2014). In what follows, we fix an arbitrary TBox $\mathcal{T}$ and signature $\Sigma \subseteq \mathsf{Sig}(\mathcal{T})$. Unless otherwise stated, our definitions and theorems are parameterised by such $\mathcal{T}$ and Σ. We assume w.l.o.g. that rules in $\mathcal{T}$ do not share existentially quantified variables. For simplicity, we also assume that $\mathcal{T}$ contains no constants (all our results can be seamlessly extended).

3.1 Overview and Main Intuitions

Our overall strategy to extract a module $\mathcal{M}$ of $\mathcal{T}$ for an inseparability relation $\equiv_{\Sigma}^{z}$, 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

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

Figure 1: Example TBox $\mathcal{T}^{ex}$ with DL translation

(ii) turning disjunctions into conjunctions while splitting them into different rules, thus replacing each function-free disjunctive rule of the form $\varphi(\mathbf{x}) \to \bigvee_{i=1}^{n} \psi_i(\mathbf{x})$ with datalog rules $\varphi(\mathbf{x}) \to \psi_1(\mathbf{x}), \ldots, \varphi(\mathbf{x}) \to \psi_n(\mathbf{x})$.

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 some supporting rule in $\mathcal{P}$. In this way, $\mathcal{M}$ is fully determined by the substitution θ and the ABoxes $\mathcal{A}_0$ and $\mathcal{A}_r$.

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, suppose that $\mathcal{T}$ is $\mathcal{T}^{ex}$ 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 $\varphi = \mathsf{D}(x) \to \mathsf{G}(x)$ is the only non-trivial Σ-implication entailed by $\mathcal{T}^{ex}$; thus, for $\mathcal{M}$ to be a module we only require that $\mathcal{M} \models \varphi$.

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 transformed into clauses

$$r_4' = \mathsf{D}(x) \to \mathsf{S}(x, f(x_3)) \qquad r_4'' = \mathsf{D}(x) \to \mathsf{E}(f(x_3))$$

Hence $\mathcal{M} = \{r_4$–$r_6\}$ is a Σ-implication inseparable module of $\mathcal{T}^{ex}$, and as $\mathsf{G}(a)$ cannot be derived from any subset of $\{r_4$–$r_6\}$, $\mathcal{M}$ is also minimal.

In our approach, 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 following datalog rules in $\mathcal{P}$:

$$\mathsf{D}(x) \to \mathsf{S}(x, c) \qquad\qquad \mathsf{D}(x) \to \mathsf{E}(c)$$

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

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 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 $\mathsf{Sig}(\mathcal{T})$-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 $r = \varphi(\mathbf{x}) \to \exists \mathbf{y}.[\bigvee_{i=1}^{n} \psi_i(\mathbf{x}, \mathbf{y})]$ in $\mathcal{T}$, the rule $\varphi \to \bot$ if $n = 0$ and all rules $\varphi \to \gamma\theta$ 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. $\diamond$

3.3 Modules for each Inseparability Relation

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

Implication Inseparability The example in Section 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 $\varphi = \mathsf{A}(x_1, \ldots, x_n) \to \mathsf{B}(x_1, \ldots, x_n)$, 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 $\mathsf{A}(c_{\mathsf{A}}^1, \ldots, c_{\mathsf{A}}^n)$ for each n-ary predicate $\mathsf{A} \in \Sigma$, and $\mathcal{A}_r$ with a fact $\mathsf{B}(c_{\mathsf{A}}^1, \ldots, c_{\mathsf{A}}^n)$ for each pair of n-ary predicates $\{\mathsf{B}, \mathsf{A}\} \subseteq \Sigma$ with $\mathsf{B} \neq \mathsf{A}$.

Definition 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 $c_{\mathsf{A}}^1, \ldots, c_{\mathsf{A}}^n$ be also fresh constants. The setting $\chi_{\mathsf{i}} = \langle \theta^{\mathsf{i}}, \mathcal{A}_0^{\mathsf{i}}, \mathcal{A}_r^{\mathsf{i}} \rangle$ is defined as follows:

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

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 $\mathcal{T}$ (Krötzsch, Rudolph, and Hitzler 2008; Stefanoni, Motik, and Horrocks 2013). 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. $\mathcal{M}^{\chi_{\mathsf{i}}} \equiv_{\Sigma}^{\mathsf{i}} \mathcal{T}$.

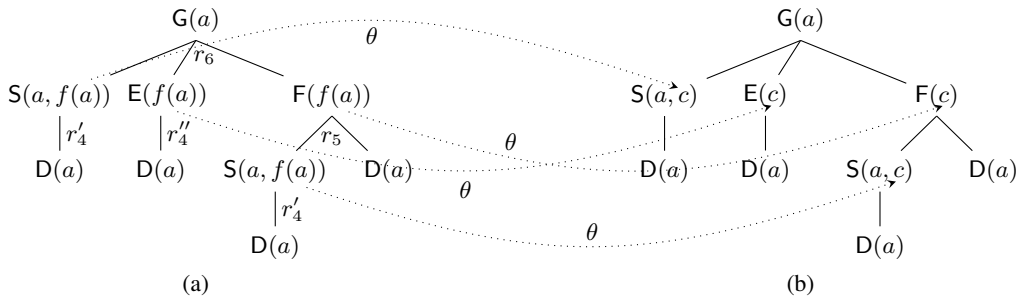

Figure 2: Proofs of $\mathsf{G}(a)$ from $\mathsf{D}(a)$ in (a) $\mathcal{T}^{ex}$ and (b) the corresponding datalog program

Fact Inseparability The setting χ_{i} in Def. 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, $\mathcal{M}^{\chi_{\mathsf{i}}}$ is only guaranteed to 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 (Marnette 2009). Every Σ-ABox can be homomorphically mapped into the critical Σ-ABox. In this way, we can show that all proofs of a Σ-fact in $\mathcal{T} \cup \mathcal{A}$ can be embedded in a proof of a relevant fact in $\mathcal{P}^{\chi} \cup \mathcal{A}_0$.

Definition 4. Let constants c_{y_i} be as in Def. 2, and let $*$ be a fresh constant. The setting $\chi_{\mathsf{f}} = \langle \theta^{\mathsf{f}}, \mathcal{A}_0^{\mathsf{f}}, \mathcal{A}_r^{\mathsf{f}} \rangle$ is defined as follows: *(i)* $\theta^{\mathsf{f}} = \theta^{\mathsf{i}}$, *(ii)* $\mathcal{A}_0^{\mathsf{f}} = \{\, \mathsf{A}(*, \ldots, *) \mid \mathsf{A} \in \Sigma \,\}$, and *(iii)* $\mathcal{A}_r^{\mathsf{f}} = \mathcal{A}_0^{\mathsf{f}}$ ◇

The datalog programs for χ_{i} and χ_{f} coincide and hence the only difference between the two settings is in the definition of their corresponding ABoxes. In our example, both $\mathcal{A}_0^{\mathsf{f}}$ and $\mathcal{A}_r^{\mathsf{f}}$ 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 5. $\mathcal{M}^{\chi_{\mathsf{f}}} \equiv_{\Sigma}^{\mathsf{f}} \mathcal{T}$.

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 that $\mathcal{T}^{ex} \cup \mathcal{A} \models q$ (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 it does not contain any 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 may 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 rule 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 6. Let constants c_{y_i} and $*$ be as in Def. 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 all Σ-facts $\mathsf{A}(a_1, \ldots, a_n)$ with each a_j either a constant c_{y_i} or $*$. ◇

Correctness is established by the following theorem:

Theorem 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 $\mathcal{M}^{\chi_{\mathsf{q}}}$. This interpretation, however, cannot be extended to a model of r_3 (and hence of $\mathcal{T}$) without reinterpreting A, R or D.

The main insight behind locality and reachability modules is to ensure that each model of the module can be extended to a model of $\mathcal{T}$ in a uniform way. Specifically, each model of a $\top\bot^*$-locality or $\top\bot^*$-reachability module can be extended to a model of $\mathcal{T}$ by interpreting all other predicates A as either $\emptyset$ or $(\Delta^{\mathcal{I}})^n$ with n the arity of A. 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 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 variable to the fresh constant $*$ and $\mathcal{A}_0^{\mathsf{m}} = \mathcal{A}_r^{\mathsf{m}} = \mathcal{A}_0^{\mathsf{f}}$. ◇

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 $\mathcal{M}^{\chi_{\mathsf{m}}} = \{r_1, r_2, r_3\}$.

To show that $\mathcal{M}^{\chi_{\mathsf{m}}}$ is a $\equiv_{\Sigma}^{\mathsf{m}}$-module we prove that all models $\mathcal{I}$ of $\mathcal{M}^{\chi_{\mathsf{m}}}$ can be extended to a model of $\mathcal{T}$ as follows: *(i)* predicates not occurring in the materialisation of $\mathcal{P}^{\chi_{\mathsf{m}}} \cup \mathcal{A}_0^{\mathsf{m}}$ are interpreted as empty; *(ii)* predicates in the support of χ_{m} (and hence occurring in $\mathcal{M}^{\chi_{\mathsf{m}}}$) are interpreted as in $\mathcal{I}$; and *(iii)* all other predicates A are interpreted as $(\Delta^{\mathcal{I}})^n$ with n the arity of A.

Theorem 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 (Armas Romero, Cuenca Grau, and Horrocks 2012; Tsarkov and Palmisano 2012; Cuenca Grau et al. 2010). In this case, it is not only required that modules are implication inseparable from $\mathcal{T}$, but also that they preserve all implications $\mathsf{A}(\mathbf{x}) \rightarrow \mathsf{B}(\mathbf{x})$ with $\mathsf{A} \in \Sigma$ but $\mathsf{B} \notin \Sigma$. This requirement can be captured as given next.

Definition 10. TBoxes $\mathcal{T}$ and $\mathcal{T}'$ are Σ-*classification inseparable* ($\mathcal{T} \equiv^{\mathsf{c}}_{\Sigma} \mathcal{T}'$) if for each φ of the form $\mathsf{A}(\mathbf{x}) \rightarrow \mathsf{B}(\mathbf{x})$ with $\mathsf{A} \in \Sigma$, and $\mathsf{B} \in \mathsf{Sig}(\mathcal{T} \cup \mathcal{T}')$ we have $\mathcal{T} \models \varphi$ iff $\mathcal{T}' \models \varphi$. ◇

Classification inseparability is a stronger requirement than implication inseparability. For $\mathcal{T} = \{\mathsf{A}(x) \rightarrow \mathsf{B}(x)\}$ 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 and Chainsaw 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 $\mathsf{A}(x) \rightarrow \mathsf{B}(x)$ but $\mathcal{M}$ does not. We can cast $\bot$-modules in our framework with the following setting, which extends χ_{m} in Def. 8 by also considering as relevant facts involving predicates not in Σ.

Definition 11. The setting $\chi_{\mathsf{b}} = \langle \theta^{\mathsf{b}}, \mathcal{A}^{\mathsf{b}}_0, \mathcal{A}^{\mathsf{b}}_r \rangle$ is as follows: $\theta^{\mathsf{b}} = \theta^{\mathsf{m}}$, $\mathcal{A}^{\mathsf{b}}_0 = \mathcal{A}^{\mathsf{m}}_0$, and $\mathcal{A}_r$ consists of 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. 2 to ensure classification inseparability. As in the case of χ_{b} in Def. 11 the only required modification is to also consider as relevant facts involving predicates outside Σ.

Definition 12. Setting $\chi_{\mathsf{c}} = (\theta^{\mathsf{c}}, \mathcal{A}^{\mathsf{c}}_0, \mathcal{A}^{\mathsf{c}}_r)$ is as follows: $\theta^{\mathsf{c}} = \theta^{\mathsf{i}}$, $\mathcal{A}^{\mathsf{c}}_0 = \mathcal{A}^{\mathsf{i}}_0$, and $\mathcal{A}^{\mathsf{c}}_r$ consists of all facts $\mathsf{B}(c^1_{\mathsf{A}}, \ldots, c^n_{\mathsf{A}})$ s.t. $\mathsf{A} \neq \mathsf{B}$ are n-ary predicates, $A \in \Sigma$ and $\mathsf{B} \in \mathsf{Sig}(\mathcal{T})$. ◇

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

Theorem 13. $\mathcal{M}^{\chi_{\mathsf{c}}} \equiv^{\mathsf{c}}_{\Sigma} \mathcal{T}$.

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 it is desirable that modules satisfy additional requirements.

In ontology reuse scenarios, it is sometimes desirable that a module $\mathcal{M}$ does not "leave any relevant information behind", in the sense that $\mathcal{T} \setminus \mathcal{M}$ does not entail any relevant Σ-consequence—a property referred to as *depletingness* (Kontchakov, Wolter, and Zakharyaschev 2010).

Definition 14. Let $\equiv^{\mathsf{z}}_{\Sigma}$ be an inseparability relation. A $\equiv^{\mathsf{z}}_{\Sigma}$-module $\mathcal{M}$ of $\mathcal{T}$ is *depleting* if $\mathcal{T} \setminus \mathcal{M} \equiv^{\mathsf{z}}_{\Sigma} \emptyset$. ◇

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

Theorem 15. *$\mathcal{M}^{\chi_{\mathsf{z}}}$ is depleting for each $\mathsf{z} \in \{\mathsf{m}, \mathsf{q}, \mathsf{f}, \mathsf{i}, \mathsf{c}\}$.*

Another common application of modules is to optimise the computation of justifications: minimal subsets of a TBox that are sufficient to entail a given formula (Kalyanpur et al. 2007; Suntisrivaraporn et al. 2008).

Definition 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 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 $\mathsf{A}(\mathbf{x}) \rightarrow \mathsf{B}(\mathbf{x})$, 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 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 $\mathsf{z} \in \{\mathsf{q}, \mathsf{f}, \mathsf{i}, \mathsf{c}\}$, $\mathcal{T} \in L$, and $r \in \mathcal{T}$, decide whether $r \in \mathcal{M}^{\chi_{\mathsf{z}}}$. The problem is solvable in polynomial time for arbitrary classes L of TBoxes if $\mathsf{z} = \mathsf{m}$.*

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 $\mathcal{P}$ (Dantsin et al. 2001). 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 $\mathsf{z} \in \{\mathsf{m}, \mathsf{q}, \mathsf{f}, \mathsf{i}, \mathsf{c}\}$, the datalog program $\mathcal{P}^{\chi_{\mathsf{z}}}$ can be computed in linear time in the size of $|\mathcal{T}|$. The number of constants n in χ_{z} (and hence in $\mathcal{P}^{\chi_{\mathsf{z}}} \cup \mathcal{A}^{\mathsf{z}}_0$) 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 (Zhou et al. 2014), 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 $\mathcal{O}(|\mathcal{T}| \cdot n^v)$, and thus tractable for ontology languages where rules have a bounded number of variables (as is the case for most DLs). Finally, if $\mathsf{z} = \mathsf{m}$ the setting χ_{m} involves a single

constant $*$ and module extraction boils down to reasoning in propositional datalog (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 19. *$\mathcal{M}^{\chi_{\mathsf{i}}} \subseteq \mathcal{M}^{\chi_{\mathsf{f}}} \subseteq \mathcal{M}^{\chi_{\mathsf{q}}} \subseteq \mathcal{M}^{\chi_{\mathsf{m}}} \subseteq \mathcal{M}^{\chi_{\mathsf{b}}}$ and $\mathcal{M}^{\chi_{\mathsf{i}}} \subseteq \mathcal{M}^{\chi_{\mathsf{c}}} \subseteq \mathcal{M}^{\chi_{\mathsf{b}}}$*

As already discussed, these containment relations are strict for many $\mathcal{T}$ and Σ.

We conclude this section by discussing whether each χ_z with $z \in \{\mathsf{q}, \mathsf{f}, \mathsf{i}, \mathsf{c}\}$ is optimal for its inseparability relation in the sense that there is no setting that produces smaller modules. To make optimality statements precise we need to consider families of module settings, that is, functions that assign a module setting for each pair of $\mathcal{T}$ and Σ.

Definition 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. We say that Ψ is *uniform* if for every Σ and pair of TBoxes $\mathcal{T}, \mathcal{T}'$ with the same number of existentially quantified variables $\Psi(\mathcal{T}, \Sigma) = \Psi(\mathcal{T}', \Sigma)$. Let $z \in \{\mathsf{i}, \mathsf{f}, \mathsf{q}, \mathsf{c}\}$; 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. $\diamond$

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 corresponding to each setting χ^z in Sections 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.

The next theorem shows that Ψ^z is optimal for implication and classification inseparability.

Theorem 21. *Ψ^z is z-optimal for $z \in \{\mathsf{i}, \mathsf{c}\}$.*

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. We can provide a family of settings that distinguishes tautological from non-tautological inferences (see extended version of the paper); however, this family yields settings of exponential size in $|\mathcal{T}|$, which is undesirable in practice.

4 Proof of Concept Evaluation

We have implemented a prototype system for module extraction that uses RDFox for datalog materialisation (Motik et al. 2014). Additionally, the ontology reasoner PAGOdA (Zhou et al. 2014) provides functionality for computing the support of an entailed fact in datalog, which we have adapted for computing modules. We have evaluated our system on representative ontologies, including SNOMED (SCT), Fly Anatomy (FLY), the Gene Ontology (GO) and BioModels (BM).[1] SCT is expressed in the EL profile of OWL 2, whereas FLY, GO and BM require expressive DLs (Horn-$\mathcal{SRI}$, $\mathcal{SHIQ}$ and $\mathcal{SRIQ}$, respectively). We have normalised all ontologies to make axioms equivalent to rules.

	FLY		SCT		GO		BM	
rules	19,830		112,833		145,083		462,120	
	gen	rnd	gen	rnd	gen	rnd	gen	rnd
$\bot, \chi_{\mathsf{b}}$	242	847	242	5,196	1,461	12,801	1,010	64,320
χ_{c}	112	446	230	3,500	309	3,990	285	14,273
$\top\bot^*$	219	796	233	5,182	1,437	12,747	963	62,897
χ_{m}	215	789	233	5,182	1,431	12,724	955	62,286
χ_{q}	109	480	123	2,329	267	4,146	447	16,905
χ_{f}	76	476	24	2,258	162	4,142	259	14,043
χ_{i}	8	7	15	235	103	2,429	105	4,107
$\lvert\Sigma\rvert$	2.7	7.8	2.7	41.9	2.4	56.6	2.5	210.5

Table 1: Results for genuine and random signatures Σ

We compared the size of our modules with the locality-based modules computed using the OWL API. We have followed the experimental methodology from (Del Vescovo et al. 2013) where two kinds of signatures are considered: *genuine signatures* corresponding to the signature of individual axioms, and *random signatures* with a given probability for a symbol to be included. For each type of signature and ontology, we took a sample of 400 runs and averaged module sizes. For random signatures we considered a probability of $1/1000$. All experiments have been performed on a server with two Intel Xeon E5-2643 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} (Section 3.4) and $\top\bot^*$-modules with those for χ_{m}, χ_{q}, χ_{f}, and χ_{i} (Section 3.3). We can see that module size consistently decreases as we consider weaker inseparability relations. In particular, the modules for χ_{c} can be 4 times smaller than $\bot$-modules. The difference between $\top\bot^*$-modules and χ_{i} modules is even bigger, especially in the case of FLY. In fact, χ_{i} modules are sometimes empty, which is not surprising since two predicates in a large ontology are unlikely to be in an implication relationship. Also note that our modules for semantic 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 and hence the quality of our approximation.

Computation times were comparable for all settings χ^z with times being slightly higher for χ_{i} and χ_{c} as they involved a larger number of constants. Furthermore, extraction times were comparable to locality-based modules for genuine signatures with average times of $0.5s$ for FLY, $0.9s$ for SCT, 4.2 for GO and $5s$ for BM.

[1] The ontologies used in our tests are available for download at `http://www.cs.ox.ac.uk/isg/ontologies/UID/` under IDs 794 (FLY), 795 (SCT), 796 (GO) and 797 (BM).

5 Conclusion and Future Work

We have proposed a novel approach to module extraction by exploiting off-the-shelf datalog reasoners, which allows us to efficiently compute approximations of minimal modules for different inseparability relations. Our results open the door to significant improvements in common applications of modules, such as computation of justifications, modular and incremental reasoning and ontology reuse, which currently rely mostly on locality-based modules.

Our approach is novel, and we see many interesting open problems. For example, the issue of optimality requires further investigation. Furthermore, it would be interesting to integrate our extraction techniques in existing modular reasoners as well as in systems for justification extraction.

Acknowledgements

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

References

Armas Romero, A.; Cuenca Grau, B.; and Horrocks, I. 2012. MORe: Modular combination of OWL reasoners for ontology classification. In *ISWC*, 1–16.

Cuenca Grau, B.; Horrocks, I.; Kazakov, Y.; and Sattler, U. 2007. Just the right amount: extracting modules from ontologies. In *WWW*, 717–726.

Cuenca Grau, B.; Horrocks, I.; Kazakov, Y.; and Sattler, U. 2008. Modular reuse of ontologies: Theory and practice. *J. Artif. Intell. Res.* 31.

Cuenca Grau, B.; Halaschek-Wiener, C.; Kazakov, Y.; and Suntisrivaraporn, B. 2010. Incremental classification of description logics ontologies. *J. Autom. Reasoning* 44(4):337–369.

Dantsin, E.; Eiter, T.; Gottlob, G.; and Voronkov, A. 2001. Complexity and expressive power of logic programming. *ACM Comput. Surv.* 33(3):374–425.

Del Vescovo, C.; Parsia, B.; Sattler, U.; and Schneider, T. 2011. The modular structure of an ontology: Atomic decomposition. In *IJCAI*, 2232–2237.

Del Vescovo, C.; Klinov, P.; Parsia, B.; Sattler, U.; Schneider, T.; and Tsarkov, D. 2013. Empirical study of logic-based modules: Cheap is cheerful. In *DL*, 144–155.

Gatens, W.; Konev, B.; and Wolter, F. 2014. Lower and upper approximations for depleting modules of description logic ontologies. In *ECAI*.

Jiménez-Ruiz, E., and Cuenca Grau, B. 2011. Logmap: Logic-based and scalable ontology matching. In *ISWC*, 273–288.

Jiménez-Ruiz, E.; Cuenca Grau, B.; Sattler, U.; Schneider, T.; and Berlanga Llavori, R. 2008. Safe and economic re-use of ontologies: A logic-based methodology and tool support. In *ESWC*, 185–199.

Kalyanpur, A.; Parsia, B.; Horridge, M.; and Sirin, E. 2007. Finding all justifications of OWL DL entailments. In *ISWC*.

Konev, B.; Lutz, C.; Walther, D.; and Wolter, F. 2013. Model-theoretic inseparability and modularity of description logic ontologies. *Artif. Intell.* 203:66–103.

Kontchakov, R.; Lutz, C.; Toman, D.; Wolter, F.; and Zakharyaschev, M. 2011. The combined approach to ontology-based data access. In *IJCAI*, 2656–2661.

Kontchakov, R.; Wolter, F.; and Zakharyaschev, M. 2010. Logic-based ontology comparison and module extraction, with an application to DL-Lite. *Artif. Intell.* 174(15):1093–1141.

Krötzsch, M.; Rudolph, S.; and Hitzler, P. 2008. ELP: tractable rules for OWL 2. In *ISWC*.

Ludwig, M. 2014. Just: a tool for computing justifications w.r.t. $\mathcal{ELH}$ ontologies. In *ORE*, 1–7.

Lutz, C., and Wolter, F. 2010. Deciding inseparability and conservative extensions in the description logic EL. *J. Symb. Comput.* 45(2):194–228.

Marnette, B. 2009. Generalized schema-mappings: from termination to tractability. In *PODS*.

Motik, B.; Nenov, Y.; Piro, R.; Horrocks, I.; and Olteanu, D. 2014. Parallel materialisation of datalog programs in centralised, main-memory RDF systems. In *AAAI*, 129–137.

Nortje, R.; Britz, K.; and Meyer, T. 2013. Reachability modules for the description logic $\mathcal{SRIQ}$. In *LPAR*, 636–652.

Sattler, U.; Schneider, T.; and Zakharyaschev, M. 2009. Which kind of module should I extract? In *DL*.

Seidenberg, J., and Rector, A. L. 2006. Web ontology segmentation: analysis, classification and use. In *WWW*, 13–22.

Stefanoni, G.; Motik, B.; and Horrocks, I. 2013. Introducing nominals to the combined query answering approaches for EL. In *AAAI*.

Stuckenschmidt, H.; Parent, C.; and Spaccapietra, S., eds. 2009. *Modular Ontologies: Concepts, Theories and Techniques for Knowledge Modularization*, volume 5445 of *LNCS*. Springer.

Suntisrivaraporn, B.; Qi, G.; Ji, Q.; and Haase, P. 2008. A modularization-based approach to finding all justifications for OWL DL entailments. In *ASWC*, 1–15.

Suntisrivaraporn, B. 2008. Module extraction and incremental classification: A pragmatic approach for ontologies. In *ESWC*, 230–244.

Tsarkov, D., and Palmisano, I. 2012. Chainsaw: a metareasoner for large ontologies. In *ORE*.

Zhou, Y.; Nenov, Y.; Cuenca Grau, B.; and Horrocks, I. 2014. Pay-as-you-go OWL query answering using a triple store. In *AAAI*, 1142–1148.