Ontology Module Extraction and Applications to Ontology Classification

Ana Armas Romero

Mansfield College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Trinity 2015

To my grandparents.

Acknowledgements

First and foremost, I want to express my deepest gratitude to my supervisors, Bernardo Cuenca Grau and Ian Horrocks. To Bernardo, for dedicating countless hours to discussing ideas and to teaching me how to write good papers. To Ian, for always finding the time for me when I needed advice and for patiently helping me put things in perspective in times of struggle. I would also like to thank them both for sharing their ideas and generously guiding me in this journey. I feel extremely lucky to have had the opportunity to work with them.

I want to thank Mark for his support inside and outside the office. Inside the office, our collaboration has been very enriching and I have learned a great deal from him. Outside the office, he has been an endless source of encouragement, inspiration and fun.

I am also grateful to my other colleagues and friends in the Knowledge Representation and Reasoning group for many interesting discussions, for their generous help on numerous occasions, and for always making lunchtime at the office something to look forward to.

Last but not least, I am grateful to my family for always believing in me.

This work was supported by the Engineering and Physical Sciences Research Council (EPSRC).

Abstract

Module extraction is the task of computing a (preferably small) fragment $\mathcal{M}$ of an ontology $\mathcal{O}$ that preserves a class of entailments over a signature of interest Σ. Existing practical approaches ensure that $\mathcal{M}$ preserves all second-order entailments of $\mathcal{O}$ over Σ, which is a stronger condition than is required in many applications. In the first part of this thesis, we propose a novel approach to module extraction which, based on a reduction to a datalog reasoning problem, makes it possible to compute modules that are tailored to preserve only specific kinds of entailments. This leads to obtaining modules that are often significantly smaller than those produced by other practical approaches, as shown in an empirical evaluation.

In the second part of this thesis, we consider the application of module extraction to the optimisation of ontology classification. Classification is a fundamental reasoning task in ontology design, and there is currently a wide range of reasoners that provide this service. Reasoners aimed at so-called lightweight ontology languages are much more efficient than those aimed at more expressive ones, but they do not offer completeness guarantees for ontologies containing axioms outside the relevant language. We propose an original approach to classification based on exploiting module extraction techniques to divide the workload between a general purpose reasoner and a more efficient reasoner for a lightweight language in such a way that the bulk of the workload is assigned to the latter. We show how the proposed approach can be realised using two particular module extrac-

tion techniques, including the one presented in the first part of the thesis. Furthermore, we present the results of an empirical evaluation that shows that this approach can lead to a significant performance improvement in many cases.

Contents

List of Figures

Part I

Foundations

Chapter 1

Introduction

Module extraction is the task of computing, given an ontology $\mathcal{O}$ and a signature of interest Σ, a (preferably small) subset $\mathcal{M}$ of $\mathcal{O}$ (a *module*) that preserves a class of entailments of $\mathcal{O}$ over the symbols in Σ. The class of entailments whose preservation is required may vary from one application to another. In all cases, the subset $\mathcal{M}$ needs to be indistinguishable from $\mathcal{O}$ w.r.t. Σ for the application at hand, making it safe for it to rely on $\mathcal{M}$ instead of $\mathcal{O}$ for tasks that concern only symbols from Σ.

Module extraction has received a great deal of attention in recent years [79, 88, 21, 56, 26, 73, 29], and modules have found numerous applications in ontology reuse [21, 43], matching [41], debugging [90, 61] and classification [94, 18].

The preservation of relevant entailments is usually formalised via *inseparability relations* [52]. The strongest such notion is *model* inseparability, which requires that it must be possible to turn any model of $\mathcal{M}$ into a model of $\mathcal{O}$ by (re-)interpreting only symbols outside Σ; in this case, $\mathcal{M}$ preserves all second-order entailments of $\mathcal{O}$ w.r.t. Σ [53]. A weaker and more flexible notion is that of *deductive* inseparability, which only requires $\mathcal{O}$ and $\mathcal{M}$ to entail the same Σ-formulas in a particular *query language*. Unfortunately, the decision problems associated with module extraction are generally of high complexity or even undecidable, especially for expressive ontology

languages. For model inseparability, checking whether $\mathcal{M}$ is a Σ-module of $\mathcal{O}$ is undecidable even if $\mathcal{O}$ is restricted to the lightweight description logic (DL) $\mathcal{EL}$ [53], for which standard reasoning is tractable [7]. For deductive inseparability, the problem is typically decidable for lightweight DLs and "reasonable" query languages, albeit still of high worst-case complexity; e.g., it is ExpTime-complete for $\mathcal{EL}$ if we consider concept inclusions as the query language [63]. The complexity of module extraction for ontology languages that are not based on DLs, such as variants of datalog$^{\pm}$ [16], remains largely unexplored.

Small modules are preferable to larger ones in all applications. Unfortunately, minimal module extraction requires decidability of the associated module checking problem. In particular, practical algorithms that ensure minimality of the extracted modules are known only for $\mathcal{ELI}$ ontologies satisfying a particular acyclicity condition [53] and for some dialects of DL-Lite [56]. Practical module extraction techniques are typically based on sound approximations, which ensure that the computed fragment $\mathcal{M}$ is a module (i.e., inseparable from $\mathcal{O}$ w.r.t. Σ), but provide no minimality guarantee. The most popular such techniques are based on a family of polynomially checkable conditions based on the notion of syntactic locality [19, 21, 77]. Each locality-based module $\mathcal{M}$ enjoys a number of desirable properties w.r.t. the signature Σ of interest:

(P1) It is *model inseparable* from $\mathcal{O}$, thus preserving all second-order Σ-entailments of $\mathcal{O}$.

(P2) It is *depleting*, in the sense that $\mathcal{O} \setminus \mathcal{M}$ is inseparable from the empty ontology; this implies that no relevant information is "left behind" after extracting $\mathcal{M}$ from $\mathcal{O}$.

(P3) It is *self-contained*, in that it preserves not only the relevant entailments over Σ, but also w.r.t. all other symbols in its signature.

(P4) It is *justification-preserving*, in the sense that each subset-minimal fragment of $\mathcal{O}$ preserving a Σ-entailment (each justification) is contained in $\mathcal{M}$.

(P5) It can be computed efficiently, even for ontologies in expressive DLs.

Model inseparability ensures that modules can be used regardless of the query language relevant to the application at hand. Depletingness and self-containment have been identified as important properties for ontology reuse and modular ontology development tasks [77, 43]. Finally, the preservation of justifications enables the use of modules for optimising debugging and explanation services [78, 44].

Locality-based module extraction 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 [25]. One way to address this issue is to develop techniques that approximate minimal modules more closely, while still fulfilling properties (P1)–(P4). Efforts in this direction have confirmed that locality-based modules can be far from optimal in practice [29]; however, these techniques apply only to rather restricted ontology languages and utilise algorithms with high worst-case complexity.

Another approach to computing smaller modules would be to weaken properties (P1)–(P4), which are stronger than many applications require. In particular, model inseparability is a very strong condition, and deductive inseparability w.r.t. a query language suitable for the application at hand would usually suffice.

1.1 Contributions of this Thesis

In this thesis we contribute to the field of ontology module extraction in two ways. On the one hand, we present a novel module extraction technique that can be parameterised by the class of consequences that must be preserved. On the other hand, we propose a new way to exploit module extraction to optimise ontology reasoning.

1.1.1 Module Extraction

Our novel approach, which has already been published in [5], reduces module extraction to a reasoning problem in the basic rule-based language *datalog* [1, 24]. The connection between module extraction and datalog was first observed in [89], where it was shown that syntactic locality $\bot$-module extraction for $\mathcal{EL}$ ontologies could be reduced to propositional datalog reasoning. Our approach takes this connection much farther, and generalises locality-based modules in an elegant way. The key distinguishing features of our approach are as follows:

- It is applicable, not only to ontology languages based on description logics, but also to more expressive rule-based knowledge representation formalisms that extend datalog with existential quantification and disjunction in the head of rules [16, 14, 2].

- It is sensitive to the different inseparability relations proposed in the literature; in particular, we can extract deductively inseparable modules w.r.t. a query language tailored to the specific requirements of the application at hand. This allows us to relax property (P1) and extract significantly smaller modules.

- In all cases, our modules are depleting and capture all justifications of relevant entailments; moreover, our approach can be adapted to either ensure or dispense with self-containment, depending on the application's needs.

- It not only ensures tractability of module extraction for DL-based ontology languages, but also enables the use of highly scalable off-the-shelf datalog engines.

We have implemented this approach in the PrisM module extractor,[1] which relies on the RDFox datalog engine [67]. Our evaluation over complex, real-world, ontologies shows that weakening the inseparability relation considered can substantially

[1] `http://www.cs.ox.ac.uk/isg/tools/PrisM/`

decrease module size (sometimes even by one or more orders of magnitude), which could significantly improve the usefulness of modules in several applications.

1.1.2 Modular Reasoning

We have applied module extraction to optimise the task of ontology classification. Classification is the problem of computing the subsumption hierarchy between the terms in the input ontology. It is a fundamental ontology reasoning service: not only can it be used by the developer to detect modelling errors, but it can also provide the user with a succinct roadmap between the terms in the ontology. For expressive ontology languages, however, the decision problems associated with classification have a very high worst-case complexity; in particular, subsumption with respect to a $\mathcal{SROIQ}$ ontology is known to be a 2NExpTime-complete problem [47, 23]. On the other hand, there exist a number of highly optimised polynomial classification algorithms for more restricted, *lightweight* ontology languages, such as $\mathcal{EL}^{++}$ [7].

Despite the discouraging complexity results, general purpose reasoners implementing classification algorithms for expressive ontology languages (such as HermiT [33], Pellet [82], Fact++ [92], RacerPro [36] or, more recently, Konclude [87]) have been highly optimised and can successfully deal with a large number of ontologies. Some ontologies, however, remain challenging for them. Such ontologies still typically contain only a relatively small number of axioms that are outside a given lightweight language. For example, the Foundational Model of Anatomy (FMA) ontology, known to be hard to classify for state-of-the-art reasoners, only contains 107 non-$\mathcal{EL}^{++}$ axioms out of a total of 211,369 axioms. Nevertheless, algorithms for lightweight languages are not guaranteed to completely classify ontologies that do not fall entirely within their corresponding language, since even a single axiom can have a significant impact on the ontology's subsumption hierarchy.

The modular classification technique proposed in this thesis, previously published

in [4], consists in using module extraction techniques to combine an efficient reasoner for a lightweight logic $\mathcal{L}$ (or $\mathcal{L}$-reasoner) and a general purpose reasoner for an expressive ontology language. We present this technique from a high level point of view, and also show how it can be realised using two particular module extraction techniques, including the one also presented in this thesis.

The proposed technique provides the following compelling features:

- It is general and flexible, as it is not tied to a particular ontology language $\mathcal{L}$ or to a particular reasoner or reasoning technique, or even to a module extraction technique.

- It is easy to implement, as reasoners are combined in a black-box manner, with no modification of their internals being required.

- It exhibits "pay-as-you-go" behaviour when an $\mathcal{L}$-ontology is extended with axioms outside $\mathcal{L}$: on the one hand, the use of an $\mathcal{L}$-reasoner is not precluded by the extension; on the other hand, the performance degrades gradually with the number of additional non-$\mathcal{L}$ axioms.

We have implemented this approach in the MORe system,[2] integrating the OWL 2 reasoner HermiT with the OWL 2 EL reasoner ELK and the datalog reasoner RDFox. An evaluation over a representative set of real-world ontologies illustrates the potential of the approach for optimising ontology classification.

1.2 Structure of this Thesis

The contents of this thesis are organised as follows:

- In the remainder of Part I we provide the foundations for the technical material presented in Parts II and III. In Chapter 2 we introduce the ontology languages

[2] `http://www.cs.ox.ac.uk/isg/tools/MORe/`

that we will consider throughout the rest of the thesis, namely first-order logic existential disjunctive rules, and description logics [8]; we also introduce the hyperresolution calculus for first-order logic [11], which we will exploit in many of our technical results. In Chapter 4 we recapitulate the key notions of inseparability relation and module that have been proposed in the context of description logics [27, 64, 61, 87, 60], and we also adapt these notions to the setting of first-order rules. In Chapter 3 we introduce the problem of ontology classification and provide a brief overview of the main different kinds of classification algorithms available in the literature.

- In Part II we present and evaluate our novel approach to module extraction. In Chapter 5 we introduce the technique. In Section 5.2, we define the notion of a *module setting*, which constitutes the core of our framework, and establish the key technical results that justify the correctness of our approach. In Sections 5.3 and Section 5.4, we provide concrete module settings for a number of prominent inseparability relations. In Section 5.5, we show that our modules are consistent with the intuition that weaker inseparability relations should lead to smaller modules. For this, we introduce a notion of homomorphism between module settings, which will allow us to establish containment relations between the modules specified in Sections 5.3 and 5.4. In Section 5.6, we study the additional properties of our modules, considering depletingness, self-containment and justification-preservation. In Section 5.7, we briefly discuss the complexity of module extraction within our framework and show tractability for DL-based ontology languages. In Section 5.8, we discuss the optimality of the module settings introduced in Sections 5.3 and 5.4. In Chapter 6 we describe and evaluate the PrisM module extractor, where we have implemented the proposed technique.

- In Part III we present and evaluate our technique for exploiting module extraction techniques to optimise ontology classification. In Chapter 7 we introduce the technique. In Section 7.1 we provide a high level intuition about how the workload of ontology classification can be split between two reasoners. In Section 7.2 we explain how this division can be performed effectively using module extraction techniques and also techniques for approximating the subsumption hierarchy of an ontology. In Sections 7.3 and 7.4 we provide more specific details about how the proposed technique can be realised using two concrete module extraction techniques; we consider one syntactic locality-based technique, and also one based on the framework presented in Chapter 5. In Section 7.5 we explain how the technique proposed can be generalised to divide the classification workload between more than two reasoners. In Chapter 8 we describe and evaluate the MORe reasoner, based on our modular classification technique.

- Finally, in Part IV we recapitulate the contributions made in this thesis and suggest some directions for future work.

Chapter 2

Preliminaries

We next introduce notation and basic definitions used in the remainder of the thesis.

In Section 2.1 we introduce the language of first-order rules, which is powerful enough to fully capture expressive rule-based ontology languages such as datalog$^{\pm}$ [16], and datalog$^{\pm,\vee}$ [14, 2], as well as all mainstream description logics [8]. Most of the results in this thesis hold for arbitrary ontologies consisting of such first-order rules, and are hence applicable to a very wide range of knowledge representation formalisms. In Section 2.2 we introduce the syntax and first-order semantics of the description logic $\mathcal{SROIQ}$ [39], which underpins the W3C standard ontology language OWL 2 [68, 23]. We then introduce a normal form for $\mathcal{SROIQ}$ and establish its correspondence to first-order rules. Finally, in Section 2.3 we briefly recall the well-known hyperresolution calculus for first-order logic [11], which we exploit in many of our technical results to show that our modules preserve the required consequences.

Throughout this thesis, we assume basic familiarity with first-order logic [28] and we use standard first-order logic notions, such as predicates, constants, variables, function symbols, interpretations and entailment (written $\models$). For clarity, we next include a few basic definitions. A *term* t is either a variable, or a constant, or an expression of the form $f(t_1, \ldots, t_n)$ with $n > 0$, $t_1, \ldots, t_n$ terms and f an n-

ary function symbol; in this last case we call t a *functional* term. An *atom* is an expression of the form $P(t_1, \ldots, t_m)$ with $m \geq 0$, $t_1, \ldots, t_m$ terms and P an m-ary predicate. A (first-order) *formula* is an expression φ of the form specified by the following grammar, where γ is an atom

$$\varphi \quad ::= \quad \gamma \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \rightarrow \varphi_2 \mid \exists x.\varphi \mid \forall x.\varphi$$

We use φ and $\varphi(\mathbf{x})$ interchangeably when $\mathbf{x}$ are the free variables in φ (those not within the scope of an existential or universal quantifier). A *ground* formula, or *sentence*, is a formula with no free variables. A *function-free* formula is a formula that mentions no function symbols.[1]

We define a *signature* as a set of predicates; furthermore, given a first-order sentence ϕ, we use $\mathsf{Sig}(\phi)$ to denote its signature, i.e. the set of predicates mentioned in ϕ. We then say that ϕ is a Σ-sentence if $\mathsf{Sig}(\phi) \subseteq \Sigma$. Analogously, we denote with $\mathsf{Ct}(\phi)$ the set of constants occurring in ϕ. These definitions extend naturally to sets of sentences. The restriction of signatures to contain just predicates, and the separate treatment of constants and function symbols, will be convenient for working with inseparability relations later on.

A set of sentences $\mathcal{F}'$ is a (model) *conservative extension* of a set $\mathcal{F}$ if for each model $\mathcal{I}$ of $\mathcal{F}$ there is a model $\mathcal{J}$ of $\mathcal{F}'$ with the same domain as $\mathcal{I}$ and such that $A^{\mathcal{I}} = A^{\mathcal{J}}$ for each $A \in \mathsf{Sig}(\mathcal{F})$ and $a^{\mathcal{I}} = a^{\mathcal{J}}$ for each $a \in \mathsf{Ct}(\mathcal{F})$.

We deviate slightly from the standard definition of first order logic in that our definition does not include the nullary symbols $\top$ and $\bot$, which are interpreted as true and false respectively in every first-order interpretation. Similarly, we consider first-order logic without equality $\approx$ and hence we do not assume that $\approx$ is interpreted as the identity relation over the domain in every interpretation. Instead, we treat

[1] In the literature, constants are sometimes regarded as function symbols of arity 0; we do not adopt this convention and allow function-free formulas to mention constants.

$\bot$, $\top$ and $\approx$ as ordinary predicates the meaning of which we axiomatise explicitly in every knowledge base. We assume that $\bot$ is nullary, $\top$ is unary and $\approx$ is binary. Given a set $\mathcal{F}$ of function-free sentences, we then define the following sets of sentences $\mathcal{F}^{\bot}$, $\mathcal{F}^{\top}$, and $\mathcal{F}^{\approx}$.

- $\mathcal{F}^{\bot}$ is empty if $\mathcal{F}$ contains no occurrences of $\bot$, and the singleton set $\{\neg\bot\}$ otherwise.
- $\mathcal{F}^{\top}$ is empty if $\mathcal{F}$ contains no occurrence of $\top$; otherwise, it is the set

$$\{\, \forall x_1, \ldots, x_n[A(x_1, \ldots, x_n) \rightarrow \top(x_i)] \mid A \in \mathsf{Sig}(\mathcal{F}) \text{ } n\text{-ary}, \ 1 \leq i \leq n \,\}$$

- $\mathcal{F}^{\approx}$ is empty if $\mathcal{F}$ contains no occurrences of $\approx$; otherwise, it consists of the sentences (EQ1)–(EQ5) given next. Sentence (EQ1) is instantiated for each constant a in $\mathsf{Ct}(\mathcal{F})$; furthermore, sentences (EQ2) and (EQ5) are instantiated for each n-ary predicate A in $\mathsf{Sig}(\mathcal{F})$ and each x_i in $\mathbf{x} = (x_1, \ldots, x_n)$:

$$\rightarrow a \approx a \tag{EQ1}$$

$$\forall \mathbf{x}\, [A(\mathbf{x}) \rightarrow x_i \approx x_i] \tag{EQ2}$$

$$\forall x, y\, [x \approx y \rightarrow y \approx x] \tag{EQ3}$$

$$\forall x, y, z\, [x \approx y \wedge y \approx z \rightarrow x \approx z] \tag{EQ4}$$

$$\forall \mathbf{x}, y\, [A(\mathbf{x}) \wedge x_i \approx y \rightarrow A(x_1, \ldots, x_{i-1}, y, x_{i+1}, \ldots, x_n)] \tag{EQ5}$$

We consider *substitutions* as functional mappings between two sets of terms. Given a substitution σ and a term t not in the domain of σ, in an abuse of notation the expression $t\sigma$ denotes t. Substitutions can be applied to formulas: given an atom $A(t_1, ..., t_n)$, it is $A(t_1, ..., t_n)\sigma = A(t_1\sigma, ..., t_n\sigma)$, and given a non-atomic formula ϕ, the formula $\phi\sigma$ is the result of applying σ to all atoms in ϕ. This application is

extended to sets of formulas in the natural way. Given two substitutions σ and τ, their composition is the substitution $\sigma\tau$ such that $t(\sigma\tau) = (t\sigma)\tau$ for each t in the domain of σ. We say that σ and τ are *compatible* if they coincide over the intersection of their domains. If σ and τ are compatible, their union is the substitution $\sigma \cup \tau$ such that $t(\sigma \cup \tau) = t\sigma$ for each t in the domain of σ and $t(\sigma \cup \tau) = t\tau$ for each t in the domain of τ. Finally, we use $\mathsf{dom}(\sigma)$ (resp. $\mathsf{range}(\sigma)$) to denote the domain (resp. the range) of σ.

2.1 Rule-Based First-Order Languages

Rule-based languages are prominent knowledge representation formalisms closely related to ontology languages [24, 15, 16]. In this thesis, we focus on monotonic formalisms and hence on rule languages that can be seen as fragments of first-order logic. We next define a general notion of first-order rule which underpins the datalog$^{\pm}$ and datalog$^{\pm,\vee}$ families of languages [16, 2].

A *fact* γ is a function-free ground atom. A finite set of facts is called a *dataset*. A *rule* r is a function-free first-order sentence of the form

$$\forall \mathbf{x}[\varphi(\mathbf{x}) \rightarrow \exists \mathbf{y} \psi(\mathbf{x}, \mathbf{y})] \tag{2.1}$$

where $\mathbf{x}$ and $\mathbf{y}$ are disjoint vectors of variables, φ is a (possibly empty) conjunction of distinct atoms over constants and variables from $\mathbf{x}$, and ψ is built from atoms over constants and variables from $\mathbf{x} \cup \mathbf{y}$ using conjunction ($\wedge$) and disjunction ($\vee$). Note that any fact is also a rule. Formula φ is the rule *body* and $\exists \mathbf{y} \psi(\mathbf{x}, \mathbf{y})$ is the rule *head*. The head of a rule can be empty, in which case we represent it as $\Box$. Universal quantifiers in rules are omitted for brevity. Rules are required to be *safe*, that is, all universally quantified variables in the head must occur in the body. A rule is *Horn* if its head contains no occurrences of $\vee$, and it is *datalog* if its head is either empty

or it consists of a single atom where all variables are universally quantified.[2] A rule is a *(simple) implication* if it is of the form $A(\mathbf{x}) \rightarrow B(\mathbf{x})$ with A and B predicates of the same arity.

A (first-order) *ontology* $\mathcal{O}$ is a finite set of rules satisfying $\mathcal{O}^{\perp} \cup \mathcal{O}^{\top} \cup \mathcal{O}^{\approx} \subseteq \mathcal{O}$. We assume w.l.o.g. that different rules in $\mathcal{O}$ do not share existentially quantified variables and that the only rule with an empty head in $\mathcal{O}$ is $\perp \rightarrow \Box$ (syntactic alternative to the sentence $\neg\perp \in \mathcal{O}^{\perp}$).

A *datalog program* is an ontology containing only datalog rules. Given a datalog program $\mathcal{P}$ and a dataset $\mathcal{D}$, their *materialisation*, denoted with $\mathcal{P}(\mathcal{D})$, is the set of facts entailed by $\mathcal{P} \cup \mathcal{D}$. Such materialisation can be computed in time polynomial in the size of $\mathcal{D}$ using forward chaining [1, 24].

To conclude this section, we define the languages typically used for querying first-order ontologies. We define a *Boolean positive existential query* (Boolean PEQ) as a non-empty sentence q built from function-free atoms using only $\exists$, $\wedge$ and $\vee$; such a query holds w.r.t. an ontology $\mathcal{O}$ if $\mathcal{O} \models q$. A Boolean PEQ is a conjunctive query (CQ) if it is disjunction-free. The following proposition, the proof of which is straightforward, establishes a useful connection between Boolean PEQ evaluation and entailment of first-order rules.

Proposition 1. *Let $\mathcal{O}$ be an ontology, $r = \bigwedge_{i=1}^{n} \gamma_i(\mathbf{x}) \rightarrow \exists \mathbf{y} \psi(\mathbf{x}, \mathbf{y})$ a rule, and let σ be a substitution mapping all universally quantified variables in r to fresh distinct constants. Furthermore, let q be the Boolean PEQ defined as $\psi\sigma$. Then we have $\mathcal{O} \models r$ iff $\mathcal{O} \cup \{\gamma_i\sigma\}_{i=1}^{n} \models q$.*

[2]Note that, for any set $\mathcal{F}$ of first-order sentences, $\mathcal{F}^{\perp} \cup \mathcal{F}^{\top} \cup \mathcal{F}^{\approx}$ contains only datalog rules.

2.2 Description Logics

Description logics (DLs) [8] are a family of knowledge representation formalisms that correspond to decidable fragments of first-order logic. DLs are the logical formalisms underpinning the standard ontology languages: OWL DL is based on the description logic $\mathcal{SHOIN}$ [40], whereas its revision OWL 2 is based on the more expressive logic $\mathcal{SROIQ}$ [39, 23, 96].

The basic building blocks in $\mathcal{SROIQ}$ are pairwise disjoint countable sets of *atomic concepts*, which correspond to unary predicates, *atomic roles*, which correspond to binary predicates, and *individuals*, which correspond to constants. A *role* R is either an atomic role or the inverse S^- of an atomic role S. Complex concepts are constructed according to the following grammar, where $\bot_c$ and $\top_c$ are the special bottom and top concepts, A is an atomic concept, R is a role, o is an individual and $n \geq 1$:

$$\begin{aligned} C \quad ::= \quad & \bot_c \mid \top_c \mid A \mid \{o\} \mid \neg C \mid C_1 \sqcap C_2 \mid C_1 \sqcup C_2 \mid \\ & \exists R.C \mid \forall R.C \mid \exists R.\mathsf{Self} \mid \geq n R.C \mid \leq n R.C \end{aligned}$$

We assume that concept expressions of the form $\geq 1 R.C$ are replaced in practice by their equivalent $\exists R.C$. A *general concept inclusion axiom* (GCI) is an expression of the form $C_1 \sqsubseteq C_2$, where C_1 and C_2 are concepts. A *role inclusion axiom* (RIA) is an expression of the form $R_1 \circ \cdots \circ R_m \sqsubseteq S$ where each R_i is a role and S is an atomic role. A *role disjointness axiom* is an expression of the form $\mathsf{Disj}(S_1, S_2)$ with S_1 and S_2 roles. Finally, a *reflexivity axiom* is an expression of the form $\mathsf{Ref}(S)$ with S a role.

A $\mathcal{SROIQ}$ ontology is a finite set of GCIs, RIAs, role disjointness and reflexivity axioms. In order to ensure the decidability of basic reasoning tasks, each $\mathcal{SROIQ}$ ontology must satisfy certain additional conditions (e.g., the set of RIAs must satisfy a regularity condition); these conditions are, however, immaterial to the results in

roles		
$\pi(R, x, y)$	$=$	$R(x, y)$
$\pi(R^-, x, y)$	$=$	$R(y, x)$
concepts		
$\pi(\bot_c, x)$	$=$	$\bot$
$\pi(\top_c, x)$	$=$	$\top(x)$
$\pi(o, x)$	$=$	$x \approx o$
$\pi(\neg C, x)$	$=$	$\neg\pi(C, x)$
$\pi(C_1 \sqcap C_2, x)$	$=$	$\pi(C_1, x) \wedge \pi(C_2, x)$
$\pi(C_1 \sqcup C_2, x)$	$=$	$\pi(C_1, x) \vee \pi(C_2, x)$
$\pi(\exists R.C, x)$	$=$	$\exists y[\pi(R, x, y) \wedge \pi(C, y)]$
$\pi(\forall R.C, x)$	$=$	$\forall y[\pi(R, x, y) \rightarrow \pi(C, y)]$
$\pi(\exists R.\mathsf{Self}, x)$	$=$	$\pi(R, x, x)$
$\pi(\geq n R.C, x)$	$=$	$\exists x_1, \ldots, x_n[\bigwedge_i(\pi(R, x, x_i) \wedge \pi(C, x_i)) \wedge \bigwedge_{i \neq j} \neg(x_i \approx x_j)]$
$\pi(\leq n R.C, x)$	$=$	$\forall x_1, \ldots, x_{n+1}[\bigwedge_i(\pi(R, x, x_i) \wedge \pi(C, x_i)) \rightarrow \bigvee_{i \neq j} x_i \approx x_j]$
axioms		
$\pi(C_1 \sqsubseteq C_2)$	$=$	$\forall x[\pi(C_1, x) \rightarrow \pi(C_2, x)]$
$\pi(R_1 \circ \cdots \circ R_m \sqsubseteq S)$	$=$	$\forall x_1, \ldots, x_{m+1}[\bigwedge_{i=1}^{m} \pi(R_i, x_i, x_{i+1}) \rightarrow \pi(S, x_1, x_{m+1})]$
$\pi(\mathsf{Disj}(S_1, S_2))$	$=$	$\forall x, y[\pi(S_1, x, y) \wedge \pi(S_2, x, y) \rightarrow \bot]$
$\pi(\mathsf{Ref}(S))$	$=$	$\forall x[\top(x) \rightarrow \pi(S, x, x)]$

Figure 2.1: Semantics of $\mathcal{SROIQ}$ via translation into first-order logic

this thesis and we refer the reader to [39] for further details.

The semantics of $\mathcal{SROIQ}$ can be given by a direct translation into first-order logic [8, 66] using the mapping function π in Figure 2.1. Given a $\mathcal{SROIQ}$ ontology $\mathcal{O}$, let $\mathcal{F}_\mathcal{O} = \{\, \pi(\alpha) \mid \alpha \in \mathcal{O} \,\}$; we then define

$$\pi(\mathcal{O}) = \mathcal{F}_\mathcal{O} \cup \mathcal{F}_\mathcal{O}^\bot \cup \mathcal{F}_\mathcal{O}^\top \cup \mathcal{F}_\mathcal{O}^\approx$$

A first-order interpretation is a model of $\mathcal{O}$ if it is a model of $\pi(\mathcal{O})$.

Note that $\pi(\mathcal{O})$ is not always a set of first-order rules as defined in Section 2.1. However, $\mathcal{O}$ can always be polynomially normalised into an entailment preserving $\mathcal{SROIQ}$ ontology $\mathcal{O}'$ such that $\pi(\mathcal{O}')$ is a set of rules. We next define normalised $\mathcal{SROIQ}$ ontologies and assume from here onwards (unless otherwise stated) that $\mathcal{SROIQ}$ ontologies are normalised.

Definition 2. A $\mathcal{SROIQ}$ ontology $\mathcal{O}$ is *normalised* if it consists only of axioms

α	$\pi(\alpha)$
$A \sqsubseteq \bot_c$	$A(x) \to \bot$
$A \sqsubseteq \{o\}$	$A(x) \to x \approx o$
$\top_c \sqsubseteq A$	$\top(x) \to A(x)$
$\{o\} \sqsubseteq A$	$(\to A(o))$
$A \sqsubseteq B_1 \sqcup B_2$	$A(x) \to B_1(x) \vee B_2(x)$
$A_1 \sqcap A_2 \sqsubseteq B$	$A_1(x) \wedge A_2(x) \to B(x)$
$A \sqsubseteq \exists S.B$	$A(x) \to \exists y[S(x,y) \wedge B(y)]$
$A \sqsubseteq \exists S.\mathsf{Self}$	$A(x) \to S(x,x)$
$\exists S.A \sqsubseteq B$	$S(x,y) \wedge A(y) \to B(x)$
$\exists S.\mathsf{Self} \sqsubseteq A$	$S(x,x) \to A(x)$
$A \sqsubseteq \leq n S.B$	$A(x) \wedge \bigwedge_{i=1}^{n+1}[S(x,y_i) \wedge B(y_i)] \to \bigvee_{i \neq j} y_i \approx y_j$
$S_1 \circ S_2 \sqsubseteq S_3$	$S_1(x,y) \wedge S_2(y,z) \to S_3(x,z)$
$S_1^- \sqsubseteq S_2$	$S_1(x,y) \to S_2(y,x)$
$\mathsf{Disj}(S_1, S_2)$	$S_1(x,y) \wedge S_2(x,y) \to \bot$
$\mathsf{Ref}(S)$	$\top(x) \to S(x,x)$

Table 2.1: Correspondence between normalised $\mathcal{SROIQ}$ axioms and rules

of the form

$$A \sqsubseteq \bot_c \quad A \sqsubseteq \{o\} \quad \top_c \sqsubseteq A \quad \{o\} \sqsubseteq A \quad A \sqsubseteq B_1 \sqcup B_2 \quad A_1 \sqcap A_2 \sqsubseteq B$$

$$A \sqsubseteq \exists S.B \quad A \sqsubseteq \exists S.\mathsf{Self} \quad \exists S.A \sqsubseteq B \quad \exists S.\mathsf{Self} \sqsubseteq A \quad A \sqsubseteq \leq n S.B$$

$$S_1 \circ S_2 \sqsubseteq S_3 \quad S_1^- \sqsubseteq S_2 \quad \mathsf{Disj}(S_1, S_2) \quad \mathsf{Ref}(S)$$

with $A_{(i)}$, $B_{(i)}$ atomic concepts, o an individual, $S_{(i)}$ atomic roles, and $n \geq 1$. $\diamond$

Table 2.1 shows the application of π to normalised axioms. Clearly, $\pi(\mathcal{O})$ is a set of rules whenever $\mathcal{O}$ is normalised. Moreover, π establishes a bijection between $\mathcal{O}$ and $\pi(\mathcal{O})$ in this case. Since $\mathcal{O}$ and $\pi(\mathcal{O})$ are semantically equivalent, it is natural to identify them, and we shall do so in the remainder of this thesis.

Proposition 3. *Let $\mathcal{O}$ be a $\mathcal{SROIQ}$ ontology and let $\mathcal{O}'$ be the result of exhaustively applying to $\mathcal{O}$ the rewriting rules in Figure 2.2. Then, $\mathcal{O}'$ satisfies the following properties: (i) it is normalised; (ii) it is of size polynomial in the size of $\mathcal{O}$; and (iii) it is a conservative extension of $\mathcal{O}$.*

$$
\begin{array}{rcl}
\bot_c \sqsubseteq C & \Rightarrow & \\
D \sqsubseteq \{o\} & \Rightarrow & X \sqsubseteq \{o\},\ D \sqsubseteq X \\
\exists S^-.D \sqsubseteq C & \Rightarrow & \exists P.X \sqsubseteq D,\ S^- \sqsubseteq P \\
\exists S.D \sqsubseteq C & \Rightarrow & \exists S.X \sqsubseteq C,\ D \sqsubseteq X \\
{\geq}mS^-.D \sqsubseteq C & \Rightarrow & {\geq}mP.D \sqsubseteq C,\ S^- \sqsubseteq P \\
{\geq}mS.D \sqsubseteq C & \Rightarrow & \top_c \sqsubseteq {\leq}(m{-}1)S.D \sqcup C \\
D \sqcap C_1 \sqsubseteq C_2 & \Rightarrow & X \sqcap C_1 \sqsubseteq C_2,\ D \sqsubseteq X \\
\neg C_1 \sqsubseteq C_2 & \Rightarrow & \top_c \sqsubseteq C_1 \sqcup C_2 \\
C_1 \sqcup C_2 \sqsubseteq C_3 & \Rightarrow & C_1 \sqsubseteq C_3,\ C_2 \sqsubseteq C_3 \\
\forall S^-.C_1 \sqsubseteq C_2 & \Rightarrow & \forall P.C_1 \sqsubseteq C_2,\ P^- \sqsubseteq S \\
\forall S.C_1 \sqsubseteq C_2 & \Rightarrow & \top_c \sqsubseteq \exists S.X \sqcup C_2,\ X \sqcap C \sqsubseteq \bot_c \\
\exists S^-.\mathsf{Self} \sqsubseteq C & \Rightarrow & \exists P.\mathsf{Self} \sqsubseteq C,\ S^- \sqsubseteq P \\
{\leq}(m{-}1)S^-.C_1 \sqsubseteq C_2 & \Rightarrow & {\leq}(m{-}1)P.C_1 \sqsubseteq C_2,\ P^- \sqsubseteq S \\
{\leq}(m{-}1)S.C_1 \sqsubseteq C_2 & \Rightarrow & \top_c \sqsubseteq {\geq}mS.C_1 \sqcup C_2 \\
C \sqsubseteq \top_c & \Rightarrow & \\
\{o\} \sqsubseteq D & \Rightarrow & \{o\} \sqsubseteq X,\ X \sqsubseteq D \\
C \sqsubseteq \exists S^-.D & \Rightarrow & C \sqsubseteq \exists P.D,\ P^- \sqsubseteq S \\
C \sqsubseteq \exists S.D & \Rightarrow & C \sqsubseteq \exists S.X,\ X \sqsubseteq D \\
C \sqsubseteq \forall S^-.D & \Rightarrow & C \sqsubseteq \forall P.D,\ S^- \sqsubseteq P \\
C \sqsubseteq \forall S.D & \Rightarrow & C \sqsubseteq \forall S.X,\ X \sqsubseteq D \\
C \sqsubseteq \exists S^-.\mathsf{Self} & \Rightarrow & C \sqsubseteq \exists P.\mathsf{Self},\ P^- \sqsubseteq S \\
C \sqsubseteq {\leq}nS^-.D & \Rightarrow & C \sqsubseteq {\leq}nP.D,\ S^- \sqsubseteq P \\
C \sqsubseteq {\leq}nS.D & \Rightarrow & C \sqsubseteq {\leq}nS.X,\ X \sqsubseteq D \\
C_1 \sqsubseteq D \sqcup C_2 & \Rightarrow & C_1 \sqsubseteq X \sqcup C_2,\ X \sqsubseteq D \\
C_1 \sqsubseteq \neg C_2 & \Rightarrow & C_1 \sqcap C_2 \sqsubseteq \bot_c \\
C_1 \sqsubseteq C_2 \sqcap C_3 & \Rightarrow & C_1 \sqsubseteq C_2,\ C_1 \sqsubseteq C_3 \\
C_1 \sqsubseteq {\geq}mS^-.C_2 & \Rightarrow & C_1 \sqsubseteq {\geq}mP.C_2,\ P^- \sqsubseteq S \\
C_1 \sqsubseteq {\geq}mS.C_2 & \Rightarrow & C_1 \sqsubseteq \exists S.X_i,\ X_i \sqsubseteq C_2,\ X_i \sqcap X_j \sqsubseteq \bot_c\ (1{\leq}i{<}j{\leq}m) \\
D_1 \sqsubseteq D_2 & \Rightarrow & D_1 \sqsubseteq X,\ X \sqsubseteq D_2 \\
R_1 \circ R_2 \circ R_3 \circ \cdots \circ R_k \sqsubseteq S & \Rightarrow & R_1 \circ R_2 \sqsubseteq P,\ P \circ R_3 \circ \cdots \circ R_k \sqsubseteq S \\
Q^- \circ R \sqsubseteq S & \Rightarrow & P \circ R \sqsubseteq S,\ Q^- \sqsubseteq P \\
R \circ Q^- \sqsubseteq S & \Rightarrow & R \circ P \sqsubseteq S,\ Q^- \sqsubseteq P \\
\mathsf{Disj}(R, Q^-) & \Rightarrow & \mathsf{Disj}(R, P),\ Q^- \sqsubseteq P \\
\mathsf{Disj}(Q^-, R) & \Rightarrow & \mathsf{Disj}(P, R),\ Q^- \sqsubseteq P \\
\mathsf{Ref}(Q^-) & \Rightarrow & \mathsf{Ref}(Q)
\end{array}
$$

Figure 2.2: Normalisation of $\mathcal{SROIQ}$ axioms, where $C_{(i)}$ are concepts, $D_{(i)}$ are non-atomic concepts different from $\bot_c$ and $\top_c$, X is a fresh atomic concept, Q and S are atomic roles, $R_{(i)}$ are roles, P is a fresh atomic role, and $m \geq 2$, $n \geq 1$, $k \geq 3$

Proof. It is easy to see that some rewrite rule is applicable to any axiom that is not normalised; furthermore, no rule is applicable to normalised axioms. Thus, $\mathcal{O}'$ is normalised. Furthermore, note that the rules in Figure 2.2 are a syntactic variant of the structural transformation in first-order logic [71]. This implies that $\mathcal{O}'$ can be computed in time polynomial in the size of $\mathcal{O}$ (assuming unary encoding of numbers), and also that it is a conservative extension of $\mathcal{O}$. □

2.3 Hyperresolution and Proofs

Reasoning w.r.t. ontologies can be realised by means of *hyperresolution* [75, 11], which generalises the forward chaining technique for datalog. Hyperresolution is applicable to sets of first-order *clauses*—universally quantified sentences of the form $\bigwedge_i \gamma_i \rightarrow \bigvee_j \delta_j$ with γ_i and δ_j atoms (possibly containing function symbols). Thus, it is only applicable to ontologies containing existentially quantified rules after Skolemisation and subsequent transformation into Conjunctive Normal Form (CNF).

For each rule r of the form (2.1) and each existentially quantified variable y in r, let f_y^r be a function symbol globally unique for r and y of arity $|\mathbf{x}|$, and let θ_{sk} be the substitution such that $\theta_{\mathsf{sk}}(y) = f_y^r(\mathbf{x})$ for each r and y. The Skolemisation of r is the sentence

$$\mathsf{sk}(r) = \varphi(\mathbf{x}) \rightarrow \psi(\mathbf{x}, \mathbf{y})\theta_{\mathsf{sk}}$$

A CNF of $\mathsf{sk}(r)$ is a set of first-order clauses that is a conservative extension of $\mathsf{sk}(r)$. Such a CNF can be obtained in polynomial time using the standard structural transformation [71]. In this thesis we consider an arbitrary but fixed function κ that maps each rule r to some CNF of $\mathsf{sk}(r)$. The function κ extends to ontologies in the obvious way, and we refer to $\kappa(\mathcal{O})$ as a *clausification* of $\mathcal{O}$. By the well-known properties of Skolemisation and the structural transformation we have that $\mathcal{O} \models \phi$ iff $\kappa(\mathcal{O}) \models \phi$ for each ontology $\mathcal{O}$ and each first-order sentence ϕ over $\mathsf{Sig}(\mathcal{O})$.

Let $r = \bigwedge_{i=1}^{n} \gamma_i \rightarrow \bigvee_{j=1}^{m} \delta_j$ be a clause and let $\varphi_i = \psi_i \vee \xi_i$ with $1 \leq i \leq n$ be ground disjunctions of atoms where ξ_i is a single atom; furthermore, let σ be an MGU of each γ_i, ξ_i. The ground[3] disjunction of atoms $\bigvee_{i=1}^{n} \psi_i \vee \bigvee_{j=1}^{m} \delta_j\sigma$ is a *hyperresolvent* of r and $\varphi_1, \ldots, \varphi_n$. This disjunction can be empty, in which case we denote it with $\square$. Let $\mathcal{C}$ be a set of clauses, $\mathcal{D}$ a dataset and φ a disjunction of ground atoms. A *hyperresolution proof* (or simply a *proof*) of φ in $\mathcal{C} \cup \mathcal{D}$ is a pair $\rho = (T, \lambda)$ where T is a directed, rooted tree, and λ is a mapping from nodes in T to disjunctions of ground atoms such that for each node v in T the following properties are satisfied:

1. if v is the root of T then $\lambda(v) = \varphi$,
2. if v is a leaf in T then either $(\rightarrow \lambda(v)) \in \mathcal{C}$ or $\lambda(v) \in \mathcal{D}$, and
3. if v has children $w_1, \ldots, w_n$ then $\lambda(v)$ is a hyperresolvent of a clause from $\mathcal{C}$ and $\lambda(w_1), \ldots, \lambda(w_n)$.

The *support* of ρ, denoted by $\mathsf{supp}(\rho)$, is the set of clauses in $\mathcal{C}$ that take part in ρ as described in properties 2 and 3 above. We write $\mathcal{C} \cup \mathcal{D} \vdash \varphi$ to indicate that there exists a proof of φ in $\mathcal{C} \cup \mathcal{D}$. Hyperresolution is sound (if $\mathcal{C} \cup \mathcal{D} \vdash \varphi$ then $\mathcal{C} \cup \mathcal{D} \models \varphi$), and complete in the following sense: if $\mathcal{C} \cup \mathcal{D} \models \varphi$ then there exists $\psi \subseteq \varphi$ such that $\mathcal{C} \cup \mathcal{D} \vdash \psi$. In particular, $\mathcal{C}$ is unsatisfiable iff $\mathcal{C} \cup \mathcal{D} \vdash \square$.

Given proofs $\rho = (T, \lambda)$ and $\rho' = (T', \lambda')$, we say that ρ is *embeddable* into ρ' if there exists a mapping $\iota : T \rightarrow T'$ satisfying the following properties for each $v \in T$: *(i)* if v is a leaf of T, then $\iota(v)$ is a leaf of T'; *(ii)* if w is an ancestor of v in T then $\iota(w)$ is an ancestor of $\iota(v)$ in T', and *(iii)* $\lambda(v) \subseteq \lambda'(\iota(v)) \cup \{\bot\}$. Furthermore, given a substitution τ, we say that ρ is embeddable into ρ' *modulo* τ if it is embeddable into the proof (T', λ'_τ), where $\lambda'_\tau(v) = \lambda'(v)\tau$ for each $v \in T'$.

[3]The given disjunction is ground due to r being safe.

Chapter 3

Ontology Classification

In this chapter, we introduce the problem of ontology classification and provide a brief overview of the different kinds of classification algorithms available in the literature.

3.1 Ontology Classification in DLs and Beyond

Ontology classification is the problem of computing the so-called *subsumption hierarchy* induced by an ontology $\mathcal{O}$. For DL ontologies, the subsumption hierarchy is usually taken as the transitive relation consisting of all pairs (A, B) such that $\mathcal{O} \models A \sqsubseteq B$ and $A \neq B$, and each of them is an atomic concept, $\bot_c$, or $\top_c$. In particular, it contains the trivial pairs $(\bot_c, A)$ and $(A, \top_c)$ for each atomic concept $A \in \mathsf{Sig}(\mathcal{O})$, and also the pair $(A, \bot_c)$ whenever A is unsatisfiable in $\mathcal{O}$. Throughout this thesis, we denote the subsumption hierarchy in $\mathcal{O}$ as $\mathsf{hierarchy}(\mathcal{O})$. Furthermore, when $(A, B) \in \mathsf{hierarchy}(\mathcal{O})$, we say that B *subsumes* A and A *is subsumed by* B, or equivalently that B is a *subsumer* of A and A is a *subsumee* of B.

Example 4. In the case of the ontology $\mathcal{O}^{ex}$ from Figure 4.1, the subsumption hierarchy between atomic concepts in $\mathcal{O}^{ex}$ contains the following non-trivial pairs: (E, C), (D, H), (F, H), (G, H). $\diamond$

Classification is a fundamental task in ontology design as it can be used to detect modelling errors that manifest as unintended subsumptions or unsatisfiable terms. Furthermore, it is also a useful service from a user point of view since it provides a succinct roadmap of the terms in the ontology.

In the more general setting of first-order ontologies considered in this thesis, we take $\mathsf{hierarchy}(\mathcal{O})$ to be the transitive relation consisting of all pairs of the form (A, B) such that $A, B \in \mathsf{Sig}(\mathcal{O})$ are distinct predicates and, either they have the same arity and $\mathcal{O} \models A(\mathbf{x}) \rightarrow B(\mathbf{x})$, or $B = \bot$ and $\mathcal{O} \models A(\mathbf{x}) \rightarrow \bot$. For convenience, we further restrict these pairs to those with $A \neq \bot$ and $B \neq \top$, i.e., those that represent non-trivial implications. We can therefore see the problem of ontology classification as that of deciding, for each pair in

$$\begin{aligned}\mathsf{possSub}(\mathcal{O}) :=\{(A, B) \in \mathsf{Sig}(\mathcal{O})^2 \mid A \neq B, \bot,\ B \neq \top \text{ and} \\ \text{either } A \text{ and } B \text{ have the same arity or } B = \bot\}\end{aligned}$$

whether or not it is in $\mathsf{hierarchy}(\mathcal{O})$. In some practical cases we may only be interested in a hierarchy involving predicates of certain arities. This is the case for unary predicates/atomic concepts in DL ontologies: although role classification has also been considered in the DL literature [32], the focus tends to be on concept classification. Whenever the interest is only on predicates of certain arities, we will assume that $\mathsf{possSub}(\mathcal{O})$ reflects this restriction. We extend our classification-specific terminology to the setting of first-order ontologies in the natural way (i.e., B subsumes A if $\mathcal{O} \models A(\mathbf{x}) \rightarrow B(\mathbf{x})$, etc.). Moreover, for simplicity, we may omit variables and write $A \rightarrow B$ instead of $A(\mathbf{x}) \rightarrow B(\mathbf{x})$ and $A \rightarrow \bot$ instead of $A(\mathbf{x}) \rightarrow \bot$.

3.2 Classification Algorithms for DL Ontologies

There have been extensive efforts directed towards finding efficient algorithms for classifying DL ontologies. The existing algorithms in the literature can be grouped in two main categories: algorithms based on performing subsumption checks for individual pairs, and algorithms that compute the complete subsumption hierarchy in one pass without the need for individual subsumption tests. The remainder of this section provides a basic overview of both groups of algorithms.

3.2.1 Individual Subsumption Tests

Individual subsumption tests are usually reduced to satisfiability checks: to find out if $\mathcal{O} \models A \sqsubseteq B$, it suffices to check whether the concept $A \sqcap \neg B$ is (un)satisfiable w.r.t. $\mathcal{O}$, i.e., whether there exists any model $\mathcal{I}$ of $\mathcal{O}$ such that $(A \sqcap \neg B)^{\mathcal{I}} \neq \emptyset$. Checking satisfiability w.r.t. an ontology $\mathcal{O}$ is a problem with high worst-case complexity when $\mathcal{O}$ is in an expressive DL. For instance, the problem is known to be 2NExpTime-complete if $\mathcal{O}$ is in $\mathcal{SROIQ}$ [47, 23], and ExpTime-complete if $\mathcal{O}$ is in $\mathcal{SHIQ}$ [91]. Despite the discouraging complexity results for these logics, there exist highly optimised satisfiability checking algorithms [69, 39] which work reasonably well in practice.

Regardless of how optimised satisfiability checking algorithms are, considering every one of the quadratically many pairs in $\mathsf{possSub}(\mathcal{O})$ would be very inefficient in practice: in the case of the FMA ontology, this would require $\sim 7,100,000,000$ individual subsumption tests. For this reason, practical classification algorithms try to reduce the number of individual subsumption tests that need to be performed. Being able to identify subsumptions easily is useful for this purpose, but it is even more useful to efficiently detect non-subsumptions since in practical cases the majority of pairs in $\mathsf{possSub}(\mathcal{O})$ do not correspond to subsumption pairs in $\mathcal{O}$. For example, out

of the $\sim 7,100,000,000$ potential subsumption pairs for FMA, only $\sim 1,000,000$ correspond to actual subsumptions. To reduce the number of individual subsumption tests, most reasoners for expressive DLs (such as HermiT [33], Pellet [82], Fact++ [92] and RacerPro [36]) implement variants of the *Enhanced Traversal* (ET) algorithm [9]. Starting from a trivial hierarchy containing only $(\bot_c, \top_c)$, the ET algorithm proceeds by finding, for each atomic concept, first its most specific subsumers, and then its most general subsumees in the current partial hierarchy. In this way, many subsumptions, and also many non-subsumptions, can be inferred by transitivity of the subsumption relation without the need for dedicated tests. Other efforts in this direction include the *told subsumptions* optimisation, which provides an inexpensive way of identifying "obvious" subsumption relationships that hold in the input ontology [10, 38]. Among optimisations used to detect non-subsumptions we can find *completely defined concepts*, which identifies a fragment of the ontology for which told subsumptions provide complete information, and also *model-merging* and other related techniques that exploit the computations performed during individual subsumption tests [93, 38, 35, 32].

However, notwithstanding extensive and ongoing research into optimisation techniques, the classification of large ontologies can still require a very large number of subsumption tests. Consequently, even if no individual test is very costly, the total amount of time required for classification can still be too large for practical purposes.

3.2.2 One Pass Classification

Among algorithms that compute all subsumptions in $\mathcal{O}$ in one pass we can find the prominent family of so-called *consequence-based* algorithms. These algorithms generally consist of a set of inference rules that are exhaustively applied to the axioms in $\mathcal{O}$ until no more applications are possible; the axiom $A \sqsubseteq B$ (or some representation thereof) is derived iff the pair (A, B) is in $\mathsf{hierarchy}(\mathcal{O})$. Such algorithms first

appeared for logics in the $\mathcal{EL}$ family of lightweight DLs [7] underlying the EL profile of OWL 2, for which they are tractable. There also exist tractable consequence-based classification algorithms for some logics related to the RL profile of OWL 2 [60, 58], which in turn corresponds to a fragment of datalog. Consequence-based classification is also possible for ontologies in other Horn DLs such as Horn $\mathcal{SHIQ}$ [48] or Horn $\mathcal{SROIQ}$ [74], although tractability is no longer guaranteed. Furthermore, in recent years, algorithms of this kind have been devised even for non-Horn DLs such as $\mathcal{SH}$ [80], $\mathcal{SHI}$ [81] and $\mathcal{SHIQ}$ [12]; these (exponential) algorithms are typically worst-case optimal, in the sense that they have the same complexity as satisfiability and subsumption checking for the corresponding logic.

Consequence-based classification algorithms have proved efficient in practice, especially those for the $\mathcal{EL}$ family: the reasoners ELK [49] and Konclude [86] implement highly optimised variations of the algorithm from [7] and are currently among the best performing reasoners for OWL 2 EL ontologies, as shown in recent independent evaluations.[1]

[1] `http://dl.kr.org/ore2015/vip.cs.man.ac.uk_8008/results.html`
`http://www.easychair.org/smart-program/VSL2014/ORE-competition.html`
`http://curation.cs.manchester.ac.uk/ore2013/ore2013.cs.manchester.ac.uk/competition/results/index.html`

Chapter 4

Module Extraction

In this section, we briefly recapitulate the key notions of *inseparability relation* and *module* that have been proposed in the description logic literature [21, 56, 53, 77, 52]. Furthermore, when required, we adapt these notions to the setting of first-order rules and prove some basic results that will be exploited throughout the thesis.

4.1 Inseparability Relations and Modules

Intuitively, given an ontology $\mathcal{O}$ and a signature Σ, a module of $\mathcal{O}$ w.r.t. Σ is a subset $\mathcal{M}$ of $\mathcal{O}$ that is indistinguishable from $\mathcal{O}$ w.r.t. tasks where only predicates in Σ are considered of interest. The indistinguishability criteria depend on the specific task at hand, and are usually formalised by means of *inseparability relations*.

Definition 5. An inseparability relation is a family $\mathcal{S} = \{ \equiv^{\mathcal{S}}_{\Sigma} \mid \Sigma \text{ a set of predicates} \}$ of equivalence relations between ontologies satisfying the following properties:

- if $\mathcal{O}'$ is a conservative extension of $\mathcal{O}$ and $\Sigma = \mathsf{Sig}(\mathcal{O})$, then $\mathcal{O} \equiv^{\mathcal{S}}_{\Sigma} \mathcal{O}'$; and
- $\mathcal{O}_1 \equiv^{\mathcal{S}}_{\Sigma} \mathcal{O}$ implies $\mathcal{O}_2 \equiv^{\mathcal{S}}_{\Sigma} \mathcal{O}$ for all $\mathcal{O}_1 \subseteq \mathcal{O}_2 \subseteq \mathcal{O}$ and Σ. ◇

The first property ensures that inseparability is stable under model-preserving transformations, whereas the second one ensures that it is consistent with the mono-

tonicity of first-order logic. The following definition captures the most common inseparability relations studied in the literature.

Definition 6. For each signature Σ, we say that ontologies $\mathcal{O}$ and $\mathcal{O}'$ are

- Σ-*model inseparable* ($\mathcal{O} \equiv^{\mathsf{m}}_{\Sigma} \mathcal{O}'$), if for every model $\mathcal{I}$ of $\mathcal{O}$ (resp. of $\mathcal{O}'$) there exists a model $\mathcal{J}$ of $\mathcal{O}'$ (resp. of $\mathcal{O}$) with the same domain s.t. $A^{\mathcal{I}} = A^{\mathcal{J}}$ for each $A \in \Sigma$.

- Σ-*query inseparable* ($\mathcal{O} \equiv^{\mathsf{q}}_{\Sigma} \mathcal{O}'$) if for each Boolean PEQ q and dataset $\mathcal{D}$ over Σ we have $\mathcal{O} \cup \mathcal{D} \models q$ iff $\mathcal{O}' \cup \mathcal{D} \models q$.

- Σ-*fact inseparable* ($\mathcal{O} \equiv^{\mathsf{f}}_{\Sigma} \mathcal{O}'$) if for each fact γ and dataset $\mathcal{D}$ over Σ we have $\mathcal{O} \cup \mathcal{D} \models \gamma$ iff $\mathcal{O}' \cup \mathcal{D} \models \gamma$.

- Σ-*implication inseparable* ($\mathcal{O} \equiv^{\mathsf{i}}_{\Sigma} \mathcal{O}'$) if for each simple Σ-implication[1] r we have $\mathcal{O} \models r$ iff $\mathcal{O}' \models r$. ◇

Example 7. Let us consider the ontology $\mathcal{O}^{ex}$ from Figure 4.1, which will serve as a running example. Let us also consider the signatures Σ_i and fragments $\mathcal{M}_i$ of $\mathcal{O}^{ex}$ given next:

$$\Sigma_1 = \{B, C, D, H\} \qquad \mathcal{M}_1 = \{r_6, r_7, r_8, r_9\}$$
$$\Sigma_2 = \{A, B\} \qquad \mathcal{M}_2 = \emptyset$$
$$\Sigma_3 = \{A, C, D, R\} \qquad \mathcal{M}_3 = \{r_1, r_2, r_4, r_5\}$$

The only non-tautological Σ_1-implication entailed by $\mathcal{O}^{ex}$ is $D(x) \to H(x)$, which also follows from $\mathcal{M}_1$; thus, $\mathcal{M}_1$ is Σ_1-implication inseparable from $\mathcal{O}^{ex}$. Furthermore, any subset of $\mathcal{O}^{ex}$ not containing $\mathcal{M}_1$ does not entail $D(x) \to H(x)$ and is hence not Σ_1-implication inseparable from $\mathcal{O}^{ex}$. As we will see later on, the requirement of fact

[1]Recall that a simple implication is a rule of the form $A(\mathbf{x}) \to B(\mathbf{x})$.

r_1 :	$A(x) \to \exists y_1[R(x, y_1) \wedge B(y_1)]$	$A \sqsubseteq \exists R.B$
r_2 :	$A(x) \to R(x, o)$	$A \sqsubseteq \exists R.\{o\}$
r_3 :	$B(x) \wedge C(x) \to D(x)$	$B \sqcap C \sqsubseteq D$
r_4 :	$R(x, y) \wedge C(y) \to E(x)$	$\exists R.C \sqsubseteq E$
r_5 :	$E(x) \to C(x)$	$E \sqsubseteq C$
r_6 :	$D(x) \to F(x) \vee G(x)$	$D \sqsubseteq F \sqcup G$
r_7 :	$F(x) \to \exists y_2 S(x, y_2)$	$F \sqsubseteq \exists S.\top_c$
r_8 :	$S(x, y) \to H(x)$	$\exists S.\top_c \sqsubseteq H$
r_9 :	$G(x) \to H(x)$	$G \sqsubseteq H$

Figure 4.1: Example ontology $\mathcal{O}^{ex}$ in both rule and DL notation

inseparability is stronger than that of implication inseparability; indeed, $\mathcal{M}_1$ is not Σ_1-fact inseparable from $\mathcal{O}^{ex}$ since, for $\mathcal{D}_1 = \{B(a), C(a)\}$, we have $\mathcal{O}^{ex} \cup \mathcal{D}_1 \models D(a)$ but $\mathcal{M}_1 \cup \mathcal{D}_1 \not\models D(a)$.

It can be readily checked that $\mathcal{M}_2$ is Σ_2-fact inseparable from $\mathcal{O}^{ex}$. It is, however, not Σ_2-query inseparable: for $\mathcal{D}_2 = \{A(a)\}$, we have $\mathcal{O}^{ex} \cup \mathcal{D}_2 \models \exists y B(y)$ but $\mathcal{M}_2 \cup \mathcal{D}_2 \not\models \exists y B(y)$.

Finally, consider $\mathcal{M}_3$ and Σ_3. As we will see later on, $\mathcal{M}_3$ is Σ_3-query inseparable from $\mathcal{O}^{ex}$; however, it is not Σ_3-model inseparable. Indeed, the interpretation $\mathcal{I}$ where $\Delta^{\mathcal{I}} = \{a, o\}$, $A^{\mathcal{I}} = E^{\mathcal{I}} = \{a\}$, $B^{\mathcal{I}} = C^{\mathcal{I}} = \{o\}$, $D^{\mathcal{I}} = \emptyset$ and $R^{\mathcal{I}} = \{(a, o)\}$ is a model of $\mathcal{M}_3$. This interpretation, however, cannot be extended to a model of r_3 (or, consequently, to a model of $\mathcal{O}$) without reinterpreting A, C, D or R. We will also see that, to ensure Σ_3-model inseparability it suffices to extend $\mathcal{M}_3$ with rule r_3. $\diamond$

Model inseparability can be characterised in terms of preservation of second-order consequences [53]: ontologies $\mathcal{O}$ and $\mathcal{O}'$ are Σ-model inseparable if and only if for each second-order formula φ over Σ we have $\mathcal{O} \models \varphi$ iff $\mathcal{O}' \models \varphi$. Additionally, as we show next, query and fact inseparability can be characterised in terms of preservation of first-order rules and datalog rules, respectively.

Proposition 8. *The following statements hold for each signature Σ and each pair of ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$:*

1. $\mathcal{O}_1 \equiv^{\mathsf{q}}_{\Sigma} \mathcal{O}_2$ *iff* $\mathcal{O}_1 \models r \Leftrightarrow \mathcal{O}_2 \models r$ *holds for each* Σ*-rule* r *with a non-empty head.*

2. $\mathcal{O}_1 \equiv^{\mathsf{f}}_{\Sigma} \mathcal{O}_2$ *iff* $\mathcal{O}_1 \models r \Leftrightarrow \mathcal{O}_2 \models r$ *holds for each datalog* Σ*-rule* r *with a non-empty head.*

Proof. We prove the first statement; the second one is analogous. Suppose $\mathcal{O}_1 \equiv^{\mathsf{q}}_{\Sigma} \mathcal{O}_2$ and consider an arbitrary rule $r = \bigwedge_{i=1}^{n} \gamma_i(\mathbf{x}) \rightarrow \exists \mathbf{y} \psi(\mathbf{x}, \mathbf{y})$ over Σ such that $\psi \neq \Box$, and a substitution σ mapping universally quantified variables in r to fresh distinct constants. Furthermore, consider the dataset $\mathcal{D} = \{\gamma_i \sigma\}_{i=1}^{n}$, and the Boolean PEQ $q = \exists \mathbf{y} \psi(\mathbf{x}, \mathbf{y}) \sigma$ ($\exists \mathbf{y} \psi(\mathbf{x}, \mathbf{y}) \sigma$ is indeed a Boolean PEQ since by hypothesis ψ is non-empty). By Proposition 1, $\mathcal{O}_i \models r$ iff $\mathcal{O}_i \cup \mathcal{D} \models q$. Together with $\mathcal{O}_1 \equiv^{\mathsf{q}}_{\Sigma} \mathcal{O}_2$, this implies that $\mathcal{O}_1 \models r \Leftrightarrow \mathcal{O}_2 \models r$.

On the other hand, suppose that $\mathcal{O}_1 \models r \Leftrightarrow \mathcal{O}_2 \models r$ holds for each Σ-rule r with a non-empty head. Let q be a Boolean PEQ over Σ and $\mathcal{D}$ a Σ-dataset and consider the rule $r = \bigwedge_{\gamma \in \mathcal{D}} \gamma \rightarrow q$. By Proposition 1 we have $\mathcal{O}_i \cup \mathcal{D} \models q$ iff $\mathcal{O}_i \models r$, and since, by assumption, $\mathcal{O}_1 \models r \Leftrightarrow \mathcal{O}_2 \models r$, it follows that $\mathcal{O}_1 \cup \mathcal{D} \models q \Leftrightarrow \mathcal{O}_2 \cup \mathcal{D} \models q$ and hence $\mathcal{O}_1$ and $\mathcal{O}_2$ are Σ-query inseparable. $\Box$

It immediately follows that the inseparability relations in Definition 6 are naturally ordered from strongest to weakest for each non-trivial Σ:

$$\equiv^{\mathsf{m}}_{\Sigma} \subsetneq \equiv^{\mathsf{q}}_{\Sigma} \subsetneq \equiv^{\mathsf{f}}_{\Sigma} \subsetneq \equiv^{\mathsf{i}}_{\Sigma}$$

Furthermore, we can now identify the classes of entailments that are relevant to each inseparability relation.

Definition 9. For each inseparability relation $\mathcal{S} \in \{\mathsf{m}, \mathsf{q}, \mathsf{f}, \mathsf{i}\}$, let $\mathsf{rel}_{\mathcal{S}}$ be the function

mapping each ontology $\mathcal{O}$ and signature Σ to a set of relevant entailments as follows:

$$\mathsf{rel}_{\mathcal{S}}(\mathcal{O},\Sigma)=\begin{cases}\{\,\phi \mid \mathcal{O}\models\phi \text{ and } \phi \text{ is a second-order } \Sigma\text{-sentence}\,\} & \text{if } \mathcal{S}=\mathsf{m}\\ \{\,r \mid \mathcal{O}\models r \text{ and } r \text{ is a } \Sigma\text{-rule with non-empty head}\,\} & \text{if } \mathcal{S}=\mathsf{q}\\ \{\,r \mid \mathcal{O}\models r \text{ and } r \text{ is a datalog } \Sigma\text{-rule with non-empty head}\,\} & \text{if } \mathcal{S}=\mathsf{f}\\ \{\,r \mid \mathcal{O}\models r \text{ and } r \text{ is of the form } A(\mathbf{x})\rightarrow B(\mathbf{x}) \text{ with } A,B\in\Sigma\,\} & \text{if } \mathcal{S}=\mathsf{i}\end{cases}$$

◇

The following theorem establishes that the inseparability relations in Definition 6 are fully characterised by the preservation of relevant Σ-entailments as in Definition 9.

Theorem 10. *Let $\mathcal{O}$ and $\mathcal{O}'$ be ontologies, Σ a signature, and let $\mathcal{S}\in\{\mathsf{m},\mathsf{q},\mathsf{f},\mathsf{i}\}$. Then, $\mathcal{O}\equiv^{\mathcal{S}}_{\Sigma}\mathcal{O}'$ if and only if $\mathsf{rel}_{\mathcal{S}}(\mathcal{O},\Sigma)=\mathsf{rel}_{\mathcal{S}}(\mathcal{O}',\Sigma)$.*

Proof. This is a direct consequence of Definitions 6 and 9, Proposition 8 and the characterisation of model inseparability in terms of second-order entailments in [53]. □

Inseparability relations allow us to formalise modules as well as their desirable properties.

Definition 11. Let $\mathcal{O}$ be an ontology, Σ a signature, and $\mathcal{S}$ an inseparability relation, and let $\mathcal{M}\subseteq\mathcal{O}$. We say that $\mathcal{M}$ is a $\equiv^{\mathcal{S}}_{\Sigma}$-*module of* $\mathcal{O}$ if it holds that $\mathcal{O}\equiv^{\mathcal{S}}_{\Sigma}\mathcal{M}$. Furthermore, we say that $\mathcal{M}$ is

- *minimal* if no $\mathcal{M}'\subsetneq\mathcal{M}$ is a $\equiv^{\mathcal{S}}_{\Sigma}$-module of $\mathcal{O}$;
- *self-contained* if $\mathcal{O}\equiv^{\mathcal{S}}_{\Sigma\cup\mathsf{Sig}(\mathcal{M})}\mathcal{M}$;
- *depleting* if $\mathcal{O}\setminus\mathcal{M}\equiv^{\mathcal{S}}_{\Sigma}\emptyset$; and *strongly depleting* if $\mathcal{O}\setminus\mathcal{M}\equiv^{\mathcal{S}}_{\Sigma\cup\mathsf{Sig}(\mathcal{M})}\emptyset$.

Finally, we define a *justification* in $\mathcal{O}$ of a sentence ϕ such that $\mathcal{O}\models\phi$ as a subset-minimal $\mathcal{O}'\subseteq\mathcal{O}$ such that $\mathcal{O}'\models\phi$. We say that $\mathcal{M}$ is *justification-preserving* if for each ϕ in $\mathsf{rel}_{\mathcal{S}}(\mathcal{O},\Sigma)$ and each justification $\mathcal{O}'$ of ϕ in $\mathcal{O}$ we have $\mathcal{O}'\subseteq\mathcal{M}$. ◇

Example 12. Consider again the ontologies and signatures in Example 7. We can see that each $\mathcal{M}_i$ is a module of $\mathcal{O}^{ex}$; in particular, $\mathcal{M}_1$ is a $\equiv^{\mathsf{i}}_{\Sigma_1}$-module, $\mathcal{M}_2$ is a $\equiv^{\mathsf{f}}_{\Sigma_2}$-module , $\mathcal{M}_3$ is a $\equiv^{\mathsf{q}}_{\Sigma_3}$-module, and $\mathcal{M}_3 \cup \{r_3\}$ is a $\equiv^{\mathsf{m}}_{\Sigma_3}$-module. $\diamond$

The inseparability requirement ensures that modules can be used instead of $\mathcal{O}$ for reasoning purposes, provided that the entailments relevant to the application at hand are captured by the given inseparability relation and contain only symbols from Σ.

Minimality ensures that the module contains as little irrelevant information as possible while still satisfying the inseparability requirement. Although minimality is clearly desirable in most applications of modules (e.g., reasoning over small ontology subsets is typically preferable to reasoning over the whole ontology), extracting modules of minimal size is invariably very hard (and often algorithmically infeasible); thus, practical techniques aim at computing modules that are typically much smaller than $\mathcal{O}$, albeit not necessarily minimal.

Self-contained modules are inseparable from $\mathcal{O}$ not only w.r.t. the relevant signature Σ, but also w.r.t. their own signature. Depletingness ensures that no relevant information is "left behind" after extracting the module from $\mathcal{O}$, i.e., that $\mathcal{O} \setminus \mathcal{M}$ is inseparable from the empty ontology. The most basic form of depletingness is formulated in terms of Σ, whereas a stronger variant requires inseparability w.r.t. the symbols in $\mathcal{M}$ as well. Self-contained and depleting modules are especially well-suited for ontology reuse and modular ontology development applications. For instance, if $\mathcal{M}$ is both self-contained and depleting, the developer of $\mathcal{O}$ can remodel the subdomain characterised by Σ by replacing $\mathcal{M}$ in $\mathcal{O}$ with a new set of axioms, with the guarantee that any changes performed will not have any unintended interactions with the rest of $\mathcal{O}$.

Justification-preservation enables the use of modules for ontology debugging and repair [78, 44, 46, 45]. The justification of an entailment is a useful form of explanation; furthermore, ontology repair services typically rely on the computation of

all justifications of an unintended entailment as a first step towards obtaining a repair plan. Computing justifications, however, is a computationally intensive task and practical module extraction techniques have been effectively exploited to optimise this process [90].

We conclude this section by briefly discussing the impact of normalisation on module extraction. As pointed out in Section 2.2, our technical results are applicable to ontologies consisting of rules; when referring to DL ontologies, we implicitly assume they are given in rule form and are therefore normalised. We argue that normalisation techniques stemming from the structural transformation preserve inseparability and hence it is possible to obtain a module for a DL ontology once a module for its normalisation has been computed.

Definition 13. A *normalisation function* norm maps $\mathcal{SROIQ}$ ontologies to normalised $\mathcal{SROIQ}$ ontologies s.t. the following holds for all ontologies in its domain:

- $\mathsf{norm}(\mathcal{O})$ is a conservative extension of $\mathcal{O}$; and
- $\mathcal{O}_1 \subseteq \mathcal{O}_2$ implies $\mathsf{norm}(\mathcal{O}_1) \subseteq \mathsf{norm}(\mathcal{O}_2)$. ◇

Definition 13 captures the standard normalisation techniques stemming from the structural transformation, such as the one discussed in Section 2.2. Furthermore, it is typically straightforward in practice to keep track of the correspondence between the axioms in the original ontology $\mathcal{O}$ and those in $\mathsf{norm}(\mathcal{O})$. As shown by the following proposition, this correspondence allows us to efficiently obtain a module of $\mathcal{O}$ once a module for $\mathsf{norm}(\mathcal{O})$ has been computed.

Proposition 14. *Let Σ be a signature, let $\mathcal{S}$ be an inseparability relation, and let* norm *be a normalisation function. Then, $\mathcal{M} \equiv^{\mathcal{S}}_{\Sigma} \mathcal{O}$ iff $\mathsf{norm}(\mathcal{M}) \equiv^{\mathcal{S}}_{\Sigma} \mathsf{norm}(\mathcal{O})$.*

Proof. By definition, $\mathcal{S}$ satisfies that $\mathcal{O}_1 \subseteq \mathcal{O}_2 \subseteq \mathcal{O}$ and $\mathcal{O}_1 \equiv^{\mathcal{S}}_{\Sigma} \mathcal{O}$ implies $\mathcal{O}_2 \equiv^{\mathcal{S}}_{\Sigma} \mathcal{O}$. Therefore, if $\mathsf{norm}(\mathcal{M})$ contains a $\equiv^{\mathcal{S}}_{\Sigma}$-module of $\mathsf{norm}(\mathcal{O})$ then $\mathsf{norm}(\mathcal{M})$ is a

α	$\bot$-local w.r.t. Σ	$\top$-local w.r.t. Σ
$A \sqsubseteq \bot_c$	if $A \notin \Sigma$	never
$A \sqsubseteq \{o\}$	if $A \notin \Sigma$	never
$\top_c \sqsubseteq A$	never	if $A \notin \Sigma$
$\{o\} \sqsubseteq A$	never	if $A \notin \Sigma$
$A \sqsubseteq B_1 \sqcup B_2$	if $A \notin \Sigma$	if $B_1 \notin \Sigma$ or $B_2 \notin \Sigma$
$A_1 \sqcap A_2 \sqsubseteq B$	if $A_1 \notin \Sigma$ or $A_2 \notin \Sigma$	if $B \notin \Sigma$
$A \sqsubseteq \exists R.B$	if $A \notin \Sigma$	if $\{R, B\} \cap \Sigma = \emptyset$
$A \sqsubseteq \exists R.\mathsf{Self}$	if $A \notin \Sigma$	if $R \notin \Sigma$
$\exists R.A \sqsubseteq B$	if $R \notin \Sigma$ or $A \notin \Sigma$	if $B \notin \Sigma$
$\exists R.\mathsf{Self} \sqsubseteq A$	if $R \notin \Sigma$	if $B \notin \Sigma$
$A \sqsubseteq \leq m R.B$	if $A \notin \Sigma$ or $R \notin \Sigma$ or $B \notin \Sigma$	never
$R_1 \circ R_2 \sqsubseteq S$	if $R_1 \notin \Sigma$ or $R_2 \notin \Sigma$	if $S \notin \Sigma$
$R \sqsubseteq S^-$	if $R \notin \Sigma$	if $S \notin \Sigma$
$\mathsf{Disj}(R, S)$	if $R \notin \Sigma$ or $S \notin \Sigma$	never
$\mathsf{Ref}(R)$	never	if $R \notin \Sigma$

Table 4.1: Syntactic locality for normalised $\mathcal{SROIQ}$ axioms

$\equiv^{\mathcal{S}}_{\Sigma}$-module of $\mathsf{norm}(\mathcal{O})$ itself. On the other hand, since $\mathsf{norm}(\mathcal{O})$ (resp. $\mathsf{norm}(\mathcal{M})$) is a conservative extension of $\mathcal{O}$ (resp. of $\mathcal{M}$), we have that $\mathsf{norm}(\mathcal{O}) \equiv^{\mathcal{S}}_{\Sigma} \mathcal{O}$ (resp. $\mathsf{norm}(\mathcal{M}) \equiv^{\mathcal{S}}_{\Sigma} \mathcal{M}$). Since $\equiv^{\mathcal{S}}_{\Sigma}$ is an equivalence relation, it follows that $\mathcal{M} \equiv^{\mathcal{S}}_{\Sigma} \mathcal{O}$ iff $\mathsf{norm}(\mathcal{M}) \equiv^{\mathcal{S}}_{\Sigma} \mathsf{norm}(\mathcal{O})$. □

4.2 Syntactic Locality

For many of the inseparability relations introduced in Section 4.1, the problem of checking whether $\mathcal{M}$ is a module for $\mathcal{O}$ w.r.t. Σ is typically of very high complexity, and often undecidable, even for rather lightweight ontology languages.

Consequently, practical module extraction techniques are typically based on approximations, which ensure that the computed $\mathcal{M}$ is a module, yet not necessarily a minimal one. One such approximation that is often exploited in practice is based on the notion of *syntactic locality* [19, 20, 77, 21].

Intuitively, a normalised $\mathcal{SROIQ}$ axiom is $\bot$-local (resp. $\top$-local) if treating all

atomic concepts and roles outside Σ as the $\bot$ (resp. $\top$) concept and role, respectively, leads to the axiom being an "obvious" tautology.

Definition 15. A normalised $\mathcal{SROIQ}$ axiom α is $\bot$-local (resp. $\top$-local) w.r.t. a signature Σ if it satisfies the conditions given in the second (resp. third) column in Table 4.1. A normalised $\mathcal{SROIQ}$ ontology $\mathcal{O}$ is $\bot$-local (resp. $\top$-local) w.r.t. Σ if each of its axioms is $\bot$-local (resp. $\top$-local) w.r.t. Σ. Finally, we say that $\mathcal{O}$ is local w.r.t. Σ if it is either $\bot$-local or $\top$-local w.r.t. Σ. $\diamond$

The notions of $\bot$- and $\top$-locality can be naturally extended to rules corresponding to normalised $\mathcal{SROIQ}$ axioms.

The key properties of $\bot$- and $\top$-locality are summarised next.

Proposition 16. *Let $\mathcal{O}$ be a $\mathcal{SROIQ}$ ontology, Σ a signature and $x \in \{\bot, \top\}$.*

1. *If $\mathcal{O}$ is x-local w.r.t. Σ then $\mathcal{O} \equiv^{\mathsf{m}}_{\Sigma} \emptyset$.*
2. *If $\mathcal{M} \subseteq \mathcal{O}$ and $\mathcal{O} \setminus \mathcal{M}$ is x-local w.r.t. $\Sigma \cup \mathsf{Sig}(\mathcal{M})$, then $\mathcal{M}$ is a self-contained, strongly-depleting, and justification-preserving $\equiv^{\mathsf{m}}_{\Sigma}$-module of $\mathcal{O}$.*

Property 2 in Proposition 16 immediately suggests the notion of locality-based module.

Definition 17. Let $\mathcal{O}$ be a normalised $\mathcal{SROIQ}$ ontology and Σ a signature. Let $x \in \{\bot, \top\}$. The x-module for $\mathcal{O}$ w.r.t. Σ, denoted $\mathcal{M}^{x}_{[\mathcal{O},\Sigma]}$, is the smallest subset $\mathcal{M} \subseteq \mathcal{O}$ such that $\mathcal{O} \setminus \mathcal{M}$ is x-local w.r.t. $\Sigma \cup \mathsf{Sig}(\mathcal{M})$.

The $\top\bot^*$-module for $\mathcal{O}$ w.r.t. Σ is the least fix-point of the sequence $\{\mathcal{M}_i\}_{i\geq 1}$ where $\mathcal{M}_1 = \mathcal{M}^{\bot}_{[\mathcal{O},\Sigma]}$ and $\mathcal{M}_i$ is defined as follows for $i \geq 2$:

$$\mathcal{M}_i \;=\; \begin{cases} \mathcal{M}^{\top}_{[\mathcal{M}_{i-1},\Sigma]} \text{ if } i \text{ is odd} \\ \mathcal{M}^{\bot}_{[\mathcal{M}_{i-1},\Sigma]} \text{ if } i \text{ is even} \end{cases}$$

$\diamond$

Algorithm 1 Extraction of x-modules ($x \in \{\bot, \top\}$)

Input: $\mathcal{O}$ an ontology, Σ a signature
Output: $\mathcal{M}$ satisfying $\mathcal{M} = \mathcal{M}^{x}_{[\mathcal{O},\Sigma]}$

```
1: M := ∅
2: O' := O
3: repeat
4:     changed := FALSE
5:     for all α ∈ O' do
6:         if α not ⊥-local w.r.t. Σ ∪ Sig(M) then
7:             M := M ∪ {α}
8:             O' := O' \ {α}
9:             changed := TRUE
10: until changed = FALSE
11: return M
```

Example 18. Consider again the ontology $\mathcal{O}^{ex}$ from Figure 4.1, and also the signature $\Sigma = \{B, C, D, R\}$. We have that $\mathcal{M}^{\bot}_{[\mathcal{O}^{ex},\Sigma]} = \{r_3{-}r_9\}$, $\mathcal{M}^{\top}_{[\mathcal{O}^{ex},\Sigma]} = \{r_1{-}r_5\}$, and $\mathcal{M}^{\top\bot^*}_{[\mathcal{O}^{ex},\Sigma]} = \{r_3{-}r_5\}$. ◇

It is possible to extract $\top$- and $\bot$-modules in polynomial time with Algorithm 1. Consequently, it is also possible to extract $\top\bot^*$-modules in polynomial time. Furthermore, by Proposition 16, they are all self-contained, strongly depleting and justification-preserving. They are, however, generally not minimal, even amongst strongly depleting and self-contained modules [29, 53].

Part II

Module Extraction

Chapter 5

Extracting Modules via Datalog Reasoning

5.1 Overview

We now provide a high-level overview of our approach to module extraction, which is based on a novel reduction to a reasoning problem in datalog. Our approach builds on recent techniques that exploit datalog engines for ontology reasoning [55, 85, 101]. The connection between module extraction and datalog was first observed in [89], where it was shown that $\bot$-module extraction for the lightweight DL $\mathcal{EL}^+$ can be reduced to propositional datalog reasoning.

Our approach takes this connection much farther by providing a unified framework that supports module extraction for arbitrary first-order ontologies, as well as for a wide range of inseparability relations. Modules obtained using our approach can be tailored to the requirements of the application at hand. In addition to being significantly smaller in practice, our modules preserve the nice features of syntactic locality modules: they are widely applicable, they can be efficiently computed in practice, and they satisfy a wide range of additional desirable properties.

In what follows, we fix w.l.o.g. an arbitrary ontology $\mathcal{O}$ and a signature $\Sigma \subseteq \mathsf{Sig}(\mathcal{O})$. Unless otherwise stated, definitions and theorems in this chapter are parameterised by such $\mathcal{O}$ and Σ. As stated in Chapter 2, we assume that rules in $\mathcal{O}$ do not share variables.

Our overall strategy to extract a module $\mathcal{M}$ of $\mathcal{O}$ can be roughly summarised by the following steps:[1]

1. Choose a substitution θ mapping all existentially quantified variables in $\mathcal{O}$ to fresh Skolem constants, and obtain a datalog program $\mathcal{P}$ from $\mathcal{O}$ by

   (a) Skolemising all rules in $\mathcal{O}$ using θ to obtain function-free rules $\varphi \to \psi$, which may contain both $\wedge$ and $\vee$ in the head; and

   (b) replacing each of the resulting rules $\varphi \to \psi$ with the set $\{ \varphi \to \gamma \mid \gamma \in \psi \}$ of datalog rules; in this way, disjunctions in the head of rules are turned into conjunctions and split into different datalog rules.

   Clearly, such a program $\mathcal{P}$ logically entails $\mathcal{O}$ and thus preserves all its consequences.

2. Choose a Σ-dataset $\mathcal{D}_0$ of "initial facts" and compute the materialisation of $\mathcal{P} \cup \mathcal{D}_0$.

3. Choose a set $\mathcal{D}_r$ of "relevant facts" in the materialisation (possibly containing symbols outside Σ), and compute the supporting rules $\mathcal{P}'$ in $\mathcal{P}$ for each such fact.

4. Output the subset $\mathcal{M} \subseteq \mathcal{O}$ of all rules in $\mathcal{O}$ that correspond to some rule in $\mathcal{P}'$.

The subset $\mathcal{M}$ described above is fully determined by the substitution θ and the datasets $\mathcal{D}_0$ and $\mathcal{D}_r$. The main intuition behind our module extraction approach is that we can pick θ, $\mathcal{D}_0$ and $\mathcal{D}_r$ (and hence also $\mathcal{M}$) such that each proof ρ of a

[1]For simplicity, in this section we overlook certain technical details such as the presence of constants in $\mathcal{O}$. These will be thoroughly addressed later on.

$$
\begin{array}{lll}
r_1': & A(x) \to R(x, c_{y_1}) & r_1'': \quad A(x) \to B(c_{y_1}) \\
r_2: & A(x) \to R(x, o) & \\
r_3: & B(x) \wedge C(x) \to D(x) & \\
r_4: & R(x, y) \wedge C(y) \to E(x) & \\
r_5: & E(x) \to C(x) & \\
r_6': & D(x) \to F(x) & r_6'': \quad D(x) \to G(x) \\
r_7': & F(x) \to S(x, c_{y_2}) & \\
r_8: & S(x, y) \to H(x) & \\
r_9: & G(x) \to H(x) &
\end{array}
$$

Figure 5.1: Datalog program obtained from $\mathcal{O}^{ex}$ using $\theta = \{y_1 \mapsto c_{y_1}, y_2 \mapsto c_{y_2}\}$

Σ-consequence φ of $\mathcal{O}$ to be preserved under the inseparability relation of interest can be embedded into a collection of proofs in $\mathcal{P} \cup \mathcal{D}_0$ of a relevant fact from $\mathcal{D}_r$. In this way, we can ensure that $\mathcal{M}$ contains all the necessary rules to entail φ.

Example 19. To illustrate how our strategy might work in practice, consider our running example ontology $\mathcal{O}^{ex}$ from Figure 4.1 and signature $\Sigma = \{B, C, D, H\}$.

Assume that our goal is to compute a module $\mathcal{M}$ that is Σ-implication inseparable from $\mathcal{O}^{ex}$. Recall from Example 7 that the sentence $\varphi = D(x) \to H(x)$ is the only non-trivial Σ-implication entailed by $\mathcal{O}^{ex}$, and therefore the only requirement for $\mathcal{M}$ is that $\mathcal{M} \models \varphi$. Furthermore, note that proving $\mathcal{O}^{ex} \models \varphi$ amounts to proving $\mathcal{O}^{ex} \cup \{D(a)\} \models H(a)$ with a a fresh constant (see Proposition 1 in Section 2.1).

Figure 5.2(a) depicts a hyperresolution proof ρ showing how $H(a)$ can be derived from $D(a)$ and the set of clauses corresponding to r_6–r_9, where rule r_7 is transformed into the clause $r_7^{\kappa} = F(x) \to S(x, f_{y_2}^{r_7}(x))$. It follows that $\mathcal{M} = \{r_6\text{–}r_9\}$ is Σ-implication inseparable from $\mathcal{O}^{ex}$ since it covers the support of ρ. Moreover, $\mathcal{M}$ is minimal since $H(a)$ cannot be derived from any subset of $\{r_6\text{–}r_9\}$.

In our approach, we take $\mathcal{D}_0$ and $\mathcal{D}_r$ to contain, respectively, the initial fact $D(a)$ and the fact $H(a)$ to be proved. We also make θ map variables y_1 and y_2 to fresh constants c_{y_1} and c_{y_2}, respectively. The resulting datalog program $\mathcal{P}$ is shown in Figure 5.1.

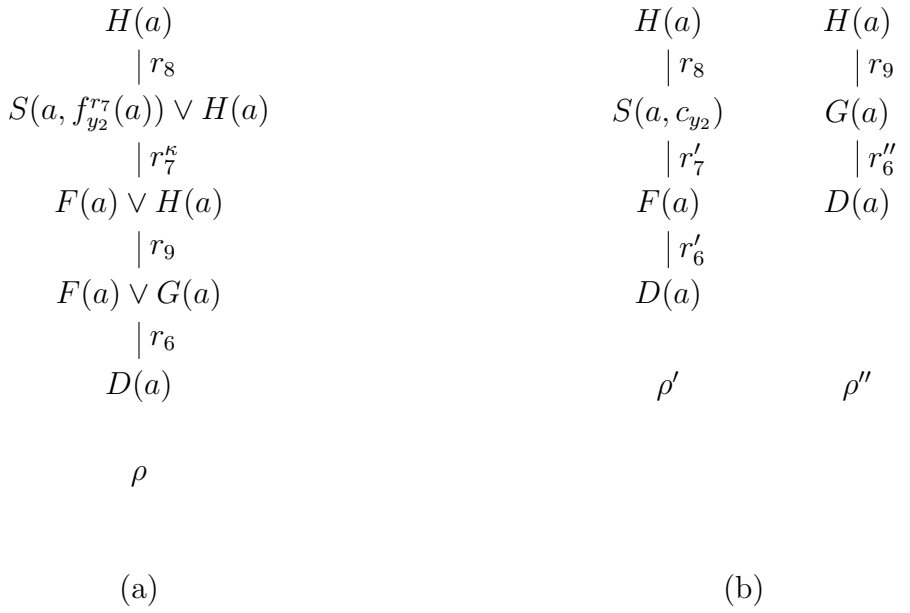


Figure 5.2: Proofs of $H(a)$ from $D(a)$ in (a) $\mathcal{O}^{ex}$ and (b) the corresponding datalog program

Figure 5.2(b) depicts proofs ρ' and ρ'' of $H(a)$ in $\mathcal{P} \cup \{D(a)\}$. The support of proof ρ'' in the datalog program consists of rules r''_6 and r_9, which stem from rules r_6 and r_9 in $\mathcal{O}^{ex}$; we can see, however, that $\{r_6, r_9\} \subsetneq \mathcal{M}$ and hence it does not entail φ. The same situation arises if we were to consider ρ' only, in which case we would recover only rules r_6–r_8. This is because the datalog program is a strengthening of $\mathcal{O}^{ex}$ and one particular proof in the datalog program may not translate back into a proof over the original ontology. Indeed, in order to compute $\mathcal{M}$, we need to consider the supports of *both* ρ' and ρ'', in which case we would successfully recover $\mathcal{M}$.

In this example, our approach would allow us to compute a minimal module. This is, however, not the case in general: since $\mathcal{P}$ is a strengthening of the given ontology there may be proofs in $\mathcal{P} \cup \mathcal{D}_0$ of facts in $\mathcal{D}_r$ that do not correspond to proofs of any Σ-consequence of the ontology, which may then lead to the inclusion of unnecessary rules in the module. $\diamond$

In the following sections we describe our approach formally.

- In Section 5.2, we define the general notion of a *module setting*, which captures the degrees of freedom of our framework and uniquely specifies the datalog program $\mathcal{P}$ and module $\mathcal{M}$ corresponding to specific choices of θ, $\mathcal{D}_0$, and $\mathcal{D}_r$. Furthermore, we establish the key correspondence between proofs over the original ontology $\mathcal{O}$ and sets of proofs over $\mathcal{P} \cup \mathcal{D}_0$, which we exploit in many of our subsequent technical results.

- In Section 5.3, we describe concrete module settings for each of the inseparability relations introduced in Section 4.1, namely implication (Section 5.3.1), fact (Section 5.3.2), query (Section 5.3.3), and model inseparability (Section 5.3.4) where we also show that syntactic locality $\bot$-modules can be precisely captured by an instantiation of our framework.

- In Section 5.4, we consider variants of the inseparability relations in Section 4.1 studied in the module extraction literature, and describe specific module settings for them. These results show that our framework can be easily adapted to capture new inseparability relations and hence illustrate the generality and versatility of our approach.

- In Section 5.5, we show that our modules are consistent with the intuition that stronger inseparability relations should lead to larger modules. For this, we introduce a notion of homomorphism between module settings, which will allow us to establish containment relations between the modules specified in Sections 5.3 and 5.4.

- In Section 5.6, we study the additional properties of our modules. We show that they are depleting and justification-preserving for all the inseparability relations in previous sections. Our modules, however, may not be strongly depleting or self-contained; although this may be beneficial, as it allows us to extract smaller modules, these properties are still important for ontology reuse

scenarios. Hence, we propose a technique that ensures that extracted modules are also strongly depleting and self-contained.

- In Section 5.7, we briefly discuss the complexity of module extraction within our framework and show tractability for DL-based ontology languages.

- Finally, in Section 5.8, we discuss the optimality of the module settings introduced in Sections 5.3 and 5.4. In particular, even though our modules are not minimal in general, we aim to determine whether the modules obtained from the settings in Sections 5.3 and 5.4 are the smallest possible within our framework.

5.2 The Notion of a Module Setting

In this section we present our framework for module extraction. The key notion is that of a *module setting*, which captures in a declarative way the main elements of our approach as discussed in Section 5.1.

Definition 20. A *module setting* for $\mathcal{O}$ and Σ is a tuple $\chi = \langle \theta, \mathcal{D}_0, \mathcal{D}_r \rangle$ with

- θ a substitution mapping each constant and existentially quantified variable in $\mathcal{O}$ to a (possibly fresh) constant;

- $\mathcal{D}_0$ a dataset mentioning only predicates from Σ; and

- $\mathcal{D}_r$ a dataset mentioning only predicates from $\mathsf{Sig}(\mathcal{O}) \cup \{\bot\}$.

For each rule $r = \varphi(\mathbf{x}) \to \exists \mathbf{y} \psi(\mathbf{x}, \mathbf{y})$ in $\mathcal{O}$, let $\Xi^\chi(r)$ be the following set of datalog rules:

$$\Xi^\chi(\varphi(\mathbf{x}) \to \exists \mathbf{y} \psi(\mathbf{x}, \mathbf{y})) = \{\, (\varphi \to \gamma)\theta \mid \gamma \in \psi \,\}$$

The *program* $\mathcal{P}^\chi$ of χ is defined as

$$\mathcal{P}^\chi = \bigcup_{r \in \mathcal{O}} \Xi^\chi(r)$$

and the *support* of χ is the set

$$\mathsf{supp}(\chi) = \{\, r \mid r \in \mathsf{supp}(\rho) \text{ with } \rho \text{ a proof in } \mathcal{P}^\chi \cup \mathcal{D}_0 \text{ of a fact from } \mathcal{D}_r \,\}.$$

Finally, if $\mathcal{F} = \{\, r \in \mathcal{O} \mid \mathsf{supp}(\chi) \cap \Xi^\chi(r) \neq \emptyset \,\}$, the *module* $\mathcal{M}^\chi$ of χ is defined as the following subset of $\mathcal{O}$:

$$\mathcal{M}^\chi = \mathcal{F} \cup \mathcal{F}^\bot \cup \mathcal{F}^\top \cup \mathcal{F}^\approx. \qquad \diamond$$

The mapping θ and the datasets $\mathcal{D}_0$ and $\mathcal{D}_r$ constitute the degrees of freedom of our framework, and Definition 20 ensures that specific choices of these parameters of the module setting χ fully determine the module $\mathcal{M}^\chi$.

The datalog program $\mathcal{P}^\chi$ is obtained by applying θ to each rule r in $\mathcal{O}$ while at the same time splitting the head atoms of r into different rules. The application of θ turns existentially quantified variables into (possibly fresh) constants and hence transforms $\mathcal{O}$ into a set of rules where all variables are universally quantified; additionally, θ maps the constants occurring in $\mathcal{O}$ into (possibly different) constants. Since θ is not required to be injective, it is possible for θ to map an existentially quantified variable and a constant from $\mathcal{O}$ to the same constant. As we will see next, $\mathcal{P}^\chi$ is a strengthening of $\mathcal{O}$ in the sense that it preserves all consequences of $\mathcal{O}$ when coupled with an arbitrary dataset. Analogous datalog strengthenings have been exploited in the literature to overestimate reasoning outcomes in description logic ontologies [60, 85, 59, 101, 100, 99].

The support $\mathsf{supp}(\chi)$ collects all datalog rules participating in any proof in $\mathcal{P}^\chi \cup \mathcal{D}_0$ of any relevant fact from $\mathcal{D}_r$. Intuitively, this support captures the "image" of the module in $\mathcal{P}^\chi$. Finally, the module $\mathcal{M}^\chi$ then consists of all the rules in $\mathcal{O}$ that have a corresponding datalog rule in the support $\mathsf{supp}(\chi)$.

Example 21. Let us reconsider Example 19 in Section 5.1, where we chose θ to

map variables y_1 and y_2 to fresh constants c_{y_1} and c_{y_2}, whereas $\mathcal{D}_0$ and $\mathcal{D}_r$ contain, respectively, the initial fact $D(a)$ and the fact $H(a)$ to be proved. Definition 20 ensures that $\mathcal{P}^\chi$ consists precisely of the datalog rules in Figure 5.1. The support $\mathsf{supp}(\chi)$ consists of all rules in the support of ρ' and ρ'' shown in Figure 5.2. Finally, $\mathcal{M}^\chi$ consists of rules r_6–r_9, as required. ◇

In subsequent technical results we reason about $\kappa(\mathcal{O})$ rather than $\mathcal{O}$. For this, we consider a natural extension of the substitution θ in χ where functional terms $f^r_y(\mathbf{t})$ occurring in $\kappa(\mathcal{O})$ are mapped to $y\theta$ (regardless of the particular $\mathbf{t}$) whenever y is in the domain of θ. By slight abuse of notation and for the sake of simplicity, we refer to such an extended substitution also as θ. In this way, for each rule $r \in \mathcal{O}$, its clausification $\kappa(r)$ is related to $\Xi^\chi(r)$ in an analogous way to how r is related to $\Xi^\chi(r)$. For example, if $\mathcal{O}$ contains rule $r = A(x) \rightarrow \exists y.R(x, y)$, whose clausification is $\kappa(r) = \{r' = A(x) \rightarrow R(x, f^r_y(x))\}$ and for which $\Xi^\chi(r) = \{r'' = A(x) \rightarrow R(x, y\theta)\}$, the (extended) substitution θ is such that $r\theta = r'\theta = r'$.

The following lemma establishes a key correspondence between hyperresolution proofs in (the clausification of) $\mathcal{O}$ and sets of proofs in the datalog program $\mathcal{P}^\chi$. Such a correspondence is already manifest in Figure 5.2 for our running example. Given an arbitrary dataset $\mathcal{D}$ and a substitution τ from constants to constants that is compatible with the substitution θ in χ (i.e., τ and θ coincide over the intersection of their domains - see Section 2.3), the lemma shows that each proof ρ of a disjunction of facts $\varphi = \bigvee_{i=1}^n \gamma_i$ in $\kappa(\mathcal{O}) \cup \mathcal{D}$ has a corresponding set $\mathcal{T}$ of proofs of each disjunct $\gamma_i\tau$ in $\mathcal{P}^\chi \cup \mathcal{D}\tau$. This implies, in particular, that $\mathcal{P}^\chi$ is indeed a strengthening of $\mathcal{O}$. Furthermore, the set $\mathcal{T}$ is both *support* and *structure preserving*: for every clause from $\kappa(r)$ participating in ρ, there is a proof in $\mathcal{T}$ with a datalog rule from $\Xi^\chi(r)$ in its support; finally, each proof in $\mathcal{T}$ is embeddable (see Section 2.3) into ρ and hence its structure is compatible with that of ρ.

Lemma 22. *Let χ be a module setting for $\mathcal{O}$ and Σ and let its corresponding substi-*

tution be θ. Let $\mathcal{D}$ be a dataset, and let τ be an arbitrary substitution from constants in $\mathcal{D}$ into constants that is compatible with θ. Finally, let φ be a (possibly empty) disjunction of facts and $\rho = (T, \lambda)$ a proof of φ in $\kappa(\mathcal{O}) \cup \mathcal{D}$. Then

(a) for each $\gamma \in \varphi$ there exists a proof ρ' in $\mathcal{P}^\chi \cup \mathcal{D}\tau$ of $\gamma\tau$ that is embeddable into ρ modulo $\tau \cup \theta$, and

(b) for each $r \in \mathcal{O}$ s.t. $\kappa(r) \cap \mathsf{supp}(\rho) \neq \emptyset$ either $r = \bot \to \Box$ or there exists $\gamma \in \varphi \cup \{\bot\}$ and a proof ρ' of $\gamma\tau$ in $\mathcal{P}^\chi \cup \mathcal{D}\tau$ that is embeddable into ρ modulo $\tau \cup \theta$, and such that $\Xi^\chi(r) \cap \mathsf{supp}(\rho') \neq \emptyset$.

Proof. In order to be able to reason by induction on the depth d of ρ, we prove that properties (a) and (b) hold even if φ is a disjunction of ground atoms (not necessarily function-free).

$d = 0$

If $\mathsf{supp}(\rho) = \emptyset$ then φ is a fact in $\mathcal{D}$ and $\varphi(\tau \cup \theta) = \varphi\tau \in \mathcal{D}\tau$ so there exists a trivial proof ρ' in $\mathcal{P}^\chi \cup \mathcal{D}\tau$ of $\varphi(\tau \cup \theta)$, which is clearly embeddable into ρ via $\tau \cup \theta$.

If $\mathsf{supp}(\rho) \neq \emptyset$ then $\kappa(\mathcal{O})$ must contain a clause of the form $(\ \to \varphi)$. Since, by assumption, the only rule in $\mathcal{O}$ with an empty head is $\bot \to \Box$, it must be the case that $\varphi \neq \Box$ and for each $\gamma \in \varphi$ there exists a proof ρ' in $\mathcal{P}^\chi \cup \mathcal{D}\tau$ of $\gamma\theta = \gamma(\tau \cup \theta)$ of depth 0 that is supported by $(\ \to \gamma\theta) \in \Xi^\chi(r)$ and embeddable into ρ modulo $\tau \cup \theta$.

In either case, both properties are satisfied.

$d > 0$

Let $\rho = (T, \lambda)$ with v the root of T and $w_1, \ldots, w_n$ the children of v. Consider a clause $s \in \kappa(\mathcal{O})$ such that φ is a hyperresolvent of s and $\lambda(w_1), \ldots, \lambda(w_n)$. Then, s must be of the form $\bigwedge_{i=1}^n \delta_i' \to \varphi'$ where

– $\lambda(w_i) = \delta_i \vee \psi_i$ for each $1 \leq i \leq n$, and

– $\varphi = \bigvee_{i=1}^{n} \psi_i \vee \varphi'\sigma$ with σ an MGU of δ_i, δ_i' for each $1 \leq i \leq n$.

(a) Let $\gamma \in \varphi$. We need to find a proof $\rho' = (T', \lambda')$ of $\gamma(\tau \cup \theta)$ in $\mathcal{P}^{\chi} \cup \mathcal{D}\tau$ that is embeddable into ρ modulo $\tau \cup \theta$. If $\gamma \in \psi_i$ then by induction hypothesis we can find such a proof. If $\gamma \in \varphi'\sigma$ then it holds that $\gamma = \gamma'\sigma$ for some $\gamma' \in \varphi'$. By induction hypothesis, we have a proof $\rho_i = (T_i, \lambda_i)$ in $\mathcal{P}^{\chi} \cup \mathcal{D}\tau$ of each $\delta_i(\tau \cup \theta)$ that is embeddable modulo $\tau \cup \theta$ into a proper subproof of ρ. Because σ is an MGU of δ_i and δ_i', with $\delta_i = \delta_i'\sigma$, we have that $\sigma(\tau \cup \theta)$ is an MGU of of $\delta_i(\tau \cup \theta)$ and $\delta_i'\theta$. Indeed, $\delta_i = \delta_i'\sigma$ implies $\delta_i(\tau \cup \theta) = (\delta_i'\sigma)(\tau \cup \theta)$. On the other hand, since neither $\mathcal{O}$ nor $\mathcal{D}$ mention function symbols, any functional term $f(\mathbf{t})$ occurring in $\delta_i = \delta_i'\sigma$ must originate in the Skolemisation of some existentially quantified variable in $\mathcal{O}$ that takes place as part of the application of κ to $\mathcal{O}$; consequently, as previously discussed, the effect of θ on $f(\mathbf{t})$ is independent of $\mathbf{t}$, so we have $\delta_i(\tau \cup \theta) = (\delta_i'\sigma)(\tau \cup \theta) = (\delta_i'(\tau \cup \theta))(\sigma(\tau \cup \theta))$. Moreover, because $\tau \cup \theta$ only extends the domain of θ to constants in $\mathcal{D}$ but not in $\mathcal{O}$, we have $(\delta_i'(\tau \cup \theta))(\sigma(\tau \cup \theta)) = (\delta_i'\theta)(\sigma(\tau \cup \theta))$, and thus $\delta_i(\tau \cup \theta) = (\delta_i'\theta)(\sigma(\tau \cup \theta))$. We can hence combine all the ρ_i with $(\bigwedge_{i=1}^{n} \delta_i' \rightarrow \gamma')\theta \in \mathcal{P}^{\chi}$, to obtain a proof of $(\gamma'\theta)(\sigma(\tau \cup \theta)) = (\gamma'\sigma)(\tau \cup \theta) = \gamma(\tau \cup \theta)$ in $\mathcal{P}^{\chi} \cup \mathcal{D}\tau$ that is clearly also embeddable into ρ modulo $\tau \cup \theta$.

(b) Consider $r \in \mathcal{O}$ such that there exists $r' \in \kappa(r) \cap \mathsf{supp}(\rho)$. We need to find $\gamma \in \varphi \cup \{\bot\}$ and a proof $\rho' = (T', \lambda')$ of $\gamma(\tau \cup \theta)$ in $\mathcal{P}^{\chi} \cup \mathcal{D}\tau$ that is embeddable into ρ modulo $\tau \cup \theta$ and has some rule from $\Xi^{\chi}(r)$ in its support.

Assume first that $r' = s$ and $r \neq \bot \rightarrow \Box$. Then it must be $\varphi' \neq \Box$ since, by assumption, $\bot \rightarrow \Box$ is the only rule in $\mathcal{O}$ with an empty head. We can

then pick any $\gamma' \in \varphi'$ and have ρ' be a proof of $(\gamma'\sigma)(\tau \cup \theta)$ in $\mathcal{P}^\chi \cup \mathcal{D}\tau$ that is supported by $(\bigwedge_{i=1}^n \delta_i' \to \gamma')\theta \in \Xi^\chi(r)$, as we saw when considering property (a).

Assume now that $r' \neq s$ and $r \neq \bot \to \Box$. Then there must be some i such that r' supports a proof $\tilde{\rho}_i$ of $\delta_i \vee \psi_i$ that is a subproof of ρ. Since $\tilde{\rho}_i$ is of depth $< d$, by i.h. there must be some $\delta_i'' \in \delta_i \vee \psi_i \cup \{\bot\}$ and a proof $\rho_i'' = (T'', \lambda'')$ of $\delta_i''(\tau \cup \theta)$ in $\mathcal{P}^\chi \cup \mathcal{D}\tau$ that is supported by some rule in $\Xi^\chi(r)$ and is embeddable into $\tilde{\rho}_i$. If $\delta_i'' \in \psi_i \cup \{\bot\} \subseteq \varphi \cup \{\bot\}$ then $\rho' = \rho''$ is the proof we are looking for. If $\delta_i'' = \delta_i$ (and $\delta_i \neq \bot$) then we can combine ρ'' with suitable proofs of each $\delta_j(\tau \cup \theta)$ for the remaining j (which we know exist by i.h.), as before, to construct a proof ρ' in $\mathcal{P}^\chi \cup \mathcal{D}\tau$ of $\gamma(\tau \cup \theta)$ for some $\gamma \in \varphi$, that is embeddable into ρ modulo $\tau \cup \theta$. $\Box$

Corollary 23. *Let χ be a module setting for $\mathcal{O}$ and Σ and let its corresponding substitution be θ. Let $\mathcal{D}$ be a dataset, and let τ be an arbitrary substitution from constants in $\mathcal{D}$ into constants that is compatible with θ. Finally, let φ be a (possibly empty) disjunction of facts and $\rho = (T, \lambda)$ a proof of φ in $\kappa(\mathcal{O}) \cup \mathcal{D}$. Then, there exists a non-empty set $\mathcal{T}$ of proofs in $\mathcal{P}^\chi \cup \mathcal{D}\tau$ satisfying the following properties:*

1. *Each $\rho' \in \mathcal{T}$ is a proof of $\gamma\tau$ for some $\gamma \in \varphi \cup \{\bot\}$. Furthermore, for each $\gamma \in \varphi$ there is some proof of $\gamma\tau$ in $\mathcal{T}$.*

2. *For each rule $r \in \mathcal{O}$ with $\kappa(r) \cap \mathsf{supp}(\rho) \neq \emptyset$ either $r = \bot \to \Box$ or there exists $\rho' \in \mathcal{T}$ satisfying $\Xi^\chi(r) \cap \mathsf{supp}(\rho') \neq \emptyset$.*

3. *Each $\rho' \in \mathcal{T}$ is embeddable into ρ modulo $\tau \cup \theta$.*

Proposition 1 and Corollary 23 establish how the datasets $\mathcal{D}_0$ and $\mathcal{D}_r$ in χ can be chosen so as to ensure that $\mathcal{M}^\chi$ preserves the required Σ-consequences.

Suppose $\mathcal{M}^\chi$ is required to preserve some Σ-consequence $r = \varphi(\mathbf{x}) \to \exists \mathbf{y} \psi(\mathbf{x}, \mathbf{y})$ of $\mathcal{O}$. By Proposition 1, given any substitution mapping variables in $\mathbf{x}$ to distinct

constants $\mathbf{c}$, it must be the case that $\mathcal{O} \cup \varphi(\mathbf{c}) \models \exists \mathbf{y} \psi(\mathbf{c}, \mathbf{y})$. Since $\mathcal{P}^\chi$ is a strengthening of $\mathcal{O}$ we also have $\mathcal{P}^\chi \cup \varphi(\mathbf{c}) \models \exists \mathbf{y} \psi(\mathbf{c}, \mathbf{y})$. Assume that we now choose $\mathcal{D}_0$ and $\mathcal{D}_r$ so as to satisfy the following requirements:

1. the instantiation $\varphi(\mathbf{c})$ of the body of r must be embeddable in $\mathcal{D}_0$; and
2. the set of all facts γ in the materialisation of $\mathcal{P}^\chi \cup \varphi(\mathbf{c})$ satisfying the instantiation $\exists \mathbf{y} \psi(\mathbf{c}, \mathbf{y})$ of the head of r must be embeddable into $\mathcal{D}_r$.

By completeness of hyperresolution, and given that $\mathcal{P}^\chi$ is a datalog program, there must exist a fact γ s.t. $\gamma \models \exists \mathbf{y} \psi(\mathbf{c}, \mathbf{y})$ and $\mathcal{P}^\chi \cup \varphi(\mathbf{c}) \vdash \gamma$. By Corollary 23 and Definition 20, the aforementioned requirements on $\mathcal{D}_0$ and $\mathcal{D}_r$ suffice to guarantee that $\mathcal{M}^\chi$ will preserve the consequence r.

Example 24. Consider again our running example in Section 5.1 and the associated module setting $\chi = \langle \theta, \mathcal{D}_0, \mathcal{D}_r \rangle$ given in Example 21. We can see that our choices of $\mathcal{D}_0 = \{D(a)\}$ and $\mathcal{D}_r = \{H(a)\}$ satisfy our sufficient requirements for $\mathcal{M}^\chi$ to entail $\varphi = D(x) \rightarrow H(x)$. First, any instantiation of the body $D(x)$ of φ is isomorphic to (and hence embeddable into) $\mathcal{D}_0$. Second, the materialisation of $\mathcal{P}^\chi \cup \{D(a)\}$ consists of facts $F(a)$, $S(a, c_{y_2})$ and $H(a)$, where the latter is isomorphic to the instantiation of the head of φ; since we chose $\mathcal{D}_r$ to consist precisely of $H(a)$, the second requirement is also satisfied. $\diamond$

The following theorem makes precise the aforementioned sufficient requirements for $\mathcal{M}^\chi$ to preserve the entailment of a Σ-rule r. Furthermore, it shows that whenever $\mathcal{M}^\chi$ entails r, it also contains all the justifications for r in the original ontology $\mathcal{O}$.

Theorem 25. *Let $r = \varphi(\mathbf{x}) \rightarrow \exists \mathbf{y} \psi(\mathbf{x}, \mathbf{y})$ be a rule and $\chi = \langle \theta, \mathcal{D}_0, \mathcal{D}_r \rangle$ a module setting for $\mathcal{O}$ and Σ such that for each substitution σ mapping all variables in $\mathbf{x}$ to pairwise distinct constants, there exists another substitution τ_σ that is compatible with θ and such that*

- $(\varphi\sigma)\tau_\sigma \subseteq \mathcal{D}_0$, *and*
- $((\psi\sigma)\tau_\sigma)\sigma' \cup \{\bot\} \subseteq \mathcal{D}_r$ *for each substitution* σ' *mapping variables in* $\mathbf{y}$ *to constants and such that* $\mathcal{P}^\chi \cup (\varphi\sigma)\tau_\sigma \models ((\psi\sigma)\tau_\sigma)\sigma'$.

Then $\mathcal{O} \models r$ *iff* $\mathcal{M}^\chi \models r$. *Furthermore, if* $\mathcal{O}'$ *is a justification for* r *in* $\mathcal{O}$, *then* $\mathcal{O}' \subseteq \mathcal{M}^\chi$.

Proof. Since $\mathcal{M}^\chi \subseteq \mathcal{O}$, it follows from monotonicity of first-order logic that $\mathcal{O} \models r$ whenever $\mathcal{M}^\chi \models r$. To prove the opposite direction of the implication, it suffices to show that if $\mathcal{O}' \subseteq \mathcal{O}$ is a justification for r in $\mathcal{O}$, then $\mathcal{O}' \subseteq \mathcal{M}^\chi$.

If $\psi = \Box$ then, by minimality of $\mathcal{O}'$, given a substitution σ mapping variables in $\mathbf{x}$ to fresh distinct constants, there must be a proof ρ of $\Box$ in $\kappa(\mathcal{O}) \cup \varphi\sigma$ such that $\mathsf{supp}(\rho) \cap \kappa(r) \neq \emptyset$ for each $r \in \mathcal{O}'$. By assumption, there exists a substitution τ_σ that is compatible with θ and such that $(\varphi\sigma)\tau_\sigma \subseteq \mathcal{D}_0$. By Corollary 23, for each $r \in \mathcal{O}'$ either it is $r = \bot \to \Box$ or there exists a proof $\rho_\bot$ in $\mathcal{P}^\chi \cup (\varphi\sigma)\tau_\sigma$ of $\bot$ such that $\mathsf{supp}(\rho_\bot) \cap \Xi^\chi(r) \neq \emptyset$. If $r = \bot \to \Box$ then, since by assumption the only rule in $\mathcal{O}$ with an empty head is $\bot \to \Box$, in particular it must also be the case that $\mathcal{O} \models \varphi(\mathbf{x}) \to \bot$. It suffices to show that in this case also $\mathcal{M}^\chi \models \varphi(\mathbf{x}) \to \bot$ (as we do next when considering $\psi \neq \Box$): then, it follows that $\bot \in \mathsf{Sig}(\mathcal{M}^\chi)$ and therefore $r = \bot \to \Box \in \mathcal{M}^\chi$ because $\mathcal{M}^\chi$ is an ontology. If $r \neq \bot \to \Box$ then, since $(\varphi\sigma)\tau_\sigma \subseteq \mathcal{D}_0$ and $\bot \in \mathcal{D}_r$, it follows that $r \in \mathcal{M}^\chi$.

If $\psi \neq \Box$ we can assume w.l.o.g. that $\psi = \bigvee_{i=1}^n \psi_i$ with $m > 0$ and each ψ_i a conjunction of atoms. For a fresh predicate Q, consider the ontology

$$\mathcal{O}_Q = \{\, \psi_i(\mathbf{x}, \mathbf{y}) \to Q(\mathbf{x}) \mid 1 \leq i \leq n \,\}$$

It is immediate that, for each subset $\mathcal{O}'' \subseteq \mathcal{O}$, it is $\mathcal{O}'' \models r$ iff $\mathcal{O}'' \cup \mathcal{O}_Q \models \varphi(\mathbf{x}) \to Q(\mathbf{x})$. Therefore, by minimality of $\mathcal{O}'$, there must be some $\mathcal{O}'_Q \subseteq \mathcal{O}_Q$ such that $\mathcal{O}' \cup \mathcal{O}'_Q$ is a justification of $\varphi(\mathbf{x}) \to Q(\mathbf{x})$ in $\mathcal{O} \cup \mathcal{O}_Q$. By minimality of $\mathcal{O}' \cup \mathcal{O}'_Q$, given a substitution

σ mapping variables in $\mathbf{x}$ to fresh distinct constants, there must be a proof ρ of $Q(\mathbf{x})\sigma$ in $\kappa(\mathcal{O} \cup \mathcal{O}_Q) \cup \varphi\sigma$ such that $\kappa(r) \cap \mathsf{supp}(\rho) \neq \emptyset$ for each $r \in \mathcal{O}' \cup \mathcal{O}'_Q$.

By assumption, there is a substitution τ_σ that is compatible with θ and such that $(\varphi\sigma)\tau_\sigma \subseteq \mathcal{D}_0$. Since $\mathcal{O}_Q$ does not contain any existentially quantified variables, or any constants that do not occur already in $\mathcal{O}$, there exists a module setting $\chi_Q = \langle \theta^Q, \mathcal{D}_0^Q, \mathcal{D}_r^Q \rangle$ for $\mathcal{O} \cup \mathcal{O}_Q$ and Σ such that $\theta^Q = \theta$, and therefore $\mathcal{P}^{\chi_Q} = \mathcal{P}^\chi \cup \mathcal{O}_Q$. By Corollary 23, for each $s \in \mathcal{O}' \subseteq \mathcal{O}' \cup \mathcal{O}'_Q$ either it is $s = \bot \to \Box$ or there exists a proof ρ' in $\mathcal{P}^{\chi_Q} \cup (\varphi\sigma)\tau_\sigma$ of either $(Q(\mathbf{x})\sigma)\tau_\sigma$ or $\bot$ such that $s' \in \Xi^{\chi_Q}(s) \cap \mathsf{supp}(\rho') \neq \emptyset$.

If $s = \bot \to \Box$ then there must be some proof in $\kappa(\mathcal{O} \cup \mathcal{O}_Q) \cup \varphi\sigma$ of a disjunction of the form $\bot \vee \phi$ (with ϕ possibly empty). By Corollary 23 there exists a proof $\rho_\bot$ in $\mathcal{P}^{\chi_Q} \cup (\varphi\sigma)\tau_\sigma$ of $\bot$. Furthermore, since $\bot$ does not mention Q, and neither does the body of any rule in $\mathcal{P}^{\chi_Q} = \mathcal{P}^\chi \cup \mathcal{O}_Q$, the proof $\rho_\bot$ must actually be a proof in $\mathcal{P}^\chi \cup (\varphi\sigma)\tau_\sigma$. Because $(\varphi\sigma)\tau_s \subseteq \mathcal{D}_0$ and $\bot \in \mathcal{D}_r$, it follows that $\mathsf{supp}(\rho_\bot) \subseteq \mathsf{supp}(\chi)$. Hence $\bot \in \mathsf{Sig}(\mathcal{M}^\chi)$, and consequently $s = \bot \to \Box \in \mathcal{M}^\chi$.

Otherwise, if ρ' is a proof of $\gamma = \bot$ then, as before, ρ' must be a proof in $\mathcal{P}^\chi \cup (\varphi\sigma)\tau_\sigma$. Then, since $(\varphi\sigma)\tau_s \subseteq \mathcal{D}_0$ and $\bot \in \mathcal{D}_r$, we have $s' \in \mathsf{supp}(\chi)$ and thus $s \in \mathcal{M}^\chi$. If ρ' is a proof of $\gamma = (Q(\mathbf{x})\sigma)\tau_\sigma$, let $\rho' = (T, \lambda)$ with v the root of T and $w_1, \ldots, w_m$ its children. The rule applied at the top of ρ' must be from $\mathcal{O}_Q$ and therefore different from s'. In particular, this rule must be of the form $\psi_i(\mathbf{x}, \mathbf{y}) \to Q(\mathbf{x})$, with $((\psi_i\tilde{\sigma})\tau_\sigma) = \bigwedge_{j=1}^m \lambda(w_j)$ for some extension $\tilde{\sigma}$ of σ to $\mathbf{y}$. Clearly, there is some substitution σ' with domain $\mathbf{y}$ such that $((\psi_i\tilde{\sigma})\tau_\sigma) = ((\psi_i\sigma)\tau_\sigma)\sigma'$. The rule s' must thus be in the support of a proof ρ'_j in $\mathcal{P}^{\chi_Q} \cup (\varphi\sigma)\tau_s$ of some $\lambda(w_j)$. Since r does not mention Q neither does $\lambda(w_j)$, and again we have that ρ' must in fact be a proof in $\mathcal{P}^\chi \cup (\varphi\sigma)\tau_\sigma$. This implies $\mathcal{P}^\chi \cup (\varphi\sigma)\tau_s \models \lambda(w_j)$ and hence by assumption $\lambda(w_j) \in \mathcal{D}_r$. Finally, since $(\varphi\sigma)\tau_\sigma \subseteq \mathcal{D}_0$, we have $s' \in \mathsf{supp}(\chi)$ and consequently $s \in \mathcal{M}^\chi$. $\Box$

5.3 Modules for each Inseparability Relation

Theorem 25 tells us how to choose a module setting χ so that its corresponding module $\mathcal{M}^\chi$ preserves a particular consequence of $\mathcal{O}$. However, in order for $\mathcal{M}^\chi$ to be a $\equiv_\Sigma^{\mathcal{S}}$-module of $\mathcal{O}$ for a given inseparability relation $\mathcal{S}$, it must preserve not one, but all of the (possibly infinitely many) relevant consequences in $\mathsf{rel}_{\mathcal{S}}(\mathcal{O}, \Sigma)$ (recall Theorem 10 in Chapter 4).

In this section we consider each inseparability relation $\mathcal{S} \in \{\mathsf{m}, \mathsf{q}, \mathsf{f}, \mathsf{i}\}$, and formulate a specific module setting $\chi_{\mathcal{S}}$ which provably yields a $\equiv_\Sigma^{\mathcal{S}}$-module of $\mathcal{O}$. Later on in Section 5.8 we will consider the optimality of these module settings—that is, whether there may exist a different setting that yields a smaller module for the relevant inseparability relation.

5.3.1 Implication Inseparability

Our running example immediately suggests a natural module setting $\chi_{\mathsf{i}} = \langle \theta^{\mathsf{i}}, \mathcal{D}_0^{\mathsf{i}}, \mathcal{D}_r^{\mathsf{i}} \rangle$ that guarantees implication inseparability.

As in our example, we pick the substitution θ^{i} to be as "general" as possible by Skolemising each existentially quantified variable to a distinct fresh constant and mapping constants occurring in $\mathcal{O}$ to themselves. To pick $\mathcal{D}_0^{\mathsf{i}}$ and $\mathcal{D}_r^{\mathsf{i}}$ we rely on the application of the sufficient conditions established in Theorem 25 to each of the (quadratically many) Σ-implications $A(\mathbf{x}) \rightarrow B(\mathbf{x})$ with $\mathbf{x} = (x_1, \ldots, x_n)$. More precisely, to capture the instantiations of the body $A(\mathbf{x})$ we define $\mathcal{D}_0$ to contain a fact $A(c_A^1, \ldots, c_A^n)$ involving fresh constants c_A^i uniquely associated to the predicate A; furthermore, to capture the head of the implication, we define $\mathcal{D}_r$ so as to contain a fact $B(c_A^1, \ldots, c_A^n)$. In this way, the dataset $\mathcal{D}_0^{\mathsf{i}}$ contains linearly many and $\mathcal{D}_r^{\mathsf{i}}$ quadratically many facts in the size of the signature Σ.

Definition 26. For each existentially quantified variable y in $\mathcal{O}$, let c_y be a fresh

constant. Furthermore, for each $A \in \Sigma$ of arity n, let $\mathbf{c}_A = (c_A^1, \ldots, c_A^n)$ be an array of fresh constants. The module setting $\chi_{\mathsf{i}} = \langle \theta^{\mathsf{i}}, \mathcal{D}_0^{\mathsf{i}}, \mathcal{D}_r^{\mathsf{i}} \rangle$ is defined as follows:

- $\theta^{\mathsf{i}} = \{\, y \mapsto c_y \mid y \text{ existentially quantified in } \mathcal{O} \,\} \cup \{\, c \mapsto c \mid c \in \mathsf{Ct}(\mathcal{O}) \,\}$,
- $\mathcal{D}_0^{\mathsf{i}} = \{\, A(\mathbf{c}_A) \mid A \in \Sigma \,\}$; and
- $\mathcal{D}_r^{\mathsf{i}} = \{\, B(\mathbf{c}_A) \mid A \neq B \text{ predicates in } \Sigma \text{ of the same arity} \,\} \cup \{\bot\}$. ◇

The module setting χ_{i} is reminiscent of the datalog encodings typically used to check whether a concept A is subsumed by another concept B w.r.t. a "lightweight" ontology $\mathcal{O}$ [60, 85]. There, existentially quantified variables in rules are also Skolemised as fresh constants to produce a datalog program $\mathcal{P}$, and then it is checked whether $\mathcal{P} \cup \{A(a)\} \models B(a)$.

The module setting χ_{i} captures implication inseparability as a straightforward consequence of Theorem 25.

Theorem 27. *$\mathcal{M}^{\chi_{\mathsf{i}}} \equiv_{\Sigma}^{\mathsf{i}} \mathcal{O}$.*

Proof. Consider an arbitrary rule of the form $A(\mathbf{x}) \to B(\mathbf{x})$ with $\mathbf{x} = (x_1, \ldots, x_n)$ a vector of distinct variables and A, B distinct n-ary predicates from Σ. Let σ be a substitution mapping $x_1, \ldots, x_n$ to distinct constants $c_1, \ldots, c_n$, and τ_σ another substitution such that $c_i \tau_\sigma = c_A^i$. By definition of χ_{i} we have $(A(\mathbf{x})\sigma)\tau_\sigma \in \mathcal{D}_0^{\mathsf{i}}$ and $(B(\mathbf{x})\sigma)\tau_\sigma \in \mathcal{D}_r^{\mathsf{i}}$, and thus, by Theorem 25, it follows that $\mathcal{O} \models A(\mathbf{x}) \to B(\mathbf{x})$ iff $\mathcal{M}^{\chi_{\mathsf{i}}} \models A(\mathbf{x}) \to B(\mathbf{x})$. □

5.3.2 Fact Inseparability

By Theorem 10, fact inseparability requires the preservation of all the datalog Σ-rules entailed by $\mathcal{O}$. Thus, in contrast to implication inseparability, it may require the preservation of a very large (and possibly even infinite) set of entailments. Unsurprisingly, the module setting χ_{i} cannot be used to capture fact inseparability as illustrated by the following example.

Example 28. Consider $\mathcal{O}^{ex}$ and $\Sigma_1 = \{B, C, D, H\}$, for which $\mathcal{M}^{\chi_\mathsf{i}} = \{r_6\text{–}r_9\}$. As seen in Example 7, for $\mathcal{D} = \{B(a), C(a)\}$ it is the case that $\mathcal{O}^{ex} \cup \mathcal{D} \models D(a)$, while we also have $\mathcal{M}^{\chi_\mathsf{i}} \cup \mathcal{D} \not\models D(a)$; hence, $\mathcal{M}^{\chi_\mathsf{i}}$ is not Σ_1-fact inseparable from $\mathcal{O}^{ex}$.

Equivalently, $\mathcal{O}^{ex}$ entails the datalog rule $r_3 = B(x) \wedge C(x) \rightarrow D(x)$, whereas $\mathcal{M}^{\chi_\mathsf{i}}$ does not and hence Theorem 25 is no longer applicable since $\mathcal{D}$, which instantiates the body of r_3, cannot be embedded into $\mathcal{D}_0^\mathsf{i} = \{B(c_B), C(c_C), D(c_D), H(c_H)\}$. $\diamond$

Thus, we will next define a suitable module setting $\chi_\mathsf{f} = \langle \theta^\mathsf{f}, \mathcal{D}_0^\mathsf{f}, \mathcal{D}_r^\mathsf{f} \rangle$ to capture fact inseparability. As in the previous case, we will exploit the sufficient conditions given in Theorem 25. To this end, we first need to make sure that $\mathcal{D}_0^\mathsf{f}$ (resp. $\mathcal{D}_r^\mathsf{f}$) captures all possible body (resp. head) instantiations of all possible datalog rules over Σ that may be entailed by $\mathcal{O}$. We achieve this by choosing $\mathcal{D}_0^\mathsf{f}$ and $\mathcal{D}_r^\mathsf{f}$ to be the "most constrained" Σ-dataset possible, which is typically referred to in the literature as the *critical dataset* [65, 22].

Definition 29. Let Σ be a signature and let $*$ be a fresh constant. The *critical* Σ*-dataset* is defined as follows:

$$\mathcal{D}_\Sigma^* = \{\, A(\overbrace{*, \ldots, *}^{n}) \mid A \text{ } n\text{-ary predicate in } \Sigma \,\} \qquad \diamond$$

Indeed, it is straightforward to see that every Σ-dataset (and hence any datalog rule instantiation) can be embedded into $\mathcal{D}_\Sigma^*$ by mapping every constant into $*$.

As in the case of χ_i, we choose the substitution θ^f to be as general as possible by mapping existentially quantified variables to distinct fresh constants. However, in contrast to χ_i, we require all constants occurring in $\mathcal{O}$ to be mapped to $*$ rather than to themselves. This choice is justified by the following example.

Example 30. Consider $\mathcal{O}^{ex}$ and $\Sigma = \{A, C, E\}$. Clearly, for $\mathcal{D} = \{A(a), C(o)\}$ we have $\mathcal{O}^{ex} \cup \mathcal{D} \models E(a)$ due to rules r_2 and r_4 in $\mathcal{O}^{ex}$. If we were to pick θ^f to be θ^i, which maps constant o in $\mathcal{O}^{ex}$ to itself, we would obtain $\chi_\mathsf{f} = \emptyset$ even if we choose

$\mathcal{D}_0^{\mathsf{f}}$ and $\mathcal{D}_r^{\mathsf{f}}$ as the critical Σ-dataset. Indeed, the relevant fact $E(*)$ would not be provable from $\mathcal{P}^{\chi_{\mathsf{f}}} \cup \mathcal{D}_0^{\mathsf{f}}$. ◇

We are now ready to define χ_{f} formally.

Definition 31. Let constants c_y be as in Definition 26, and let $*$ be a fresh constant. The module setting $\chi_{\mathsf{f}} = \langle \theta^{\mathsf{f}}, \mathcal{D}_0^{\mathsf{f}}, \mathcal{D}_r^{\mathsf{f}} \rangle$ is defined as follows:

- $\theta^{\mathsf{f}} = \{\, y \mapsto c_y \mid y \text{ existentially quantified in } \mathcal{O} \,\} \cup \{\, c \mapsto * \mid c \in \mathsf{Ct}(\mathcal{O}) \,\}$,
- $\mathcal{D}_0^{\mathsf{f}} = \mathcal{D}_\Sigma^*$, and
- $\mathcal{D}_r^{\mathsf{f}} = \mathcal{D}_\Sigma^* \cup \{\bot\}$. ◇

Example 32. The datalog program generated by θ^{f} for $\mathcal{O}^{ex}$ coincides with that in Figure 5.1 in all rules except for r_2, which now becomes $r_2' : \ A(x) \to R(x, *)$. If we consider again $\Sigma_1 = \{B, C, D, H\}$, and $\mathcal{D} = \{B(a), C(a)\}$ from Example 28, we clearly have $\mathcal{P}^{\chi_{\mathsf{f}}} \cup \mathcal{D}_0^{\mathsf{f}} \vdash D(*) \in \mathcal{D}_r^{\mathsf{f}}$ since $\{B(*), C(*)\} \subseteq \mathcal{D}_0^{\mathsf{f}}$. The unique proof ρ of $D(*)$ in $\mathcal{P}^{\chi_{\mathsf{f}}} \cup \mathcal{D}_0^{\mathsf{f}}$ is only supported by r_3; this guarantees that $r_3 \in \mathcal{M}^{\chi_{\mathsf{f}}}$ and thus ρ corresponds directly to a proof of $D(a)$ in $\mathcal{M}^{\chi_{\mathsf{f}}} \cup \mathcal{D}$. Hence, we have $\mathcal{M}^{\chi_{\mathsf{f}}} \cup \mathcal{D} \models D(a)$, as required.

If we now consider again the signature $\Sigma = \{A, C, E\}$ from Example 30, we can observe in Figure 5.3 how our choice of mapping constant o to $*$ ensures that the module contains the necessary rules r_2 and r_4. ◇

We can now exploit Theorem 25 once more to show that χ_{f} captures fact inseparability.

Theorem 33. $\mathcal{M}^{\chi_{\mathsf{f}}} \equiv_\Sigma^{\mathsf{f}} \mathcal{O}$.

Proof. Let $r = \varphi \to \gamma$ be a datalog rule over Σ. Let σ be a substitution mapping all variables in r to pairwise distinct constants, and τ_* a substitution mapping each constant in the range of σ to $*$. By definition of χ_{f} we have $(\varphi\sigma)\tau_* \subseteq \mathcal{D}_0^{\mathsf{f}}$ and

(a) (b)

Figure 5.3: Proofs of (a) $E(a)$ in $\mathcal{O}^{ex} \cup \{A(a), C(o)\}$ and (b) $E(*)$ in $\mathcal{P}^{\chi_f} \cup \mathcal{D}_0^f$

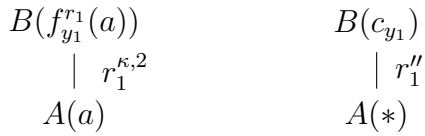
$(\gamma\sigma)\tau_* \subseteq \mathcal{D}_r^f$ and thus, by Theorem 25 it follows that $\mathcal{O} \models r$ iff $\mathcal{M}^{\chi_f} \models r$. Finally, by Proposition 8, this implies $\mathcal{M}^{\chi_f} \equiv_\Sigma^f \mathcal{O}$. □

5.3.3 Query Inseparability

Positive existential queries constitute a much richer query language than facts as they allow for existentially quantified variables. As a result, the query inseparability requirement inevitably leads to larger modules.

Example 34. Consider $\mathcal{O}^{ex}$ and $\Sigma = \{A, B\}$. Given the Σ-dataset $\mathcal{D} = \{A(a)\}$ and the Σ-query $q = \exists y B(y)$, we have that $\mathcal{O}^{ex} \cup \mathcal{D} \models q$ (due to rule r_1). In this case, however, $\mathcal{M}^{\chi_f}$ is empty and thus $\mathcal{M}^{\chi_f} \cup \mathcal{D} \not\models q$. Indeed, the only additional facts in the materialisation of $\mathcal{P}^{\chi_f} \cup \{A(*), B(*)\}$ are $R(*, c_{y_1})$ and $B(c_{y_1})$, and hence neither r_1' nor r_1'' are in $\mathsf{supp}(\chi_f)$. This suggests that, although the critical Σ-dataset $\mathcal{D}_\Sigma^*$ is constrained enough to embed every Σ-dataset, we may need to consider additional relevant facts to capture all proofs of all Σ-queries. In particular, rule r_1 implies that B has a non-empty extension whenever A does: a dependency that is then checked by q. This can be captured by considering fact $B(c_{y_1})$ as relevant, in which case r_1 would be included in the module. ◇

By Theorem 10, query inseparability requires the preservation of all Σ-rules entailed by $\mathcal{O}$ (and not just of those that are datalog). In particular, first-order rules



(a) (b)

Figure 5.4: Proofs of (a) $B(f_{y_1}^{r_1}(a))$ in $\mathcal{O}^{ex} \cup \{A(a)\}$ and (b) $B(c_{y_1})$ in $\mathcal{P}^{\chi_\mathsf{q}} \cup \mathcal{D}_0^\mathsf{q}$

may involve existentially quantified variables, which correspond in our framework to Skolem constants. This naturally suggests a module setting χ_q that differs from χ_f only in that Σ-facts involving Skolem constants (and not just those mentioning only $*$) are also considered relevant.

Definition 35. Let constants c_y and $*$ be as in Definition 31. We define the module setting $\chi_\mathsf{q} = \langle \theta^\mathsf{q}, \mathcal{D}_0^\mathsf{q}, \mathcal{D}_r^\mathsf{q} \rangle$ as follows:

- $\theta^\mathsf{q} = \theta^\mathsf{f}$,
- $\mathcal{D}_0^\mathsf{q} = \mathcal{D}_\Sigma^*$, and
- $\mathcal{D}_r^\mathsf{q} = \{\, A(a_1, \ldots, a_n) \mid A \in \Sigma, \text{ each } a_j \text{ is either } * \text{ or some } c_y \,\} \cup \{\bot\}$. ◇

Example 36. Coming back to Example 34, we can observe in Figure 5.4 how a proof of $B(f_{y_1}^{r_1}(a))$ in $\kappa(\mathcal{O}^{ex}) \cup \{A(a)\}$ that is supported by $r_1^{\kappa,2} = A(x) \to B(f_{y_1}^{r_1}(x))$ can be recovered from a proof of $B(c_{y_1})$ in $\mathcal{P}^{\chi_\mathsf{q}} \cup \mathcal{D}_0^\mathsf{q}$ that is supported by r_1''. The definition of χ_q ensures that $B(c_{y_1}) \in \mathcal{D}_r^\mathsf{q}$, and hence $r_1 \in \mathcal{M}^{\chi_\mathsf{q}}$. ◇

Theorem 37. *$\mathcal{M}^{\chi_\mathsf{q}} \equiv_\Sigma^\mathsf{q} \mathcal{O}$.*

Proof. Let $r = \varphi \to \exists \mathbf{y} \psi$ be a rule over Σ with a non-empty head. Let σ be a substitution mapping all variables in r to pairwise distinct constants and τ_* a substitution that maps each constant in the range of σ to $*$. By definition of χ_q we have $(\varphi\sigma)\tau_* \subseteq \mathcal{D}_0^\mathsf{q}$, and also $\psi\sigma' \subseteq \mathcal{D}_r^\mathsf{q}$ for each substitution σ' mapping all variables in r to constants in $\mathsf{Ct}(\mathcal{D}_0^\mathsf{q} \cup \mathcal{D}_r^\mathsf{q}) \cup \mathsf{range}(\theta^\mathsf{q})$. Thus, by Theorem 25, it follows that $\mathcal{O} \models r$ iff $\mathcal{M}^{\chi_\mathsf{q}} \models r$. By Proposition 8, this implies $\mathcal{M}^{\chi_\mathsf{q}} \equiv_\Sigma^\mathsf{q} \mathcal{O}$. □

5.3.4 Model Inseparability

Model inseparability differs substantially from all the previous inseparability relations: rather than just the preservation of rule-shaped consequences, it requires the preservation of models (and hence of second-order consequences). Theorem 25, which we have repeatedly exploited to show that our modules preserve the required entailments, relies on the properties of hyperresolution (a first-order logic calculus); hence, it is not applicable to show preservation of second-order logic consequences. In particular, as the following example illustrates, the modules generated by χ_{q} may not be Σ-model inseparable from $\mathcal{O}$.

Example 38. Consider $\mathcal{O}^{ex}$ and $\Sigma = \{A, C, D, R\}$, in which case $\mathcal{M}^{\chi_{\mathsf{q}}} = \{r_1, r_2, r_4, r_5\}$. As we saw in Example 7, the interpretation $\mathcal{I}$ where $\Delta^{\mathcal{I}} = \{a, o\}$, $A^{\mathcal{I}} = E^{\mathcal{I}} = \{a\}$, $B^{\mathcal{I}} = C^{\mathcal{I}} = \{o\}$, $D^{\mathcal{I}} = \emptyset$ and $R^{\mathcal{I}} = \{(a, o)\}$ is a model of $\mathcal{M}^{\chi_{\mathsf{q}}}$; however, it cannot be extended to a model of $\mathcal{O}$ without reinterpreting A, C, D or R. ◇

Intuitively, when constructing a model of $\mathcal{O}$, fixing the interpretation of certain predicates restricts the ways in which the remaining predicates can be interpreted. These restrictions are obviously determined by the dependencies introduced by the rules in $\mathcal{O}$. To capture model inseparability, we need to ensure that $\mathcal{M}^{\chi}$ preserves all such relevant dependencies between predicates in Σ. For this, we pick θ and $\mathcal{D}_0$ in such a way that any model of $\mathcal{O}$ can be embedded in the materialisation of $\mathcal{P}^{\chi} \cup \mathcal{D}_0$; in turn, we choose $\mathcal{D}_r$ in such a way that it captures the Σ-reducts of all those models. If this is the case, the proofs in $\mathcal{P}^{\chi} \cup \mathcal{D}_0$ of facts from $\mathcal{D}_r$ will then capture all the dependencies between predicates in Σ.

Example 39. Note that $\mathcal{I}$ in Example 38 cannot be embedded in the materialisation of $\mathcal{P}^{\chi_{\mathsf{q}}} \cup \mathcal{D}_0^{\mathsf{q}}$ since the only facts over $\{B, C\}$ that it contains are $C(*)$ and $B(c_{y_1})$, and the constant o cannot be mapped to both c_{y_1} and $*$. ◇

To capture all models of $\mathcal{O}$, we once more pick $\mathcal{D}_0 = \mathcal{D}_\Sigma^*$; but now, in contrast to

χ_{q}, we choose a substitution θ that maps both existentially quantified variables and constants in $\mathcal{O}$ to $*$. Furthermore, to ensure that $\mathcal{D}_r$ captures the Σ-reducts of all those models, we pick $\mathcal{D}_r$ as $\mathcal{D}^*_\Sigma$ as well.

Definition 40. The module setting $\chi_{\mathsf{m}} = \langle \theta^{\mathsf{m}}, \mathcal{D}_0^{\mathsf{m}}, \mathcal{D}_r^{\mathsf{m}} \rangle$ is as follows:

- $\theta^{\mathsf{m}} = \{\, y \mapsto * \mid y \text{ existentially quantified in } \mathcal{O} \,\} \cup \{\, c \mapsto * \mid c \in \mathsf{Ct}(\mathcal{O}) \,\}$,
- $\mathcal{D}_0^{\mathsf{m}} = \mathcal{D}^*_\Sigma$, and
- $\mathcal{D}_r^{\mathsf{m}} = \mathcal{D}^*_\Sigma \cup \{\bot\}$. ◇

Example 41. Consider $\mathcal{O}^{ex}$, Σ and $\mathcal{I}$ as in Example 38. The substitution θ^{m} maps the existentially quantified variables in r_1 and r_7 to $*$. Thus, rules r_1 and r_7 in $\mathcal{O}^{ex}$ correspond to the following rules in $\mathcal{P}^{\chi_{\mathsf{m}}}$:

$$\begin{aligned} r_1 \quad &\rightsquigarrow \quad r_1^* : \;\; A(x) \to R(x,*), \quad r_1^{**} : \;\; A(x) \to B(*) \\ r_7 \quad &\rightsquigarrow \quad r_7^* : \;\; F(x) \to S(x,*) \end{aligned}$$

The materialisation of $\mathcal{P}^{\chi_{\mathsf{m}}} \cup \mathcal{D}_0^{\mathsf{m}}$ contains facts $A(*)$, $B(*)$, $C(*)$, $D(*)$, $E(*)$ and $R(*,*)$. Consequently, it is now possible to embed the interpretation $\mathcal{I}$ into the aforementioned materialisation by mapping both a and o to $*$.

Figure 5.5 shows all (non-trivial) proofs in $\mathcal{P}^{\chi_{\mathsf{m}}} \cup \mathcal{D}_0^{\mathsf{m}}$ of facts from $\mathcal{D}_r^{\mathsf{m}}$. We can observe that the module $\mathcal{M}^{\chi_{\mathsf{m}}}$ consists of rules r_1–r_5. Clearly, any model of $\mathcal{M}^{\chi_{\mathsf{m}}}$ can be extended to a model of $\mathcal{O}^{ex}$ since no predicates from $\mathcal{M}^{\chi_{\mathsf{m}}}$ occur in the head of any rule in $\mathcal{O}^{ex} \setminus \mathcal{M}^{\chi_{\mathsf{m}}}$. ◇

Theorem 42 shows that the module $\mathcal{M}^{\chi_{\mathsf{m}}}$ is Σ-model inseparable from $\mathcal{O}$. Indeed, any model $\mathcal{I}$ of $\mathcal{M}^{\chi_{\mathsf{m}}}$ can be extended to a model of $\mathcal{O}$ in the following way: *(i)* predicates not occurring in the materialisation of $\mathcal{P}^{\chi_{\mathsf{m}}} \cup \mathcal{D}_0^{\mathsf{m}}$ 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 with arity n are interpreted as $(\Delta^{\mathcal{I}})^n$.

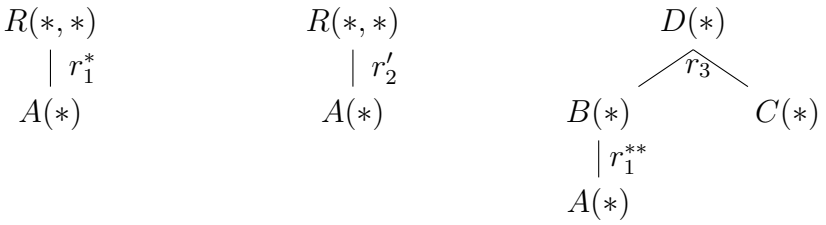


Figure 5.5: All proofs in $\mathcal{P}^{\chi_{\mathsf{m}}} \cup \mathcal{D}_0^{\mathsf{m}}$ of facts from $\mathcal{D}_r^{\mathsf{m}}$

Theorem 42. $\mathcal{M}^{\chi_{\mathsf{m}}} \equiv_{\Sigma}^{\mathsf{m}} \mathcal{O}$.

Proof. Let $\mathcal{I}$ be a model of $\mathcal{M}^{\chi_{\mathsf{m}}}$. W.l.o.g., we assume that $\mathcal{I}$ is defined over all of $\mathsf{Sig}(\mathcal{O})$. Let $\mathcal{J}$ be an interpretation with the same domain Δ as $\mathcal{I}$, and such that

$$A^{\mathcal{J}} = \begin{cases} A^{\mathcal{I}} & \text{if } A \in \Sigma \cup \mathsf{Sig}(\mathsf{supp}(\chi_{\mathsf{m}})) \\ \Delta^{\mathsf{arity}(A)}/\textsc{True} & \text{if } A \in \mathsf{Sig}(\mathcal{P}^{\chi_{\mathsf{m}}}(\mathcal{D}_0^{\mathsf{m}})) \setminus (\Sigma \cup \mathsf{Sig}(\mathsf{supp}(\chi_{\mathsf{m}}))) \\ \emptyset/\textsc{False} & \text{otherwise} \end{cases}$$

Note that $\Sigma \cup \mathsf{Sig}(\mathsf{supp}(\chi_{\mathsf{m}})) \subseteq \mathsf{Sig}(\mathcal{P}^{\chi_{\mathsf{m}}}(\mathcal{D}_0^{\mathsf{m}}))$.

Consider $r : \varphi \to \psi \in \mathcal{O}$. We now show that $\mathcal{J} \models r$.

Assume first $\psi = \Box$. Then $r = \bot \to \Box$ and we need to check that $\bot^{\mathcal{J}} = \textsc{False}$. If $\bot \notin \mathsf{Sig}(\mathcal{P}^{\chi_{\mathsf{m}}}(\mathcal{D}_0^{\mathsf{m}}))$ then it is $\bot^{\mathcal{J}} = \textsc{False}$ by definition of $\mathcal{J}$. If $\bot \in \mathsf{Sig}(\mathcal{P}^{\chi_{\mathsf{m}}}(\mathcal{D}_0^{\mathsf{m}}))$ then, since $\bot \in \mathcal{D}_r^{\mathsf{m}}$, it must be $\bot \in \mathsf{Sig}(\mathsf{supp}(\chi_{\mathsf{m}}))$ and therefore also $\bot \in \mathsf{Sig}(\mathcal{M}^{\chi_{\mathsf{m}}})$. This implies that $\bot \to \Box \in \mathcal{M}^{\chi_{\mathsf{m}}}$ and thus, since $\mathcal{I}$ is a model of $\mathcal{I}$, it must be $\bot^{\mathcal{I}} = \textsc{False}$. Since $\bot \in \mathsf{Sig}(\mathsf{supp}(\chi_{\mathsf{m}}))$, by definition of $\mathcal{J}$ we then have that $\bot^{\mathcal{J}} = \bot^{\mathcal{I}} = \textsc{False}$.

Assume now $\psi \neq \Box$. If $\mathsf{Sig}(\psi) \not\subseteq \mathsf{Sig}(\mathcal{P}^{\chi_{\mathsf{m}}}(\mathcal{D}_0^{\mathsf{m}}))$ then, because $\mathcal{P}^{\chi_{\mathsf{m}}} \cup \mathcal{D}_0^{\mathsf{m}}$ only mentions one constant (namely, $*$), it follows that also $\mathsf{Sig}(\varphi) \not\subseteq \mathsf{Sig}(\mathcal{P}^{\chi_{\mathsf{m}}}(\mathcal{D}_0^{\mathsf{m}}))$ and therefore $\varphi^{\mathcal{J}} = \emptyset$ and $\mathcal{J} \models r$. Hence, in the following we assume $\mathsf{Sig}(\psi) \subseteq \mathsf{Sig}(\mathcal{P}^{\chi_{\mathsf{m}}}(\mathcal{D}_0^{\mathsf{m}}))$.

If $\mathsf{Sig}(\psi) \cap (\Sigma \cup \mathsf{Sig}(\mathsf{supp}(\chi_{\mathsf{m}}))) = \emptyset$, then $A^{\mathcal{J}} = \Delta^{\mathsf{arity}(A)}$ for each $A \in \mathsf{Sig}(\psi)$ and it is immediate that $\mathcal{J} \models r$. Otherwise suppose that there exists a substitution σ over all variables in r such that $\mathcal{J} \models \varphi\sigma$ (if no such substitution exists then $\mathcal{J} \models r$

holds trivially). Then $\varphi^{\mathcal{J}} \neq \emptyset$ and it must be $\mathsf{Sig}(\varphi) \subseteq \mathsf{Sig}(\mathcal{P}^{\chi_{\mathsf{m}}}(\mathcal{D}_0^{\mathsf{m}}))$. Because $*$ is the only constant in $\mathcal{P}^{\chi_{\mathsf{m}}} \cup \mathcal{D}_0^{\mathsf{m}}$, the latter implies $\varphi\sigma_* \subseteq \mathcal{P}^{\chi_{\mathsf{m}}}(\mathcal{D}_0^{\mathsf{m}})$ and thus also $\psi\sigma_* \subseteq \mathcal{P}^{\chi_{\mathsf{m}}}(\mathcal{D}_0^{\mathsf{m}})$, with σ_* a substitution that maps all variables to $*$.

By assumption, there exists $\gamma = A(*, \ldots, *) \in \psi\sigma_*$ with $A \in \Sigma \cup \mathsf{Sig}(\mathsf{supp}(\chi_{\mathsf{m}}))$. Because $\varphi\sigma_* \subseteq \mathcal{P}^{\chi_{\mathsf{m}}}(\mathcal{D}_0^{\mathsf{m}})$, there is a proof $\rho_{\gamma,r}$ of γ in $\mathcal{P}^{\chi_{\mathsf{m}}} \cup \mathcal{D}_0^{\mathsf{m}}$ that is supported by a rule from $\Xi^{\chi_{\mathsf{m}}}(r)$. If $A \in \Sigma$ then, by definition of χ_{m}, it is $\gamma \in \mathcal{D}_r^{\chi_{\mathsf{m}}}$ and consequently $r \in \mathcal{M}^{\chi_{\mathsf{m}}}$. If, on the other hand, $A \notin \Sigma$, there must be $\gamma' \in \mathcal{D}_r^{\mathsf{m}}$ and a proof ρ' of γ' in $\mathcal{P}^{\chi_{\mathsf{m}}} \cup \mathcal{D}_0^{\mathsf{m}}$ such that some proof of $A(*, \ldots, *)$ is a subproof of ρ'. Replacing this subproof with $\rho_{A,r}$ results in another proof of γ' in $\mathcal{P}^{\chi_{\mathsf{m}}} \cup \mathcal{D}_0^{\mathsf{m}}$ that is supported by a rule in $\Xi^{\chi_{\mathsf{m}}}(r)$. Thus $r \in \mathcal{M}^{\chi_{\mathsf{m}}}$ in this case as well. Rules in $\Xi^{\chi_{\mathsf{m}}}(r)$ have the same body as r, so $r \in \mathcal{M}^{\chi_{\mathsf{m}}}$ implies $\mathsf{Sig}(\varphi) \subseteq \mathsf{Sig}(\mathsf{supp}(\chi_{\mathsf{m}}))$, and thus $\mathcal{I}$ and $\mathcal{J}$ agree over $\mathsf{Sig}(\varphi)$. By assumption, $\mathcal{J} \models \varphi\sigma$, so we have $\mathcal{I} \models \varphi\sigma$ as well, and, since $r \in \mathcal{M}^{\chi_{\mathsf{m}}}$, also $\mathcal{I} \models \psi\sigma$. Finally, since $\psi\sigma_* \subseteq \mathcal{P}^{\chi_{\mathsf{m}}}(\mathcal{D}_0^{\mathsf{m}})$, we have that $\mathsf{Sig}(\psi) \subseteq \mathsf{Sig}(\mathcal{P}^{\chi_{\mathsf{m}}}(\mathcal{D}_0^{\mathsf{m}}))$ and therefore $\psi^{\mathcal{I}} \subseteq \psi^{\mathcal{J}}$, so $\mathcal{J} \models \psi\sigma$. Since σ is arbitrary, we can conclude that $\mathcal{J} \models r$. □

The modules generated by χ_{m} are similar in spirit to locality-based modules in that certain symbols outside the signature of the module are interpreted as either the empty set or the universal relation (of the relevant arity) over the interpretation domain. As we will show later on, $\mathcal{M}^{\chi_{\mathsf{m}}} \subseteq \mathcal{M}^{\bot}_{[\mathcal{O},\Sigma]}$ whenever $\mathcal{O}$ is a normalised $\mathcal{SROIQ}$ ontology. However, as illustrated by the following example, our modules are incomparable to $\top$- and $\top\bot^*$-modules.

Example 43. For $\mathcal{O} = \mathcal{O}^{ex}$ and $\Sigma = \{D, F\}$ we have the following:

$$\mathcal{M}^{\chi_{\mathsf{m}}} = \{r_6\} \qquad \mathcal{M}^{\top}_{[\mathcal{O},\Sigma]} = \{r_1\text{–}r_5\} \qquad \mathcal{M}^{\top\bot^*}_{[\mathcal{O},\Sigma]} = \emptyset.$$

Consequently, $\mathcal{M}^{\chi_{\mathsf{m}}}$ is neither contained in $\mathcal{M}^{\top}_{[\mathcal{O},\Sigma]}$, nor in $\mathcal{M}^{\top\bot^*}_{[\mathcal{O},\Sigma]}$. To see that the converse is also true, consider $\mathcal{O} = \{r_1, r_{10}\}$, with $r_{10} = C(x) \rightarrow B(x)$, and

$\Sigma = \{A, C, R\}$. Then, we have the following:

$$\mathcal{M}^{\chi_{\mathsf{m}}} = \{r_1\} \qquad \mathcal{M}^{\top}_{[\mathcal{O},\Sigma]} = \mathcal{M}^{\top\bot^*}_{[\mathcal{O},\Sigma]} = \mathcal{O} \qquad \diamond$$

The modules generated by χ_{m} are also related to reachability-based modules [73] since, due to the use of a single constant in the range of θ^{m} and in $\mathcal{D}_0^{\mathsf{m}}$, the extraction process depends solely on the syntactic dependencies between predicates in $\mathcal{O}$.

We have already mentioned that modules generated by χ_{m} are included in locality $\bot$-modules. To conclude this section, we show how χ_{m} can be modified to precisely capture $\bot$-modules. For this, it suffices to modify χ_{m} by making $\mathcal{D}_r$ the critical dataset over the entire signature of $\mathcal{O}$ (instead of just Σ) as given in the following definition.

Definition 44. The module setting $\chi_{\mathsf{b}} = \langle \theta^{\mathsf{b}}, \mathcal{D}_0^{\mathsf{b}}, \mathcal{D}_r^{\mathsf{b}} \rangle$ is as follows:

- $\theta^{\mathsf{b}} = \theta^{\mathsf{m}}$,
- $\mathcal{D}_0^{\mathsf{b}} = \mathcal{D}^*_{\Sigma}$, and
- $\mathcal{D}_r^{\mathsf{b}} = \mathcal{D}^*_{\mathsf{Sig}(\mathcal{O})} \cup \{\bot\}$. $\diamond$

The following proposition shows that $\mathcal{M}^{\chi_{\mathsf{b}}}$ coincides with the $\bot$-locality module of a $\mathcal{SROIQ}$ ontology $\mathcal{O}$ relative to Σ.

Proposition 45. *If $\mathcal{O}$ is a normalised $\mathcal{SROIQ}$ ontology, then $\mathcal{M}^{\chi_{\mathsf{b}}} = \mathcal{M}^{\bot}_{[\mathcal{O},\Sigma]}$.*

Proof. By definition, $\mathcal{M}^{\bot}_{[\mathcal{O},\Sigma]}$ is the smallest subset $\mathcal{M} \subseteq \mathcal{O}$ such that every axiom in $\mathcal{O} \setminus \mathcal{M}$ is $\bot$-local w.r.t. $\Sigma \cup \mathsf{Sig}(\mathcal{M})$. To show that $\mathcal{M}^{\bot}_{[\mathcal{O},\Sigma]} \subseteq \mathcal{M}^{\chi_{\mathsf{b}}}$, it suffices to show that, for each $r \in \mathcal{O} \setminus \mathcal{M}^{\chi_{\mathsf{b}}}$, r is $\bot$-local w.r.t. $\Sigma \cup \mathsf{Sig}(\mathcal{M}^{\chi_{\mathsf{b}}})$. Consider $r \in \mathcal{O} \setminus \mathcal{M}^{\chi_{\mathsf{b}}}$. Since $\mathcal{D}_r^{\mathsf{b}}$ contains all facts that can occur in $\mathcal{P}^{\chi_{\mathsf{b}}}(\mathcal{D}_0^{\mathsf{b}})$, we have that $\mathsf{supp}(\chi_{\mathsf{b}})$ consists of all rules in the support of any proof in $\mathcal{P}^{\chi_{\mathsf{b}}} \cup \mathcal{D}_0^{\mathsf{b}}$. Furthermore, we have that $\mathsf{Sig}(\mathcal{M}^{\chi_{\mathsf{b}}}) \cup \Sigma = \mathsf{Sig}(\mathcal{P}^{\chi_{\mathsf{b}}}(\mathcal{D}_0^{\mathsf{b}}))$. Therefore, there can be no proof in $\mathcal{P}^{\chi_{\mathsf{b}}} \cup \mathcal{D}_0^{\mathsf{b}}$ that has

a rule from $\Xi^{\chi_\mathsf{b}}(r)$ in its support. Because the only constant mentioned in $\mathcal{P}^{\chi_\mathsf{b}} \cup \mathcal{D}_0^\mathsf{b}$ is $*$, this means that some predicate from the body of r does not occur at all in $\mathsf{Sig}(\mathcal{P}^{\chi_\mathsf{b}}(\mathcal{D}_0^\mathsf{b})) = \mathsf{Sig}(\mathsf{supp}(\chi_\mathsf{b})) \cup \Sigma$. As can be observed in Tables 2.1 and 4.1, this implies that r is $\bot$-local w.r.t. $\mathsf{Sig}(\mathsf{supp}(\chi_\mathsf{b})) \cup \Sigma$.

To see that $\mathcal{M}^{\chi_\mathsf{b}} \subseteq \mathcal{M}^\bot_{[\mathcal{O},\Sigma]}$, consider $r \in \mathcal{M}^{\chi_\mathsf{b}}$. There must exist $r' \in \Xi^{\chi_\mathsf{b}}(r)$ such that $r' \in \mathsf{supp}(\rho)$ for some proof ρ in $\mathcal{P}^{\chi_\mathsf{b}} \cup \mathcal{D}_0^\mathsf{b}$. To show that $r \in \mathcal{M}^\bot_{[\mathcal{O},\Sigma]}$, let us reason by induction on the depth d of ρ.

$d = 0$ Then the body of r is empty and thus r is not $\bot$-local w.r.t. any signature. It follows that $r \in \mathcal{M}^\bot_{[\mathcal{O},\Sigma]}$.

$d > 0$ It suffices to consider the case where r' is the rule applied at the top of ρ, since we already know by induction hypothesis that, for each $s \in \mathcal{O}$ such that a rule from $\Xi^{\chi_\mathsf{b}}(s)$ is in the support of a (proper) subproof of ρ, it is $s \in \mathcal{M}^\bot_{[\mathcal{O},\Sigma]}$. Since $\mathsf{Sig}(\mathcal{D}_0^\mathsf{b}) = \Sigma$, this implies that, if $\rho = (T, \lambda)$ with v the root of T and $w_1, \ldots, w_n$ its children, then $\mathsf{Sig}(\lambda(w_i)) \subseteq \mathsf{Sig}(\mathcal{M}^\bot_{[\mathcal{O},\Sigma]}) \cup \Sigma$. Consequently, the body of r must have its signature fully contained in $\mathsf{Sig}(\mathcal{M}^\bot_{[\mathcal{O},\Sigma]}) \cup \Sigma$. It follows that r is not $\bot$-local w.r.t. $\mathsf{Sig}(\mathcal{M}^\bot_{[\mathcal{O},\Sigma]}) \cup \Sigma$ and hence $r \in \mathcal{M}^\bot_{[\mathcal{O},\Sigma]}$. □

5.4 Additional Inseparability Relations

The bulk of the research on module extraction has focused on the inseparability relations considered in Section 5.3. In this section we show how our framework can be adapted to other interesting inseparability relations.

5.4.1 Classification Inseparability

Classification—the problem of identifying the subsumption hierarchy between all pairs of atomic concepts in a DL ontology—is a fundamental reasoning task in ontology

engineering. As mentioned in Chapter 3, classifying a first-order ontology $\mathcal{O}$ amounts to computing, for each predicate A in $\mathsf{Sig}(\mathcal{O})$, all the entailed implications of the form $A(\mathbf{x}) \rightarrow \bot$ or $A(\mathbf{x}) \rightarrow B(\mathbf{x})$, where B is a predicate of the same arity as A.

In recent years, locality $\bot$-modules have been successfully exploited for optimising classification of DL ontologies [94, 18, 4]. In addition to being model-inseparable from the given ontology and satisfying the properties in Proposition 16 from Section 4.2, $\bot$-modules enjoy an additional property that makes them especially well-suited for optimising classification [19, 18]:

Proposition 46. *Let $\mathcal{O}$ be an ontology, Σ a signature, and r a rule of the form $A(\mathbf{x}) \rightarrow \psi$ where $A \in \Sigma$ and either $\psi = \bot$ or $\psi = B(\mathbf{x})$ with $B \in \mathsf{Sig}(\mathcal{O})$. Then, $\mathcal{O} \models r$ iff $\mathcal{M}^{\bot}_{[\mathcal{O},\Sigma]} \models r$.*

It follows from Proposition 46 that the $\bot$-module for $\Sigma = \{A\}$ in $\mathcal{O}$ captures all subsumers of A in $\mathcal{O}$, and hence it is indistinguishable from $\mathcal{O}$ w.r.t. to all implications having A in the body. We can capture this additional property of $\bot$-modules by means of the following inseparability relation.

Definition 47. Ontologies $\mathcal{O}$ and $\mathcal{O}'$ are Σ-*classification inseparable* ($\mathcal{O} \equiv^{\mathsf{c}}_{\Sigma} \mathcal{O}'$) if for each rule r of the form $A(\mathbf{x}) \rightarrow \psi$, where $A \in \Sigma$ and either $\psi = \bot$ or $\psi = B(\mathbf{x})$ with $B \in \mathsf{Sig}(\mathcal{O} \cup \mathcal{O}')$, we have $\mathcal{O} \models r$ iff $\mathcal{O}' \models r$. Furthermore, $\mathsf{rel}_{\mathsf{c}}$ is the function mapping each ontology $\mathcal{O}$ and signature Σ to the following set of rules:

$$\mathsf{rel}_{\mathsf{c}}(\mathcal{O}, \Sigma) = \{r = A(\mathbf{x}) \rightarrow \psi \mid \mathcal{O} \models r,\ A \in \Sigma \text{ and } \psi = \bot \text{ or } \psi = B(\mathbf{x}) \text{ with } B \in \mathsf{Sig}(\mathcal{O})\} \quad \diamond$$

It follows straightforwardly from the definition that $\mathcal{O} \equiv^{\mathsf{c}}_{\Sigma} \mathcal{O}'$ holds iff $\mathsf{rel}_{\mathsf{c}}(\mathcal{O}, \Sigma)$ coincides with $\mathsf{rel}_{\mathsf{c}}(\mathcal{O}', \Sigma)$; hence, Theorem 10 in Chapter 4 trivially extends to classification inseparability. Furthermore, it can be readily checked that $\equiv^{\mathsf{c}}_{\Sigma} \subsetneq \equiv^{\mathsf{i}}_{\Sigma}$ for each non-trivial signature Σ, and hence classification inseparability is (as expected) a stronger requirement than implication inseparability. Finally, although $\bot$-modules

ensure classification inseparability from $\mathcal{O}$, they are also model-inseparable (a much stricter requirement) and hence they are typically much larger than necessary for ontology classification.

Example 48. Consider our example ontology $\mathcal{O}^{ex}$. We can observe how classification inseparability is a stronger requirement than implication inseparability by considering $\Sigma = \{G\}$. Clearly, $\mathcal{M} = \emptyset$ is Σ-implication inseparable from $\mathcal{O}^{ex}$, whereas Σ-classification inseparability requires rule r_9 to be contained in $\mathcal{M}$. To see how $\bot$-modules differ from minimal classification-inseparable modules consider $\Sigma = \{A\}$; in this case, we have that $\mathcal{M}^{\bot}_{[\mathcal{O}^{ex},\Sigma]} = \{r_1, r_2\}$, but A has no subsumers in $\mathcal{O}^{ex}$ and therefore the empty ontology is already Σ-classification inseparable from $\mathcal{O}^{ex}$. $\diamond$

The following module setting extends χ_{i} from Definition 26 to capture classification inseparability. As one would naturally expect, the only required modification is to extend $\mathcal{D}^{\mathsf{i}}_r$ with facts involving predicates outside Σ.

Definition 49. For each $A \in \Sigma$ of arity n, let $\mathbf{c}_A = (c^1_A, \ldots, c^n_A)$ be an array of fresh constants. The module setting $\chi_{\mathsf{c}} = (\theta^{\mathsf{c}}, \mathcal{D}^{\mathsf{c}}_0, \mathcal{D}^{\mathsf{c}}_r)$ is defined as follows:

- $\theta^{\mathsf{c}} = \theta^{\mathsf{i}}$,
- $\mathcal{D}^{\mathsf{c}}_0 = \mathcal{D}^{\mathsf{i}}_0$, and
- $\mathcal{D}^{\mathsf{c}}_r = \{\, B(\mathbf{c}_A) \mid A \in \Sigma \text{ and } B \in \mathsf{Sig}(\mathcal{O}) \text{ distinct predicates of the same arity} \,\} \cup \{\bot\}$.

$\diamond$

Example 50. Consider again $\mathcal{O} = \mathcal{O}^{ex}$ and $\Sigma = \{A\}$. Since no fact from $\mathcal{D}^{\mathsf{c}}_r$ is provable in $\mathcal{P}^{\chi_{\mathsf{c}}} \cup \mathcal{D}^{\mathsf{c}}_0$ the module $\mathcal{M}^{\chi_{\mathsf{c}}}$ is empty and thus also minimal in this case. $\diamond$

We can show that χ_{c} captures implication inseparability using an argument analogous to that in the proof of Theorem 27.

Theorem 51. $\mathcal{M}^{\chi_{\mathsf{c}}} \equiv^{\mathsf{c}}_{\Sigma} \mathcal{O}$.

5.4.2 Weak Query Inseparability

One of the possible applications of modules based on query inseparability is to optimise query answering. In particular, if $\mathcal{M}$ is Σ-query inseparable from $\mathcal{O}$, then $\mathcal{O} \cup \mathcal{D} \models q$ iff $\mathcal{M} \cup \mathcal{D} \models q$ for any Σ-query q and Σ-dataset $\mathcal{D}$; thus, we can replace $\mathcal{O}$ with $\mathcal{M}$ to answer an arbitrary query w.r.t. arbitrary data provided that only symbols in Σ are deemed relevant.

While the aforementioned notion of query inseparability is useful for situations where the data is unknown or frequently changing, in many situations the data in an ontology can be considered fixed and hence one could potentially extract smaller modules by not requiring $\mathcal{M}$ to be robust under extensions with arbitrary data.

Botoeva et al. [13] investigated a restricted notion of query inseparability that is well-suited for cases where the data in an ontology can be considered fixed. In this thesis, we will refer to this restricted notion as weak query inseparability.

Definition 52. Ontologies $\mathcal{O}$ and $\mathcal{O}'$ are Σ-*weak query inseparable* ($\mathcal{O} \equiv_{\Sigma}^{\mathsf{wq}} \mathcal{O}'$) if for each Boolean PEQ q over Σ we have $\mathcal{O} \models q$ iff $\mathcal{O}' \models q$. Furthermore, $\mathsf{rel}_{\mathsf{wq}}$ is the function mapping each ontology $\mathcal{O}$ and signature Σ to the set

$$\mathsf{rel}_{\mathsf{wq}}(\mathcal{O}, \Sigma) = \{\, q \mid \mathcal{O} \models q \text{ and } q \text{ is a Boolean PEQ such that } \mathsf{Sig}(q) \subseteq \Sigma \,\} \qquad \diamond$$

Again, Theorem 10 extends naturally to weak query inseparability; indeed, it is the case that $\mathcal{O} \equiv_{\Sigma}^{\mathsf{wq}} \mathcal{O}'$ if and only if $\mathsf{rel}_{\mathsf{wq}}(\mathcal{O}, \Sigma)$ coincides with $\mathsf{rel}_{\mathsf{wq}}(\mathcal{O}', \Sigma)$. Moreover, the requirements in Definition 52 are weaker than those of query inseparability and hence $\equiv_{\Sigma}^{\mathsf{q}} \subsetneq \equiv_{\Sigma}^{\mathsf{wq}}$ for each Σ. As a result, weak query inseparability typically yields smaller modules.

We next propose a module setting χ_{wq} that captures weak query inseparability. The main difference between χ_{wq} and χ_{q} in Section 5.3.3 is that the initial dataset $\mathcal{D}_0^{\mathsf{wq}}$ is chosen as empty rather than $\mathcal{D}_{\Sigma}^{*}$. This is a natural choice given that we no

longer have the requirement that arbitrary Σ-datasets must be embeddable into $\mathcal{D}_0^{\mathsf{wq}}$. Furthermore, θ^{wq} differs from θ^{q} in that it maps constants from $\mathsf{Ct}(\mathcal{O})$ to themselves (same as θ^{i}).

Definition 53. Let constants c_y, for each existentially quantified variable y in $\mathcal{O}$, be as in Definition 26. The module setting $\chi_{\mathsf{wq}} = \langle \theta^{\mathsf{wq}}, \mathcal{D}_0^{\mathsf{wq}}, \mathcal{D}_r^{\mathsf{wq}} \rangle$ is defined as follows:

- $\theta^{\mathsf{wq}} = \theta^{\mathsf{i}}$,
- $\mathcal{D}_0^{\mathsf{wq}} = \emptyset$, and
- $\mathcal{D}_r^{\mathsf{wq}} = \{\, A(a_1, \ldots, a_n) \mid A \in \Sigma,$ each a_j either in $\mathsf{Ct}(\mathcal{O})$ or equals some $c_y \,\} \cup \{\bot\}$.

$\diamond$

Example 54. Consider the extension of $\mathcal{O}^{ex}$ with the fact ($\rightarrow D(i)$) (seen as a ground rule) and the signature $\Sigma = \{B, C, D, H\}$. It can be readily checked that $\mathcal{M}^{\chi_{\mathsf{q}}} = \{r_3, r_6\text{–}r_9\}$, whereas $\mathcal{M}^{\chi_{\mathsf{wq}}} = \{r_6\text{–}r_9\}$. $\diamond$

Theorem 55. $\mathcal{M}^{\chi_{\mathsf{wq}}} \equiv_{\Sigma}^{\mathsf{wq}} \mathcal{O}$.

Proof. Any Boolean PEQ q can be seen as the rule ($\rightarrow q$). Consequently, $\mathcal{M}^{\chi_{\mathsf{wq}}} \equiv_{\Sigma}^{\mathsf{wq}} \mathcal{O}$ follows from Theorem 25 by a similar argument to that in the proof of Theorem 37. $\square$

5.5 Module Containment

Intuitively, the more expressive the language for which preservation of consequences is required, the larger the modules need to be. For instance, since $\equiv_{\Sigma}^{\mathsf{f}} \subsetneq \equiv_{\Sigma}^{\mathsf{i}}$, it is to be expected that the module $\mathcal{M}^{\chi_{\mathsf{i}}}$ obtained for implication inseparability is contained in the module $\mathcal{M}^{\chi_{\mathsf{f}}}$ for fact inseparability. We next show that all our modules in Sections 5.3 and 5.4 are consistent with this intuition.

Our first step will be to introduce a notion of homomorphism between module settings, which will then allow us to establish a containment relation between the corresponding modules.

Definition 56. For a module setting $\chi = \langle \theta, \mathcal{D}_0, \mathcal{D}_r \rangle$, let $\mathsf{Ct}(\chi)$ denote the set of constants occurring in $\mathcal{D}_0$, $\mathcal{D}_r$, and in the range of θ. A substitution $\mu : \mathsf{Ct}(\chi) \to \mathsf{Ct}(\chi')$ is a *homomorphism from χ to χ'* if the following conditions hold:

- $\theta\mu = \theta'$;
- $\mathcal{D}_0\mu \subseteq \mathcal{D}'_0$; and
- $\mathcal{D}_r\mu \subseteq \mathcal{D}'_r$.

We write $\chi \hookrightarrow \chi'$ to denote that a homomorphism from χ to χ' exists. $\diamond$

The fact that $\chi \hookrightarrow \chi'$ (witnessed by some homomorphism μ) implies that χ' is "more general" than χ, in the sense that any proof in $\mathcal{P}^{\chi} \cup \mathcal{D}_0$ of a fact in $\mathcal{D}_r$ can be embedded (via μ) in a support-preserving way into a proof in $\mathcal{P}^{\chi'} \cup \mathcal{D}'_0$ of a fact in $\mathcal{D}'_r$. It follows that $\mathsf{supp}(\chi)$ is contained in $\mathsf{supp}(\chi')$ (modulo μ) and hence we also have $\mathcal{M}^{\chi} \subseteq \mathcal{M}^{\chi'}$.

Theorem 57. *If χ, χ' are s.t. $\chi \hookrightarrow \chi'$, then $\mathcal{M}^{\chi} \subseteq \mathcal{M}^{\chi'}$.*

Proof. Let $\chi = \langle \theta, \mathcal{D}_0, \mathcal{D}_r \rangle$ and $\chi' = \langle \theta', \mathcal{D}'_0, \mathcal{D}'_r \rangle$, and let $\mu : \mathsf{Ct}(\chi) \to \mathsf{Ct}(\chi')$ be a homomorphism from χ to χ'. Since $\mathcal{D}_r\mu \subseteq \mathcal{D}'_r$, it suffices to show that for any rule $r \in \mathcal{O}$ and any proof ρ in $\mathcal{P}^{\chi} \cup \mathcal{D}_0$ of a fact $\gamma \in \mathcal{D}_r$ such that $\mathsf{supp}(\rho) \cap \Xi^{\chi}(r) \neq \emptyset$, there exists a proof ρ' in $\mathcal{P}^{\chi'} \cup \mathcal{D}'_0$ of $\gamma\mu$ such that $\mathsf{supp}(\rho') \cap \Xi^{\chi'}(r) \neq \emptyset$. We prove the following more general claim.

Let r be a rule in $\mathcal{O}$ and ρ a proof of a fact γ (not necessarily in $\mathcal{D}_r$) in $\mathcal{P}^{\chi} \cup \mathcal{D}_0$ such that $\mathsf{supp}(\rho) \cap \Xi^{\chi}(r) \neq \emptyset$. We show that there is a proof ρ' of $\gamma\mu$ in $\mathcal{P}^{\chi'} \cup \mathcal{D}'_0$ such that $\mathsf{supp}(\rho') \cap \Xi^{\chi'}(r) \neq \emptyset$ by induction on the depth d of ρ.

$d = 0$

Since, by assumption, $\mathsf{supp}(\rho) \cap \Xi^{\chi}(r) \neq \emptyset$, we have that r is of the form $r = (\ \rightarrow \psi)$ and there exists $(\ \rightarrow \gamma) \in \Xi^{\chi}(r)$ with $\gamma = \delta\theta$ for some $\delta \in \psi$, and also $(\ \rightarrow \delta\theta') \in \Xi^{\chi'}(r)$. Since θ is defined over all constants in $\mathcal{O}$, it holds that $(\delta\theta)\mu = \delta(\theta\mu) = \delta\theta'$, and hence we have a proof of $\gamma\mu$ in $\mathcal{P}^{\chi'} \cup \mathcal{D}'_0$ supported by a rule in $\Xi^{\chi'}(r)$.

$d > 0$

Let $\rho = (T, \lambda)$ with v the root of T and $w_1, \ldots, w_n$ the children of v. Let s be the rule used to derive $\lambda(v)$ from $\lambda(w_1), \ldots, \lambda(w_n)$. Finally, for each i let ρ_i be the subproof of $\lambda(w_i)$. If for any i we have $\mathsf{supp}(\rho_i) \cap \Xi^{\chi}(r) \neq \emptyset$, the claim follows by the induction hypothesis since $s\mu \in \mathcal{P}^{\chi'}$. Otherwise, we have $s \in \Xi^{\chi}(r)$. Moreover, by the induction hypothesis, every $\lambda(w_i)\mu$ has a proof in $\mathcal{P}^{\chi'} \cup \mathcal{D}'_0$, and the claim follows since $s\mu \in \Xi^{\chi'}(r)$. □

It is straightforward to construct homomorphisms between the module settings in Sections 5.3 and 5.4 in accordance with the containment relationship of their corresponding inseparability relations. The following result, which establishes the intuitive relationships between our modules, then follows immediately from Theorem 57.

Corollary 58.

$$\mathcal{M}^{\chi_{\mathsf{i}}} \subseteq \mathcal{M}^{\chi_{\mathsf{f}}} \subseteq \mathcal{M}^{\chi_{\mathsf{q}}} \subseteq \mathcal{M}^{\chi_{\mathsf{m}}} \subseteq \mathcal{M}^{\chi_{\mathsf{b}}}$$

$$\mathcal{M}^{\chi_{\mathsf{i}}} \subseteq \mathcal{M}^{\chi_{\mathsf{c}}} \subseteq \mathcal{M}^{\chi_{\mathsf{b}}}$$

$$\mathcal{M}^{\chi_{\mathsf{wq}}} \subseteq \mathcal{M}^{\chi_{\mathsf{q}}}$$

Furthermore, as already illustrated by examples throughout Sections 5.3 and 5.4, these containment relations are strict for many $\mathcal{O}$ and Σ.

5.6 Depletingness, Self-Containment, and Justification-Preservation

The minimal requirement on a module is to preserve all relevant consequences w.r.t. a given inseparability relation. As we argued in Chapter 4, however, in some applications it is desirable that modules satisfy additional properties. In this section, we establish whether our modules as defined in Sections 5.3 and 5.4 satisfy the (strong) depletingness, self-containment, and justification-preservation properties enjoyed by locality-based modules.

To establish our results, it is convenient to abstract away from the notion of module setting for a fixed $\mathcal{O}$ and Σ and consider instead *families* of module settings; that is, functions that assign a module setting to each pair of $\mathcal{O}$ and Σ.

Definition 59. A *module setting family* is a function Ψ that maps each pair of ontology $\mathcal{O}$ and signature Σ to a module setting for $\mathcal{O}$ and Σ. Given an inseparability relation $\mathcal{S}$, we say that Ψ is $\mathcal{S}$-*admissible* if, for each pair of $\mathcal{O}$ and Σ, $\mathcal{M}^{\Psi(\mathcal{O},\Sigma)}$ is a $\equiv^{\mathcal{S}}_{\Sigma}$-module of $\mathcal{O}$. Furthermore, we say that Ψ is depleting (resp. strongly depleting, self-contained, justification-preserving) if so is $\mathcal{M}^{\Psi(\mathcal{O},\Sigma)}$ for each $\mathcal{O}$ and Σ.

Finally, for each $\mathcal{S} \in \{\mathsf{m}, \mathsf{q}, \mathsf{f}, \mathsf{i}, \mathsf{c}, \mathsf{wq}\}$, we denote with $\Psi^{\mathcal{S}}$ the ($\mathcal{S}$-admissible) family induced by the module setting $\chi^{\mathcal{S}}$ as defined in Sections 5.3 and 5.4. $\diamond$

5.6.1 Depletingness and Justification-Preservation

As discussed in Chapter 4, depletingness of $\mathcal{M}$ ensures that $\mathcal{O} \setminus \mathcal{M}$ is inseparable from the empty ontology and hence no relevant information is left behind in $\mathcal{O}$ after extracting $\mathcal{M}$. As illustrated by the following example, not all modules are depleting.

Example 60. Consider $\mathcal{O}$ consisting of the following rules and let $\Sigma = \{A, B\}$:

$$s_1 = A(x) \rightarrow B(x) \wedge C(x) \qquad s_2 = A(x) \rightarrow D(x) \wedge E(x) \qquad s_3 = D(x) \rightarrow B(x)$$

Clearly, both $\mathcal{M}_1 = \{s_1\}$ and $\mathcal{M}_2 = \{s_2, s_3\}$ are implication-inseparable from $\mathcal{O}$ and hence $\equiv^{\mathsf{i}}_{\Sigma}$-modules. However, neither of them is depleting. $\diamond$

We next show that all the modules we defined in Sections 5.3 and 5.4 are depleting.

Proposition 61. *$\Psi^{\mathcal{S}}$ is depleting for each $\mathcal{S} \in \{\mathsf{m}, \mathsf{q}, \mathsf{f}, \mathsf{i}, \mathsf{c}, \mathsf{wq}\}$.*

Proof. Let $\mathcal{O}$ and Σ be arbitrary and let $\mathcal{M} = \mathcal{M}^{\Psi^{\mathcal{S}}(\mathcal{O},\Sigma)}$. By Theorems 27, 33, 37, 42, 51 and 55, it suffices to show $\mathcal{M}^{\Psi^{\mathcal{S}}(\mathcal{O}\setminus\mathcal{M},\Sigma)} = \emptyset$. By the definition of a module (cf. Def. 20), it holds that $\mathcal{M}^{\Psi^{\mathcal{S}}(\mathcal{O}\setminus\mathcal{M},\Sigma)} \subseteq \mathcal{O}\setminus\mathcal{M}$. Furthermore, since $\mathcal{O}\setminus\mathcal{M} \subseteq \mathcal{O}$, it follows that $\mathcal{M}^{\Psi^{\mathcal{S}}(\mathcal{O}\setminus\mathcal{M},\Sigma)} \subseteq \mathcal{M}$. Consequently, $\mathcal{M}^{\Psi^{\mathcal{S}}(\mathcal{O}\setminus\mathcal{M},\Sigma)} = \emptyset$. $\square$

We next show that our modules are also justification-preserving and hence can be seamlessly exploited in ontology debugging applications [44, 90]. Furthermore, since debugging applications typically require only implication inseparability, we can exploit the fine grained modules in Definition 26 rather than the much coarser grained $\bot$-locality modules underpinning current implementations.

Proposition 62. *$\Psi^{\mathcal{S}}$ is justification-preserving for each $\mathcal{S} \in \{\mathsf{m}, \mathsf{q}, \mathsf{f}, \mathsf{i}, \mathsf{c}, \mathsf{wq}\}$.*

Proof. Let $\phi \in \mathsf{rel}_{\mathcal{S}}(\mathcal{O}, \Sigma)$ and let $\mathcal{O}'$ be a justification of ϕ in $\mathcal{O}$. We need to check that $\mathcal{O}' \subseteq \mathcal{M}^{\Psi^{\mathcal{S}}(\mathcal{O},\Sigma)}$. Since $\phi \in \mathsf{rel}_{\mathcal{S}}(\mathcal{O}, \Sigma)$ and $\mathcal{O}' \models \phi$, it follows by definition of $\mathsf{rel}_{\mathcal{S}}$ that $\phi \in \mathsf{rel}_{\mathcal{S}}(\mathcal{O}', \Sigma)$. By Theorems 27, 33, 37, 42, 51 and 55, we have $\mathcal{M}^{\Psi^{\mathcal{S}}(\mathcal{O}',\Sigma)} \equiv^{\mathcal{S}}_{\Sigma} \mathcal{O}'$, and therefore $\mathcal{M}^{\Psi^{\mathcal{S}}(\mathcal{O}',\Sigma)} \models \phi$. By the definition of a module (cf. Def. 20) it is immediate that $\mathcal{O}' \subseteq \mathcal{O}$ implies $\mathcal{M}^{\Psi^{\mathcal{S}}(\mathcal{O}',\Sigma)} \subseteq \mathcal{M}^{\Psi^{\mathcal{S}}(\mathcal{O},\Sigma)}$. Finally, by minimality of $\mathcal{O}'$, we have $\mathcal{M}^{\Psi^{\mathcal{S}}(\mathcal{O}',\Sigma)} = \mathcal{O}'$ and therefore $\mathcal{O}' \subseteq \mathcal{M}^{\Psi^{\mathcal{S}}(\mathcal{O},\Sigma)}$. $\square$

We conclude by addressing the effect of normalisation on these properties. Similarly to our treatment of normalisation in Proposition 14 from Chapter 4, we show

that we can recover a depleting and justification-preserving module for a $\mathcal{SROIQ}$ ontology $\mathcal{O}$ from one such module for its normalisation $\mathsf{norm}(\mathcal{O})$.

Proposition 63. *Let $\mathcal{S}$ be an inseparability relation, and let Ψ be a module setting family that is $\mathcal{S}$-admissible and depleting. Let norm be a normalisation function. Let $\mathcal{O}$ be a $\mathcal{SROIQ}$ ontology and let $\mathcal{M} \subseteq \mathcal{O}$ be such that the following holds:*

1. $\mathcal{M}^{\Psi(\mathsf{norm}(\mathcal{O}),\Sigma)} \subseteq \mathsf{norm}(\mathcal{M})$ *and*
2. $\mathsf{norm}(\mathcal{O}\backslash\mathcal{M}) \subseteq \mathsf{norm}(\mathcal{O})\backslash\mathsf{norm}(\mathcal{M})$.

Then, $\mathcal{M}$ is a depleting and justification-preserving $\equiv^{\mathcal{S}}_{\Sigma}$-module of $\mathcal{O}$.

Proof. Let $\mathcal{O}' = \mathsf{norm}(\mathcal{O})$. By $\mathcal{S}$-admissibility of Ψ we have that $\mathcal{M}^{\Psi(\mathcal{O}',\Sigma)}$ is a $\equiv^{\mathcal{S}}_{\Sigma}$-module of $\mathcal{O}'$. Since $\mathcal{M}^{\Psi(\mathcal{O}',\Sigma)} \subseteq \mathsf{norm}(\mathcal{M})$, by monotonicity of first-order logic $\mathsf{norm}(\mathcal{M})$ is also a $\equiv^{\mathcal{S}}_{\Sigma}$-module of $\mathcal{O}'$. By Proposition 14 it follows that $\mathcal{M}$ is a $\equiv^{\mathcal{S}}_{\Sigma}$-module of $\mathcal{O}$.

We next show that $\mathcal{M}$ is depleting. Since $\mathcal{M}^{\Psi(\mathcal{O}',\Sigma)} \subseteq \mathsf{norm}(\mathcal{M})$, we have that $\mathcal{O}'\backslash\mathsf{norm}(\mathcal{M}) \subseteq \mathcal{O}'\backslash\mathcal{M}^{\Psi(\mathcal{O}',\Sigma)}$. Proposition 61 implies that $\mathcal{O}'\backslash\mathcal{M}^{\Psi(\mathcal{O}',\Sigma)} \equiv^{\mathcal{S}}_{\Sigma} \emptyset$, and hence by monotonicity of first-order logic also $\mathcal{O}'\backslash\mathsf{norm}(\mathcal{M}) = \mathsf{norm}(\mathcal{O}\backslash\mathcal{M}) \equiv^{\mathcal{S}}_{\Sigma} \emptyset$. Since $\mathsf{norm}(\mathcal{O}\backslash\mathcal{M})$ is a conservative extension of $\mathcal{O}\backslash\mathcal{M}$, by Definition 5 we have $\mathsf{norm}(\mathcal{O}\backslash\mathcal{M}) \equiv^{\mathcal{S}}_{\Sigma} \mathcal{O}\backslash\mathcal{M}$, and thus $\mathcal{O}\backslash\mathcal{M} \equiv^{\mathcal{S}}_{\Sigma} \emptyset$.

To show that $\mathcal{M}$ is justification-preserving, let $\varphi \in \mathsf{rel}_{\mathcal{S}}(\mathcal{O},\Sigma)$ and consider a justification $\tilde{\mathcal{O}} \subseteq \mathcal{O}$ of φ in $\mathcal{O}$. Suppose there is some $\alpha \in \tilde{\mathcal{O}}\backslash\mathcal{M}$. Then $\alpha \in \mathcal{O}\backslash\mathcal{M}$ and $\mathsf{norm}(\alpha) \subseteq \mathsf{norm}(\mathcal{O}\backslash\mathcal{M}) = \mathcal{O}'\backslash\mathsf{norm}(\mathcal{M})$. Since $\mathcal{M}^{\Psi(\mathcal{O}',\Sigma)} \subseteq \mathsf{norm}(\mathcal{M})$, this implies $\mathsf{norm}(\alpha) \cap \mathcal{M}^{\Psi(\mathcal{O}',\Sigma)} = \emptyset$. On the other hand, since $\mathcal{O}'$ is a conservative extension of $\mathcal{O}$, we have $\varphi \in \mathsf{rel}_{\mathcal{S}}(\mathcal{O}',\Sigma)$. Also, because $\mathsf{norm}(\tilde{\mathcal{O}})$ is a conservative extension of $\tilde{\mathcal{O}}$, we have $\mathsf{norm}(\tilde{\mathcal{O}}) \models \varphi$ and there must be a justification $\tilde{\mathcal{O}}'$ of φ in $\mathsf{norm}(\tilde{\mathcal{O}}) \subseteq \mathcal{O}'$. By Proposition 62, $\mathcal{M}^{\Psi(\mathcal{O}',\Sigma)}$ is justification-preserving, and consequently $\tilde{\mathcal{O}}' \subseteq \mathcal{M}^{\Psi(\mathcal{O}',\Sigma)}$. Furthermore, by minimality of $\tilde{\mathcal{O}}$ there is no proper subset of $\tilde{\mathcal{O}}$ whose normalisation includes $\tilde{\mathcal{O}}'$. It follows that $\mathsf{norm}(\alpha) \cap \tilde{\mathcal{O}}' \neq \emptyset$ and hence $\mathsf{norm}(\alpha) \cap \mathcal{M}^{\Psi(\mathcal{O}',\Sigma)} \neq \emptyset$. This

is a contradiction that stems from assuming that $\alpha \in \tilde{\mathcal{O}} \setminus \mathcal{M}$. Therefore $\tilde{\mathcal{O}} \subseteq \mathcal{M}$, i.e., $\mathcal{M}$ is justification-preserving. □

5.6.2 Self-Containment and Strong Depletingness

In contrast to locality-based modules, the modules obtained using our approach are neither strongly depleting, nor self-contained. To see this, consider the following example.

Example 64. Let $\Sigma = \{A, D\}$ and $\mathcal{O} = \{r_{11}–r_{16}\}$ with

$$\begin{aligned} r_{11} &= (\rightarrow A(o)) \\ r_{12} &= A(x) \rightarrow \exists y.[R(x,y) \wedge B(y)] \\ r_{13} &= A(x) \rightarrow \exists y.[R(x,y) \wedge C(y)] \\ r_{14} &= R(x,y) \rightarrow D(x) \\ r_{15} &= B(x) \rightarrow C(x) \\ r_{16} &= (\rightarrow C(i)) \end{aligned}$$

Let $\mathcal{M}_1 = \{r_{11}–r_{14}\}$ and $\mathcal{M}_2 = \{r_{12}–r_{14}\}$. We can check that

$$\mathcal{M}^{\chi_{\mathsf{f}}} = \mathcal{M}^{\chi_{\mathsf{q}}} = \mathcal{M}^{\chi_{\mathsf{m}}} = \mathcal{M}^{\chi_{\mathsf{wq}}} = \mathcal{M}_1$$
$$\mathcal{M}^{\chi_{\mathsf{i}}} = \mathcal{M}^{\chi_{\mathsf{c}}} = \mathcal{M}_2$$

Clearly, $\mathcal{O} \models C(i)$ but $\mathcal{M}^{\chi_{\mathsf{wq}}} \not\models C(i)$; since C is in the signature of $\mathcal{M}^{\chi_{\mathsf{wq}}}$, we have that $\mathcal{M}^{\chi_{\mathsf{wq}}}$ is not self-contained. Furthermore, $\mathcal{M}^{\chi_{\mathsf{wq}}}$ is not strongly depleting since $\mathcal{O} \backslash \mathcal{M}^{\chi_{\mathsf{wq}}} \models C(i)$. For the remaining inseparability relations, observe that $\mathcal{O} \models B(x) \rightarrow C(x)$ and $\mathcal{M}_i \not\models B(x) \rightarrow C(x)$ for $1 \leq i \leq 2$. Since both B and C are in the signatures of $\mathcal{M}_1$ and $\mathcal{M}_2$ it follows that none of our modules is self-contained (note that implications are relevant consequences for all relations other than wq). Furthermore, since we also have that $\mathcal{O} \setminus \mathcal{M}_i \models B(x) \rightarrow C(x)$ we can conclude that

our modules are also not strongly depleting. ◇

Self-containment and strong-depletingness are strong requirements that are not always needed for applications. Hence, the fact that our modules do not satisfy them by default can be beneficial as it may allow us to compute smaller modules.

However, as mentioned in Chapter 4, these properties can be useful in certain ontology reuse scenarios. We next show that our framework can be adapted so as to satisfy these properties whenever they are required. This can be achieved via a fix-point construction where modules are computed w.r.t. iterative extensions of the initial signature. Such fix-point constructions are reminiscent of the standard algorithms for computing locality modules [19, 21].

Definition 65. Let $\Psi^{\mathcal{S}}$ be a module setting family for an inseparability relation $\mathcal{S}$. We define the family $\Psi^{\mathcal{S}}_{self}$ as the function mapping each $\mathcal{O}$ and Σ to the least fix-point of the sequence $\{\mathcal{M}_i\}_{i\geq 0}$ as defined next:

$$\Sigma_0 = \Sigma \qquad \mathcal{M}_i = \mathcal{M}^{\Psi^{\mathcal{S}}(\mathcal{O},\Sigma_i)} \text{ for } i \geq 0$$
$$\Sigma_i = \Sigma_{i-1} \cup \mathsf{Sig}(\mathcal{M}_{i-1}) \text{ for } i > 0$$

◇

The aforementioned fix-point is well-defined: since $\Sigma_i \subseteq \Sigma \cup \mathsf{Sig}(\mathcal{O})$ for each $i \geq 0$ and $\Sigma \cup \mathsf{Sig}(\mathcal{O})$ is finite, there must be some $i_0 \geq 0$ such that $\Sigma_{i_0} = \Sigma_j$ and $\mathcal{M}_{i_0} = \mathcal{M}_j$ for each $j > i_0$. We show that, with this adaptation, our modules satisfy the required properties.

Proposition 66. *The family $\Psi^{\mathcal{S}}_{self}$ is self-contained and strongly depleting for each $\mathcal{S} \in \{\mathsf{m}, \mathsf{q}, \mathsf{f}, \mathsf{i}, \mathsf{c}, \mathsf{wq}\}$.*

Proof. Let $\mathcal{M} = \Psi^{\mathcal{S}}_{self}(\mathcal{O}, \Sigma)$. It is immediate that $\mathcal{M}$ is a self-contained $\equiv^{\mathcal{S}}_{\Sigma}$-module of $\mathcal{O}$. Strong depletingness of $\mathcal{M}$ follows from Proposition 61 for $\mathcal{S} \in \{\mathsf{m}, \mathsf{q}, \mathsf{f}, \mathsf{i}, \mathsf{c}, \mathsf{wq}\}$. □

The construction in Definition 65 can straightforwardly be adapted to the case of non-normalised $\mathcal{SROIQ}$ ontologies by following the same approach as in Proposition 63.

5.7 Complexity of Module Extraction

In this section, we argue that our modules can be efficiently computed in many practically relevant cases. For this, we analyse the complexity of the following decision problem.

Definition 67. Let L be a class of ontologies and let $\mathcal{S} \in \{\mathsf{m}, \mathsf{q}, \mathsf{f}, \mathsf{i}, \mathsf{c}, \mathsf{b}, \mathsf{wq}\}$ be an inseparability relation. The decision problem $\mathsf{isInModule}_{[L,\mathcal{S}]}$ is as follows:

- *Input*: an ontology $\mathcal{O} \in L$, a signature Σ and a rule $r \in \mathcal{O}$.
- *Output*: TRUE if and only if $r \in \mathcal{M}^{\Psi^{\mathcal{S}}(\mathcal{O},\Sigma)}$. ◇

Furthermore, we consider the following classes of ontologies, which are strongly connected to DL-based ontology languages.

Definition 68. Let k be a fixed non-negative integer. The class L^k_{arity} consists of all ontologies where predicates have arity at most k.

The *graph* of a conjunction of atoms φ is the undirected graph $G_\varphi = (V, E)$ such that V is the set of variables occurring in φ, and E contains an edge between each pair of variables that occur together in some atom in φ. A *tree decomposition* of G_φ is a tree $T = (W, F)$, such that there exists a labelling λ mapping each vertex $w \in W$ to some subset $\lambda(w) \subseteq V$, and the following conditions are satisfied:

- for each $v \in V$, there exists $v \in W$ with $v \in \lambda(w)$,
- for each $\{v, v'\} \in E$, there exists $w \in W$ with $\{v, v'\} \subseteq \lambda(w)$, and
- for each $v \in V$, the set $\{\, w \in W \mid v \in \lambda(w) \,\}$ induces a (connected) subtree of T.

The *width* of the tree decomposition T is $\max_{w \in W}(|\lambda(w)| - 1)$. The *treewidth* of φ is the minimum width over all the tree decompositions of G_φ. The treewidth of a rule is defined as the treewidth of its body. Finally, the class $L^{k'}_{tw}$ consists of all ontologies where each rule has treewidth at most k'. $\diamond$

The rules corresponding to $\mathcal{SROIQ}$ ontologies are not only of fixed predicate arity, but also their bodies are tree-shaped (see Section 2.2). The latter implies that rules stemming from $\mathcal{SROIQ}$ ontologies have treewidth at most one.

As already discussed, an appealing feature of our approach is that module extraction can be delegated to an off-the-shelf datalog reasoner, regardless of the language in which ontologies are expressed. The following proposition establishes that both the datalog program and the initial dataset exploited in our approach are of polynomial size; furthermore, the datalog transformation Ξ^χ in the definition of a module setting (see Definition 20 in Section 5.2) does not alter the shape of rules in the original ontology in any significant way.

Proposition 69. *Let $\mathcal{O}$ be an ontology and $\Sigma \subseteq \mathsf{Sig}(\mathcal{O})$ a signature. Furthermore, let $\mathcal{S} \in \{\mathsf{m}, \mathsf{q}, \mathsf{f}, \mathsf{i}, \mathsf{c}, \mathsf{b}, \mathsf{wq}\}$ and $\Psi_\mathcal{S}(\mathcal{O}, \Sigma) = \chi = \langle \theta, \mathcal{D}_0, \mathcal{D}_r \rangle$. Then, $\mathcal{P}^\chi$ and $\mathcal{D}_0$ are of size linear in $|\mathcal{O}|$. Furthermore, if $\mathcal{O}$ is in L^k_{arity} (resp. in L^k_{tw}) for some fixed k, then so is $\mathcal{P}^\chi$.*

Proof. It is clear from Definition 5.2 that $\Xi^\chi(r)$ contains a datalog rule for each atom in the head of r. Thus, $\mathcal{P}^\chi$ is clearly of size linear w.r.t. $|\mathcal{O}|$. Furthermore, $\mathcal{D}_0$ contains one fact for each predicate in Σ, and since $\Sigma \subseteq \mathsf{Sig}(\mathcal{O})$, it follows that $\mathcal{D}_0$ is also of size linear w.r.t. $|\mathcal{O}|$. Finally, the transformation Ξ^χ does not increase the arity of predicates and the body of each rule $\Xi^\chi(r)$ coincides with that of r and hence preserves its treewidth. $\square$

The computational properties of datalog programs with bounded arity and/or treewidth are well-understood: fact entailment is NP-complete in combined com-

plexity for programs of bounded arity, and the complexity drops to PTIME if we additionally restrict ourselves to programs of bounded treewidth. Bounded arity of predicates implies that the corresponding materialisation is polynomially bounded in size, and thus can be computed in a polynomial number of steps (i.e., applications of the immediate consequence operator); furthermore, bounded treewidth of rule bodies implies that each such step can be performed in polynomial time [34, 17].

The complexity of module extraction, however, is not only determined by that of datalog reasoning, but also by the complexity of computing the support of all proofs involved in a relevant entailment.

The following theorem, proved in [101], establishes that computing such support can also be reduced to standard datalog reasoning. Given a program $\mathcal{P}$, a dataset $\mathcal{D}$, and a set of facts $\mathcal{F}$, the main idea is to extend $\mathcal{P}$ and $\mathcal{D}$ with additional rules and facts that are responsible for computing the support of all proofs of facts from $\mathcal{F}$ in $\mathcal{P} \cup \mathcal{D}$. Such support is "recorded" by means of fresh predicates: auxiliary predicates $\bar{Q}$ are used to record relevant facts in $\mathcal{D}$ of the form $Q(\mathbf{c})$; furthermore, each rule $r \in \mathcal{P}$ is represented by a fresh constant d_r, and a fresh unary predicate Rel is used to capture the relevant rules from $\mathcal{P}$ in the support.

Theorem 70 (Zhou et al. [101])**.** *Let $\mathcal{P}$ be a datalog program, let $\mathcal{D}$ be a dataset, and let $\mathcal{F}$ be a set of facts in the materialisation of $\mathcal{P} \cup \mathcal{D}$. Let Rel be a fresh unary predicate and, for each predicate Q occurring in $\mathcal{P} \cup \mathcal{D}$, let $\bar{Q}$ be a fresh predicate of the same arity. Furthermore, let d_r be a fresh constant for each $r \in \mathcal{P}$.*

Let $\Delta(\mathcal{F})$ be the dataset $\Delta(\mathcal{F}) = \{\, \bar{P}(\mathbf{c}) \mid P(\mathbf{c}) \in \mathcal{F} \,\}$ and let $\Delta(\mathcal{P})$ be the smallest datalog program including $\mathcal{P}$ and containing all of the following rules for each

$r = \bigwedge_{j=1}^{m} B_1(\mathbf{x}_j) \rightarrow H(\mathbf{x})$ *in* $\mathcal{P}$*:*

$$\begin{aligned}
\bar{H}(\mathbf{x}) \wedge B_1(\mathbf{x}_1) \wedge \ldots \wedge B_m(\mathbf{x}_m) &\rightarrow \mathsf{Rel}(d_r) \\
\bar{H}(\mathbf{x}) \wedge B_1(\mathbf{x}_1) \wedge \ldots \wedge B_m(\mathbf{x}_m) &\rightarrow \bar{B}_i(\mathbf{x}_1) \\
&\vdots \\
\bar{H}(\mathbf{x}) \wedge B_1(\mathbf{x}_1) \wedge \ldots \wedge B_m(\mathbf{x}_m) &\rightarrow \bar{B}_i(\mathbf{x}_m)
\end{aligned}$$

Then, a rule $s \in \mathcal{P}$ *is in the support of some proof in* $\mathcal{P} \cup \mathcal{D}$ *of a fact from* $\mathcal{F}$ *iff* $\mathsf{Rel}(d_s)$ *is in the materialisation of* $\Delta(\mathcal{P}) \cup (\mathcal{D} \cup \Delta(F))$*.*

Example 71. Consider the program $\mathcal{P}$ consisting of rules r'_6 and r'_7 from Figure 5.1. The program $\Delta(\mathcal{P})$ consists of the following rules:

$$\begin{aligned}
&D(x) \rightarrow F(x) && (r'_6) \\
&\bar{F}(x) \wedge D(x) \rightarrow \mathsf{Rel}(d_{r'_6}) \\
&\bar{F}(x) \wedge D(x) \rightarrow \bar{D}(x) \\
&F(x) \rightarrow S(x, c_{y_2}) && (r'_7) \\
&\bar{S}(x, c_{y_2}) \wedge F(x) \rightarrow \mathsf{Rel}(d_{r'_7}) \\
&\bar{S}(x, c_{y_2}) \wedge F(x) \rightarrow \bar{F}(x)
\end{aligned}$$

Consider also the dataset $\mathcal{D} = \{D(a)\}$. It is immediate that the materialisation of $\mathcal{P} \cup \mathcal{D}$ contains the fact $\gamma = S(a, c_{y_2})$ and that there exists a proof ρ of γ in $\mathcal{O} \cup \mathcal{D}$ whose support is all of $\mathcal{P}$. Given $\mathcal{F} = \{S(a, c_{y_2})\}$ we have that $\Delta(\mathcal{F}) = \{\bar{S}(a, c_{y_2})\}$, and it is easy to see that, as follows from Theorem 70, the materialisation of $\Delta(\mathcal{P}) \cup (\mathcal{D} \cup \Delta(\mathcal{F}))$ contains the facts $\mathsf{Rel}(d_{r'_6})$ and $\mathsf{Rel}(d_{r'_7})$. $\diamond$

The datalog program $\Delta(\mathcal{P})$ (resp. the dataset $\mathcal{D} \cup \Delta(\mathcal{F})$) in Theorem 70 is of size polynomial in $|\mathcal{P}|$ (resp. in $|\mathcal{D}|$ and $|\mathcal{F}|$) and does not use any predicates with arity greater than that of predicates used in $\mathcal{P}$. However, $\Delta(\mathcal{P})$ may have an arbitrarily larger treewidth than that of $\mathcal{P}$. We next argue that in the case of normalised $\mathcal{SROIQ}$

ontologies the increase in treewidth is bounded.

Proposition 72. *Let $\mathcal{O}$ be a normalised $\mathcal{SROIQ}$ ontology and $\Sigma \subseteq \mathsf{Sig}(\mathcal{O})$ a signature. Furthermore, let χ be a module setting for $\mathcal{O}$ and Σ. Then $\Delta(\mathcal{P}^{\chi})$ has treewidth at most 2.*

Proof. For each rule $r \in \mathcal{O}$ whose head is formed only by (one or more) atoms that are unary, or mention no more than one variable, it is straightforward that $\Delta(\Xi^{\chi}(r))$ still consists only of rules with tree-shaped bodies. For each $r \in \mathcal{O}$ that does not mention more than two variables, it is also straightforward that $\Delta(\Xi^{\chi}(r))$ also consists only of rules with tree-shaped bodies. Finally, if r mentions more than two variables, and its head contains atoms that mention more than one variable, then r must be of one of the following forms:

- $A(x) \wedge \bigwedge_{i=1}^{m+1}[R(x, y_i) \wedge B(y_i)] \rightarrow \bigvee_{i \neq j} y_i \approx y_j$
  Then the bodies of the rules in $\Delta(\Xi^{\chi}(r))$ will be of the form $A(x) \wedge \bigwedge_{i=1}^{m+1}[R(x, y_i) \wedge B(y_i)] \wedge y_{i_1} \approx y_{i_2}$ with $1 \leq i_1 < i_2 \leq m$, and hence will have treewidth 2 due to the cycle of length 3 formed by $R(x, y_{i_1})$, $R(x, y_{i_2})$ and $y_{i_1} \approx y_{i_2}$.

- $R_1(x, y) \wedge R_2(y, z) \rightarrow S(x, z)$
  Then the body of the single rule in $\Delta(\Xi^{\chi}(r))$ will be of the form $R_1(x, y) \wedge R_2(y, z) \wedge S(x, z)$, which has treewidth 2 due to the cycle of length 3 formed by its three atoms.

- $x \approx y \wedge y \approx z \rightarrow x \approx z$
  Analogous to the previous case.

- $A(x_1, x_2) \wedge x_1 \approx y \rightarrow A(y, x_2)$
  Idem.

- $A(x_1, x_2) \wedge x_2 \approx y \rightarrow A(x_1, y)$
  Idem. □

The following theorem establishes two practically relevant cases for which module extraction in our framework can be performed in polynomial time. We first show that modules $\mathcal{M}^{\chi_{\mathsf{m}}}$ ensuring model-inseparability are computable in polynomial time for arbitrary ontologies and signatures. Then, we establish that modules $\mathcal{M}^{\chi}$ for the remaining inseparability relations considered in this thesis are also computable in polynomial time for all classes of ontologies whose extended datalog program $\Delta(\mathcal{P}^{\chi})$ in Definition 70 can be bounded in both predicate arity and treewidth.

Theorem 73. *Let L be a class of ontologies. and let $\mathcal{S} \in \{\mathsf{m}, \mathsf{q}, \mathsf{f}, \mathsf{i}, \mathsf{c}, \mathsf{wq}\}$. Furthermore, let $L_{\mathcal{S},\Delta}$ be the following class of datalog programs:*

$$L_{\mathcal{S},\Delta} = \{\, \Delta(\mathcal{P}^{\Psi^{\mathcal{S}}(\mathcal{O},\Sigma)}) \mid \mathcal{O} \in L, \Sigma \subseteq \mathsf{Sig}(\mathcal{O}) \,\}$$

The problem $\mathsf{isInModule}_{[L,\mathcal{S}]}$ is decidable in linear time if $\mathcal{S} = \mathsf{m}$, and in polynomial time if $L_{\mathcal{S},\Delta} \subseteq L^{k}_{arity} \cap L^{k'}_{tw}$ for some fixed, non-negative integers k and k'.

Proof. Let $\mathcal{O}$ be an arbitrary ontology $\mathcal{O}$ and Σ a signature. Consider the module setting $\Psi^{\mathcal{S}}(\mathcal{O}, \Sigma) = \langle \theta, \mathcal{D}_0, \mathcal{D}_r \rangle$ and let $\mathcal{P} = \mathcal{P}^{\Psi^{\mathcal{S}}(\mathcal{O},\Sigma)}$ and $\mathcal{F} = \mathcal{D}_r \cap \mathcal{P}(\mathcal{D}_0)$. By Proposition 69, $\mathcal{P}$ and $\mathcal{D}_0$ can be computed in time polynomial in the size of $\mathcal{O}$, and so can $\Delta(\mathcal{P})$. Therefore, it suffices to show that $\mathcal{D}_0 \cup \Delta(\mathcal{F})$ can be obtained in polynomial time and so can the materialisation of $\Delta(\mathcal{P}) \cup (\mathcal{D}_0 \cup \Delta(\mathcal{F}))$.

If $\mathcal{S} = \mathsf{m}$ then $\mathcal{D}_r$ and $\mathcal{D}_0 \cup \Delta(\mathcal{F})$ can be computed in linear time. Furthermore, it is easy to see that computing the materialisation of $\Delta(\mathcal{P}) \cup (\mathcal{D}_0 \cup \Delta(\mathcal{F}))$ is also feasible in linear time since it boils down to propositional datalog reasoning.

Consider now the case of $\mathcal{S} \in \{\mathsf{q}, \mathsf{f}, \mathsf{i}, \mathsf{c}, \mathsf{wq}\}$ and $L_{\mathcal{S},\Delta} \subseteq L^{k}_{arity} \cap L^{k'}_{tw}$. Note that $\Delta(\mathcal{P}) \in L^{k}_{arity} \cap L^{k'}_{tw}$ implies $\mathcal{P} \in L^{k}_{arity} \cap L^{k'}_{tw}$. Hence, $\mathcal{P}(\mathcal{D}_0)$ can be computed in time polynomial in the size of $\mathcal{O}$. In this case $\mathcal{F}$ is always a set of facts in $\mathcal{P}(\mathcal{D}_0)$ over predicates from Σ, and therefore it can be computed in polynomial time. Moreover, the dataset $\mathcal{D}_0 \cup \Delta(\mathcal{F})$ can be computed in polynomial time as well and thus, since

$\Delta(\mathcal{P}) \in L^k_{arity} \cap L^{k'}_{tw}$, so can the materialisation of $\Delta(\mathcal{P}) \cup (\mathcal{D}_0 \cup \Delta(\mathcal{F}))$. □

Tractability of module extraction w.r.t. $\mathcal{SROIQ}$ ontologies is now an immediate consequence of Theorem 73 and Proposition 72.

Corollary 74. *Let $\mathcal{S} \in \{\mathsf{m}, \mathsf{q}, \mathsf{f}, \mathsf{i}, \mathsf{c}, \mathsf{wq}\}$ and let $\mathcal{O}$ be a normalised $\mathcal{SROIQ}$ ontology. The module $\mathcal{M}^{\chi_\mathcal{S}}$ is computable in polynomial time.*

5.8 Optimality

As already discussed, in general, our modules are not minimal for their corresponding inseparability relation. It is, however, of interest to determine which module setting families yield the smallest possible modules for a given inseparability relation within the limits of our framework. To this end, we next present and study a suitable notion of *optimality* applicable to module setting families.

Our notion of module setting family in Definition 59 is rather general in that it does not establish any relationship between the different module settings in the family. In order to study optimality, it makes sense to restrict ourselves to families satisfying certain *uniformity* conditions. Roughly speaking, we consider a family as uniform if *(i)* existentially quantified variables and constants in ontologies are treated homogeneously within a setting (i.e., different existential variables receive the same treatment, and so do different constants) as well as consistently across different settings; and *(ii)* signatures are treated monotonically across settings (i.e., if χ and χ' are members of a family for some ontology $\mathcal{O}$ and signatures Σ and Σ' with $\Sigma \subseteq \Sigma'$, then they treat predicates in Σ and $\mathsf{Sig}(\mathcal{O}) \backslash \Sigma'$ in exactly the same way).

Definition 75. A module setting family Ψ is *uniform* if, for each pair of ontologies $\mathcal{O}, \mathcal{O}'$ and signatures Σ, Σ', the module settings $\Psi(\mathcal{O}, \Sigma) = \langle \theta, \mathcal{D}_0, \mathcal{D}_r \rangle$ and $\Psi(\mathcal{O}', \Sigma') = \langle \theta', \mathcal{D}'_0, \mathcal{D}'_r \rangle$ satisfy the following properties, where $\mathsf{Ex}(\mathcal{F})$ denotes the set of existentially quantified variables in $\mathcal{F}$:

1. If $\Sigma = \Sigma'$, $|\mathsf{Ct}(\mathcal{O})| \leq |\mathsf{Ct}(\mathcal{O}')|$ and $|\mathsf{Ex}(\mathcal{O})| \leq |\mathsf{Ex}(\mathcal{O}')|$, then for each injective substitution $\nu : \mathsf{dom}(\theta) \to \mathsf{dom}(\theta')$ mapping variables to variables and constants to constants we have

   - $\theta = \nu\theta'$,
   - $\mathcal{D}_0 = \{\, A(\mathbf{c}) \mid A(\mathbf{c}\nu) \in \mathcal{D}'_0,\ \mathbf{c}$ has only constants from $\mathsf{Ct}(\Psi(\mathcal{O}, \Sigma))\,\}$, and
   - $\mathcal{D}_r = \{\, A(\mathbf{c}) \mid A(\mathbf{c}\nu) \in \mathcal{D}'_r,\ \mathbf{c}$ has only constants from $\mathsf{Ct}(\Psi(\mathcal{O}, \Sigma))\,\}$.

2. If $\mathcal{O} = \mathcal{O}'$ and $\Sigma \subseteq \Sigma'$, then

   - $\theta = \theta'$,
   - $\mathcal{D}_0 = \{\, A(\mathbf{c}) \in \mathcal{D}'_0 \mid A \in \Sigma\,\}$,
   - $\{\, A(\mathbf{c}) \in \mathcal{D}_r \mid A \in \Sigma\,\} = \{\, A(\mathbf{c}) \in \mathcal{D}'_r \mid A \in \Sigma\,\}$, and
   - $\{\, A(\mathbf{c}) \in \mathcal{D}_r \mid A \in \mathsf{Sig}(\mathcal{O})\backslash\Sigma'\,\} = \{\, A(\mathbf{c}) \in \mathcal{D}'_r \mid A \in \mathsf{Sig}(\mathcal{O})\backslash\Sigma'\,\}$.

Let $\mathcal{S}$ be an inseparability relation and let $\mathbf{\Psi}_{\mathcal{S}}$ be the class of all uniform module setting families that are $\mathcal{S}$-admissible. We say that Ψ is *$\mathcal{S}$-optimal* if $\Psi \in \mathbf{\Psi}_{\mathcal{S}}$ and $\mathcal{M}^{\Psi(\mathcal{O},\Sigma)} \subseteq \mathcal{M}^{\Psi'(\mathcal{O},\Sigma)}$ for every $\Psi' \in \mathbf{\Psi}_{\mathcal{S}}$ and each pair of $\mathcal{O}$ and Σ. $\diamond$

It is easy to see that each of the $\mathcal{S}$-admissible families $\Psi^{\mathcal{S}}$, with $\mathcal{S} \in \{\mathsf{m}, \mathsf{q}, \mathsf{f}, \mathsf{i}, \mathsf{c}, \mathsf{wq}\}$, is uniform. Furthermore, the following theorem establishes that Ψ^{m}, Ψ^{i}, Ψ^{c}, and Ψ^{wq} are also optimal for their respective inseparability relations. The proof of the theorem is rather technical and is deferred to the appendix.

Theorem 76. *The family $\Psi^{\mathcal{S}}$ is $\mathcal{S}$-optimal for $\mathcal{S} \in \{\mathsf{m}, \mathsf{i}, \mathsf{c}, \mathsf{wq}\}$.*

In contrast, the families Ψ^{f} and Ψ^{q} for fact and query inseparability are not optimal. To see this, consider the ontology $\mathcal{O}$ consisting of the following rules:

$$A(x) \to B(x) \qquad B(x) \to A(x) \qquad C(x) \to D(x)$$

Furthermore, let $\Sigma = \{A, C, D\}$. The module setting $\chi_{\mathsf{f}} = \Psi^{\mathsf{f}}(\mathcal{O}, \Sigma)$ yields $\mathcal{M}^{\chi_{\mathsf{f}}} = \mathcal{O}$. Indeed, for $\mathcal{D} = \{A(a)\}$ we have a non-trivial proof of $A(a)$ in $\mathcal{O} \cup \mathcal{D}$ that involves rules $A(x) \to B(x)$ and $B(x) \to A(x)$, which are then included in the module. However, it is clear that $\mathcal{M} = \{C(x) \to D(x)\}$ is already Σ-fact inseparable from $\mathcal{O}$. We can in fact define a module setting $\chi = \langle \theta, \mathcal{D}_0, \mathcal{D}_r \rangle$ whose corresponding module is precisely $\mathcal{M}$. For this, the idea is to define $\mathcal{D}_0$ and $\mathcal{D}_r$ in such a way that the aforementioned proofs of tautological statements are avoided. Consider $\mathcal{D}_0$ and $\mathcal{D}_r$ as follows:

$$\begin{aligned} \mathcal{D}_0 &= \{X(c_Y^0) \mid X, Y \in \Sigma\} \cup \{X(c_Y^1) \mid X, Y \in \Sigma \text{ and } X \neq Y\} \\ \mathcal{D}_r &= \{Y(c_Y^1) \mid Y \in \Sigma\} \end{aligned}$$

Datasets $\mathcal{D}_0$ and $\mathcal{D}_r$ are disjoint, which guarantees that proofs of Σ-tautologies are not taken into account and therefore $\mathcal{M}^{\chi}$ is indeed $\mathcal{M}$. The construction of $\mathcal{D}_0$ and $\mathcal{D}_r$ given for this example ontology and signature can be generalised so as to define a uniform module setting family Ψ that provides a counter-example to the optimality of Ψ^{f}. There is, however, a price to pay for such smaller modules, namely an increase in the size of module settings. Indeed, $\Psi(\mathcal{O}, \Sigma)$ is of size exponential in Σ, whereas the size of $\Psi^{\mathsf{f}}(\mathcal{O}, \Sigma)$ remains polynomial. Such exponential blow-up is clearly undesirable in practice.

The following theorem establishes that Ψ^{f} and Ψ^{q} are not optimal. They do, however, work well in practice, as shown in the evaluation presented in Chapter 6. The proof of the theorem works by proposing "better" module setting families which, in both cases, incur the aforementioned exponential blow-up. We conjecture that such a blow-up is unavoidable in any optimal module setting family for fact or query inseparability (if such a family exists). As in the case of Theorem 76, the proof is technical and is deferred to the appendix.

Theorem 77. *The family $\Psi^{\mathcal{S}}$ is not $\mathcal{S}$-optimal for $\mathcal{S} \in \{\mathsf{f}, \mathsf{q}\}$.*

5.9 Related Work

Module extraction has received a great deal of attention in the literature. In Section 5.9.1 we discuss the complexity of inseparability checking for different ontology languages and inseparability relations. In Section 5.9.2 we recapitulate existing module extraction techniques based on inseparability, and provide a brief overview of their practical applications. Finally, in Section 5.9.3 we discuss a number of problems, such as forgetting and interpolation, which are closely related to inseparability checking and module extraction.

5.9.1 Inseparability Relations

Inseparability relations originate in the notions of model and deductive conservative extensions for description and modal logics [3, 30, 31], and constitute the foundation of module extraction techniques [52].

Model inseparability is undecidable for all description logics that extend $\mathcal{EL}$ [63]. It is, however, tractable for $\mathcal{ELI}$ ontologies that are acyclic if one of them is the empty ontology [53]; furthermore, it is coNExpTime$^{\text{NP}}$-complete for the description logic $\mathcal{ALCI}$ if signatures are restricted to consist of atomic concepts only [53]. For *DL-Lite*$^{\mathcal{N}}_{bool}$ and *DL-Lite*$^{\mathcal{N}}_{horn}$ it is coNExpTime-hard [56], but no matching upper bound is known to the best of our knowledge.

The complexity of query inseparability has been studied mainly for lightweight DLs. It is ExpTime-complete for $\mathcal{EL}$ [63], Π^p_2-complete and coNP-complete for *DL-Lite*$^{\mathcal{N}}_{bool}$ and *DL-Lite*$^{\mathcal{N}}_{horn}$, respectively [56], and ExpTime-complete for *DL-Lite*$^{\mathcal{H}}_{core}$ and *DL-Lite*$^{\mathcal{H}}_{horn}$ [50, 13]. Baader et al. [6] considered a variant of query inseparability where the signature of datasets is restricted to Σ but the signature of queries is not, and identified decidable sufficient conditions for such inseparability in $\mathcal{ELI}$, which can be checked in polynomial time for $\mathcal{EL}$. The complexity of weak query

inseparability (see Section 5.4.2) was studied by Botoeva et al. [13]; it is known to be P-complete for *DL-Lite*$_{core}$, *DL-Lite*$_{horn}$ and $\mathcal{ELH}$, ExpTime-complete for *DL-Lite*$^{\mathcal{H}}_{core}$ and *DL-Lite*$^{\mathcal{H}}_{horn}$, and 2ExpTime-complete for both Horn-$\mathcal{ALCHI}$ and Horn-$\mathcal{ALCI}$.

The complexity of implication and classification inseparability coincides with that of standard reasoning tasks such as subsumption checking [52]. In the description logic literature implication inseparability has been studied in a more general form: given $\mathcal{L}$-ontologies $\mathcal{O}$ and $\mathcal{O}'$ and a signature Σ, the problem is to determine whether $\mathcal{O}$ and $\mathcal{O}'$ entail the same concept inclusion axioms $C \sqsubseteq D$ where C and D are (possibly complex) $\mathcal{L}$-concepts over Σ. In this setting, inseparability has been found to be Π^p_2-complete for *DL-Lite*$^{\mathcal{N}}_{bool}$, coNP-complete for *DL-Lite*$^{\mathcal{N}}_{horn}$ [56], ExpTime-complete for $\mathcal{EL}$ [63], 2ExpTime-complete for $\mathcal{ALC}$ [30], $\mathcal{ALCI}$, $\mathcal{ALCQ}$ and $\mathcal{ALCQI}$ [52], and undecidable for $\mathcal{ALCQIO}$ [52]. Note that this variant of implication inseparability is highly dependent on the ontology language $\mathcal{L}$, whereas our results are largely logic-independent. We leave the investigation of such inseparability relations within our framework as an interesting problem for future work.

5.9.2 Module Extraction

Practical module extraction techniques are typically based on approximations, which ensure that the computed module is (model) inseparable from the given ontology, yet not necessarily minimal. One such approximation, which we discussed in detail in Chapter 4, is based on syntactic locality [19, 21, 77]. An implementation of $\bot$-, $\top$- and $\top\bot^*$-module extraction is integrated in the OWLAPI,[2] and an alternative implementation can be downloaded as a separate Java library.[3] The semantic counterpart of syntactic locality, semantic locality, was proposed in [20]. Deciding semantic locality is, for any given DL, as hard as checking satisfiability w.r.t. the empty TBox,

[2] `http://owlapi.sourceforge.net/`

[3] `https://www.cs.ox.ac.uk/isg/tools/ModuleExtractor/`

and hence it is only tractable for logics with restricted expressivity. For this reason modules based on syntactic locality, which can be extracted in polynomial time, have been the preferred choice in practice. Furthermore, an exhaustive comparison of syntactic and semantic locality modules [25] revealed that the difference between them is not significant in most practical cases.

Reachability-based modules [89, 72, 73] can be seen as a refinement of syntactic locality modules. Available in the same three flavours ($\bot$-, $\top$- and $\top\bot^*$-reachability), they are also modules for model inseparability and can be extracted from $\mathcal{SROIQ}$ ontologies in polynomial time. While $\bot$-reachability modules coincide with $\bot$-modules, $\top$- and $\top\bot^*$-reachability modules are generally a subset of their syntactic locality counterparts. This refinement comes at the cost of losing depletingness, although reachability modules are still self-contained and preserve all justifications for consequences over the reference signature Σ.

Konev et al. [53] and Gatens et al. [29] developed module extraction techniques for model inseparability in acyclic $\mathcal{ELI}$ and acyclic $\mathcal{ALCQI}$, respectively. These techniques ensure that the extracted modules are self-contained and depleting, and in the case of $\mathcal{ELI}$ also minimal. The polynomial algorithm for acyclic $\mathcal{ELI}$ is implemented in the system MEX,[4] and the more general, non-tractable algorithm for acyclic $\mathcal{ALCQI}$ is implemented in the system AMEX.[5] In contrast to locality and reachability modules, the applicability of these techniques is limited to a relatively restricted class of ontologies, and tractability is only guaranteed for an even more restricted class.

Kontchakov et al. [56] exploited the decidability of query inseparability for the logics *DL-Lite*$^{\mathcal{N}}_{bool}$ and *DL-Lite*$^{\mathcal{N}}_{horn}$ for module extraction. The module extraction techniques in [56] produce minimal or minimal depleting modules and have the same complexity as the corresponding inseparability relation (they are Π^p_2-complete and

[4] `http://cgi.csc.liv.ac.uk/~konev/software/`
[5] `http://www.csc.liv.ac.uk/~wgatens/software/amex.html`

coNP-complete, respectively).

Konev et al. [50] devised a technique for extracting minimal depleting modules for query inseparability from ontologies in a dialiect of DL-Lite in polynomial time.

Baader et al. [6] proposed exponential time algorithms to extract modules from $\mathcal{ELI}$ ontologies that preserve a variant of query inseparability. Furthermore, they showed that computing such modules is feasible in polynomial time for $\mathcal{EL}$ ontologies.

Recently, Rousset and Ulliana [76] studied modularity in the context of *deductive triple stores*, that is, RDF triple stores equipped with a set of datalog rules. The preservation properties of the modules in [76], however, are very different from the ones considered in our work.

Del Vescovo et al. [26] considered the problem of finding a polynomial representation of all modules of an ontology, for a particular notion of module. The proposed representation is called *atomic decomposition* and is applicable to any notion of a module that satisfies certain properties that include self-containment and depletingness. The atomic decomposition of an ontology for a suitable notion of module can be computed in polynomial time using a module extraction algorithm as an oracle.

Module extraction has been identified as a key task to support knowledge reuse [21, 43]. Modules have also been exploited to optimise ontology matching [41], as well as the computation of justifications [90, 61] for ontology debugging and explanation. Finally, module extraction techniques have been successfully applied to optimising ontology classification [94, 89, 18] and have been integrated in the ontology reasoner Chainsaw.[6]

5.9.3 Related Problems

Module extraction is strongly related to the notions of *forgetting* and *uniform interpolation* [27, 62, 57, 54, 70, 98]. A uniform interpolant of an $\mathcal{L}$-ontology $\mathcal{O}$ and

[6] `http://sourceforge.net/projects/chainsaw/`

a signature Σ is an $\mathcal{L}$-ontology $\mathcal{O}'$ that only mentions symbols from Σ and which is inseparable from $\mathcal{O}$ w.r.t. Σ for a given inseparability relation. In contrast to modules, uniform interpolants are not required to be subsets of $\mathcal{O}$ and they cannot contain any symbol outside Σ (all remaining symbols are thus *forgotten*). The latter requirement implies that uniform interpolants for a given Σ and $\mathcal{O}$ may not always exist [54, 64, 97].

Konev et al. [51] studied the problem of computing the *logical difference* of ontologies $\mathcal{O}$ and $\mathcal{O}'$—that is, the set of of queries that receive different answers w.r.t. $\mathcal{O}$ and $\mathcal{O}'$. Computing the logical difference (or a concise representation thereof) has been identified as a valuable resource for ontology versioning tasks [42] and is closely related to inseparability checking; indeed, inseparable ontologies are those that have an empty difference.

Finally, Zhou et al. [101] proposed a hybrid approach to ontology-based query answering where the bulk of the computation is delegated to a datalog reasoner. Given an ontology $\mathcal{O}$, dataset $\mathcal{D}$, query $q(\mathbf{x})$, and candidate answer tuple $\mathbf{c}$, a core technique in this approach is to compute fragments $\mathcal{O}' \subseteq \mathcal{O}$ and $\mathcal{D}' \subseteq \mathcal{D}$ such that $\mathcal{O} \cup \mathcal{D} \models q(\mathbf{c})$ iff $\mathcal{O}' \cup \mathcal{D}' \models q(\mathbf{c})$. Similarly to our modules, these fragments are computed by first strengthening $\mathcal{O}$ into a datalog program $\mathcal{P}$ and then exploiting the datalog reasoner to identify the axioms and facts responsible for the validity of $q(\mathbf{c})$. In contrast to query inseparable modules, however, the fragment $\mathcal{O}' \cup \mathcal{D}'$ is only guaranteed to preserve the fixed query $q(\mathbf{c})$ w.r.t. the fixed dataset $\mathcal{D}$, rather than all queries w.r.t. all datasets over a reference signature.

Chapter 6

The PrisM Parameterised Module Extractor

6.1 System Description

The PrisM[1] system implements the module extraction technique described in Chapter 5. PrisM is written in Java and is available under academic license. It integrates the datalog reasoner (or *triple store*), RDFox[2] [67], which is used as a "black box", and also several modules of the PAGOdA[3] query answering system [101]. In addition to these, our system makes use of the OWLAPI.[4]

PrisM takes as input an OWL 2 ontology and a signature, as well as a parameter indicating the inseparability relation that sets the requirements for the module to be extracted; the inseparability relation can be any $\mathcal{S} \in \{\mathsf{i}, \mathsf{f}, \mathsf{q}, \mathsf{m}, \mathsf{c}, \mathsf{wq}\}$, as defined in Chapters 4 and 5. Upon this input, PrisM returns a module of (a normalisation of) the given ontology for the specified signature and inseparability relation. The system currently offers restricted support for OWL 2 datatypes: any (possibly negated) facts

[1] `http://www.cs.ox.ac.uk/isg/tools/PrisM/`
[2] `http://www.cs.ox.ac.uk/isg/tools/RDFox/`
[3] `http://www.cs.ox.ac.uk/isg/tools/PAGOdA/`
[4] `http://owlapi.sourceforge.net/`

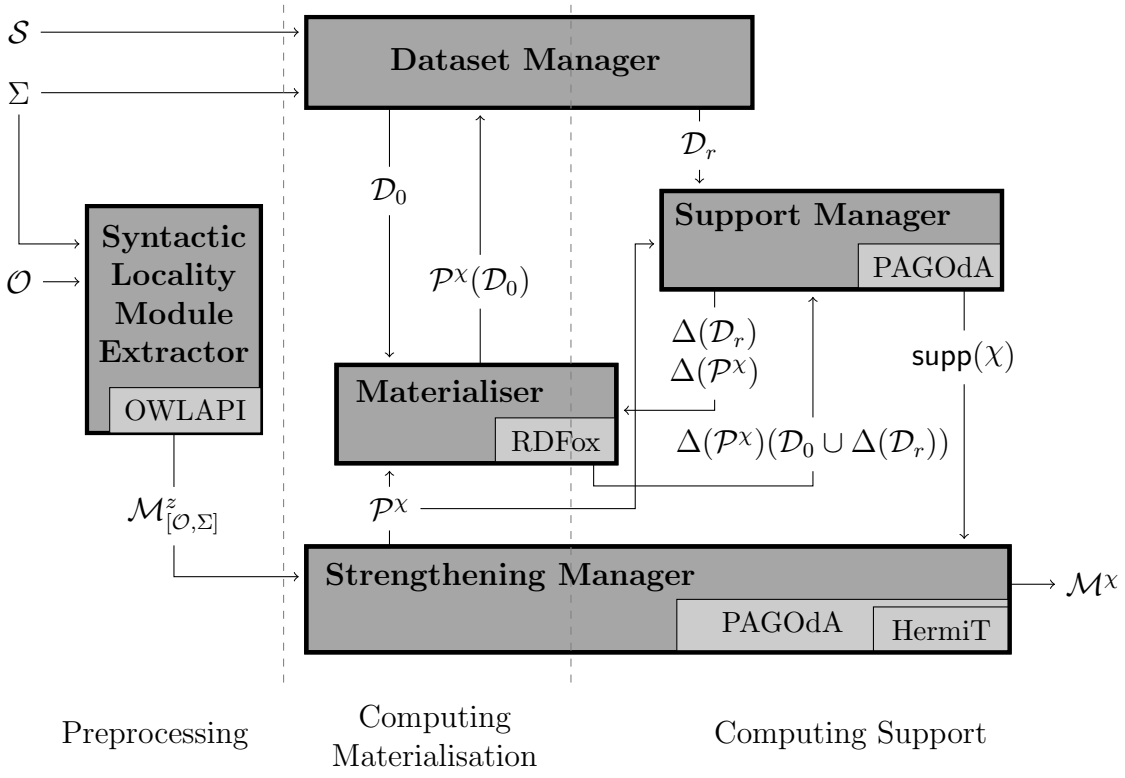


Figure 6.1: Workflow in the PrisM system

involving a data property (a special kind of binary predicate) are ignored.

Figure 6.1 depicts the architecture of the PrisM system. Different components are represented in different boxes, and the use of any external systems is indicated in the lower right corner of the corresponding box. Since RDFox is used in a black box manner, it would be possible in principle to use any other materialisation-based datalog reasoner with very basic query answering support. The PAGOdA system, on the other hand, is integrated in a tighter way: several of its components are reused and extended.

The workflow inside PrisM can be divided into three stages: a preprocessing stage, a stage for computing the materialisation, and a final stage for computing the support of the module setting and obtaining the module. We next describe each stage in more detail, as well as the corresponding components.

Stage 1: Preprocessing. The input ontology $\mathcal{O}$ is taken as a preloaded instance of the OWLOntology class from the OWLAPI. The signature Σ must be a set made up of instances of OWLClass (which represent atomic concepts in the OWLAPI) or OWLObjectProperty (which represent atomic roles). The inseparability relation $\mathcal{S}$ is chosen from an enumeration provided by the system. In order to speed up the extraction, the SyntacticLocalityModuleExtractor class from the OWLAPI is used to reduce $\mathcal{O}$ to a suitable preliminary module $\mathcal{M}^{z}_{[\mathcal{O},\Sigma]}$. If $\chi = \chi_{\mathsf{c}}$, then z is taken to be $\bot$ in order to guarantee Σ-classification inseparability from $\mathcal{O}$; otherwise, it is $z = \top\bot^*$ since $\mathcal{M}^{\top\bot^*}_{[\mathcal{O},\Sigma]}$ is generally smaller than $\mathcal{M}^{\bot}_{[\mathcal{O},\Sigma]}$ while still being Σ-model inseparable from $\mathcal{O}$. The fact that model inseparability (m) is stronger than any $\mathcal{S} \in \{\mathsf{i}, \mathsf{f}, \mathsf{q}, \mathsf{wq}\}$, together with the transitivity of inseparability relations, guarantees that this preprocessing step does not compromise the correctness of the overall procedure.

Stage 2: Computing Materialisation. A suitable module setting $\chi{=}\langle\theta, \mathcal{D}_0, \mathcal{D}_r\rangle$ for $\mathcal{M}^{x}_{[\mathcal{O},\Sigma]}$ and Σ is chosen (in accordance with the inseparability relation $\mathcal{S}$ provided as input). The Strengthening Manager, which is in charge of computing the program $\mathcal{P}^{\chi}$, is an extension of PAGOdA's component for computing a strengthening of an ontology, which, in turn, uses an extension of HermiT's clausification component [69] to normalise DL axioms and convert them into rule form before computing the strengthening. The Dataset Manager takes care of generating the set $\mathcal{D}_0$ of initial facts. Both $\mathcal{P}^{\chi}$ and $\mathcal{D}_0$ are then passed on to the Materialiser, which computes the materialisation of $\mathcal{P}^{\chi}$ and $\mathcal{D}_0$ using an instance of RDFox.

Stage 3: Computing Support. This stage begins with the computation of the set $\mathcal{D}_r$ of relevant facts, performed by the Dataset Manager. For efficiency reasons, rather than computing the whole set $\mathcal{D}_r$, the Dataset Manager queries the Materialiser for $\mathcal{P}^{\chi}(\mathcal{D}_0)$ and computes only the facts in $\mathcal{D}_r$ that have actually been materialised. The Support Extraction Manager then computes $\Delta(\mathcal{P}^{\chi})$ and $\Delta(\mathcal{D}_r)$ from $\mathcal{P}^{\chi}$ and $\mathcal{D}_r$ as explained in Section 5.7 using an extension of another component from PAGOdA.

The additional rules and facts obtained are added to RDFox (within the Materialiser), and the materialisation is extended accordingly, resulting in the whole materialisation of $\Delta(\mathcal{P}^\chi) \cup (\mathcal{D}_0 \cup \Delta(\mathcal{D}_r))$. The support of χ is read by the Support Extraction Manager from the materialisation of $\Delta(\mathcal{P}^\chi) \cup (\mathcal{D}_0 \cup \Delta(\mathcal{D}_r))$, and it is passed on to the Strengthening Manager, which identifies and finally outputs the corresponding axioms that constitute $\mathcal{M}^\chi$. The current prototypical implementation does not reverse the normalisation performed in the second stage; hence, if $\mathcal{O}$ was not in normal form, the result is not guaranteed to be a subset of $\mathcal{O}$, but rather of a normalisation of $\mathcal{O}$.

6.2 Evaluation

We have evaluated PrisM on a set of test ontologies identified in [33] as non-trivial for standard reasoning. All ontologies have been normalised prior to module extraction to make DL axioms equivalent to rules. Further details on these ontologies are given in Table 6.1.[5] The first and second columns in the table indicate the ontology ID and name in the Oxford Ontology Repository.[6] The third and fourth columns provide the number of predicates and rules in the resulting ontology after normalisation. The fifth and sixth columns specify how many of these rules contain disjunction and existential quantification in the head. Finally, the last column indicates the DL expressivity[7] of the normalised ontology as given by the the OWLAPI.

All experiments have been performed on a server with 2 Intel Xeon E5-2670 2.60GHz processors, each of which has 8 physical cores that serve 2 virtual cores each, making a total of 32 virtual cores. In our experiments we allocated 90GB of RAM, and RDFox was always run on 16 threads. We have compared the module sizes and extraction times using our system with those for locality-based $\bot$- and

[5]The ontologies used in our experiments are available for download at `https://krr-nas.cs.ox.ac.uk/2015/AnaArmas_thesis/testOntologies_MORe.zip`

[6]`http://www.cs.ox.ac.uk/isg/ontologies/UID/`

[7]We refer the reader to [8] for a detailed account on DL naming conventions.

ID	name	predicates	rules	disj.	exist.	expressivity
00001	ACGT-v1.0	2,019	5,512	105	259	$\mathcal{SROIQ}(D)$
00004	BAMS-simplified	1,199	18,976	0	16,782	$\mathcal{ALEHIF}^+$
00024	DOLCE	603	2,148	53	184	$\mathcal{SHOIN}(D)$
00026	GALEN-no-FIT	29,073	66,191	0	26,973	$\mathcal{ALEH}$
00029	GALEN-doctored	3,740	7,447	0	2,367	$\mathcal{ALEHIF}^+$
00032	GALEN-undoctored	3,762	7,818	0	2,715	$\mathcal{ALEHIF}^+$
00347	LUBM-one-uni	68	84,771	0	8	$\mathcal{ALEHI}^+(D)$
00350	OBI	2,965	10,952	77	1,168	$\mathcal{SHOIN}(D)$
00351	AERO	355	669	11	100	$\mathcal{SROIQ}(D)$
00354	NIF-gross-anatomy	4,166	7,134	51	1,506	$\mathcal{SROIF}(D)$
00463	Fly-anatomy-XP	8,047	42,107	0	9,433	$\mathcal{ALERI}^+$
00471	FMA-lite	78,986	168,828	0	42,734	$\mathcal{ALEH}^+$
00477	Gazetteer	150,981	382,158	0	156,743	$\mathcal{ALE}^+$
00512	Lipid	1,289	5,222	541	893	$\mathcal{ALCHIN}$
00545	Molecule-role	9,222	153,020	0	6,276	$\mathcal{ALE}^+$
00774	RNA-v0.2	338	938	34	90	$\mathcal{SRIQ}(D)$
00775	Roberts-family	183	2,020	1	73	$\mathcal{SROIQ}(D)$
00778	SNOMED	54,982	191,891	18,323	60,377	$\mathcal{SH}$
00786	NCI-v12.04e	93,628	193,453	65	76,957	$\mathcal{SH}(D)$

Table 6.1: Test ontologies

$\top\bot^*$-modules, computed using the OWL API. We have followed the experimental methodology from [25], where two kinds of signatures are considered:

- **genuine signatures**, which correspond to the signature of individual axioms, and
- **random signatures**, which include the signatures of several axioms.

Unlike Del Vescovo et al. [25], who defined random signatures simply as random subsets of the ontology signature, we extracted such signatures using a randomised graph sampling algorithm. We first represented the syntactic dependencies between symbols in the (normalised) ontology as a graph, and then traversed the graph in a randomised way until we visited a set number n of nodes. The symbols corresponding to the visited nodes were then taken as the random signature.[8] The advantage of this approach is that it yields signatures that are "semantically connected", which we

[8]The functionality required to perform random walks is currently integrated in RDFox.

	00001	00004	00024	00026	00029	00032	00347	00350	00351	00354
total	5,512	18,976	2,148	66,191	7,447	7,818	84,771	10,952	669	7,134
$\bot$	678	18,306	1,000	14,253	187	690	84,726	803	133	826
χ_{c}	558	16,942	883	9,799	94	479	23,651	558	74	618
$\top\bot^*$	674	18,297	990	13,749	114	596	58,186	768	130	786
χ_{m}	584	18,297	910	13,686	112	592	44,244	624	97	675
χ_{q}	563	17,151	884	9,448	96	533	44,368	596	77	626
χ_{wq}	514	0	875	0	0	0	43,761	538	64	111
χ_{f}	563	17,108	884	5,962	96	533	31,322	596	77	626
χ_{i}	558	655	882	3,279	18	130	11,234	558	67	617
$\lvert\Sigma\rvert$	2	3	2	3	3	3	2	2	2	2

	00463	00471	00477	00512	00545	00774	00775	00778	00786
total	42,107	168,828	382,158	5,222	153,020	938	2,020	191,891	193,453
$\bot$	22,348	47,192	214,820	261	143,399	80	1,916	433	1,140
χ_{c}	112	12	<1	86	6	76	1,491	426	390
$\top\bot^*$	221	20	9	34	2	80	1,913	427	1,138
χ_{m}	217	12	8	32	1	80	1,498	426	1,138
χ_{q}	107	12	8	29	1	78	1,492	426	385
χ_{wq}	0	0	0	0	0	0	1,490	0	0
χ_{f}	80	1	<1	29	<1	78	1,492	426	371
χ_{i}	12	1	<1	27	<1	76	1,491	397	120
$\lvert\Sigma\rvert$	3	2	3	2	3	2	2	3	3

Table 6.2: Module sizes for genuine signatures

believe is likely to be the case in practical applications. The number n was chosen by default as 0.1% of the total graph and then increased by up to two orders of magnitude in cases where the resulting signatures contained less than 15 predicates on average and thus were too small to provide additional information w.r.t. genuine signatures.

For each kind of signature and each ontology, we have considered a sample of 400 runs and averaged module sizes and module extraction times. On the one hand, we have compared the modules produced by χ_{c} (Section 5.4) with $\bot$-modules, which are the only kind of modules in the literature that ensure classification inseparability. On the other hand, we have compared the modules produced by χ_{m}, χ_{q}, χ_{wq}, χ_{f}, and χ_{i} (Sections 5.3 and 5.4) with $\top\bot^*$-modules. As discussed in Section 5.9.2, no other system is (to the best of our knowledge) capable of computing modules spe-

	00001	00004	00024	00026	00029	00032	00347	00350	00351	00354
total	5,512	18,976	2,148	66,191	7,447	7,818	84,771	10,952	669	7,134
$\bot$	857	18,904	1,053	27,771	1,890	3,279	84,732	1,795	315	1,537
χ_{c}	691	17,607	933	17,879	1,223	2,483	58,203	1,084	208	1,240
$\top\bot^*$	854	18,894	1,044	27,184	1,726	3,108	80,153	1,758	311	1,501
χ_{m}	759	18,894	964	27,175	1,719	3,101	63,031	1,611	274	1,388
χ_{q}	736	18,579	942	18,315	1,380	2,633	63,031	1,389	265	1,279
χ_{wq}	517	0	875	0	0	0	62,455	539	81	113
χ_{f}	735	18,536	942	18,255	1,364	2,620	58,980	1,389	241	1,278
χ_{i}	688	2,511	931	17,646	1,060	2,314	50,362	1,080	191	1,238
$\lvert\Sigma\rvert$	43	82	20	107	104	107	11	92	56	79
%	1	1	1	0.1	1	1	10	1	10	1

	00463	00471	00477	00512	00545	00774	00775	00778	00786
total	42,107	168,828	382,158	5,222	153,020	938	2,020	191,891	193,453
$\bot$	23,139	49,345	215,886	1,555	143,448	371	1,979	11,766	16,820
χ_{c}	595	402	38	1,199	28	338	1,527	11,342	7,974
$\top\bot^*$	982	1,658	1,050	837	16	371	1,977	11,762	16,817
χ_{m}	973	1,450	1,049	819	14	369	1,561	11,651	16,817
χ_{q}	757	1,450	1,049	774	14	368	1,557	11,644	8,969
χ_{wq}	0	0	0	0	0	0	1,506	0	0
χ_{f}	664	74	16	766	5	368	1,557	11,342	8,415
χ_{i}	333	74	16	467	5	338	1,526	11,342	6,228
$\lvert\Sigma\rvert$	28	154	312	66	19	58	42	202	326
%	0.1	0.1	0.1	1	0.1	10	10	0.1	0.1

Table 6.3: Module sizes for random signatures

cific to the deductive inseparability relations considered in this thesis. Furthermore, other module extraction systems that ensure model inseparability, such as MEX and AMEX, are only applicable to rather restricted ontology languages. Consequently, $\top\bot^*$-modules seemed the best available option for comparison to our approach.

Tables 6.2 and 6.3 provide the average number of rules in each kind of module for genuine and random signatures, respectively. In both tables, the total number of rules in the normalised ontology is provided at the top for comparison purposes, whereas the average size of the signatures considered is specified towards the bottom. Table 6.3 additionally includes the percentage n of the dependency graph covered by the random walks from which random signatures were obtained.

We can observe that module size consistently decreases as we consider weaker in-

	00001	00004	00024	00026	00029	00032	00347	00350	00351	00354
$\bot$	27	64	11	273	24	27	274	95	3	31
χ_{c}	846	36,784	1,066	30,256	295	989	14,823	747	130	6,588
$\top\bot^*$	43	95	19	418	39	42	463	124	5	43
χ_{m}	831	18,244	1,013	25,884	231	826	15,268	724	126	722
χ_{q}	857	62,680	1,074	29,051	241	903	14,688	798	136	3,530
χ_{wq}	857	32,972	1,069	27,272	229	861	10,677	787	140	767
χ_{f}	846	62,797	1,074	28,254	246	910	14,495	784	131	3,840
χ_{i}	847	35,158	1,072	28,499	234	890	10,285	789	132	3,826

	00463	00471	00477	00512	00545	00774	00775	00778	00786
$\bot$	122	506	1,154	18	363	5	10	722	569
χ_{c}	4,176	9,729	52,529	378	34,705	161	963	1,657	3,449
$\top\bot^*$	199	805	1,851	30	616	11	18	1,056	899
χ_{m}	775	1,543	4,995	166	2,783	147	857	1,590	3,340
χ_{q}	485	824	1,792	225	595	161	955	1,708	3,475
χ_{wq}	455	790	1,753	210	552	154	966	1,612	3,419
χ_{f}	463	788	1,772	229	580	164	967	1,700	3,526
χ_{i}	456	792	1,759	229	579	164	965	1,691	3,479

Table 6.4: Extraction times in milliseconds for genuine signatures

separability relations. The modules produced by χ_{c} can be several orders of magnitude smaller than $\bot$-modules, as in the cases of 00463, 00471, 00477 or 00545. Although those are rather extreme cases, we observed in most cases at least a 75% decrease in size (see 00026, 00029, 00032, 00347, 00350, 00351, 00786). Our modules for model inseparability improve reasonably on $\top\bot^*$-modules in most cases, although the greatest difference in size is of course between $\top\bot^*$-modules and χ_{i}-modules, reaching one order of magnitude for some ontologies (see 00471 and 00477, and also 0004, 00463 and 00786 with genuine signatures). In realistic ontologies, only a very small proportion of predicate pairs are related by atomic implication, and this is often still the case when considering a datalog overestimation of the ontology; thus, the difference in size between $\top\bot^*$-modules and χ_{i} is rather unsurprising. There is naturally also a big difference in size between χ_{wq}-modules and all other modules whenever ontologies do not mention any constants, since the former are in that case obviously empty. It is worth observing that, even though there are several cases where χ_{m} modules and χ_{q} modules are of very similar size (e.g. 00471), there are also cases where they differ

	00001	00004	00024	00026	00029	00032	00347	00350	00351	00354
$\bot$	26	58	10	284	35	39	276	107	4	29
χ_{c}	1,076	36,376	1,162	56,125	2,989	5,737	15,708	2,065	423	13,664
$\top\bot^*$	101	84	18	457	54	60	485	144	9	45
χ_{m}	1,011	18,114	1,077	49,792	2,675	4,900	20,885	1,936	384	1,467
χ_{q}	1,064	171,497	1,132	55,343	2,814	5,533	20,654	2,077	423	8,591
χ_{wq}	1,056	32,474	1,153	51,342	2,670	5,170	15,390	2,010	390	1,591
χ_{f}	1,058	179,759	1,122	54,854	2,785	5,477	20,822	2,067	417	8,358
χ_{i}	1,078	34,866	1,144	54,068	2,765	5,368	15,063	2,108	427	8,376

	00463	00471	00477	00512	00545	00774	00775	00778	00786
$\bot$	138	549	1,172	23	342	5	10	841	698
χ_{c}	5,441	13,291	51,707	20,712	31,616	768	1,064	34,980	44,463
$\top\bot^*$	192	793	1,768	37	576	10	20	1,210	1,070
χ_{m}	1,615	3,202	6,073	2,188	2,504	640	967	20,409	41,283
χ_{q}	1,431	2,669	3,191	2,939	591	750	1,046	34,330	43,785
χ_{wq}	1,332	2,638	3,157	2,756	567	715	1,028	20,293	43,007
χ_{f}	1,391	2,640	3,157	2,964	569	748	1,047	33,565	44,604
χ_{i}	1,382	2,664	3,223	2,891	562	767	1,048	34,774	44,168

Table 6.5: Extraction times in milliseconds for random signatures

significantly (e.g. 00786). Similarly, χ_{q} modules and χ_{f} modules have similar size in some cases (e.g. 00350) but not in others (e.g. 00471), and the same happens with χ_{q} and χ_{wq} (exemplified by ontologies 00775 and 00354), and with χ_{f} and χ_{i} (see ontologies 00512 and 00004). These observations suggest that our modules faithfully reflect the differences between the inseparability relations we considered, and that they could offer significant advantages for practical applications

Tables 6.4 and 6.5 provide the average module extraction time (in milliseconds) for genuine and random signatures, respectively. The extraction of our modules is consistently slower than that of locality-based modules; however, the average extraction time rarely exceeds 1 minute, and is very often below 10 seconds (especially for genuine signatures). This suggests that our modules are feasible for practical applications. Furthermore, since most of the extraction time is invariably spent by the datalog reasoner, future advancements in the area of datalog reasoning can lead to further performance gains for systems implementing our technique.

Part III

Modular Reasoning

Chapter 7

Modular Classification of Ontologies

This chapter presents a modular approach to ontology classification where module extraction techniques are exploited to split the workload between a general-purpose reasoner and a reasoner specific for a lightweight logic $\mathcal{L}$ (or $\mathcal{L}$-*reasoner*).

As mentioned in Chapter 3, the decision problems associated with classification have a high worst-case complexity for very expressive ontology languages. In contrast, they can be decided in polynomial time for lightweight DLs such as those in the $\mathcal{EL}$ family. Many existing ontologies, however, do not fall within any lightweight description logic. Nevertheless, such ontologies often contain only a relatively small number of axioms that are outside one of these lightweight logics. For example, out of the 211,369 axioms in the Foundational Model of Anatomy (FMA) ontology, only 107 are not in $\mathcal{EL}^{++}$. Unfortunately, even if an ontology $\mathcal{O}$ contains only one axiom outside a logic $\mathcal{L}$, an $\mathcal{L}$-reasoner is not guaranteed to completely classify $\mathcal{O}$; furthermore, each axiom outside $\mathcal{L}$ could have a huge effect on the ontology's subsumption hierarchy.

While reasoners for lightweight logics can typically classify ontologies in *one-pass*, classification algorithms for expressive logics usually need to deal with many sub-

sumption tests individually. As mentioned in Chapter 3, many optimisations in general-purpose reasoners are oriented at reducing the total number of subsumption tests. Using a suitable $\mathcal{L}$-reasoner to efficiently decide most subsumption pairs for a given ontology can significantly reduce the number of subsumption tests that a general-purpose reasoner needs to perform, and thus lead to a significant speed up in classification time.

This chapter is structured as follows:

- In Section 7.1 we provide a high level intuition about how the workload of ontology classification can be split between two reasoners.
- In Section 7.2 we explain how the division of the classification workload can be performed effectively using module extraction techniques and also techniques for approximating the subsumption hierarchy of an ontology.
- In Sections 7.3 and 7.4 we provide more specific details about how the proposed technique can be realised using two concrete module extraction techniques; we consider one syntactic locality-based technique, and also one based on the framework introduced in Chapter 5.
- In Section 7.5 we explain how the technique proposed can be generalised to divide the classification workload between more than two reasoners.
- Finally, Section 7.6 contains an overview of other approaches to ontology classification based on exploiting algorithms for restricted logics in the classification of ontologies that do not fall within those logics.

7.1 Overview

This section provides a general overview of our approach for dividing the workload of ontology classification between a general-purpose reasoner and a reasoner specific

for a more restricted logic in order to better exploit the particular computational capabilities of each one of them.

In what follows, let $\mathcal{O}$ be an arbitrary but fixed ontology, $\mathcal{L}$ an arbitrary but fixed (probably lightweight) logic, $\mathcal{O}_\mathcal{L}$ the set of rules in $\mathcal{O}$ that fall within $\mathcal{L}$, R a general purpose reasoner and $\mathsf{R}_\mathcal{L}$ a reasoner specific for $\mathcal{L}$.

The main ideas behind the approach we propose to compute the subsumption hierarchy of $\mathcal{O}$ using R and $\mathsf{R}_\mathcal{L}$ in combination are captured by the following steps:

1. Find a subset $P_\mathcal{L} \subseteq \mathsf{possSub}(\mathcal{O})$ of potential subsumption pairs that can be decided by looking only at $\mathcal{O}_\mathcal{L}$. More formally, find $P_\mathcal{L} \subseteq \mathsf{possSub}(\mathcal{O})$ satisfying the property (♣) below.

$$\mathcal{O} \models A \to B \text{ iff } \mathcal{O}_\mathcal{L} \models A \to B \text{ for each } (A, B) \in P_\mathcal{L} \qquad (\clubsuit)$$

2. Find a subset $\mathcal{M} \subseteq \mathcal{O}$ such that the remaining pairs in $\mathsf{possSub}(\mathcal{O}) \setminus P_\mathcal{L}$ can be decided by looking only at $\mathcal{M}$. That is, find $\mathcal{M} \subseteq \mathcal{O}$ satisfying the property (♠) below.

$$\mathcal{O} \models A \to B \text{ iff } \mathcal{M} \models A \to B \text{ for each } (A, B) \in \mathsf{possSub}(\mathcal{O}) \setminus P_\mathcal{L} \qquad (\spadesuit)$$

3. Use $\mathsf{R}_\mathcal{L}$ to compute $\mathsf{hierarchy}(\mathcal{O}_\mathcal{L})$ and R to compute $\mathsf{hierarchy}(\mathcal{M})$.

We refer to such a set as $P_\mathcal{L}$ as a set of $\mathcal{L}$-*pairs* for $\mathcal{O}$.

Conditions (♣) and (♠) are trivially satisfied when $P_\mathcal{L} = \mathsf{hierarchy}(\mathcal{O}_\mathcal{L})$ and $\mathcal{M} = \mathcal{O}$. This choice of $P_\mathcal{L}$ and $\mathcal{M}$, however, does not lead to any decrease in the workload assigned to R. The following example shows that it is sometimes possible to choose $P_\mathcal{L}$ and $\mathcal{M}$ in a more advantageous way.

Example 78. Consider $\mathcal{O}^{ex}$ from Figure 4.1 and $\mathcal{L}$ = datalog. The largest datalog

program contained in $\mathcal{O}^{ex}$ is $\mathcal{O}^{ex}_{\mathcal{L}} = \{r_2{-}r_5, r_8, r_9\}$ and they satisfy

$$\mathsf{hierarchy}(\mathcal{O}^{ex}) = \{(E,C),(D,H),(F,H),(G,H)\}$$
$$\mathsf{hierarchy}(\mathcal{O}^{ex}_{\mathcal{L}}) = \{(E,C),(G,H)\}$$

The sets $P_{\mathcal{L}} = \mathsf{possSub}(\mathcal{O}^{ex})\backslash\{(D,H),(F,H)\}$ and $\mathcal{M} = \{r_6{-}r_9\}$ clearly fulfil conditions (♣) and (♠). ◇

The following proposition shows that our approach is correct in the sense that it leads to a sound and complete classification of $\mathcal{O}$.

Proposition 79. *Let $P_{\mathcal{L}} \subseteq \mathsf{possSub}(\mathcal{O})$ and $\mathcal{M} \subseteq \mathcal{O}$ satisfy conditions (♣) and (♠). Then, $\mathsf{hierarchy}(\mathcal{O}) = \mathsf{hierarchy}(\mathcal{O}_{\mathcal{L}}) \cup \mathsf{hierarchy}(\mathcal{M})$.*

Proof. Since $\mathcal{O}_{\mathcal{L}} \subseteq \mathcal{O}$ and $\mathcal{M} \subseteq \mathcal{O}$ by hypothesis, by monotonicity of first-order logic it follows that $\mathsf{hierarchy}(\mathcal{O}_{\mathcal{L}}) \cup \mathsf{hierarchy}(\mathcal{M}) \subseteq \mathsf{hierarchy}(\mathcal{O})$.

Now consider $(A,B) \in \mathsf{hierarchy}(\mathcal{O})$. By definition $\mathsf{hierarchy}(\mathcal{O}) \subseteq \mathsf{possSub}(\mathcal{O})$, so it must be either $(A,B) \in P_{\mathcal{L}}$ or $(A,B) \in \mathsf{possSub}(\mathcal{O})\backslash P_{\mathcal{L}}$. If $(A,B) \in P_{\mathcal{L}}$ then, by (♣), it follows that $(A,B) \in \mathsf{hierarchy}(\mathcal{O}_{\mathcal{L}})$. If, on the other hand, it is $(A,B) \in \mathsf{possSub}(\mathcal{O})\backslash P_{\mathcal{L}}$, then by (♠) we have that $(A,B) \in \mathsf{hierarchy}(\mathcal{M})$. □

This approach is motivated by the fact that general purpose reasoners are in some cases not efficient enough at performing ontology classification. It is therefore natural to think that the key to obtaining an improvement in performance with this approach is in finding a suitable $\mathcal{M}$ that is as small as possible. Note that $\mathcal{M}$ directly depends on $P_{\mathcal{L}}$; intuitively, a larger set $P_{\mathcal{L}}$ (with a smaller complementary set $\mathsf{possSub}(\mathcal{O})\backslash P_{\mathcal{L}}$) should lead to the existence of a smaller suitable subset $\mathcal{M}$. There are, thus, two challenges to make this approach useful in practice: the first one is maximising the size of $P_{\mathcal{L}}$ and the second one is, once $P_{\mathcal{L}}$ is fixed, minimising that of $\mathcal{M}$.

7.2 Effectively Dividing the Workload

In this section we explain how module extraction techniques, as well as techniques for efficiently approximating the subsumption hierarchy of $\mathcal{O}$, can be used to make the approach introduced in Section 7.1 useful in practice.

The conditions (♣) and (♠) characterising $P_{\mathcal{L}}$ and $\mathcal{M}$ are reminiscent of inseparability relations, introduced in Chapter 4. These conditions, however, are given in terms of a set of pairs from $\mathsf{possSub}(\mathcal{O})$ rather than a signature. We next introduce a natural generalisation of the notion of implication inseparability that captures precisely the conditions that we are considering.

Definition 80. Given a set of pairs $P \subseteq \mathsf{possSub}(\mathcal{O})$, we say that $\mathcal{O}$ and $\mathcal{O}'$ are *P-implication inseparable* ($\mathcal{O} \equiv^{\mathsf{i}}_{P} \mathcal{O}'$) if, for each $(A, B) \in P$, we have $\mathcal{O} \models A \to B$ iff $\mathcal{O}' \models A \to B$. Furthermore, if $\mathcal{M} \subseteq \mathcal{O}$ satisfies $\mathcal{O} \equiv^{\mathsf{i}}_{P} \mathcal{M}$ we say that $\mathcal{M}$ is a *$\equiv^{\mathsf{i}}_{P}$-module* of $\mathcal{O}$. ◇

We refer to this new family of relations between ontologies as *generalised implication inseparability*. It is easy to see that the conditions (♣) and (♠) given in Section 7.1 to characterise $P_{\mathcal{L}}$ and $\mathcal{M}$ can be equivalently reformulated in terms of generalised implication inseparability: (♣) is equivalent to $\mathcal{O} \equiv^{\mathsf{i}}_{P_{\mathcal{L}}} \mathcal{O}_{\mathcal{L}}$, and (♠) is equivalent to $\mathcal{O} \equiv^{\mathsf{i}}_{\mathsf{possSub}(\mathcal{O}) \setminus P_{\mathcal{L}}} \mathcal{M}$. Hence, the problem of finding $\mathcal{M}$ can be seen as a module extraction problem.

Modularity also provides a sufficient condition to identify a set $P_{\mathcal{L}}$ of $\mathcal{L}$-pairs: if $P_{\mathcal{L}}$ is such that there exists a $\equiv^{\mathsf{i}}_{P_{\mathcal{L}}}$-module $\mathcal{M}'$ of $\mathcal{O}$ satisfying $\mathcal{M}' \subseteq \mathcal{O}_{\mathcal{L}}$, then it is guaranteed that $\mathcal{O} \equiv^{\mathsf{i}}_{P_{\mathcal{L}}} \mathcal{O}_{\mathcal{L}}$. In the presence of a module extraction technique that produces modules for generalised implication inseparability, this suggests a simple algorithm for computing a (maximal) suitable $P_{\mathcal{L}}$: consider all subsets of $\mathsf{possSub}(\mathcal{O})$ in decreasing size order and, for each of them, check whether the corresponding module falls within $\mathcal{L}$. This could, however, be quite costly. Since the ultimate goal

is to optimise ontology classification, in practice a more goal directed algorithm is desirable, even if it does not compute optimal solutions.

Example 81. Consider again $\mathcal{O}^{ex}$ and $P_{\mathcal{L}} = \mathsf{possSub}(\mathcal{O}^{ex}) \backslash \{(D, H), (F, H)\}$ (with $\mathcal{L} =$ datalog) from Example 78. The subset $\mathcal{M}' = \{r_5, r_9\}$ of $\mathcal{O}^{ex}_{\mathcal{L}}$ is a $\equiv^{\mathsf{i}}_{P_{\mathcal{L}}}$-module of $\mathcal{O}^{ex}$. ◇

Another way to identify a suitable $P_{\mathcal{L}}$ is via finding a complete (but potentially unsound) approximation, or *overapproximation*, of $\mathsf{hierarchy}(\mathcal{O})$, i.e., a superset P_o of $\mathsf{hierarchy}(\mathcal{O})$. Together with $\mathsf{hierarchy}(\mathcal{O}_{\mathcal{L}})$, which is a sound (but potentially incomplete) approximation, or *underapproximation*, of $\mathsf{hierarchy}(\mathcal{O})$, the overapproximation P_o leads to the set of pairs $\mathsf{hierarchy}(\mathcal{O}_{\mathcal{L}}) \cup (\mathsf{possSub}(\mathcal{O}) \backslash P_o)$, which is indeed a set of $\mathcal{L}$-pairs: for each $(A, B) \in \mathsf{hierarchy}(\mathcal{O}_{\mathcal{L}})$ it is $\mathcal{O}_{\mathcal{L}} \models A \to B$ as well as $\mathcal{O} \models A \to B$, hence it holds that $\mathcal{O}_{\mathcal{L}} \models A \to B$ iff $\mathcal{O} \models A \to B$; moreover, for each $(A, B) \in (\mathsf{possSub}(\mathcal{O}) \backslash P_o)$ it is $\mathcal{O} \not\models A \to B$ and therefore also $\mathcal{O}_{\mathcal{L}} \not\models A \to B$, and again it holds that $\mathcal{O}_{\mathcal{L}} \models A \to B$ iff $\mathcal{O} \models A \to B$.

Example 82. Consider once more $\mathcal{O}^{ex}$ and $\mathcal{L} =$ datalog and let $\mathcal{O}'$ consist of the set of rules from Figure 5.1 (Section 5.1). The ontology $\mathcal{O}'$ is a datalog strengthening of $\mathcal{O}$ and it satisfies $\mathsf{hierarchy}(\mathcal{O}') = \mathsf{hierarchy}(\mathcal{O}^{ex}) \cup \{(D, F), (D, G)\}$. The set

$$\begin{aligned} &\mathsf{hierarchy}(\mathcal{O}^{ex}_{\mathcal{L}}) \cup (\mathsf{possSub}(\mathcal{O}^{ex}) \backslash \mathsf{hierarchy}(\mathcal{O}')) \\ &= \mathsf{possSub}(\mathcal{O}^{ex}) \backslash \{(D, H), (F, H), (D, F), (D, G)\} \end{aligned}$$

is a set of $\mathcal{L}$-pairs. ◇

In the following two sections we show how to realise the proposed approach to ontology classification with the help of two different module extraction techniques that produce modules for generalised implication inseparability.

7.3 Using Syntactic Locality Modules

This section shows how the approach presented in Section 7.1 can be used in practice with the help of syntactic locality modules, and more specifically of $\bot$-modules.

Since the task at hand is optimising the classification of ontologies that are written in expressive ontology languages, syntactic locality modules are natural candidates due to their extremely efficient extraction algorithms and their applicability to any ontology falling within the description logic $\mathcal{SROIQ}$. We focus in particular on $\bot$-modules, which enjoy the particularly convenient property of being modules for classification inseparability, as mentioned in Section 5.4, and have already been successfully used for classification-related tasks in the past [89, 18].

The following proposition establishes a relation between classification inseparability and generalised implication inseparability:

Proposition 83. *Let* $\Sigma \subseteq \mathsf{Sig}(\mathcal{O})$ *and* $P \subseteq \mathsf{possSub}(\mathcal{O}) \cap \Sigma \times \mathsf{Sig}(\mathcal{O})$. *Let* $\mathcal{O}'$ *be an ontology such that* $\mathcal{O} \equiv^{\mathsf{c}}_{\Sigma} \mathcal{O}'$. *Then* $\mathcal{O} \equiv^{\mathsf{i}}_{P} \mathcal{O}'$.

Proof. Let $(A, B) \in P \cap (\mathsf{possSub}(\mathcal{O}) \cup \mathsf{possSub}(\mathcal{O}'))$. By Definition 47, from $\mathcal{O} \equiv^{\mathsf{c}}_{\Sigma} \mathcal{O}'$ it follows that $\mathcal{O} \models A \to B$ iff $\mathcal{O}' \models A \to B$ and hence $\mathcal{O} \equiv^{\mathsf{i}}_{P} \mathcal{O}'$. □

As a direct consequence of Proposition 83, given a set $P_{\mathcal{L}}$ of $\mathcal{L}$-pairs, the signature $\Sigma = \{\, A \mid (A, B) \in \mathsf{possSub}(\mathcal{O}) \backslash P_{\mathcal{L}} \text{ for some } B \,\}$ is such that the subset $\mathcal{M}^{\bot}_{[\mathcal{O},\Sigma]}$ is $\mathsf{possSub}(\mathcal{O}) \backslash P_{\mathcal{L}}$-implication inseparable from $\mathcal{O}$, and thus fulfils the requirements for $\mathcal{M}$ given in the previous section.

The properties of $\bot$-modules also suggest that, to find a set of $\mathcal{L}$-pairs, it suffices to find a signature $\Sigma_{\mathcal{L}} \subseteq \mathsf{Sig}(\mathcal{O})$ such that $\mathcal{M}^{\bot}_{[\mathcal{O},\Sigma_{\mathcal{L}}]} \subseteq \mathcal{O}_{\mathcal{L}}$.

Definition 84. Given a signature $\Sigma \subseteq \mathsf{Sig}(\mathcal{O})$, we say that Σ is an $\mathcal{L}$-*signature* for $\mathcal{O}$ w.r.t. $\bot$-modules (or, simply, an $\mathcal{L}$-signature) if $\mathcal{M}^{\bot}_{[\mathcal{O},\Sigma]} \subseteq \mathcal{O}_{\mathcal{L}}$. ◇

Every $\mathcal{L}$-signature $\Sigma_{\mathcal{L}}$ induces a set $P_{\Sigma_{\mathcal{L}}} = \Sigma_{\mathcal{L}} \times \mathsf{Sig}(\mathcal{O})$ which is a set of $\mathcal{L}$-pairs: by Proposition 83 it is $\mathcal{M}^{\bot}_{[\mathcal{O},\Sigma_{\mathcal{L}}]} \equiv^{\mathsf{i}}_{P_{\Sigma_{\mathcal{L}}}} \mathcal{O}$ and, since $\mathcal{M}^{\bot}_{[\mathcal{O},\Sigma_{\mathcal{L}}]} \subseteq \mathcal{O}_{\mathcal{L}} \subseteq \mathcal{O}$, it clearly

follows that $\mathcal{O}_{\mathcal{L}} \equiv^{\mathsf{i}}_{P_{\Sigma_{\mathcal{L}}}} \mathcal{O}$. The remainder of this section deals with the non-trivial task of finding an $\mathcal{L}$-*signature* for $\mathcal{O}$.

7.3.1 Finding an $\mathcal{L}$-signature

We begin by pointing out that every $\mathcal{L}$-signature $\Sigma_{\mathcal{L}}$ *must* satisfy the property ($\star$) below. If ($\star$) does not hold, then $\mathcal{M}^{\perp}_{[\mathcal{O},\Sigma_{\mathcal{L}}]}$ contains some non-$\mathcal{L}$ rule.

$$\mathcal{O} \setminus \mathcal{O}_{\mathcal{L}} \text{ is } \perp\text{-local w.r.t. } \Sigma_{\mathcal{L}} \qquad (\star)$$

Example 85. Consider again our example ontology $\mathcal{O}^{ex}$ and $\mathcal{L}$ = datalog. As already mentioned, the only non-$\mathcal{L}$ rules in $\mathcal{O}^{ex}$ are r_1, r_6 and r_7. One might think that the signature of $\mathcal{O}^{ex}_{\mathcal{L}}$ is a $\mathcal{L}$-signature, which would make the computation of a maximal $\mathcal{L}$-signature trivial. This is, however, not the case. The signature of $\mathcal{O}^{ex}_{\mathcal{L}}$, namely $\mathsf{Sig}(\mathcal{O}^{ex}_{\mathcal{L}}) = \mathsf{Sig}(\mathcal{O}^{ex}) \setminus \{F\}$, is not an $\mathcal{L}$-signature for $\mathcal{O}^{ex}$. Indeed, r_1 and r_6 are not $\perp$-local w.r.t. $\mathsf{Sig}(\mathcal{O}^{ex}_{\mathcal{L}})$. In contrast, r_1, r_6 and r_7 are all $\perp$-local w.r.t. $\Sigma = \{C, G, R\}$. Furthermore, $\mathcal{M}^{\perp}_{[\mathcal{O}^{ex},\Sigma]} = \{r_4, r_9\} \subseteq \mathcal{O}^{ex}_{\mathcal{L}}$ and thus Σ is an $\mathcal{L}$-signature for $\mathcal{O}^{ex}$. $\diamond$

Although Example 85 might suggest that property ($\star$) is also a *sufficient* condition for $\Sigma_{\mathcal{L}}$ to be an $\mathcal{L}$-signature in $\mathcal{O}$, this is unfortunately not the case, as illustrated by the following example.

Example 86. Consider now signature $\Sigma' = \{B, C, G\}$ instead. Clearly, r_1, r_6 and r_7 are also $\perp$-local w.r.t. Σ'. However, Σ' is not an $\mathcal{L}$-signature for $\mathcal{O}^{ex}$ since the module $\mathcal{M}^{\perp}_{[\mathcal{O}^{ex},\Sigma']} = \{r_3, r_6{-}r_9\}$ contains r_6 and r_7. One way to address this problem would be to reduce Σ' to $\Sigma' \setminus \{B\}$. The corresponding $\perp$-module only contains r_9, and therefore such reduced signature is an $\mathcal{L}$-signature for $\mathcal{O}^{ex}$. $\diamond$

Example 86 suggests a strategy for computing an $\mathcal{L}$-signature for $\mathcal{O}$ which is summarised by the following steps:

1. Reduce $\Sigma_0 = \mathsf{Sig}(\mathcal{O})$ to a subset $\Sigma_1 \subseteq \Sigma_0$ such that $\tilde{\mathcal{O}}_0 = \mathcal{O} \setminus \mathcal{O}_{\mathcal{L}}$ is $\perp$-local w.r.t. Σ_1 (thus satisfying ($\star$)).

2. Compute the set $\tilde{\mathcal{O}}_1$ of rules in $\mathcal{M}^{\perp}_{[\mathcal{O},\Sigma_1]}$ that mention symbols outside Σ_1.

3. Reduce Σ_1 to a subset $\Sigma_2 \subseteq \Sigma_1$ such that $\tilde{\mathcal{O}}_1$ is $\perp$-local w.r.t. Σ_2.

4. Repeat steps [2-3] to find subsequent sets $\tilde{\mathcal{O}}_i$ and Σ_i until an empty $\tilde{\mathcal{O}}_i$ is found.

Since $\mathcal{O}$ is finite, the sequence $\{\tilde{\mathcal{O}}_i\}_{i\geq 0}$ is guaranteed to converge to the empty set, and thus $\{\Sigma_i\}_{i\geq 0}$ must also converge to a fix-point that is an $\mathcal{L}$-signature.[1] Before further formalising the proposed technique and proving its correctness, the following example shows how its application would work in practice.

Example 87. Consider once more our example ontology $\mathcal{O}^{ex}$ and $\mathcal{L} =$ datalog. As already mentioned, $\mathcal{O}^{ex}_{\mathcal{L}} = \mathcal{O}^{ex}\backslash\{r_1, r_6, r_7\}$, so we start with

$$\Sigma_0 = \mathsf{Sig}(\mathcal{O}^{ex}) \qquad\qquad \tilde{\mathcal{O}}_0 = \{r_1, r_6, r_7\}$$

In order to make all rules in $\tilde{\mathcal{O}}_0$ $\perp$-local we must remove A, D and F from Σ_0:

$$\Sigma_1 = \Sigma_0 \setminus \{A, D, F\}$$

Next, we obtain $\mathcal{M}^{\perp}_{[\mathcal{O}^{ex},\Sigma_1]} = \mathcal{O}^{ex}\backslash\{r_1\}$, therefore

$$\tilde{\mathcal{O}}_1 = \{r_3, r_6, r_7\}$$

Note that r_3 is not $\perp$-local w.r.t. Σ_1. This implies $r_3 \in \mathcal{M}^{\perp}_{[\mathcal{O}^{ex},\Sigma_1]}$ and consequently $D \in \mathsf{Sig}(\mathcal{M}^{\perp}_{[\mathcal{O}^{ex},\Sigma_1]})$, which results in r_6 not being $\perp$-local w.r.t. $\Sigma_1 \cup \mathsf{Sig}(\mathcal{M}^{\perp}_{[\mathcal{O}^{ex},\Sigma_1]})$. Similarly, $r_6 \in \mathcal{M}^{\perp}_{[\mathcal{O}^{ex},\Sigma_1]}$ leads to $F \in \mathsf{Sig}(\mathcal{M}^{\perp}_{[\mathcal{O}^{ex},\Sigma_1]})$, resulting in $r_7 \in \mathcal{M}^{\perp}_{[\mathcal{O}^{ex},\Sigma_1]}$ too. Therefore, as long as r_3 is in the module, so will r_6 and r_7. Hence, we need to make

[1] For simplicity we do not consider here the possibility that a suitable reduction of Σ_1 might not exist; this case is duly addressed when the technique is properly formalised.

r_3 $\bot$-local. One way to do this is to take

$$\Sigma_2 = \Sigma_1 \setminus \{B\}$$

Now $\mathcal{M}^{\bot}_{[\mathcal{O}^{ex},\Sigma_2]} = \{r_2, r_4, r_5, r_8, r_9\}$, which satisfies $\mathsf{Sig}(\mathcal{M}^{\bot}_{[\mathcal{O}^{ex},\Sigma_2]}) \subseteq \Sigma_2$. Hence we have $\tilde{\mathcal{O}}_2 = \emptyset$. Furthermore, $\mathcal{M}^{\bot}_{[\mathcal{O}^{ex},\Sigma_2]} \subseteq \mathcal{O}^{ex}_{\mathcal{L}}$ and thus Σ_2 is an $\mathcal{L}$-signature for $\mathcal{O}^{ex}$. $\diamond$

Note that there can be many ways to perform the signature reduction required in Steps 1 and 3. In Example 87, for instance, we could also have taken $\Sigma_2 = \Sigma_1 \setminus \{C\}$, or even any subset thereof. The following example illustrates how different choices can lead to very different $\mathcal{L}$-signatures.

Example 88. As already mentioned, in Example 87 we could have alternatively chosen to take $\Sigma_2 = \Sigma_1 \setminus \{C\}$. This would have lead to $\mathcal{M}^{\bot}_{[\mathcal{O}^{ex},\Sigma_2]} = \mathcal{O}^{ex} \setminus \{r_1\}$ again, and to $\tilde{\mathcal{O}}_3 = \tilde{\mathcal{O}}_2 \cup \{r_5\}$. We could now take $\Sigma_3 = \Sigma_2 \setminus \{E\}$, and we would finally have $\tilde{\mathcal{O}}_4 = \emptyset$ and $\mathcal{M}^{\bot}_{[\mathcal{O}^{ex},\Sigma_3]} \subseteq \mathcal{O}^{ex}_{\mathcal{L}}$. The $\mathcal{L}$-signature Σ_3 obtained in this case is smaller, and hence less appealing, than the one obtained in Example 87. $\diamond$

The signature reductions shown all satisfy certain properties that make them "acceptable", but making reasonable choices requires good heuristics. We postpone the discussion of such heuristics to Section 7.3.2 and for now limit ourselves to providing a characterisation of acceptable reductions.

Definition 89. Given an ontology $\mathcal{O}$, a *signature reduction* is a function

$$\mathsf{reduce} : 2^{\mathsf{Sig}(\mathcal{O})} \times 2^{\mathcal{O}} \to 2^{\mathsf{Sig}(\mathcal{O})}$$

Algorithm 2 Extraction of an $\mathcal{L}$-signature

Input: $\mathcal{O}$ an ontology
Output: $\Sigma_{\mathcal{L}}$ an $\mathcal{L}$-signature for $\mathcal{O}$

1: $\Sigma_{\mathcal{L}} := \mathsf{Sig}(\mathcal{O})$
2: $\tilde{\mathcal{O}} := \mathcal{O} \setminus \mathcal{O}_{\mathcal{L}}$
3: **while** $\tilde{\mathcal{O}} \neq \emptyset$ and $\Sigma_{\mathcal{L}} \neq \emptyset$ **do**
4: $\quad \Sigma_{\mathcal{L}} := \mathsf{reduce}(\Sigma_{\mathcal{L}}, \tilde{\mathcal{O}})$
5: $\quad$ **if** $\Sigma_{\mathcal{L}} \neq \emptyset$ **then**
6: $\quad\quad \tilde{\mathcal{O}} := \{\, r \in \mathcal{M}^{\perp}_{[\mathcal{O},\Sigma_{\mathcal{L}}]} \mid \mathsf{Sig}(r) \not\subseteq \Sigma_{\mathcal{L}} \,\}$
7: **return** $\Sigma_{\mathcal{L}}$

such that, for $\Sigma \subseteq \mathsf{Sig}(\mathcal{O})$ and $\tilde{\mathcal{O}} \subseteq \mathcal{O}$ not $\perp$-local w.r.t. Σ, $\mathsf{reduce}(\Sigma, \tilde{\mathcal{O}})$ satisfies

$$\mathsf{reduce}(\Sigma, \tilde{\mathcal{O}}) = \begin{cases} \Sigma & \text{if } \tilde{\mathcal{O}} = \emptyset \\ \Sigma' \subset \Sigma \text{ s.t. } \tilde{\mathcal{O}} \text{ is } \perp\text{-local w.r.t. } \Sigma' & \text{if } \tilde{\mathcal{O}} \neq \emptyset \text{ and } \Sigma' \text{ exists} \\ \emptyset & \text{otherwise} \end{cases} \quad \diamond$$

Cases 1 and 3 correspond to the cases where $\tilde{\mathcal{O}} = \emptyset$ or there is no satisfactory way of reducing Σ, respectively. Case 2 constitutes the essence of the reduction: to compute a strict subset of the signature that makes the given set of rules $\perp$-local.

Given a particular signature reduction reduce, Algorithm 2 can be used to extract an $\mathcal{L}$-signature of $\mathcal{O}$. It is easy to see that, given a reduce function that can be computed in time polynomial in the size of its arguments, it is possible to compute an $\mathcal{L}$-signature for an ontology $\mathcal{O}$ in polynomial time. The following theorem proves the correctness of Algorithm 2.

Theorem 90. *Let $\mathcal{O}$ be an ontology and reduce a signature reduction function. Furthermore, let the sequences $\{\tilde{\mathcal{O}}_i\}_{i\geq 0}$ and $\{\Sigma_i\}_{i\geq 0}$ be defined as follows:*

$$\Sigma_0 = \mathsf{Sig}(\mathcal{O}) \qquad \tilde{\mathcal{O}}_0 = \mathcal{O} \setminus \mathcal{O}_{\mathcal{L}}$$
$$\Sigma_i = \mathsf{reduce}(\Sigma_{i-1}, \tilde{\mathcal{O}}_{i-1}) \qquad \tilde{\mathcal{O}}_i = \{r \in \mathcal{M}_{[\mathcal{O},\Sigma_i]} \mid \mathsf{Sig}(r) \not\subseteq \Sigma_i\} \qquad \text{for } i \geq 1$$

Then there exists $k < |\mathsf{Sig}(\mathcal{O})|$ such that either $\Sigma_k = \emptyset$ or Σ_k is an $\mathcal{L}$-signature.

Proof. Suppose $\Sigma_i \neq \emptyset$ for each $i \geq 0$. A straightforward inductive argument would show that $\Sigma_j \subseteq \Sigma_i$ for each $j > i \geq 0$. Furthermore, $\Sigma_0 = \mathsf{Sig}(\mathcal{O})$, so it cannot be the case that $\Sigma_j \subsetneq \Sigma_i$ for each $0 \leq i < j \leq |\mathsf{Sig}(\mathcal{O})|$. It follows that there must be some $k < |\mathsf{Sig}(\mathcal{O})|$ such that $\Sigma_{k+1} = \Sigma_k$. By the definition of reduce, this implies that $\mathsf{Sig}(\mathcal{M}^{\perp}_{[\mathcal{O},\Sigma_k]}) \subseteq \Sigma_k$ and $\tilde{\mathcal{O}}_k = \emptyset$. Therefore, to show that $\mathcal{M}^{\perp}_{[\mathcal{O},\Sigma_k]} \subseteq \mathcal{O}_{\mathcal{L}}$ it suffices to prove that each $r \in \mathcal{O} \backslash \mathcal{O}_{\mathcal{L}}$ is $\perp$-local w.r.t. Σ_k. Because $\Sigma_j \subseteq \Sigma_i$ for each $j > i \geq 0$ and, by hypothesis, $\Sigma_k \neq \emptyset$, in particular it must be $\Sigma_0 \neq \emptyset$. By definition of reduce, either $\mathcal{O} \setminus \mathcal{O}_{\mathcal{L}} = \emptyset$, in which case it is immediate that $\mathcal{M}^{\perp}_{[\mathcal{O},\Sigma_{\mathcal{L}}]} \subseteq \mathcal{O}_{\mathcal{L}}$, or every rule in $\tilde{\mathcal{O}}_0 = \mathcal{O} \setminus \mathcal{O}_{\mathcal{L}}$ is $\perp$-local w.r.t. $\Sigma_1 = \mathsf{reduce}(\Sigma_0, \tilde{\mathcal{O}}_0)$. By definition of $\perp$-locality, it clearly follows that each $r \in \mathcal{O} \setminus \mathcal{O}_{\mathcal{L}}$ must also be $\perp$-local w.r.t. $\Sigma_k \subseteq \Sigma_1$. □

7.3.2 Heuristics and Optimisations

The construction in Theorem 90 is inherently greedy and depends to a great extent on the choice of the reduce function, as we saw in Example 88. As mentioned earlier, good heuristics are necessary to guide our choice of reduce.

A reasonable heuristic when dealing with DL ontologies is to favour the retention of binary predicates (atomic roles) in the $\mathcal{L}$-signature since, in practice, most DL ontologies contain fewer binary predicates than unary predicates. Furthermore, each binary predicate usually appears in more rules than any unary predicate, therefore removing a binary predicate from Σ_i is likely to lead to a greater number of rules in the next $\tilde{\mathcal{O}}_{i+1}$, and hence eventually to a smaller $\mathcal{L}$-signature.

Another sensible heuristic is to base the choice of reduce on the set Γ of *global predicates* in $\mathcal{O}$, i.e., predicates that are necessarily in the signature of any $\perp$-module of $\mathcal{O}$. This set can be identified in time polynomial in the size of $\mathcal{O}$ by

(i) starting from $\Gamma = \mathsf{Sig}(\mathcal{G})$, where $\mathcal{G}$ is the set of all rules in $\mathcal{O}$ that cannot be made $\perp$-local, (e.g. rules of the form $\top(x) \rightarrow A(x)$), and *(ii)* exhaustively extending Γ with the signatures of all the rules in $\mathcal{O}$ that are not $\perp$-local w.r.t. Γ (and thus

w.r.t. any superset of Γ). Knowing Γ, we can restrict our choice of reduce to signature reduction functions that only return supersets of Γ (or the empty set). The following example shows how this can sometimes mark the difference between finding a non-empty $\mathcal{L}$-signature or not.

Example 91. Consider the ontology $\mathcal{O}' = \mathcal{O}^{ex} \cup \{\top(x) \to B(x)\}$ and $\mathcal{L} =$ datalog. Ontologies $\mathcal{O}^{ex}$ and $\mathcal{O}'$ share the same non-$\mathcal{L}$ rules, and the first steps towards computing an $\mathcal{L}$-signature are similar for both: we start with $\Sigma_0 = \mathsf{Sig}(\mathcal{O}')$ and $\tilde{\mathcal{O}}_0 = \{r_1, r_6, r_7\}$, and from them we can obtain $\Sigma_1 = \Sigma_0 \setminus \{A, D, F\}$, which leads to $\mathcal{M}^{\perp}_{[\mathcal{O}', \Sigma_1]} = \mathcal{O}' \backslash \{r_1\}$, and $\tilde{\mathcal{O}}_1 = \{r_3, r_6, r_7\}$. Similarly to the case of $\mathcal{O}^{ex}$ in Examples 88 and 87, we could now obtain Σ_2 by removing from Σ_1 either one of B or C. Even though B was a better option for $\mathcal{O}^{ex}$, the extra rule in $\mathcal{O}'$ makes this no longer be the case: taking $\Sigma_2 = \Sigma_1 \setminus \{B\}$ leads to the module $\mathcal{M}^{\perp}_{[\mathcal{O}', \Sigma_2]} = \{r_2, r_4, r_5, r_8, r_9, \top(x) \to B(x)\}$ and hence to $\tilde{\mathcal{O}}_2 = \{\top(x) \to B(x)\}$. As shown in Table 4.1, $\top(x) \to B(x)$ cannot be made local, so it must next inevitably be $\Sigma_3 = \emptyset$. It can be readily checked, however, that taking $\Sigma_2 = \Sigma_1 \setminus \{C\}$ still leads to a non-empty $\mathcal{L}$-signature in the case of $\mathcal{O}'$.

The set Γ of global predicates in $\mathcal{O}$ can help us make the right choice in this case. The set $\mathcal{G}$ of global rules in $\mathcal{O}'$ contains only $\top(x) \to B(x)$, therefore we start by taking $\Gamma = \{\top, B\}$; since all rules in $\mathcal{O}'$ are $\perp$-local w.r.t. Γ, there is no need to extend it further. Clearly, the condition $\Gamma \subseteq \Sigma_2$ immediately rules out the possibility of taking $\Sigma_2 = \Sigma_1 \setminus \{B\}$, as desired. $\diamond$

It is possible to find a reduce function that is computable in polynomial time and incorporates the heuristics above. It is easy to see from Table 4.1 that, for each rule corresponding to a normalised $\mathcal{SROIQ}$ axiom, one can identify in linear time all the subset-minimal sets of predicates whose removal from a signature would make the rule $\perp$-local w.r.t. the aforementioned signature. It is also possible to discard the undesirable solutions: those that contain predicates from Γ, and, as long as there

are other options, those that contain binary predicates. Then, $\mathsf{reduce}(\Sigma_i, \tilde{\mathcal{O}}_i)$ can be computed in polynomial time by *(i)* collecting all the acceptable solutions for each rule in $\tilde{\mathcal{O}}_i$, and *(ii)* starting from $\mathsf{reduce}(\Sigma_i, \tilde{\mathcal{O}}_i) = \emptyset$, considering each rule $r \in \tilde{\mathcal{O}}_i$ in turn and extending $\mathsf{reduce}(\Sigma_i, \tilde{\mathcal{O}}_i)$ with the solution for r that increments its size the least. Since the strategy described is a greedy one, the resulting reduce function is not optimal for each iteration, but it is computable in polynomial time.

Finally, it is also possible to further refine the given strategy to find an $\mathcal{L}$-signature by reducing the dependence on the particular choice of reduce. For this it suffices to consider the following complementary subsets of $\tilde{\mathcal{O}}_i$:

- $\tilde{\mathcal{O}}_i^{>1}$, containing those rules in $\tilde{\mathcal{O}}_i$ for which there are multiple (subset-minimal) ways in which Σ_i can be reduced to make them $\bot$-local, and
- $\tilde{\mathcal{O}}_i^{1}$, containing those for which there is at most one (subset-minimal) way in which Σ_i can be reduced to make them $\bot$-local.

Now, in each iteration, the subsequent signature Σ_{i+1} should be obtained by applying reduce only to $\tilde{\mathcal{O}}_i^1$ and Σ_i whenever $\tilde{\mathcal{O}}_i^1 \neq \emptyset$, and to $\tilde{\mathcal{O}}_i^{>1}$ and Σ_i in any other case. As a consequence of this optimisation we are no longer guaranteed to make all rules outside $\mathcal{L}$ $\bot$-local w.r.t. Σ_1, therefore it is necessary to also include in each $\tilde{\mathcal{O}}_i$ any non-$\mathcal{L}$ rules in $\mathcal{M}^{\bot}_{[\mathcal{O},\Sigma_i]}$. Note that, thanks to the simplicity of the conditions for $\bot$-locality, it is easy to ensure that the chosen reduce function provides the optimal reduction for $\tilde{\mathcal{O}}_i^1$ and Σ_i when $\tilde{\mathcal{O}}_i^1$ is not empty. Modifying the strategy in this way reduces the dependency on the way that the chosen reduce function deals with the cases with several subset-minimal solutions, as shown in the following example.

Example 92. Consider $\mathcal{L}$ = datalog and the ontology $\mathcal{O}''$ consisting of the following

rules:

$$\begin{aligned} s_4: \quad A(x) \wedge B(x) &\rightarrow \exists y[R(x,y) \wedge C(y)] \\ s_5: \quad A(x) &\rightarrow D(x) \\ s_6: \quad D(x) &\rightarrow E(x) \vee F(x) \end{aligned}$$

Since $\mathcal{O}'_{\mathcal{L}} = \{s_5\}$, we start with $\Sigma_0 = \mathsf{Sig}(\mathcal{O}')$ and $\tilde{\mathcal{O}}_0 = \{s_4, s_6\}$. There are two minimal sets whose removal from Σ_0 would make s_4 $\bot$-local: $\{A\}$ and $\{B\}$; on the other hand, there is only one such set for s_6, namely $\{D\}$. Therefore, $\tilde{\mathcal{O}}_0$ breaks into $\tilde{\mathcal{O}}_0^1 = \{s_6\}$ and $\tilde{\mathcal{O}}_0^{>1} = \{s_4\}$, and we get $\Sigma_1 = \Sigma_0 \setminus \{D\}$. Next, we obtain $\mathcal{M}^{\bot}_{[\mathcal{O}'', \Sigma_1]} = \mathcal{O}''$ and $\tilde{\mathcal{O}}_1 = \{s_4, s_5, s_6\}$ (s_4 is not in $\mathcal{L}$, and s_5 and s_6 mention D), which breaks into $\tilde{\mathcal{O}}_1^1 = \{s_6\}$ and $\tilde{\mathcal{O}}_2^{>1} = \{s_4, s_5\}$. Rule s_4 is already $\bot$-local w.r.t. Σ_1, and the single minimal solution for s_5 is $\{A\}$, therefore we get $\Sigma_2 = \Sigma_1 \setminus \{A\}$. This leads to $\mathcal{M}^{\bot}_{[\mathcal{O}', \Sigma_2]} = \emptyset$ and thus, thanks to the optimisation, we have found an optimal $\mathcal{L}$-signature (Σ_2) for $\mathcal{O}'$ in this case. $\diamond$

7.4 Using PrisM Modules

In this section we present another possible realisation of the approach introduced in Section 7.1, this time exploiting the module extraction framework proposed in Chapter 5.

In the previous section we saw that there is a close relation between classification inseparability and generalised implication inseparability. By definition, there is clearly also a close relation between implication inseparability and generalised implication inseparability: for each signature $\Sigma \subseteq \mathsf{Sig}(\mathcal{O})$ and each $P \subseteq \mathsf{possSub}(\mathcal{O}) \cap \Sigma \times \Sigma$, if $\mathcal{O} \equiv^{\mathsf{i}}_{\Sigma} \mathcal{O}'$ then $\mathcal{O} \equiv^{\mathsf{i}}_{P} \mathcal{O}'$.

Example 93. Consider $\mathcal{O} = \mathcal{O}^{ex}$ and the set $P = \{(D, E), (E, F), (F, D)\}$, as well as the signature $\Sigma = \{D, E, F\}$. It can be readily checked that in this case $\mathcal{M}^{\chi_{\mathsf{i}}} = \{r_6\}$ and $\mathcal{M}^{\chi_{\mathsf{c}}} = \{r_6 - r_9\}$. However, $\emptyset$ is already P-implication inseparable from $\mathcal{O}^{ex}$. $\diamond$

Even though the module settings χ_{i} and χ_{c} from Sections 5.3 and 5.4 can be used to obtain modules for generalised implication inseparability, the framework from Chapter 5 can also be exploited to extract modules specifically tailored for this purpose. Given a set $P \subseteq \mathsf{possSub}(\mathcal{O})$ of pairs, consider the signature $\Sigma = \{ A \mid (A, B) \in P \}$. To obtain a module setting that leads to a $\equiv^{\mathsf{i}}_P$-module of $\mathcal{O}$ it suffices to modify the module setting χ_{c} for Σ-classification inseparability defined in Section 5.4.1 by restricting $\mathcal{D}_r$ according to the set P, as specified next.

Definition 94. Let $\mathcal{O}$ be an ontology. Let $P \subseteq \mathsf{possSub}(\mathcal{O})$. For each existentially quantified variable y in $\mathcal{O}$, let c_y be a fresh constant. Furthermore, for each $A \in \mathsf{Sig}(\mathcal{O})$ of arity n, let $\mathbf{c}_A = (c^1_A, \ldots, c^n_A)$ be an array of fresh constants. The module setting $\chi^{\mathsf{i}_g} = \langle \theta^{\mathsf{i}_g}, \mathcal{D}^{\mathsf{i}_g}_0, \mathcal{D}^{\mathsf{i}_g}_r \rangle$ is defined as follows:

- $\theta^{\mathsf{i}_g} = \{ y \mapsto c_y \mid y \text{ existentially quantified in } \mathcal{O} \} \cup \{ c \mapsto c \mid c \in \mathsf{Ct}(\mathcal{O}) \}$,
- $\mathcal{D}^{\mathsf{i}_g}_0 = \{ A(\mathbf{c}_A) \mid (A, B) \in P \}$; and
- $\mathcal{D}^{\mathsf{i}_g}_r = \{ B(\mathbf{c}_A) \mid (A, B) \in P,\ A \text{ and } B \text{ of the same arity} \} \cup \{\bot\}$. ◇

Example 95. Again for $\mathcal{O} = \mathcal{O}^{ex}$ and $P = \{(D, E), (E, F), (F, D)\}$ it is easy to check that $\mathcal{M}^{\chi^{\mathsf{i}_g}} = \emptyset$ is in this case the smallest $\equiv^{\mathsf{i}}_P$-module of $\mathcal{O}^{ex}$. ◇

An argument similar to that in the proof of Theorem 27 shows that the module setting χ^{i_g} indeed captures generalised implication inseparability.

Theorem 96. *Let $P \subseteq \mathsf{possSub}(\mathcal{O})$. Then $\mathcal{M}^{\chi_{\mathsf{i}}} \equiv^{\mathsf{i}}_P \mathcal{O}$.*

We naturally generalise the notion of a module setting family to take as one of its arguments a set of pairs $P \subseteq \mathsf{possSub}(\mathcal{O})$ rather than a signature, and consider the i-acceptable, uniform family Ψ^{i_g} induced by χ^{i_g} from Definition 94.

An approach analogous to the one in Section 7.3 would consist in computing a set of pairs P satisfying $\mathcal{M}^{\Psi^{\mathsf{i}_g}(\mathcal{O}, P)} \subseteq \mathcal{O}_{\mathcal{L}}$. We conclude this section by showing that,

if $\mathcal{L}$ = datalog, such an approach can be easily overtaken by one based on under- and overapproximating $\mathsf{hierarchy}(\mathcal{O})$ with the help of $\mathcal{O}_\mathcal{L}$ and a strengthening $\mathcal{O}_s$ of $\mathcal{O}$, as explained at the end of Section 7.2. Indeed, as we prove next, the datalog strengthening $\mathcal{P}^{\Psi^{\mathsf{ig}}(\mathcal{O},\mathsf{possSub}(\mathcal{O}))}$ of $\mathcal{O}$ leads to a set of $\mathcal{L}$-pairs that contains any set of pairs P with $\mathcal{M}^{\Psi^{\mathsf{ig}}(\mathcal{O},P)} \subseteq \mathcal{O}_\mathcal{L}$.

Proposition 97. *Let $\mathcal{L}$ = datalog and $P \subseteq \mathsf{possSub}(\mathcal{O})$ satisfying $\mathcal{M}^{\Psi^{\mathsf{ig}}(\mathcal{O},P)} \subseteq \mathcal{O}_\mathcal{L}$. Then there exists a strengthening $\mathcal{O}_s$ of $\mathcal{O}$ in $\mathcal{L}$ such that*

$$P \subseteq \mathsf{hierarchy}(\mathcal{O}_\mathcal{L}) \cup (\mathsf{possSub}(\mathcal{O}) \backslash \mathsf{hierarchy}(\mathcal{O}_s))$$

Proof. The ontology $\mathcal{O}_s = \mathcal{P}^{\Psi^{\mathsf{ig}}(\mathcal{O},\mathsf{possSub}(\mathcal{O}))}$ is clearly a datalog strengthening of $\mathcal{O}$. Also, since by assumption the only rule in $\mathcal{O}$ with an empty head is $\bot \to \Box$, it is $\Xi^{\Psi^{\mathsf{ig}}(\mathcal{O},P_\mathcal{L})}(\mathcal{O}_\mathcal{L}) = \mathcal{O}_\mathcal{L}$.

Let $(A, B) \in P$, it is easy to see that one of the following properties must hold:

(i) $B \neq \bot$ (resp. $B = \bot$) and $\mathsf{supp}(\rho) \subseteq \mathcal{O}_\mathcal{L}$ for each proof ρ in $\mathcal{P}^{\Psi^{\mathsf{ig}}(\mathcal{O},P)} \cup \{A(\mathbf{c}_A)\}$ of $B(\mathbf{c}_A)$ (resp. of $\bot$).

Then it follows that $(A, B) \in \mathsf{hierarchy}(\mathcal{O}_\mathcal{L})$.

(ii) $B \neq \bot$ (resp. $B = \bot$) and $B(\mathbf{c}_A)$ (resp. $\bot$) is not in the materialisation of $\mathcal{P}^{\Psi^{\mathsf{ig}}(\mathcal{O},P)} \cup \{A(\mathbf{c}_A)\}$.

By definition of Ψ^{ig}, it is easy to see that $\mathcal{P}^{\Psi^{\mathsf{ig}}(\mathcal{O},P)} = \mathcal{P}^{\Psi^{\mathsf{ig}}(\mathcal{O},\mathsf{possSub}(\mathcal{O}))}$. It follows that $(A, B) \notin \mathsf{hierarchy}(\mathcal{O}_s)$ and hence $(A, B) \in \mathsf{possSub}(\mathcal{O}) \backslash \mathsf{hierarchy}(\mathcal{O}_s)$. □

As we will explain in more detail in Chapter 8, when $\mathcal{L}$ = datalog the computation of a set $P_\mathcal{L}$ of $\mathcal{L}$-pairs by under- and overapproximation of $\mathsf{hierarchy}(\mathcal{O})$ using $\mathcal{O}_\mathcal{L}$ and $\mathcal{P}^{\Psi^{\mathsf{ig}}(\mathcal{O},\mathsf{possSub}(\mathcal{O}))}$ can be combined with the extraction of $\mathcal{M}^{\Psi^{\mathsf{ig}}(\mathcal{O},\mathsf{possSub}(\mathcal{O})\backslash P_\mathcal{L})}$. Computing $P_\mathcal{L}$ in this way is no harder than extracting $\mathcal{M}^{\Psi^{\mathsf{ig}}(\mathcal{O},\mathsf{possSub}(\mathcal{O})\backslash P_\mathcal{L})}$. Hence, in the cases where $\mathcal{M}^{\Psi^{\mathsf{ig}}(\mathcal{O},\mathsf{possSub}(\mathcal{O})\backslash P_\mathcal{L})}$ can be extracted in polynomial time (see

Algorithm 3 Modular Classification with $\mathsf{R}_{\mathcal{L}_1}, \ldots, \mathsf{R}_{\mathcal{L}_n}$ and R

Input: $\mathcal{O}$ an ontology
Output: $\mathcal{H} = \mathsf{hierarchy}(\mathcal{O})$

1: $\mathcal{H} := \emptyset$
2: $P := \mathsf{possSub}(\mathcal{O})$
3: $\mathcal{M} := \mathcal{O}$
4: **for** $i = 1$ to n **do**
5: $\quad \mathcal{H} := \mathcal{H} \cup \mathsf{hierarchy}(\mathcal{M}_{\mathcal{L}_i})$ ▷ computed with $\mathsf{R}_{\mathcal{L}_i}$
6: $\quad P := P \setminus \mathsf{find}\mathcal{L}_i\mathsf{pairs}(\mathcal{M})$
7: $\quad \mathcal{M} := \mathsf{extractModule}_i(\mathcal{M}, P)$
8: $\mathcal{H} := \mathcal{H} \cup \mathsf{hierarchy}(\mathcal{M})$ ▷ computed with R
9: **return** $\mathcal{H}$

Section 5.7), it is possible to obtain a suitable $P_{\mathcal{L}}$ in this way in polynomial time as well. It is therefore also unlikely that a strategy analogous to the one in Section 7.3 could be more efficient in practice.

7.5 Exploiting Several Efficient Reasoners

In this section we explain how the technique for modular classification proposed in this chapter can be generalised to exploit several reasoners for restricted logics.

It is easy to see how, once we have found $P_{\mathcal{L}}$ and $\mathcal{M}$ satisfying the established requirements, instead of passing $\mathcal{M}$ on to R directly, we could repeat the process for another restricted logic $\mathcal{L}'$ and a reasoner $\mathsf{R}_{\mathcal{L}'}$ for $\mathcal{L}'$. More formally, consider a general purpose reasoner R and a finite set of reasoners $\mathsf{R}_{\mathcal{L}_1}, \ldots, \mathsf{R}_{\mathcal{L}_n}$ for the logics $\mathcal{L}_1, \ldots, \mathcal{L}_n$, respectively. Additionally, for each $i = 1, \ldots, n$, let $\mathsf{find}\mathcal{L}_i\mathsf{pairs}$ and $\mathsf{extractModule}_i$ be functions such that

- $\mathsf{find}\mathcal{L}_i\mathsf{pairs}(\mathcal{O})$ is a set of $\mathcal{L}_i$-pairs for $\mathcal{O}$, and
- given a suitable set P of pairs, $\mathsf{extractModule}_i(\mathcal{O}, P)$ is a $\equiv^{\mathsf{i}_g}_P$-module of $\mathcal{O}$.

Algorithm 3 generalises our modular classification technique to combine $\mathsf{R}_{\mathcal{L}_1}, \ldots, \mathsf{R}_{\mathcal{L}_n}$ and R. Its correctness follows trivially from Proposition 79.

7.6 Related Work

The extraction of $\bot$-modules has previously been exploited to improve the performance of incremental classification [89, 18], as well to optimise the extraction of justifications [90, 37]. Tsarkov and Palmisano [94] also used $\bot$-modules to perform modular reasoning. Their technique, implemented in the Chainsaw[2] reasoner, consists in, first, precomputing the atomic decomposition [26] of the input ontology, and then, extracting dedicated modules for particular queries and delegating the problem to the "best suited" reasoner in each case, depending on the expressivity of the module obtained. Modules are extracted only for certain kinds of queries, namely for "entailment" queries, either concerned with checking the entailment of a particular axiom, or with finding all the subsumers or subsumees of some atomic concept. For the task of computing the whole subsumption hierarchy of the ontology, however, the full ontology is given to a reasoner that can handle its expressivity.

Song et al. recently developed two techniques aimed at exploiting an $\mathcal{ALCH}$ reasoner (such as ConDor [80]) to completely classify ontologies beyond this logic. The first of these techniques [83] deals with classifying an $\mathcal{ALCHO}$ ontology $\mathcal{O}$; it consists in computing the subsumption hierarchies entailed by a weakening and a strengthening of $\mathcal{O}$, both in $\mathcal{ALCH}$, and then resolving any uncertain subsumptions with a general purpose reasoner (using no module extraction techniques to support this phase). An intermediate step tries to avoid the need to consider the strengthening of $\mathcal{O}$ by attempting to turn a canonical model of the weakening into a model of $\mathcal{O}$ that witnesses the lack of any additional entailed subsumptions. The second such technique [84] consists in rewriting an $\mathcal{ALCHI}(D)$ ontology into one in $\mathcal{ALCH}$ that provably entails the same subsumption hierarchy.

The work by Zhou et al. [101] on query answering also follows a weakening-strengthening approach where the uncertain answers are checked with a general pur-

[2] `http://sourceforge.net/projects/chainsaw/`

pose reasoner. In this case the approximated ontologies are processed with the help of a datalog engine.

Finally, Steigmiller et al. [86] recently devised an ontology classification technique where consequence-based reasoning for DLs in the $\mathcal{EL}$ family is tightly coupled with individual subsumption tests. In contrast with our approach, this coupling takes place within a single reasoner rather than as a result of combining several reasoners as black boxes.

Chapter 8

The MORe Reasoner

8.1 System Description

We have implemented the modular approach to classification proposed in Chapter 7 in the MORe system.[1] MORe is written in Java and it is available under an academic license. Built on top of the OWLAPI, the system implements its OWLReasoner interface. MORe integrates

- the reasoner ELK (v0.4.2)[2], which supports a large fragment of OWL 2 EL;
- the datalog reasoner RDFox[3]; and
- an OWL 2 reasoner, which is set by default to be HermiT (v1.3.8)[4], but can be set as any other OWL 2 reasoner that implements the OWLReasoner interface.

In addition to these systems, which are all used as black boxes, our system also integrates a modified version of the PrisM module extractor, whose architecture is described in detail in Chapter 6.

[1] http://www.cs.ox.ac.uk/isg/tools/MORe/
[2] https://code.google.com/p/elk-reasoner/
[3] http://www.cs.ox.ac.uk/isg/tools/RDFox/
[4] http://hermit-reasoner.com/

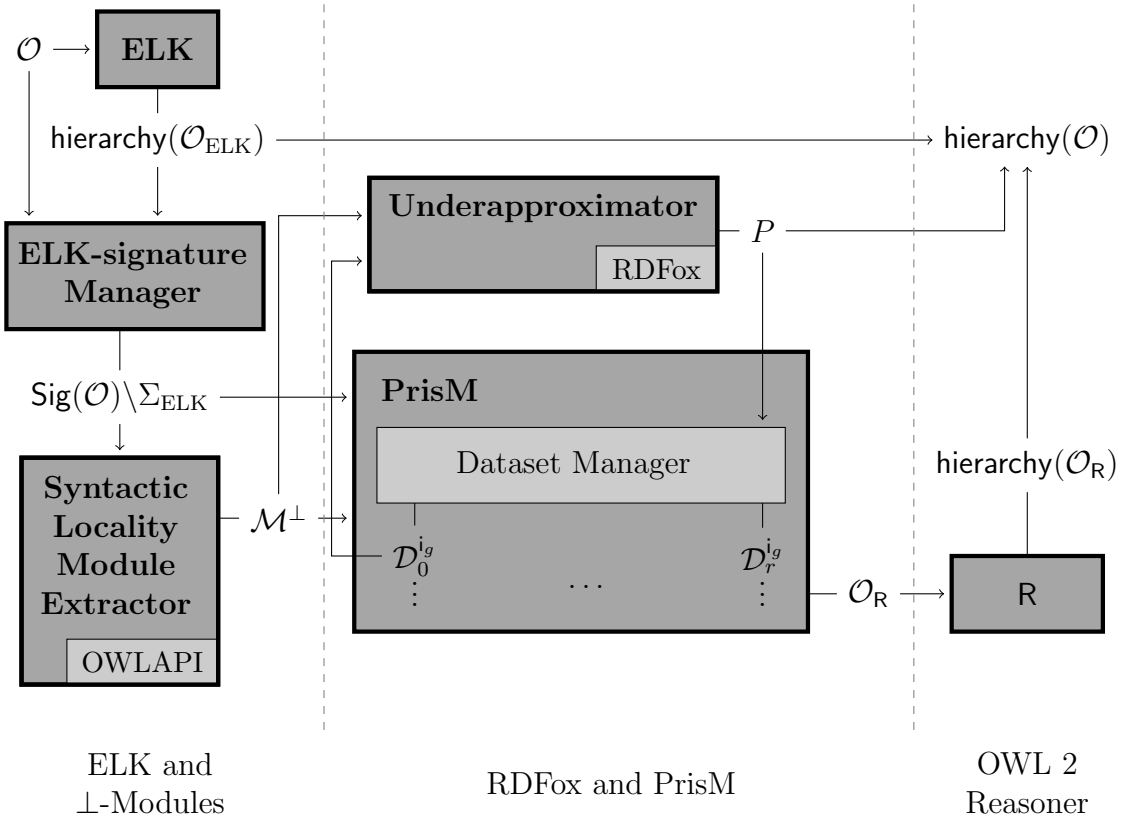


Figure 8.1: Workflow in MORe

MORe accepts as input arbitrary OWL 2 ontologies. It supports classification of the "terminological" part of the input ontology $\mathcal{O}$, i.e., the maximal subset of $\mathcal{O}$ that corresponds to a set of rules that are not equivalent to (possibly negated) facts. It also supports consistency checking (checking whether there exists a model of $\mathcal{O}$ with a non-empty domain), atomic concept satisfiability checking, and entailment checking for axioms of the form $A \sqsubseteq B$ with A and B atomic concepts.

The architecture of MORe is represented in Figure 8.1. Each box in the diagram represents a component of MORe, and any external systems used within a component are indicated in its lower right corner. The workflow of MORe can be divided in three stages: a stage that exploits ELK with the help of ⊥-modules, one where RDFox is exploited with the help of PrisM, and a final stage where an OWL 2 reasoner is used to finish the work and ensure completeness of the final outcome. We next further

describe each stage, as well as the corresponding components.

Stage 1: ELK and $\bot$-Modules. This stage is based on Section 7.3. The input ontology $\mathcal{O}$, preloaded with the OWLAPI, is preprocessed in order to encode away property range axioms (which are in OWL 2 EL but not accepted by ELK). This is done by incorporating the range information into existential restrictions; e.g., axioms of the form $A \sqsubseteq \exists R.B$ get rewritten into $A \sqsubseteq \exists R.(B \sqcap C)$ when the range of R is C. For simplicity, we identify with $\mathcal{O}$ the resulting ontology. ELK then computes $\mathsf{hierarchy}(\mathcal{O}_{\mathrm{ELK}})$,[5] and the ELK-Signature Manager takes care of identifying an ELK-signature Σ_{ELK} as explained in Sections 7.3.1 and 7.3.2. Any concepts already identified as unsatisfiable by ELK are added to Σ_{ELK}. Then its complement signature $\mathsf{Sig}(\mathcal{O})\backslash\Sigma_{\mathrm{ELK}}$ is passed on to the SyntacticLocalityModuleExtractor in the OWLAPI, together with $\mathcal{O}$, to obtain their corresponding $\bot$-module, $\mathcal{M}^{\bot}_{[\mathcal{O},\mathsf{Sig}(\mathcal{O})\backslash\Sigma_{\mathrm{ELK}}]}$, which we denote as $\mathcal{M}^{\bot}$ from here on for simplicity.

Stage 2: RDFox and PrisM. This stage is based on Sections 7.4 and 7.5. The PrisM component is a modified version of the system described in Section 6.1, adapted to extract modules for generalised implication inseparability (as defined in Section 7.2) using the module setting χ^{i_g} from Section 7.4.[6] Let S denote the set of pairs that remain to be decided at this point. The role of the Underapproximator component in this stage is to compute a subset $P \subseteq \mathsf{hierarchy}(\mathcal{M}^{\bot}) \cap S$, and the role of PrisM is to compute an ontology $\mathcal{O}_{\mathsf{R}}$ that is $S\backslash P$-implication inseparable from $\mathcal{M}^{\bot}$. The two components perform their designated tasks in close collaboration as follows:

- PrisM goes through its materialisation stage considering S (which is fully determined by $\mathcal{M}^{\bot}$ and $\mathsf{Sig}(\mathcal{O})\backslash\Sigma_{\mathrm{EL}}$) as its input set of pairs. Note that PrisM naturally narrows down the set of pairs considered to $S \cap \mathcal{P}^{\chi^{\mathsf{i}_g}}(\mathcal{D}_0^{\mathsf{i}_g})$, which, by Proposition 1 in Section 2.1, can be seen as computing an overapproximation of $\mathsf{hierarchy}(\mathcal{M}^{\bot}) \cap S$.

[5] We identify ELK with the fragment of OWL 2 EL that it can handle.

[6] For the sake of simplicity we identify χ^{i_g} with its corresponding module setting family.

- The Underapproximator computes a datalog weakening $\mathcal{O}_w$ of $\mathcal{M}^{\bot}$ by rewriting as datalog rules those axioms in $\mathcal{M}^{\bot}$ for which this is possible. Borrowing the set $\mathcal{D}_0^{\mathsf{i}_g}$ generated by PrisM, it then computes $P = \mathsf{hierarchy}(\mathcal{O}_w) \cap S$ using an independent instance of RDFox (again exploiting Proposition 1).

- Once P has been computed, PrisM performs its last stage considering $S \backslash P$ rather than S, and finishes the extraction of $\mathcal{O}_{\mathsf{R}}$. The fact that $S \backslash P \subseteq S$ guarantees that the outcome ontology $\mathcal{O}_{\mathsf{R}}$ is the same as if PrisM had considered $S \backslash P$ from the beginning.

Stage 3: OWL 2 Reasoner. In this stage, the ontology $\mathcal{O}_{\mathsf{R}}$ produced by PrisM is given to the chosen OWL 2 reasoner R to classify, and the union of $\mathsf{hierarchy}(\mathcal{O}_{\mathrm{ELK}})$ (computed by ELK), P (computed by RDFox within the Underapproximator), and $\mathsf{hierarchy}(\mathcal{O}_{\mathsf{R}})$ (computed by R) is taken as the whole subsumption hierarchy of $\mathcal{O}$. The separate sets of pairs are not integrated in a unique data structure, but rather combined in real time when subsumption queries are posed to the reasoner.

The second stage of MORe currently only supports OWL 2 ontologies whose translation into rules does not use the equality ($\approx$) predicate, which corresponds to $\mathcal{SROI}(D)$ ontologies where nominals are only used in positive existential restrictions. This is not an intrinsic limitation of the technique and full support for equality is already under development. MORe can be configured to skip the second stage and take $\mathcal{O}_{\mathsf{R}}$ as the $\bot$-module of $\mathcal{O}$ extracted in the first stage. We refer to this configuration as MORe_1 and depict its workflow in Figure 8.2. For symmetry, from here on we refer to the configuration that goes through all three stages as MORe_2.

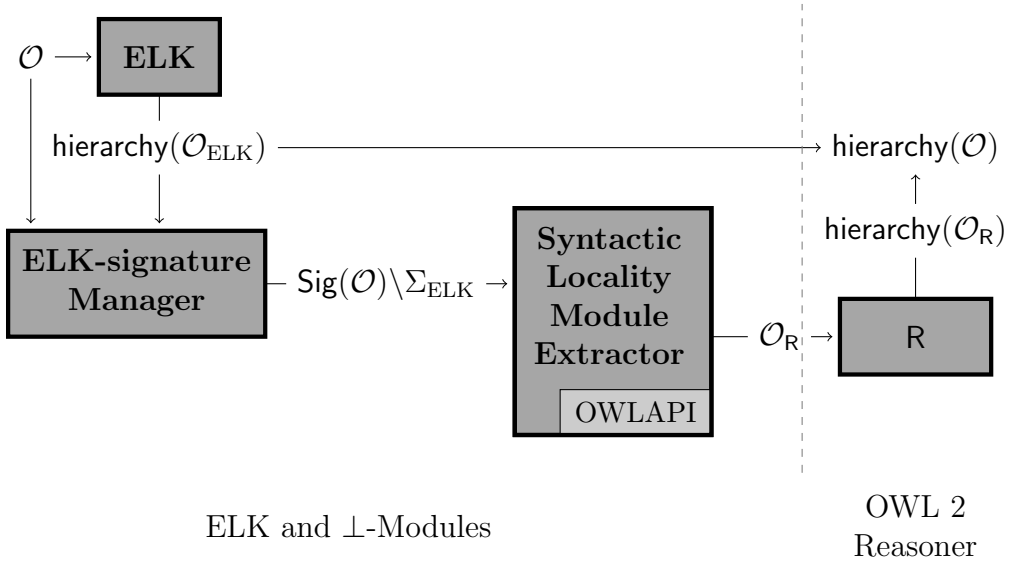


Figure 8.2: Workflow in MORe_1

8.2 Evaluation

We have evaluated MORe on the same set of ontologies used for the evaluation of PrisM in Chapter 6 (identified in [33] as non-trivial for standard reasoning), as well as two additional ontologies that are considerably challenging for most state-of-the-art OWL 2 reasoners. Since MORe only takes into account the terminological part of ontologies and MORe_2 does not currently support equality, we work with filtered versions of our test ontologies, where all axioms violating these restrictions have been removed. Details on the resulting filtered ontologies are given in Table 8.1.[7] The first column in the table indicates the ontology ID in the Oxford Ontology Repository,[8] when applicable. The second column indicates the name of the ontology. The third and fourth columns provide the number of axioms and atomic concepts in the resulting filtered ontology. Finally, the last column indicates the DL expressivity of the filtered ontology, as given by the OWLAPI.[9]

[7]The ontologies used in our experiments are available for download at `https://krr-nas.cs.ox.ac.uk/2015/AnaArmas_thesis/testOntologies_MORe.zip`

[8]`http://www.cs.ox.ac.uk/isg/ontologies/UID/`

[9]Again, we refer the reader to [8] for a detailed account on DL naming conventions.

ID	name	axioms	atomic concepts	expressivity
00001	ACGT-v1.0	5,023	1,752	$\mathcal{SROI}(D)$
00004	BAMS-simplified	18,818	1,112	$\mathcal{SHI}$
00024	DOLCE	1,510	206	$\mathcal{SHI}(D)$
00026	GALEN-no-FIT	36,489	23,138	$\mathcal{ALEH}$
00029	GALEN-doctored	4,586	2,750	$\mathcal{ALEHI}^+$
00032	GALEN-undoctored	4,828	2,750	$\mathcal{ALEHI}^+$
00347	LUBM-one-uni	93	45	$\mathcal{ALEHI}^+(\mathcal{D})$
00350	OBI	9,887	2,635	$\mathcal{SHOI}(D)$
00351	AERO	474	276	$\mathcal{SRI}(D)$
00354	NIF-gross-anatomy	6,597	4,043	$\mathcal{SRI}(D)$
00463	Fly-anatomy-XP	16,040	8,020	$\mathcal{SRI}$
00471	FMA-lite	121,712	78,979	$\mathcal{ALEH}^+$
00477	Gazetteer	167,351	150,978	$\mathcal{ALE}^+$
00512	Lipid	2,192	712	$\mathcal{ALCHI}$
00545	Molecule-role	9,629	9,219	$\mathcal{ALE}^+$
00774	RNA-v0.2	639	245	$\mathcal{SRI}(D)$
00775	Roberts-family	307	60	$\mathcal{SROI}(D)$
00778	SNOMED	54,977	54,975	$\mathcal{SH}$
00786	NCI-v12.04e	130,945	93,415	$\mathcal{SH}(D)$
-	Biomodels-21	439,238	187,516	$\mathcal{SRI}$
-	FMA	84,371	211,369	$\mathcal{ALCHI}$

Table 8.1: Test ontologies

All experiments have been performed on a server with 2 Intel Xeon E5-2670 2.60GHz processors, each of which has 8 physical cores that serve 2 virtual cores each, making a total of 32 virtual cores. Experiments were allocated 90GB of RAM, and, whenever RDFox was used, it was run on 16 threads.

To evaluate MORe we have considered, not only its integrated OWL 2 reasoner, HermiT v1.3.8, but also three additional state-of-the-art OWL 2 reasoners: JFact (v1.2.3),[10] Pellet (v2.3.0)[11] and Konclude (v0.6.1)[12]. For each ontology $\mathcal{O}$ and each OWL 2 reasoner R, we have compared the following:

- the time taken by R to classify $\mathcal{O}$;
- the time taken by MORe_1 to classify $\mathcal{O}$ using R as its OWL 2 reasoner;

[10] `http://sourceforge.net/projects/jfact/`
[11] `https://github.com/complexible/pellet`
[12] `http://www.derivo.de/produkte/konclude.html`

- the time taken by R to classify $\mathcal{O}_{\mathsf{R}}$ within MORe_1;
- the time taken by MORe_2 to classify $\mathcal{O}$ using R as its OWL 2 reasoner; and
- the time taken by R to classify $\mathcal{O}_{\mathsf{R}}$ within MORe_2.

The details of this comparison are given in Tables 8.4-8.7. In order to be able to include in our evaluation Konclude (which does not implement the OWLReasoner interface from the OWLAPI), we modified MORe so that, once $\mathcal{O}_{\mathsf{R}}$ has been computed, it is saved to a file instead of directly passed on to R from within the system. MORe_1 and MORe_2 were therefore run in this way a single time for each ontology, and then each R was run directly from the command line[13] on each of the resulting modules. In all cases, we applied a timeout of 5,000 seconds to completely classify $\mathcal{O}$. In order to better understand the potential of our modular classification technique and, moreover, the results obtained with the OWL 2 reasoners considered, we have also recorded the details of the workload division in MORe_1 and MORe_2 for each ontology, and we present them in Tables 8.2 and 8.3.

8.2.1 Division of the Workload

Table 8.2 contains details about the workload division in MORe_1. The second and third column provide the total number of axioms and the number of axioms accepted by ELK in each ontology, respectively. The fourth column specifies the size of the ontology $\mathcal{O}_{\mathsf{R}}$ that gets assigned to R in MORe_1. The fifth and sixth columns give the percentage of pairs from $\mathsf{possSub}(\mathcal{O})$ that get decided by ELK and those that remain to be decided by R, respectively. Finally, the seventh column indicates the time (in seconds) taken by MORe_1 up to the end of its first stage.

Table 8.3, in turn, contains details about the workload division in MORe_2. The details about axioms and pairs assigned to ELK are omitted since they are already

[13]In the case of JFact we had to implement our own command line interface.

$\mathcal{O}$	$\lvert\mathcal{O}\rvert$	$\lvert\mathcal{O}_{\text{ELK}}\rvert$	$\lvert\mathcal{O}_{\mathsf{R}}\rvert$	%pairs ELK	%pairs R	time stage 1
00001	5,023	4,563	4,556	0.11	99.89	1.69
00004	18,818	17,832	0	100	0	9.98
00024	1,510	929	0	100	0	1.2
00026	36,489	36,489	0	100	0	7.9
00029	4,586	4,379	0	100	0	2.37
00032	4,828	4,621	0	100	0	2.39
00347	93	69	0	100	0	0.76
00350	9,887	9,702	160	99.85	0.15	2.61
00351	474	411	250	78.62	21.38	1.05
00354	6,597	6,331	3,007	84.24	15.76	1.99
00463	16,040	16,030	15,681	9.45	90.55	3.1
00471	121,712	121,712	0	100	0	5.64
00477	167,351	167,351	0	100	0	11.53
00512	2,192	2,026	0	100	0	1.39
00545	9,629	9,629	0	100	0	1.65
00774	639	489	365	0.82	99.18	0.94
00775	307	195	290	11.67	88.33	0.99
00778	54,977	36,654	54,976	0.01	99.99	35.31
00786	130,945	130,766	66,472	70.16	29.84	28.85
Biomodels	439,238	417,136	178,569	78.66	21.34	86.65
FMA	211,369	211,199	151,143	10.72	89.28	31.08

Table 8.2: Workload division with MORe_1

given in Table 8.2; those ontologies for which ELK performs the whole classification are also not included in this table for obvious reasons. The second column of Table 8.3 provides again, for comparison purposes, the total number of axioms in each ontology. The third column indicates the size of the $\bot$-module $\mathcal{M}^{\bot}$ to be processed in the second stage (note that $\mathcal{M}^{\bot}$ here refers to the same otology as $\mathcal{O}_{\mathsf{R}}$ in Table 8.2). The fourth column gives the number of axioms in $\mathcal{M}^{\bot}$ that can be rewritten into equisatisfiable datalog rules. The fifth column contains the size of the ontology $\mathcal{O}_{\mathsf{R}}$ destined for R in this case. The sixth and seventh columns specify the percentage of pairs from $\mathsf{possSub}(\mathcal{O})$ that get decided by RDFox and those that remain to be decided by R, respectively. The eighth column indicates the time (in seconds) taken by MORe_2 up to the end of the second stage.

We can observe that, in the majority of cases where the ontology can mostly be

$\mathcal{O}$	$\lvert\mathcal{O}\rvert$	$\lvert\mathcal{M}^{\bot}\rvert$	$\lvert\mathcal{M}^{\bot}_{\text{datalog}}\rvert$	$\lvert\mathcal{O}_{\mathsf{R}}\rvert$	%pairs RDFox	%pairs R	time stages 1-2
00001	5,023	4,556	4,229	128	99.88	0.01	15.3
00350	9,887	160	132	0	0.15	0	4.34
00351	474	250	195	29	21.35	0.03	2.57
00354	6,597	3,007	1,619	0	15.76	0	20.34
00463	16,040	15,681	6,500	16,937	90.53	0.02	72.06
00774	639	365	308	24	99.11	0.07	3.41
00775	307	290	241	540	84.22	4.11	3.68
00778	54,977	54,976	3,244	n/a	n/a	n/a	timeout
00786	130,945	66,472	28,064	n/a	n/a	n/a	timeout
Biomodels	439,238	178,569	52,187	1,680	21.34	∼0	341.82
FMA	211,369	151,143	89,434	453	89.28	∼0	2,105.18

Table 8.3: Workload division with MORe_2

handled by ELK, it is guaranteed to decide over 70% of the pairs in $\mathsf{possSub}(\mathcal{O})$. There are, however, some such cases where ELK is only guaranteed to decide a very small proportion of pairs: for 00463 and FMA, even though more than 99.9% of the axioms can be handled by ELK, less than 11% of the subsumption pairs are assigned to it. On the other hand, there are cases where, even though there are some axioms in $\mathcal{O}$ that ELK cannot handle, it is still capable of fully classifying $\mathcal{O}$ (e.g. 00004, 00024, 00029, 00032, 00347, 00512). This is due to the encoding of property range axioms mentioned in Section 8.1. The sizes of the ontologies $\mathcal{O}_{\mathsf{R}}$ assigned by MORe_1 to R are generally consistent with the proportion of pairs left to classify after using ELK: the fewer pairs left to decide, the smaller $\mathcal{O}_{\mathsf{R}}$ is in comparison with $\mathcal{O}$; for example, for 00350 only less than 2% of the ontology is necessary to decide 0.15% of the potential subsumption pairs, and for 00351,00354, 00786 and Biomodels taking around half of the ontology is enough to decide between 15% and 30% of the pairs. Furthermore, the time taken by MORe_1 to perform the division of the workload typically stays under half a minute, and even when dealing with an ontology as large as Biomodels ($> 400,000$ axioms) does not exceed 1.5 minutes.

In the case of MORe_2, the under-overapproximation approach turns out to be very effective in assigning most of the remaining pairs to RDFox, even in cases where

less than half of the axioms in $\mathcal{M}^{\perp}$ can be rewritten into equisatisfiable datalog rules, such as 00463 or Biomodels. The size of the ontology $\mathcal{O}_{\mathsf{R}}$ computed by PrisM is therefore typically very small. This is, however, not the case for 000463 and 00775, where $\mathcal{O}_{\mathsf{R}}$ is in fact larger than $\mathcal{M}^{\perp}$; this is because, as mentioned in Section 6.1, the normalisation performed by PrisM is never reverted. Note that this can sometimes be beneficial in this particular context since it may allow to consider only the "relevant part" of some complex axioms. The RDFox/PrisM stage of MORe_2 generally takes longer than the ELK/$\perp$-modules stage, and even times out in a couple of occasions (the majority of the time in this stage is spent in the PrisM module extractor, and in particular in the RDFox instance integrated in it). This supports our decision to perform the ELK/$\perp$-modules stage before the RDFox/PrisM stage: since the processing performed in the latter is significantly more expensive, the overall performance would be affected negatively if it was applied directly to the whole ontology and the whole initial set of potential subsumption pairs.

8.2.2 Classification Times

Table 8.4 presents the time (in seconds) taken to classify each ontology with HermiT (second column); with MORe_1 using HermiT, indicating both the total time and the time taken by HermiT on the assigned module (third and fourth columns, respectively); and with MORe_2 using HermiT, giving again the total time and the time taken by HermiT on the assigned module (fifth and sixth columns, respectively). Tables 8.5, 8.6 and 8.7 provide analogous information for JFact, Pellet and Konclude, respectively. Whenever ELK completely handled the classification, we have not included additional details about MORe_2 (but instead written "-"), since in those cases it behaves just like MORe_1. Furthermore, the word "error" indicates those cases where a runtime error has occurred, and the word "timeout" indicates those where the computation has continued past the allocated $5,000$ seconds.

$\mathcal{O}$	HermiT	MORe_1 + HermiT		MORe_2 + HermiT	
		total	HermiT	total	HermiT
00001	5.02	5.7	4.01	17.58	1.01
00004	403.61	9.98	0	-	-
00024	43.07	1.2	0	-	-
00026	timeout	7.9	0	-	-
00029	5.01	2.37	0	-	-
00032	33.06	2.39	0	-	-
00347	1.01	0.76	0	-	-
00350	8.02	3.62	1.01	5.35	1.01
00351	1.01	2.06	1.01	3.58	1.01
00354	timeout	timeout	timeout	21.35	1.01
00463	timeout	timeout	timeout	79.08	7.02
00471	20.04	5.64	0	-	-
00477	60.1	11.53	0	-	-
00512	21.12	1.39	0	-	-
00545	3.01	1.65	0	-	-
00774	2.01	1.95	1.01	4.42	1.01
00775	timeout	timeout	timeout	timeout	timeout
00778	timeout	timeout	timeout	timeout	n/a
00786	83.13	89.95	61.1	timeout	n/a
Biomodels	748.10	518.33	431.68	346.83	5.01
FMA	timeout	49.12	18.04	2,106.77	1.01

Table 8.4: Classification times in seconds for HermiT alone, with MORe_1 and with MORe_2

We can observe that MORe_1 is very rarely slower than R for R = HermiT, JFact or Pellet. In fact there are numerous cases where MORe_1 improves their classification time by more than an order of magnitude, or helps overcome execution errors: see HermiT on 00004, 00032, 00512 and FMA; JFact on 00004, 00026, 00029, 00032, 00471, 00477 and FMA; or Pellet on 00004, 00024, 00029, 00032, 00350, 00471 and FMA. Furthermore, in the cases where MORe_1 is slower, the overhead is usually no more than a few seconds. For R = Konclude, which already integrates consequence based reasoning in its implementation, and generally shows an outstanding performance, MORe_1 is typically slower, although, again, rarely more than a few seconds. Furthermore, Konclude still substantially benefits from the support of MORe_1 on 00354 and FMA. In most of the aforementioned speed-up cases, the ontology $\mathcal{O}_{\mathsf{R}}$ left

$\mathcal{O}$	JFact	MORe_1 + JFact		MORe_2 + JFact	
		total	JFact	total	JFact
00001	2.14	3.72	2.03	15.93	0.63
00004	256.20	9.98	0	-	-
00024	1.42	1.2	0	-	-
00026	error	7.9	0	-	-
00029	97.84	2.37	0	-	-
00032	error	2.39	0	-	-
00347	0.60	0.76	0	-	-
00350	18.39	3.40	0.79	4.91	0.57
00351	1.03	1.84	0.79	3.13	0.56
00354	error	error	error	20.39	0.5
00463	634.98	618.46	615.45	482.55	410.49
00471	417.36	5.64	0	-	-
00477	timeout	11.53	0	-	-
00512	2.03	1.39	0	-	-
00545	14.07	1.65	0	-	-
00774	1.22	1.68	0.74	4.97	0.56
00775	7.26	7.99	7	error	error
00778	4052.02	3976.14	3945.83	timeout	n/a
00786	566.86	205.66	176.51	timeout	n/a
Biomodels	timeout	timeout	timeout	343.64	1.82
FMA	error	524.06	492.98	2,105.25	0.7

Table 8.5: Classification times in seconds for JFact alone, with MORe_1 and with MORe_2

for R is empty or very small, and hence the speed-up is not surprising; that is not the case for FMA, however, where $\mathcal{O}_\mathsf{R}$ contains almost 3/4 of the axioms in $\mathcal{O}$, and yet the speed-up is of more than an order of magnitude, even for Konclude.

There are a number of cases where MORe_2 substantially improves the classification time of R, and even that of MORe_1 (or helps overcome execution errors). This is the case for R = HermiT, JFact with 00354, 00463 or Biomodels; for R = Pellet with 00351, 00354, 00463 or Biomodels; and for Konclude with 00354. The ontology $\mathcal{O}_\mathsf{R}$ extracted in the case of 00775 does not seem to lead to any improvement for any of the considered reasoners; this is likely due to the unreverted normalisation of axioms performed by PrisM. However, in all other cases where MORe_2 does not timeout, the time taken by R after the division of the workload is smaller than by R alone or

$\mathcal{O}$	Pellet	MORe_1 + Pellet		MORe_2 + Pellet	
		total	Pellet	total	Pellet
00001	4.78	6.26	4.57	17.61	1.04
00004	1895.44	9.98	0	-	-
00024	timeout	1.2	0	-	-
00026	61.01	7.9	0	-	-
00029	error	2.37	0	-	-
00032	timeout	2.39	0	-	-
00347	0.92	0.76	0	-	-
00350	timeout	3.64	1.03	5.21	0.87
00351	error	error	error	3.43	0.86
00354	timeout	timeout	timeout	21.18	0.84
00463	timeout	error	error	3,605.31	3,533.25
00471	99.64	5.64	0	-	-
00477	77.97	11.53	0	-	-
00512	3.93	1.39	0	-	-
00545	4.84	1.65	0	-	-
00774	1.38	2.08	1.14	4.27	0.86
00775	timeout	timeout	timeout	timeout	timeout
00778	timeout	timeout	timeout	timeout	n/a
00786	236.13	252.69	223.84	timeout	n/a
Biomodels	timeout	timeout	timeout	359.91	18.09
FMA	timeout	132.59	101.01	2,106.33	1.25

Table 8.6: Classification times in seconds for Pellet alone, with MORe_1 and with MORe_2

as part of MORe_1. Even in some cases where the overall performance of MORe_2 is not better than that of MORe_1 (with the same R), we can see that the workload of R in MORe_2 has been drastically reduced: see HermiT, JFact and Pellet with FMA and Konclude with Biomodels and FMA. This is hardly surprising considering the size of $\mathcal{O}_{\mathsf{R}}$ in those cases.

These results suggest an interesting way of implementing our technique: making MORe_1 and MORe_2 run in parallel once the ELK/$\bot$-modules stage is over, until one of them finishes. This would avoid the overhead that comes with MORe_2 when R can deal with $\mathcal{M}^{\bot}$ fast enough, while still reaping its benefits in those cases where it really makes a difference. Furthermore, future developments in datalog engines can clearly greatly improve the performance of systems implementing our technique.

$\mathcal{O}$	Konclude	MORe_1 + Konclude		MORe_2 + Konclude	
		total	Konclude	total	Konclude
00001	0.42	2.17	0.48	15.06	0.03
00004	0.70	9.98	0	-	-
00024	0.18	1.2	0	-	-
00026	2.16	7.9	0	-	-
00029	0.26	2.37	0	-	-
00032	0.34	2.39	0	-	-
00347	0.03	0.76	0	-	-
00350	1.15	2.66	0.05	4.36	0.02
00351	0.05	1.09	0.04	2.59	0.02
00354	error	460.99	459	20.37	0.03
00463	1.37	3.99	0.89	72.13	0.7
00471	3.76	5.64	0	-	-
00477	6.74	11.53	0	-	-
00512	0.85	1.39	0	-	-
00545	0.39	1.65	0	-	-
00774	0.08	1.02	0.08	3.44	0.03
00775	0.15	1.25	0.26	6.02	2.34
00778	160.27	194.65	159.34	timeout	n/a
00786	10.17	36.09	7.24	timeout	n/a
Biomodels	20.32	94.18	7.53	341.96	0.14
FMA	561.97	37.75	6.67	2,105.21	0.03

Table 8.7: Classification times in seconds for Konclude alone, with MORe_1 and with MORe_2

It is also worth mentioning that MORe_1 has been independently evaluated on several occasions due to taking part in the ORE workshop competitions, where it has repeatedly received awards in several categories. More specifically:

- it placed second in OWL DL Classification and third in OWL EL Classification in the 2015 edition;[14]

- it placed second in OWL EL Classification and third in OWL EL Consistency and OWL DL Classification in the 2014 edition, which granted it a bronze Kurt Gödel medal at the 1st FLoC Olympic Games;[15] and

- it placed first in OWL RL Satisfiability, second in OWL DL Classification,

[14] `http://dl.kr.org/ore2015/vip.cs.man.ac.uk_8008/results.html`
[15] `http://www.easychair.org/smart-program/VSL2014/ORE-competition.html`

OWL DL Satisfiability and OWL EL satisfiability, and third in OWL EL Classification, as well as winning the Best Newcomer award in the 2013 edition.[16]

[16] http://curation.cs.manchester.ac.uk/ore2013/ore2013.cs.manchester.ac.uk/competition/results/index.html

Part IV

Discussion

Chapter 9

Conclusion

In this chapter we look back at the contributions presented in this thesis, as well as discuss some future work directions suggested by these contributions.

9.1 Summary

There are two main contributions in this thesis:

- In Part II we have presented a novel approach to module extraction, based on a reduction of the problem to datalog reasoning. In contrast with existing techniques, our approach is applicable, not only to description logics, but also to other highly expressive first-order rule formalisms. We have explored how the framework provided by our approach can be instantiated in different ways in order to cater for different inseparability relations studied in the literature. We have been able to successfully adapt our instantiation of the framework in each case to produce smaller modules for more relaxed inseparability relations. Moreover, for most of the inseparability relations considered we have been able to find the optimal instantiation of the framework. We have also shown that our modules satisfy many desirable properties which make them well-suited to

applications such as ontology reuse, debugging, modular ontology development, and reasoning optimisation. Last, but not least, we have shown that our modules can be efficiently computed by reusing off-the-shelf datalog reasoners and our experimental evaluation confirms their suitability in practice. Furthermore, since most of the workload of module extraction with our approach relies on the assisting datalog reasoner, future advances in optimisation of datalog reasoners have the potential to lead to an even better performance of systems implementing our technique.

- In Part III we have presented a novel approach to ontology classification. This approach is based on exploiting module extraction techniques to divide the workload of classifying an ontology $\mathcal{O}$ between a general purpose reasoner and one specifically designed for some restricted logic $\mathcal{L}$. We have devised a criterion to do this division in an advantageous way, and we have formulated the associated problems in terms of modularity of ontologies, reducing part of the problem to module extraction for a suitable (generalised) inseparability relation. We have then explored how $\bot$-modules can be utilised to apply this criterion in practice when $\mathcal{O}$ is a $\mathcal{SROIQ}$ ontology, and we have further argued that the algorithm proposed for this runs in polynomial time. Similarly, we have investigated how the module extraction technique introduced in Part II can be adapted to this task, and we have argued that, when $\mathcal{L} = \mathit{datalog}$, a sensible division of the workload can be done with no increased complexity w.r.t. module extraction by means of computing different approximations to the subsumption hierarchy of $\mathcal{O}$. Finally, we have shown that the proposed technique can lead to improved performance in practice for a collection of state-of-the-art OWL 2 reasoners.

9.2 Future Work

We see several directions for future work, which we outline next.

- As mentioned in Section 7.6, the Chainsaw metareasoner [95] works by building a $\bot$-module-based atomic decomposition [26] of an input ontology $\mathcal{O}$, and then extracting, for each entailment query received, a module of $\mathcal{O}$ to answer it using a suitable delegate reasoner. We believe that the notion of an atomic decomposition can be naturally applied to our modules, and it would be interesting to see if using our framework to extract modules more specifically tailored to each input query could lead to a performance improvement in Chainsaw.

- So far, the use of modules for optimising data reasoning tasks, such as fact entailment and conjunctive query answering, has been rather limited. Indeed, it is well-known that $\bot$-modules are not well-suited for such tasks [21]. PAGOdA is the only reasoning system we know of that exploits techniques akin to module extraction for data reasoning [101]. It would be interesting to investigate how our techniques could be exploited to improve PAGOdA's performance. Furthermore, we also envision potential applications to stream reasoning, where data is frequently changing but queries and ontologies can be seen as fixed.

- It remains to confirm our conjecture that optimal module setting families for fact and query inseparability necessarily incur an exponential blow-up w.r.t. the ones that we have chosen.

- Module extraction has been previously used for incremental classification [89, 18]. We believe that our modules for classification inseparability (Section 5.4.1) could significantly improve the performance of existing techniques, especially in combination with our modular approach to classification.

- The use of $\bot$-modules to exploit debugging and explanation systems for DL

ontologies has proved rather successful [90]. Given that our modules are also justification-preserving, it would be interesting to evaluate their effectiveness for implication inseparability in this setting.

- The use of module extraction techniques has so far been largely constrained to description logics. Our techniques are, however, widely applicable and could be exploited in a number of reasoning tasks for other ontology languages such as datalog$^{\pm}$ and datalog$^{\pm,\vee}$, which are currently gaining significant momentum.

- Finally, the current implementations of both techniques, PrisM and MORe, are prototypical, and there is room for a number of improvements. We plan to modify PrisM so the final result is a subset of the input ontology rather than of its normalisation, as well as add the functionality to extract self-contained modules. We also plan to integrate MORe_1 and MORe_2 in a single configuration where they run in parallel as explained at the end of Section 8.2, and to include in MORe_2 the functionality to deal with ontologies that use equality. Furthermore, we would like to incorporate in both PrisM and MORe some fine grained optimisations from the PAGOdA system that are not currently integrated in our systems.

Bibliography

[1] Serge Abiteboul, Richard Hull, and Victor Vianu. *Foundations of Databases*. Addison-Wesley, 1995.

[2] Mario Alviano, Wolfgang Faber, Nicola Leone, and Marco Manna. Disjunctive datalog with existential quantifiers: Semantics, decidability, and complexity issues. *Theory and Practice of Logic Programming*, 12(4-5):701–718, 2012.

[3] Grigoris Antoniou and Athanasios Kehagias. A note on the refinement of ontologies. *International Journal of Intelligent Systems*, 15(7):623–632, 2000.

[4] Ana Armas Romero, Bernardo Cuenca Grau, and Ian Horrocks. MORe: Modular combination of OWL reasoners for ontology classification. In Philippe Cudré-Mauroux, Jeff Heflin, Evren Sirin, Tania Tudorache, Jérôme Euzenat, Manfred Hauswirth, Josiane Xavier Parreira, Jim Hendler, Guus Schreiber, Abraham Bernstein, and Eva Blomqvist, editors, *Proceedings of the 11th International Semantic Web Conference, Part I*, volume 7649 of *Lecture Notes in Computer Science*, pages 1–16. Springer, 2012.

[5] Ana Armas Romero, Mark Kaminski, Bernardo Cuenca Grau, and Ian Horrocks. Ontology module extraction via datalog reasoning. In Blai Bonet and Sven Koenig, editors, *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, pages 1410–1416. AAAI Press, 2015.

[6] Franz Baader, Meghyn Bienvenu, Carsten Lutz, and Frank Wolter. Query and predicate emptiness in description logics. In Fangzhen Lin, Ulrike Sattler, and Miroslaw Truszczynski, editors, *Proceedings of the 12th International Conference on Principles of Knowledge Representation and Reasoning*. AAAI Press, 2010.

[7] Franz Baader, Sebastian Brandt, and Carsten Lutz. Pushing the $\mathcal{EL}$ envelope. In Leslie Pack Kaelbling and Alessandro Saffiotti, editors, *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, pages 364–369. Professional Book Center, 2005.

[8] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.

[9] Franz Baader, Bernhard Hollunder, Bernhard Nebel, Hans-Jürgen Profitlich, and Enrico Franconi. Am empirical analysis of optimization techniques for terminological representation systems. *Applied Intelligence*, 4(2):109–132, 1994.

[10] Franz Baader, Bernhard Hollunder, Bernhard Nebel, Hans Jüurgen Profitlich, and Enrico Franconi. An empirical analysis of optimization techniques for terminological representation systems. *Applied Intelligence*, 4(2):109–132, 1994.

[11] Leo Bachmair and Harald Ganzinger. Resolution theorem proving. In John Alan Robinson and Andrei Voronkov, editors, *Handbook of Automated Reasoning*, pages 19–99. Elsevier and MIT Press, 2001.

[12] Andrew Bate, Boris Motik, Bernardo Cuenca Grau, František Simančík, and Ian Horrocks. Extending consequence-based reasoning to $\mathcal{SHIQ}$. In Diego Calvanese and Boris Konev, editors, *Proceedings of the 28th International Work-*

shop on Description Logics, volume 1350 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2015.

[13] Elena Botoeva, Roman Kontchakov, Vladislav Ryzhikov, Frank Wolter, and Michael Zakharyaschev. Query inseparability for description logic knowledge bases. In Chitta Baral, Giuseppe De Giacomo, and Thomas Eiter, editors, *Proceedings of the 14th International Conference on Principles of Knowledge Representation and Reasoning*. AAAI Press, 2014.

[14] Pierre Bourhis, Michael Morak, and Andreas Pieris. The impact of disjunction on query answering under guarded-based existential rules. In Francesca Rossi, editor, *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*. IJCAI/AAAI, 2013.

[15] François Bry, Norbert Eisinger, Thomas Eiter, Tim Furche, Georg Gottlob, Clemens Ley, Benedikt Linse, Reinhard Pichler, and Fang Wei. Foundations of rule-based query answering. In Grigoris Antoniou, Uwe Aßmann, Cristina Baroglio, Stefan Decker, Nicola Henze, Paula-Lavinia Patranjan, and Robert Tolksdorf, editors, *Proceedings of the 3rd International Reasoning Web Summer School, Tutorial Lectures*, volume 4636 of *Lecture Notes in Computer Science*, pages 1–153. Springer, 2007.

[16] Andrea Calì, Georg Gottlob, Thomas Lukasiewicz, Bruno Marnette, and Andreas Pieris. Datalog+/-: A family of logical knowledge representation and query languages for new applications. In *Proceedings of the 25th Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 228–242. IEEE Computer Society, 2010.

[17] Chandra Chekuri and Anand Rajaraman. Conjunctive query containment revisited. *Theoretical Computer Science*, 239(2):211–229, 2000.

[18] Bernardo Cuenca Grau, Christian Halaschek-Wiener, Yevgeny Kazakov, and Boontawee Suntisrivaraporn. Incremental classification of description logics ontologies. *Journal of Automated Reasoning*, 44(4):337–369, 2010.

[19] Bernardo Cuenca Grau, Ian Horrocks, Yevgeny Kazakov, and Ulrike Sattler. Just the right amount: Extracting modules from ontologies. In Carey L. Williamson, Mary Ellen Zurko, Peter F. Patel-Schneider, and Prashant J. Shenoy, editors, *Proceedings of the 16th International World Wide Web Conference*, pages 717–726. ACM, 2007.

[20] Bernardo Cuenca Grau, Ian Horrocks, Yevgeny Kazakov, and Ulrike Sattler. A logical framework for modularity of ontologies. In Manuela M. Veloso, editor, *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 298–303, 2007.

[21] Bernardo Cuenca Grau, Ian Horrocks, Yevgeny Kazakov, and Ulrike Sattler. Modular reuse of ontologies: Theory and practice. *Journal of Artificial Intelligence Research*, 31:273–318, 2008.

[22] Bernardo Cuenca Grau, Ian Horrocks, Markus Krötzsch, Clemens Kupke, Despoina Magka, Boris Motik, and Zhe Wang. Acyclicity notions for existential rules and their application to query answering in ontologies. *Journal of Artificial Intelligence Research*, 47:741–808, 2013.

[23] Bernardo Cuenca Grau, Ian Horrocks, Boris Motik, Bijan Parsia, Peter F. Patel-Schneider, and Ulrike Sattler. OWL 2: The next step for OWL. *Journal of Web Semantics*, 6(4):309–322, 2008.

[24] Evgeny Dantsin, Thomas Eiter, Georg Gottlob, and Andrei Voronkov. Complexity and expressive power of logic programming. *ACM Computing Surveys*, 33(3):374–425, 2001.

[25] Chiara Del Vescovo, Pavel Klinov, Bijan Parsia, Ulrike Sattler, Thomas Schneider, and Dmitry Tsarkov. Empirical study of logic-based modules: Cheap is cheerful. In Harith Alani, Lalana Kagal, Achille Fokoue, Paul T. Groth, Chris Biemann, Josiane Xavier Parreira, Lora Aroyo, Natasha F. Noy, Chris Welty, and Krzysztof Janowicz, editors, *Proceedings of the 12th International Semantic Web Conference, Part I*, volume 8218 of *Lecture Notes in Computer Science*, pages 84–100. Springer, 2013.

[26] Chiara Del Vescovo, Bijan Parsia, Ulrike Sattler, and Thomas Schneider. The modular structure of an ontology: Atomic decomposition. In Toby Walsh, editor, *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, pages 2232–2237. IJCAI/AAAI, 2011.

[27] Thomas Eiter, Giovambattista Ianni, Roman Schindlauer, Hans Tompits, and Kewen Wang. Forgetting in managing rules and ontologies. In *Proceedings of the 2006 IEEE / WIC / ACM International Conference on Web Intelligence*, pages 411–419. IEEE Computer Society, 2006.

[28] Melvin Fitting. *First-Order Logic and Automated Theorem Proving*. Texts and Monographs in Computer Science. Springer, 1990.

[29] William Gatens, Boris Konev, and Frank Wolter. Lower and upper approximations for depleting modules of description logic ontologies. In Torsten Schaub, Gerhard Friedrich, and Barry O'Sullivan, editors, *Proceedings of the 21st European Conference on Artificial Intelligence*, volume 263 of *Frontiers in Artificial Intelligence and Applications*, pages 345–350. IOS Press, 2014.

[30] Silvio Ghilardi, Carsten Lutz, and Frank Wolter. Did I damage my ontology? A case for conservative extensions in description logics. In Patrick Doherty, John Mylopoulos, and Christopher A. Welty, editors, *Proceedings of the 10th Inter-*

national Conference on Principles of Knowledge Representation and Reasoning, pages 187–197. AAAI Press, 2006.

[31] Silvio Ghilardi, Carsten Lutz, Frank Wolter, and Michael Zakharyaschev. Conservative extensions in modal logic. In Guido Governatori, Ian M. Hodkinson, and Yde Venema, editors, *Proceedings of the 6th Advances in Modal Logic Conference*, pages 187–207. College Publications, 2006.

[32] Birte Glimm, Ian Horrocks, Boris Motik, Rob Shearer, and Giorgos Stoilos. A novel approach to ontology classification. *Journal of Web Semantics*, 14:84–101, 2012.

[33] Birte Glimm, Ian Horrocks, Boris Motik, Giorgos Stoilos, and Zhe Wang. HermiT: An OWL 2 reasoner. *Journal of Automated Reasoning*, 53(3):245–269, 2014.

[34] Martin Grohe, Thomas Schwentick, and Luc Segoufin. When is the evaluation of conjunctive queries tractable? In Jeffrey Scott Vitter, Paul G. Spirakis, and Mihalis Yannakakis, editors, *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing*, pages 657–666. ACM, 2001.

[35] Volker Haarslev and Ralf Möller. High performance reasoning with very large knowledge bases: A practical case study. In Bernhard Nebel, editor, *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, pages 161–168. Morgan Kaufmann, 2001.

[36] Volker Haarslev and Ralf Möller. RACER system description. In Rajeev Goré, Alexander Leitsch, and Tobias Nipkow, editors, *Proceedings of the 1st International Joint Conference on Automated Reasoning*, volume 2083 of *Lecture Notes in Computer Science*, pages 701–706. Springer, 2001.

[37] Matthew Horridge. *Justification Based Explanation in Ontologies*. PhD thesis, University of Manchester, Manchester, UK, 2011.

[38] Ian Horrocks. Implementation and optimisation techniques. In Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors, *The Description Logic Handbook: Theory, Implementation, and Applications*, chapter 9, pages 306–346. Cambridge University Press, 2003.

[39] Ian Horrocks, Oliver Kutz, and Ulrike Sattler. The even more irresistible $\mathcal{SROIQ}$. In Patrick Doherty, John Mylopoulos, and Christopher A. Welty, editors, *Proceedings of the 10th International Conference on Principles of Knowledge Representation and Reasoning*, pages 57–67. AAAI Press, 2006.

[40] Ian Horrocks, Peter F. Patel-Schneider, and Frank van Harmelen. From SHIQ and RDF to OWL: the making of a web ontology language. *Journal of Web Semantics*, 1(1):7–26, 2003.

[41] Ernesto Jiménez-Ruiz and Bernardo Cuenca Grau. LogMap: Logic-based and scalable ontology matching. In Lora Aroyo, Chris Welty, Harith Alani, Jamie Taylor, Abraham Bernstein, Lalana Kagal, Natasha Fridman Noy, and Eva Blomqvist, editors, *Proceedings of the 10th International Semantic Web Conference, Part I*, volume 7031 of *Lecture Notes in Computer Science*, pages 273–288. Springer, 2011.

[42] Ernesto Jiménez-Ruiz, Bernardo Cuenca Grau, Ian Horrocks, and Rafael Berlanga Llavori. Supporting concurrent ontology development: Framework, algorithms and tool. *Data & Knowledge Engineering*, 70(1):146–164, 2011.

[43] Ernesto Jiménez-Ruiz, Bernardo Cuenca Grau, Ulrike Sattler, Thomas Schneider, and Rafael Berlanga Llavori. Safe and economic re-use of ontologies: A logic-based methodology and tool support. In Sean Bechhofer, Manfred

Hauswirth, Jörg Hoffmann, and Manolis Koubarakis, editors, *Proceedings of the 5th European Semantic Web Conference*, volume 5021 of *Lecture Notes in Computer Science*, pages 185–199. Springer, 2008.

[44] Aditya Kalyanpur, Bijan Parsia, Matthew Horridge, and Evren Sirin. Finding all justifications of OWL DL entailments. In Karl Aberer, Key-Sun Choi, Natasha Fridman Noy, Dean Allemang, Kyung-Il Lee, Lyndon J. B. Nixon, Jennifer Golbeck, Peter Mika, Diana Maynard, Riichiro Mizoguchi, Guus Schreiber, and Philippe Cudré-Mauroux, editors, *Proceedings of the 6th International Semantic Web Conference and 2nd Asian Semantic Web Conference*, volume 4825 of *Lecture Notes in Computer Science*, pages 267–280. Springer, 2007.

[45] Aditya Kalyanpur, Bijan Parsia, Evren Sirin, and Bernardo Cuenca Grau. Repairing unsatisfiable concepts in OWL ontologies. In York Sure and John Domingue, editors, *Proceedings of the 3rd European Semantic Web Conference*, volume 4011 of *Lecture Notes in Computer Science*, pages 170–184. Springer, 2006.

[46] Aditya Kalyanpur, Bijan Parsia, Evren Sirin, and James A. Hendler. Debugging unsatisfiable classes in OWL ontologies. *Journal of Web Semantics*, 3(4):268–293, 2005.

[47] Yevgeny Kazakov. $\mathcal{RIQ}$ and $\mathcal{SROIQ}$ are harder than $\mathcal{SHOIQ}$. In Gerhard Brewka and Jérôme Lang, editors, *Proceedings of the 11th International Conference on Principles of Knowledge Representation and Reasoning*, pages 274–284. AAAI Press, 2008.

[48] Yevgeny Kazakov. Consequence-driven reasoning for horn $\mathcal{SHIQ}$ ontologies. In Craig Boutilier, editor, *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, pages 2040–2045, 2009.

[49] Yevgeny Kazakov, Markus Krötzsch, and František Simančík. The incredible ELK - from polynomial procedures to efficient reasoning with $\mathcal{EL}$ ontologies. *Journal of Automated Reasoning*, 53(1):1–61, 2014.

[50] Boris Konev, Roman Kontchakov, Michel Ludwig, Thomas Schneider, Frank Wolter, and Michael Zakharyaschev. Conjunctive query inseparability of OWL 2 QL tboxes. In Wolfram Burgard and Dan Roth, editors, *Proceedings of the 25th AAAI Conference on Artificial Intelligence*. AAAI Press, 2011.

[51] Boris Konev, Michel Ludwig, Dirk Walther, and Frank Wolter. The logical difference for the lightweight description logic $\mathcal{EL}$. *Journal of Artificial Intelligence Research*, 44:633–708, 2012.

[52] Boris Konev, Carsten Lutz, Dirk Walther, and Frank Wolter. Formal properties of modularisation. In Heiner Stuckenschmidt, Christine Parent, and Stefano Spaccapietra, editors, *Modular Ontologies: Concepts, Theories and Techniques for Knowledge Modularization*, volume 5445 of *Lecture Notes in Computer Science*, pages 25–66. Springer, 2009.

[53] Boris Konev, Carsten Lutz, Dirk Walther, and Frank Wolter. Model-theoretic inseparability and modularity of description logic ontologies. *Artificial Intelligence*, 203:66–103, 2013.

[54] Boris Konev, Dirk Walther, and Frank Wolter. Forgetting and uniform interpolation in large-scale description logic terminologies. In Craig Boutilier, editor, *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, pages 830–835, 2009.

[55] Roman Kontchakov, Carsten Lutz, David Toman, Frank Wolter, and Michael Zakharyaschev. The combined approach to ontology-based data access. In

Toby Walsh, editor, *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, pages 2656–2661. IJCAI/AAAI, 2011.

[56] Roman Kontchakov, Frank Wolter, and Michael Zakharyaschev. Logic-based ontology comparison and module extraction, with an application to DL-Lite. *Artificial Intelligence*, 174(15):1093–1141, 2010.

[57] Patrick Koopmann and Renate A. Schmidt. Count and forget: Uniform interpolation of $\mathcal{SHQ}$-ontologies. In Stéphane Demri, Deepak Kapur, and Christoph Weidenbach, editors, *Proceedings of the 7th International Joint Conference on Automated Reasoning*, volume 8562 of *Lecture Notes in Computer Science*, pages 434–448. Springer, 2014.

[58] Markus Krötzsch. The not-so-easy task of computing class subsumptions in OWL RL. In Philippe Cudré-Mauroux, Jeff Heflin, Evren Sirin, Tania Tudorache, Jérôme Euzenat, Manfred Hauswirth, Josiane Xavier Parreira, Jim Hendler, Guus Schreiber, Abraham Bernstein, and Eva Blomqvist, editors, *Proceedings of the 11th International Semantic Web Conference, Part I*, volume 7649 of *Lecture Notes in Computer Science*, pages 279–294. Springer, 2012.

[59] Markus Krötzsch, Sebastian Rudolph, and Pascal Hitzler. Description logic rules. In Malik Ghallab, Constantine D. Spyropoulos, Nikos Fakotakis, and Nikolaos M. Avouris, editors, *Proceedings of the 18th European Conference on Artificial Intelligence*, volume 178 of *Frontiers in Artificial Intelligence and Applications*, pages 80–84. IOS Press, 2008.

[60] Markus Krötzsch, Sebastian Rudolph, and Pascal Hitzler. ELP: Tractable rules for OWL 2. In Amit P. Sheth, Steffen Staab, Mike Dean, Massimo Paolucci, Diana Maynard, Timothy W. Finin, and Krishnaprasad Thirunarayan, editors,

Proceedings of the 7th International Semantic Web Conference, volume 5318 of *Lecture Notes in Computer Science*, pages 649–664. Springer, 2008.

[61] Michel Ludwig. Just: A tool for computing justifications w.r.t. $\mathcal{ELH}$ ontologies. In Samantha Bail, Birte Glimm, Ernesto Jiménez-Ruiz, Nicolas Matentzoglu, Bijan Parsia, and Andreas Steigmiller, editors, *Proceedings of the 3rd International Workshop on OWL Reasoner Evaluation*, volume 1207 of *CEUR Workshop Proceedings*, pages 1–7. CEUR-WS.org, 2014.

[62] Michel Ludwig and Boris Konev. Practical uniform interpolation and forgetting for $\mathcal{ALC}$ tboxes with applications to logical difference. In Chitta Baral, Giuseppe De Giacomo, and Thomas Eiter, editors, *Proceedings of the 14th International Conference on Principles of Knowledge Representation and Reasoning*. AAAI Press, 2014.

[63] Carsten Lutz and Frank Wolter. Deciding inseparability and conservative extensions in the description logic EL. *Journal of Symbolic Computation*, 45(2):194–228, 2010.

[64] Carsten Lutz and Frank Wolter. Foundations for uniform interpolation and forgetting in expressive description logics. In Toby Walsh, editor, *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, pages 989–995. IJCAI/AAAI, 2011.

[65] Bruno Marnette. Generalized schema-mappings: From termination to tractability. In Jan Paredaens and Jianwen Su, editors, *Proceedings of the 28th ACM SIGMOD Symposium on Principles of Database Systems*, pages 13–22. ACM, 2009.

[66] Boris Motik. *Reasoning in Description Logics Using Resolution and Deductive Databases*. PhD thesis, Univesität Karlsruhe (TH), Karlsruhe, Germany, 2006.

[67] Boris Motik, Yavor Nenov, Robert Piro, Ian Horrocks, and Dan Olteanu. Parallel materialisation of datalog programs in centralised, main-memory RDF systems. In Carla E. Brodley and Peter Stone, editors, *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, pages 129–137. AAAI Press, 2014.

[68] Boris Motik, Peter F. Patel-Schneider, and Bijan Parsia. OWL 2 web ontology language structural specification and functional-style syntax, 2012.

[69] Boris Motik, Rob Shearer, and Ian Horrocks. Hypertableau reasoning for description logics. *Journal of Artificial Intelligence Research*, 36:165–228, 2009.

[70] Nadeschda Nikitina and Sebastian Rudolph. (Non-)succinctness of uniform interpolants of general terminologies in the description logic $\mathcal{EL}$. *Artificial Intelligence*, 215:120–140, 2014.

[71] Andreas Nonnengart and Christoph Weidenbach. Computing small clause normal forms. In John Alan Robinson and Andrei Voronkov, editors, *Handbook of Automated Reasoning*, pages 335–367. Elsevier and MIT Press, 2001.

[72] Riku Nortje, Katarina Britz, and Thomas Meyer. A normal form for hypergraph-based module extraction for $\mathcal{SROIQ}$. In Aurona Gerber, Kerry Taylor, Thomas Meyer, and Mehmet Orgun, editors, *Proceedings of the 8th Australasian Ontology Workshop*, volume 969 of *CEUR Workshop Proceedings*, pages 40–51. CEUR-WS.org, 2012.

[73] Riku Nortje, Katarina Britz, and Thomas Meyer. Reachability modules for the description logic $\mathcal{SRIQ}$. In Kenneth L. McMillan, Aart Middeldorp, and Andrei Voronkov, editors, *Proceedings of the 19th International Conference on Logic for Programming, Artificial Intelligence and Reasoning*, volume 8312 of *Lecture Notes in Computer Science*, pages 636–652. Springer, 2013.

[74] Magdalena Ortiz, Sebastian Rudolph, and Mantas Simkus. Worst-case optimal reasoning for the horn-dl fragments of OWL 1 and 2. In Fangzhen Lin, Ulrike Sattler, and Miroslaw Truszczynski, editors, *Proceedings of the 12th International Conference on Principles of Knowledge Representation and Reasoning*. AAAI Press, 2010.

[75] John Alan Robinson. Automatic deduction with hyper-resolution. *International Journal of Computer Mathematics*, 1(3):227–234, 1965.

[76] Marie-Christine Rousset and Federico Ulliana. Extracting bounded-level modules from deductive RDF triplestores. In Blai Bonet and Sven Koenig, editors, *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, pages 268–274. AAAI Press, 2015.

[77] Ulrike Sattler, Thomas Schneider, and Michael Zakharyaschev. Which kind of module should I extract? In Bernardo Cuenca Grau, Ian Horrocks, Boris Motik, and Ulrike Sattler, editors, *Proceedings of the 22nd International Workshop on Description Logics*, volume 477 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2009.

[78] Stefan Schlobach and Ronald Cornet. Non-standard reasoning services for the debugging of description logic terminologies. In Georg Gottlob and Toby Walsh, editors, *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, pages 355–362. Morgan Kaufmann, 2003.

[79] Julian Seidenberg and Alan L. Rector. Web ontology segmentation: Analysis, classification and use. In Les Carr, David De Roure, Arun Iyengar, Carole A. Goble, and Michael Dahlin, editors, *Proceedings of the 15th International World Wide Web Conference*, pages 13–22. ACM, 2006.

[80] František Simančík, Yevgeny Kazakov, and Ian Horrocks. Consequence-based reasoning beyond horn ontologies. In Toby Walsh, editor, *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, pages 1093–1098. IJCAI/AAAI, 2011.

[81] František Simančík, Boris Motik, and Ian Horrocks. Consequence-based and fixed-parameter fractable reasoning in description logics. *Artificial Intelligence*, 209:29–77, 2014.

[82] Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur, and Yarden Katz. Pellet: A practical OWL DL reasoner. *Journal of Web Semantics*, 5(2):51–53, 2007.

[83] Weihong Song, Bruce Spencer, and Weichang Du. Complete classification of complex $\mathcal{ALCHO}$ ontologies using a hybrid reasoning approach. In Thomas Eiter, Birte Glimm, Yevgeny Kazakov, and Markus Krötzsch, editors, *Proceedings of the 26th International Workshop on Description Logics*, volume 1014 of *CEUR Workshop Proceedings*, pages 942–961. CEUR-WS.org, 2013.

[84] Weihong Song, Bruce Spencer, and Weichang Du. A transformation approach for classifying $\mathcal{ALCHI}(d)$ ontologies with a consequence-based $\mathcal{ALCH}$ reasoner. In Samantha Bail, Birte Glimm, Rafael S. Gonçalves, Ernesto Jiménez-Ruiz, Yevgeny Kazakov, Nicolas Matentzoglu, and Bijan Parsia, editors, *Proceedings of the 2nd International Workshop on OWL Reasoner Evaluation*, volume 1015 of *CEUR Workshop Proceedings*, pages 39–45. CEUR-WS.org, 2013.

[85] Giorgio Stefanoni, Boris Motik, and Ian Horrocks. Introducing nominals to the combined query answering approaches for $\mathcal{EL}$. In Marie desJardins and Michael L. Littman, editors, *Proceedings of the 27th AAAI Conference on Artificial Intelligence*. AAAI Press, 2013.

[86] Andreas Steigmiller, Birte Glimm, and Thorsten Liebig. Coupling tableau algorithms for expressive description logics with completion-based saturation procedures. In Stéphane Demri, Deepak Kapur, and Christoph Weidenbach, editors, *Proceedings of the 7th International Joint Conference on Automated Reasoning*, volume 8562 of *Lecture Notes in Computer Science*, pages 449–463. Springer, 2014.

[87] Andreas Steigmiller, Thorsten Liebig, and Birte Glimm. Konclude: System description. *Journal of Web Semantics*, 27:78–85, 2014.

[88] Heiner Stuckenschmidt, Christine Parent, and Stefano Spaccapietra, editors. *Modular Ontologies: Concepts, Theories and Techniques for Knowledge Modularization*, volume 5445 of *Lecture Notes in Computer Science*. Springer, 2009.

[89] Boontawee Suntisrivaraporn. Module extraction and incremental classification: A pragmatic approach for $\mathcal{EL}+$ ontologies. In Sean Bechhofer, Manfred Hauswirth, Jörg Hoffmann, and Manolis Koubarakis, editors, *Proceedings of the 5th European Semantic Web Conference*, volume 5021 of *Lecture Notes in Computer Science*, pages 230–244. Springer, 2008.

[90] Boontawee Suntisrivaraporn, Guilin Qi, Qiu Ji, and Peter Haase. A modularization-based approach to finding all justifications for OWL DL entailments. In John Domingue and Chutiporn Anutariya, editors, *Proceedings of the 3rd Asian Semantic Web Conference*, volume 5367 of *Lecture Notes in Computer Science*, pages 1–15. Springer, 2008.

[91] Stephan Tobies. *Complexity Results and Practical Algorithms for Logics in Knowledge Representation*. PhD thesis, RWTH Aachen, Germany, 2001.

[92] Dmitry Tsarkov and Ian Horrocks. FaCT++ description logic reasoner: System description. In Ulrich Furbach and Natarajan Shankar, editors, *Proceedings of*

the 3rd International Joint Conference on Automated Reasoning, volume 4130 of *Lecture Notes in Computer Science*, pages 292–297. Springer, 2006.

[93] Dmitry Tsarkov, Ian Horrocks, and Peter F. Patel-Schneider. Optimizing terminological reasoning for expressive description logics. *Journal of Automated Reasoning*, 39(3):277–316, 2007.

[94] Dmitry Tsarkov and Ignazio Palmisano. Chainsaw: A metareasoner for large ontologies. In Ian Horrocks, Mikalai Yatskevich, and Ernesto Jiménez-Ruiz, editors, *Proceedings of the 1st International Workshop on OWL Reasoner Evaluation*, volume 858 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2012.

[95] Dmitry Tsarkov and Ignazio Palmisano. Divide et impera: Metareasoning for large ontologies. In Pavel Klinov and Matthew Horridge, editors, *Proceedings of the 9th OWL: Experiences and Directions Workshop*, volume 849 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2012.

[96] W3C OWL Working Group. OWL 2 web ontology language document overview (second edition). W3C recommendation, World Wide Web Consortium, 2012.

[97] Kewen Wang, Zhe Wang, Rodney W. Topor, Jeff Z. Pan, and Grigoris Antoniou. Eliminating concepts and roles from ontologies in expressive descriptive logics. *Computational Intelligence*, 30(2):205–232, 2014.

[98] Zhe Wang, Kewen Wang, Rodney W. Topor, and Jeff Z. Pan. Forgetting for knowledge bases in DL-Lite. *Annals of Mathematics and Artificial Intelligence*, 58(1-2):117–151, 2010.

[99] Yujiao Zhou, Bernardo Cuenca Grau, Ian Horrocks, Zhe Wu, and Jay Banerjee. Making the most of your triple store: Query answering in OWL 2 using an RL reasoner. In Daniel Schwabe, Virgílio A. F. Almeida, Hartmut Glaser,

Ricardo A. Baeza-Yates, and Sue B. Moon, editors, *Proceedings of the 22nd International World Wide Web Conference*, pages 1569–1580. International World Wide Web Conferences Steering Committee / ACM, 2013.

[100] Yujiao Zhou, Yavor Nenov, Bernardo Cuenca Grau, and Ian Horrocks. Complete query answering over horn ontologies using a triple store. In Harith Alani, Lalana Kagal, Achille Fokoue, Paul T. Groth, Chris Biemann, Josiane Xavier Parreira, Lora Aroyo, Natasha F. Noy, Chris Welty, and Krzysztof Janowicz, editors, *Proceedings of the 12th International Semantic Web Conference, Part I*, volume 8218 of *Lecture Notes in Computer Science*, pages 720–736. Springer, 2013.

[101] Yujiao Zhou, Yavor Nenov, Bernardo Cuenca Grau, and Ian Horrocks. Pay-as-you-go OWL query answering using a triple store. In Carla E. Brodley and Peter Stone, editors, *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, pages 1142–1148. AAAI Press, 2014.

Appendix A

Proofs for Section 5.8

Theorem 76. *The family $\Psi^{\mathcal{S}}$ is $\mathcal{S}$-optimal for $\mathcal{S} \in \{\mathsf{m}, \mathsf{i}, \mathsf{c}, \mathsf{wq}\}$.*

We prove the result for each $\mathcal{S} \in \{\mathsf{m}, \mathsf{i}, \mathsf{c}, \mathsf{wq}\}$ separately.

Theorem 98. *Ψ^{m} is m-optimal.*

Proof. Suppose Ψ^{m} is not m-optimal. Then there must be some m-admissible, uniform module setting family Ψ and some $\mathcal{O}$ and Σ such that $\mathcal{M}^{\Psi^{\mathsf{m}}(\mathcal{O},\Sigma)} \not\subseteq \mathcal{M}^{\Psi(\mathcal{O},\Sigma)}$. Let $\Psi^{\mathsf{m}}(\mathcal{O}, \Sigma) = \langle \theta^{\mathsf{m}}, \mathcal{D}_0^{\mathsf{m}}, \mathcal{D}_r^{\mathsf{m}} \rangle$ and $\Psi(\mathcal{O}, \Sigma) = \langle \theta, \mathcal{D}_0, \mathcal{D}_r \rangle$. By Theorem 57, we have $\Psi^{\mathsf{m}}(\mathcal{O}, \Sigma) \not\rightarrow \Psi(\mathcal{O}, \Sigma)$, hence for each mapping $\mu : \mathsf{Ct}(\Psi^{\mathsf{m}}(\mathcal{O}, \Sigma)) \rightarrow \mathsf{Ct}(\Psi(\mathcal{O}, \Sigma))$ it must be either $\theta^{\mathsf{m}}\mu \neq \theta$ or $\mathcal{D}_0^{\mathsf{m}}\mu \not\subseteq \mathcal{D}_0$ or $\mathcal{D}_r^{\mathsf{m}}\mu \not\subseteq \mathcal{D}_r$.

Suppose θ is such that there are two existentially quantified variables y_1 and y_2 in $\mathcal{O}$ such that $y_1\theta \neq y_2\theta$. Consider the ontology $\mathcal{O}'$ consisting of the following rules:

$$
\begin{aligned}
p_1 &: A(x) \rightarrow S(x, b) \\
p_2 &: A(x) \rightarrow \exists y_1[R(x, y_1) \wedge B(y_1)] \\
p_3 &: A(x) \rightarrow \exists y_2[R(x, y_2) \wedge C(y_2)] \\
p_4 &: S(x, b) \wedge R(x, z) \wedge B(z) \wedge C(z) \rightarrow D(x)
\end{aligned}
$$

and the signature $\Sigma' = \{A, D, R\}$ and let $\Psi(\mathcal{O}', \Sigma') = \langle \theta', \mathcal{D}'_0, \mathcal{D}'_r \rangle$. By the second property of uniformity, θ' is the same as the substitution in $\Psi(\mathcal{O}', \emptyset)$ and $\Psi(\mathcal{O}', \Sigma)$,

and therefore, by the first property of uniformity, $y_1\theta' \neq y_2\theta'$. It follows that p_4 can never be applied on $\mathcal{P}^{\Psi(\mathcal{O}',\Sigma')}(\mathcal{D}'_0)$ and hence it is not in the support of $\Psi(\mathcal{O}',\Sigma')$ and $\mathcal{M}^{\Psi(\mathcal{O}',\Sigma')} \subseteq \{p_1, p_2, p_3\}$. The interpretation $\mathcal{I}$ such that $\Delta^{\mathcal{I}} = \{i\}$, $A^{\mathcal{I}} = B^{\mathcal{I}} = C^{\mathcal{I}} = \{i\}$, $D^{\mathcal{I}} = \emptyset$, $a^{\mathcal{I}} = i$ and $R^{\mathcal{I}} = S^{\mathcal{I}} = \{(i,i)\}$ is a model of $\mathcal{M}^{\Psi(\mathcal{O}',\Sigma')}$, but it cannot be extended to a model of $\mathcal{O}'$ without changing the interpretation of A, D or R. It follows that $\mathcal{M}^{\Psi(\mathcal{O}',\Sigma')}$ is not a $\equiv^{\mathsf{m}}_{\Sigma'}$-module of $\mathcal{O}'$, contradicting our hypothesis that Ψ is m-admissible. The origin of this contradiction is in the assumption that θ does not map all existentially quantified variables to the same constant, therefore there must be some constant c such that $y\theta = c$ for each variable y universally quantified in $\mathcal{O}$. Suppose now that θ is such that there exists a constant b' in $\mathcal{O}$ such that $b'\theta \neq c$. By the first property of uniformity θ' must also not map b to the same constant as y_1 and y_2. But then again p_4 can never be applied on $\mathcal{P}^{\Psi(\mathcal{O}',\Sigma')}(\mathcal{D}'_0)$ and $\mathcal{M}^{\Psi(\mathcal{O}',\Sigma')} \subseteq \{p_1, p_2, p_3\}$, which, as we have already shown, is a contradiction. Consequently, θ must map all constants in $\mathcal{O}$ to c as well.

This means that there exists some substitution $\mu : \mathsf{Ct}(\Psi^{\mathsf{m}}(\mathcal{O},\Sigma)) \to \mathsf{Ct}(\Psi(\mathcal{O},\Sigma))$ such that $\theta^{\mathsf{m}}\mu = \theta$. In particular it must be $*\mu = c$. By hypothesis, it must be the case that either $\mathcal{D}^{\mathsf{m}}_0\mu \not\subseteq \mathcal{D}_0$ or $\mathcal{D}^{\mathsf{m}}_r\mu \not\subseteq \mathcal{D}_r$.

Suppose $\mathcal{D}^{\mathsf{m}}_0\mu \not\subseteq \mathcal{D}_0$. There must be some predicate $X \in \Sigma$ with $X(c,\ldots,c) \notin \mathcal{D}_0$. Consider the ontology $\mathcal{O}''$ consisting of the following rules:

$$\begin{aligned} p_5 &: X(x,\ldots,x) \to \exists y R(x,y) \\ p_6 &: R(x,x) \to A(x) \end{aligned}$$

and the signature $\Sigma'' = \{X, A\}$, and let $\Psi(\mathcal{O}'',\Sigma'') = \langle \theta'', \mathcal{D}''_0, \mathcal{D}''_r \rangle$. By uniformity of Ψ, θ'' maps y to c, and the fact $X(c,\ldots,c)$ is not in the initial dataset of $\Psi(\mathcal{O},\{X\})$, or in that of $\Psi(\mathcal{O},\Sigma'')$, or in $\mathcal{D}''_0$. Consider the dataset

$$\hat{\mathcal{D}} = \{A(a), A(c)\} \cup \{\, X(\mathbf{t}) \mid \mathbf{t} \in \{a,c\}^{\mathsf{arity}(X)}, \mathbf{t} \neq (c,\ldots,c) \,\}$$

with a a fresh constant. The substitution $\hat{\mu}$ that maps c to itself and all other constants in $\Psi(\mathcal{O}'', \Sigma'')$ to a is a homomorphism from $\Psi(\mathcal{O}'', \Sigma'')$ to $\hat{\chi} = \langle \theta'', \hat{\mathcal{D}}, \mathcal{D}''_r \hat{\mu} \rangle$. By Theorem 57, this implies $\mathcal{M}^{\Psi(\mathcal{O}'', \Sigma'')} \subseteq \mathcal{M}^{\hat{\chi}}$. Clearly, the only R-fact in $\mathcal{P}^{\hat{\chi}}(\hat{\mathcal{D}})$ is $R(a, c)$, so p_5 is not in the support of $\hat{\chi}$ and therefore $\mathcal{M}^{\Psi(\mathcal{O}'', \Sigma'')} \subseteq \mathcal{M}^{\hat{\chi}} \subseteq \{p_5\}$. The interpretation $\mathcal{J}$ such that $\Delta^{\mathcal{J}} = \{i\}$, $X^{\mathcal{J}} = (i, \ldots, i)$, $A^{\mathcal{J}} = \emptyset$ and $R^{\mathcal{J}} = \{(i, i)\}$ is a model of $\mathcal{M}^{\Psi(\mathcal{O}', \Sigma'')}$, but it cannot be extended to a model of $\mathcal{O}''$ without changing the interpretation of X or A. It follows that $\mathcal{M}^{\Psi(\mathcal{O}'', \Sigma'')}$ is not a $\equiv^{\mathsf{m}}_{\Sigma''}$-module of $\mathcal{O}''$, which contradicts our hypothesis that Ψ is m-admissible. This contradiction stems from the assumption that $\mathcal{D}^{\mathsf{m}}_0 \mu \not\subseteq \mathcal{D}_0$, therefore it must be $\mathcal{D}^{\mathsf{m}}_0 \mu \subseteq \mathcal{D}_0$.

By hypothesis, this implies $\mathcal{D}^{\mathsf{m}}_r \mu \not\subseteq \mathcal{D}_r$, so there must be some predicate $Y \in \Sigma$ such that $Y(c, \ldots, c) \notin \mathcal{D}_r$. Consider the ontology $\mathcal{O}'''$ consisting of the following rules:

$$\begin{aligned} p_7 &: A(x) \rightarrow \exists y R(x, y) \\ p_8 &: R(x, x) \rightarrow Y(x, \ldots, x) \end{aligned}$$

and the signature $\Sigma''' = \{A, Y\}$, and let $\Psi(\mathcal{O}''', \Sigma''') = \langle \theta''', \mathcal{D}'''_0, \mathcal{D}'''_r \rangle$. By uniformity of Ψ, θ''' maps y to c. Consider the dataset $\check{\mathcal{D}} = \{A(c), Y(c, \ldots, c), A(b), Y(b, \ldots, b)\}$, with a again a fresh constant. The mapping $\check{\mu}$ that maps c to itself and all other constants in $\Psi(\mathcal{O}''', \Sigma''')$ to b is a homomorphism from $\Psi(\mathcal{O}''', \Sigma''')$ to $\check{\chi} = \langle \theta''', \check{\mathcal{D}}, \mathcal{D}'''_r \check{\mu} \rangle$. By Theorem 57, this implies $\mathcal{M}^{\Psi(\mathcal{O}''', \Sigma'')} \subseteq \mathcal{M}^{\check{\chi}}$. It is easy to see that the only R-facts in the materialisation of $\mathcal{P}^{\check{\chi}} \cup \check{\mathcal{D}}$ are $R(a, c)$ and $R(c, c)$, so the only proof in $\mathcal{P}^{\check{\chi}} \cup \check{\mathcal{D}}$ supported by p_8 is a proof of $Y(c, \ldots, c)$. However, by uniformity of Ψ, $Y(c, \ldots, c)$ is not in the relevant facts of $\Psi(\mathcal{O}, \{Y\})$, or in those of $\Psi(\mathcal{O}, \{A, Y\})$, or in $\mathcal{D}'''_r$. It follows that p_8 is not in the support of $\check{\chi}$ and $\mathcal{M}^{\Psi(\mathcal{O}''', \Sigma''')} \subseteq \mathcal{M}^{\check{\chi}} \subseteq \{p_7\}$. The interpretation $\mathcal{K}$ such that $\Delta^{\mathcal{K}} = \{i\}$, $A^{\mathcal{K}} = \{i\}$, $R^{\mathcal{K}} = \{(i, i)\}$ and $Y^{\mathcal{K}} = \emptyset$ is a model of $\mathcal{M}^{\Psi(\mathcal{O}', \Sigma'')}$, but it cannot be extended to a model of $\mathcal{O}'''$ without changing the interpretation of A or Y. It follows that $\mathcal{M}^{\Psi(\mathcal{O}''', \Sigma''')}$ is not a $\equiv^{\mathsf{m}}_{\Sigma'''}$-module of $\mathcal{O}'''$, which again contradicts our hypothesis that Ψ is m-admissible. The origin of this

contradiction is in the assumption that $\mathcal{D}_r^{\mathsf{m}}\mu \not\subseteq \mathcal{D}_r$, therefore it must be $\mathcal{D}_r^{\mathsf{m}}\mu \subseteq \mathcal{D}_r$. The mapping μ is thus a witness that $\Psi^{\mathsf{m}}(\mathcal{O},\Sigma) \hookrightarrow \Psi(\mathcal{O},\Sigma)$, ultimately contradicting the assumption that Ψ^{m} is not m-optimal. □

Theorem 99. *Ψ^{i} is i-optimal.*

Proof. Consider the module setting family Ψ_0^{i} such that for each pair of ontology $\mathcal{O}$ and signature Σ, $\Psi_0^{\mathsf{i}}(\mathcal{O},\Sigma) = \langle \theta^{\mathsf{i}_0}, \mathcal{D}_0^{\mathsf{i}_0}, \mathcal{D}_r^{\mathsf{i}_0} \rangle$ is as follows:

- $\theta^{\mathsf{i}_0} = \{\, y \mapsto c_y \mid y \text{ existentially quantified in } \mathcal{O} \,\} \cup \{\, c \mapsto c \mid c \in \mathsf{Sig}(\mathcal{O}) \text{ a constant} \,\}$
- $\mathcal{D}_0^{\mathsf{i}_0} = \{\, A(\mathbf{c}_{A,B}) \mid A \neq B \text{ predicates in } \Sigma \text{ of the same arity} \,\}$
- $\mathcal{D}_r^{\mathsf{i}_0} = \{\, B(\mathbf{c}_{A,B}) \mid A \neq B \text{ predicates in } \Sigma \text{ of the same arity} \,\} \cup \mathcal{D}_{r_\bot}^{\mathsf{i}_0}$

where each c_y is a fresh constant, $\mathbf{c}_{A,B} = c^1_{A,B}, \ldots, c^n_{A,B}$, is an array of fresh constants for each pair $A, B \in \Sigma$ of distinct n-ary predicates, and $\mathcal{D}_{r_\bot}^{\mathsf{i}_0} = \{\bot\}$ if Σ contains two distinct predicates of the same arity and $\mathcal{D}_{r_\bot}^{\mathsf{i}_0} = \emptyset$ otherwise.

First, we show that Ψ^{i} is i-optimal iff Ψ_0^{i} is i-optimal by proving that, for each $\mathcal{O}$ and Σ, it is $\mathcal{M}^{\Psi^{\mathsf{i}}(\mathcal{O},\Sigma)} = \mathcal{M}^{\Psi_0^{\mathsf{i}}(\mathcal{O},\Sigma)}$. Let us fix arbitrary $\mathcal{O}$ and Σ, and let $\Psi^{\mathsf{i}}(\mathcal{O},\Sigma) = \langle \theta^{\mathsf{i}}, \mathcal{D}_0^{\mathsf{i}}, \mathcal{D}_r^{\mathsf{i}} \rangle$ and $\Psi_0^{\mathsf{i}}(\mathcal{O},\Sigma) = \langle \theta^{\mathsf{i}_0}, \mathcal{D}_0^{\mathsf{i}_0}, \mathcal{D}_r^{\mathsf{i}_0} \rangle$. It is easy to see that $\Psi_0^{\mathsf{i}}(\mathcal{O},\Sigma) \hookrightarrow \Psi^{\mathsf{i}}(\mathcal{O},\Sigma)$, therefore by Theorem 57 we have that $\mathcal{M}^{\Psi_0^{\mathsf{i}}(\mathcal{O},\Sigma)} \subseteq \mathcal{M}^{\Psi^{\mathsf{i}}(\mathcal{O},\Sigma)}$. For showing that $\mathcal{M}^{\Psi^{\mathsf{i}}(\mathcal{O},\Sigma)} \subseteq \mathcal{M}^{\Psi_0^{\mathsf{i}}(\mathcal{O},\Sigma)}$, first note that $\mathcal{P}^{\Psi^{\mathsf{i}}(\mathcal{O},\Sigma)} = \mathcal{P}^{\Psi_0^{\mathsf{i}}(\mathcal{O},\Sigma)}$. We can assume w.l.o.g. that Σ contains at least two predicates A, B of the same arity—otherwise there are no non-trivial Σ-implications. Given a proof $\rho = (T, \lambda)$ in $\mathcal{P}^{\Psi^{\mathsf{i}}(\mathcal{O},\Sigma)} \cup \mathcal{D}_0^{\mathsf{i}}$ of $B(\mathbf{c}_A)$ (resp. of $\bot$) the proof $\rho_0 = (T, \lambda_0)$ such that $\lambda_0(v) = \lambda(v)\nu$ for each node v in T, with ν a substitution such that $\mathbf{c}_A\nu = \mathbf{c}_{A,B}$, is a proof of $B(\mathbf{c}_{A,B})$ (resp. of $\bot$) in $\mathcal{P}^{\Psi^{\mathsf{i}_0}(\mathcal{O},\Sigma)} \cup \mathcal{D}_0^{\mathsf{i}_0}$ satisfying $\mathsf{supp}(\rho_0) = \mathsf{supp}(\rho)$. Consequently, $\mathsf{supp}(\Psi^{\mathsf{i}}(\mathcal{O},\Sigma)) \subseteq \mathsf{supp}(\Psi_0^{\mathsf{i}}(\mathcal{O},\Sigma))$ and hence $\mathcal{M}^{\Psi^{\mathsf{i}}(\mathcal{O},\Sigma)} \subseteq \mathcal{M}^{\Psi_0^{\mathsf{i}}(\mathcal{O},\Sigma)}$.

Now, suppose Ψ_0^{i} is not i-optimal. There must be a uniform, i-admissible family Ψ and some $\mathcal{O}$ and Σ such that $\mathcal{M}^{\Psi_0^{\mathsf{i}}(\mathcal{O},\Sigma)} \not\subseteq \mathcal{M}^{\Psi(\mathcal{O},\Sigma)}$. Let $\Psi(\mathcal{O},\Sigma) = \langle \theta, \mathcal{D}_0, \mathcal{D}_r \rangle$.

Since θ^{i_0} is injective, there exists a mapping $\mu : \mathsf{Ct}(\Psi_0^{\mathsf{i}}(\mathcal{O}, \Sigma)) \to \mathsf{Ct}(\Psi(\mathcal{O}, \Sigma))$ such that $\theta^{\mathsf{i}_0}\mu = \theta$. This condition only determines the effect of μ on $\mathsf{dom}(\theta^{\mathsf{i}_0})$, and since $\mathsf{dom}(\theta^{\mathsf{i}_0})$ is disjoint with the set $\{ \mathbf{c}_{A,B} \mid A \neq B$ predicates in Σ of the same arity $\}$, we can assume that either μ is such that $\mathcal{D}_0^{\mathsf{i}_0}\mu \subseteq \mathcal{D}_0$, or there are two distinct predicates $A, B \in \Sigma$ of the same arity such that $\mathcal{D}_0$ contains no A-facts. Suppose the latter is the case, and consider the ontology $\mathcal{O}' = \{A(\mathbf{x}) \to B(\mathbf{x})\}$. By uniformity of Ψ, the initial dataset of $\Psi(\mathcal{O}', \Sigma)$ also contains no A-facts and therefore the support of $\Psi(\mathcal{O}', \Sigma)$ is empty and $\mathcal{M}^{\Psi(\mathcal{O}', \Sigma)} = \emptyset \not\models A(\mathbf{x}) \to B(\mathbf{x})$. This contradicts the i-admissibility of Ψ, hence it must be $\mathcal{D}_0^{\mathsf{i}_0}\mu \subseteq \mathcal{D}_0$. Finally, suppose $\mathcal{D}_r^{\mathsf{i}_0}\mu \not\subseteq \mathcal{D}_r$; then in particular Σ must contain two distinct n-ary predicates, since otherwise it is $\mathcal{D}_r^{\mathsf{i}_0} = \emptyset$ and thus trivially $\mathcal{D}_r^{\mathsf{i}_0}\mu \not\subseteq \mathcal{D}_r$. In this case there are two possible situations:

- $\bot \notin \mathcal{D}_r$.

  Let $A, B \in \Sigma$ be distinct predicates of the same arity and consider the ontology $\mathcal{O}'' = \{A(\mathbf{x}) \to C(\mathbf{x}) \vee D(\mathbf{x}), C(\mathbf{x}) \to B(\mathbf{x}), D(\mathbf{x}) \to \bot\}$ with C and D fresh predicates. Clearly, $\mathcal{O}'' \models A(\mathbf{x}) \to B(\mathbf{x})$ By uniformity of Ψ, $\bot$ is not in the set of relevant facts of $\Psi(\mathcal{O}'', \Sigma)$, and therefore $D(\mathbf{x}) \to \bot \notin \mathcal{M}^{\Psi(\mathcal{O}'', \Sigma)}$. Consequently we have $\mathcal{M}^{\Psi(\mathcal{O}'', \Sigma)} \not\models A(\mathbf{x}) \to B(\mathbf{x})$, contradicting the assumption that Ψ is i-admissible.

- $B(\mathbf{c}_{A,B})\mu \notin \mathcal{D}_r$ for some pair of distinct $A, B \in \Sigma$ of the same arity.

  Consider $\mathcal{O}''' = \{A(\mathbf{x}) \to B(\mathbf{x})\}$. By uniformity of Ψ, there are no B-facts in the set of relevant facts of $\Psi(\mathcal{O}''', \{A, B\})$. It follows that $\mathcal{M}^{\Psi(\mathcal{O}''', \{A,B\})} = \emptyset$ and hence $\mathcal{M}^{\Psi(\mathcal{O}'', \{A,B\})} \not\models A(\mathbf{x}) \to B(\mathbf{x})$, contradicting the i-admissibility of Ψ.

It follows that $\mathcal{D}_r^{\mathsf{i}_0}\mu \subseteq \mathcal{D}_r$, so μ is a homomorphism from $\Psi(\mathcal{O}, \Sigma)$ to $\Psi_0^{\mathsf{i}}(\mathcal{O}, \Sigma)$, and thus $\mathcal{M}^{\Psi_0^{\mathsf{i}}(\mathcal{O}, \Sigma)} \subseteq \mathcal{M}^{\Psi(\mathcal{O}, \Sigma)}$, which ultimately contradicts the assumption that Ψ_0^{i} (resp. Ψ^{i}) is not i-optimal. □

Theorem 100. Ψ^{c} *is* c*-optimal.*

Proof. Analogous to Theorem 99 □

Theorem 101. Ψ^{wq} *is* wq*-optimal.*

Proof. Suppose Ψ^{wq} is not wq-optimal. There must be some uniform and wq-admissible family Ψ and some $\mathcal{O}$ and Σ s.t. $\mathcal{M}^{\Psi^{\mathsf{wq}}(\mathcal{O},\Sigma)} \not\subseteq \mathcal{M}^{\Psi(\mathcal{O},\Sigma)}$. Let $\Psi(\mathcal{O},\Sigma) = \langle \theta, \mathcal{D}_0, \mathcal{D}_r \rangle$ and $\Psi^{\mathsf{wq}}(\mathcal{O},\Sigma) = \langle \theta^{\mathsf{wq}}, \mathcal{D}_0\mathsf{wq}, \mathcal{D}_r^{\mathsf{wq}} \rangle$. By Theorem 57, given an arbitrary mapping $\mu : \mathsf{Ct}(\Psi^{\mathsf{wq}}) \to \mathsf{Ct}(\Psi)$, it must be either $\theta^{\mathsf{wq}}\mu \neq \theta$ or $\mathcal{D}_0^{\mathsf{wq}}\mu \not\subseteq \mathcal{D}_0$ or $\mathcal{D}_r^{\mathsf{wq}}\mu \not\subseteq \mathcal{D}_r$.

Since θ^{wq} is injective and $\mathcal{D}^{\mathsf{wq}} = \emptyset$, we can assume that μ is such that $\theta^{\mathsf{wq}}\mu = \theta$, $\mathcal{D}_0^{\mathsf{wq}}\mu \subseteq \mathcal{D}_0$ and $\mathcal{D}_r^{\mathsf{wq}}\mu \not\subseteq \mathcal{D}_r$. But then there must be some predicate $X \in \Sigma$ and an array $\mathbf{c}$ of size $\mathsf{arity}(X)$ of constants from $\mathsf{Ct}(\mathcal{O}) \cup \{\, c_y \mid y \text{ exist. quant. in } \mathcal{O} \,\}$ such that $X(\mathbf{c})\mu \notin \mathcal{D}_r$. Consider the ontology $\mathcal{O}' = \{(\to X(\mathbf{c}))\}$; clearly, $\mathcal{O}' \models X(\mathbf{c})$. By uniformity, $X(\mathbf{c})$ is also not in the relevant facts of $\Psi(\mathcal{O}',\Sigma)$; this implies $\mathcal{M}^{\Psi(\mathcal{O}',\Sigma)} = \emptyset$ and consequently $\mathcal{M}^{\Psi(\mathcal{O}',\Sigma)} \not\models X(\mathbf{c})$. This contradicts the wq-admissibility of Ψ, hence it must be $\mathcal{D}_r^{\mathsf{wq}}\mu \subseteq \mathcal{D}_r$, which makes μ a homomorphism. It follows that the family Ψ^{wq} is wq-optimal. □

Theorem 77. *The family* $\Psi^{\mathcal{S}}$ *is not* $\mathcal{S}$*-optimal for* $\mathcal{S} \in \{\mathsf{f}, \mathsf{q}\}$.

Again, we prove the result for each $\mathcal{S} \in \{\mathsf{f}, \mathsf{q}\}$ separately.

Proposition 102. *The family* Ψ^{f} *is not* f*-optimal.*

Proof. Consider an arbitrary but fixed pair of $\mathcal{O}$ and Σ. Let $\mathsf{Ct}(\mathcal{O}) = \{c_1, \ldots, c_n\}$. Furthermore, for each predicate $B \in \Sigma$ and each array $\mathbf{v} \in \{1, \ldots, \mathsf{arity}(B)+n\}^{\mathsf{arity}(B)}$ consider a set of constants $\{\, *^i_{B,\mathbf{v}} \mid 0 \leq i \leq \mathsf{arity}(B) + n \,\}$ such that

1. $*^0_{B,\mathbf{v}}, \ldots, *^{\mathsf{arity}}_{B,\mathbf{v}}(B)$ are fresh constants and
2. $*^{\mathsf{arity}}_{B,\mathbf{v}}(B), \ldots, *^{\mathsf{arity}(B)+n}_{B,\mathbf{v}}$ are such that $*^{\mathsf{arity}(B)+i}_{B,\mathbf{v}} = c_i \in \mathsf{Ct}(\mathcal{O})$ for each $1 \leq i \leq n$.

Let $*^{\mathbf{w}}_{B,\mathbf{v}}$ denote the array $(*^{w_1}_{B,\mathbf{v}}, \ldots, *^{w_n}_{B,\mathbf{v}})$ whenever $\mathbf{w} = (w_1, \ldots, w_n)$, and consider the uniform module setting family Ψ^{f}_0 such that $\Psi^{\mathsf{f}}_0(\mathcal{O},\Sigma) = \langle \theta^{\mathsf{f}_0}, \mathcal{D}_0^{\mathsf{f}_0}, \mathcal{D}_r^{\mathsf{f}_0} \rangle$ is as follows:

- $\theta^{\mathsf{f}_0} = \{\, y \mapsto c_y \mid y \text{ existentially quantified in } \mathcal{O} \,\} \cup \{\, c \mapsto c \mid c \in \mathsf{Ct}(\mathcal{O}) \,\}$
- $\mathcal{D}_0^{\mathsf{f}_0} = \{A(*^{\mathbf{w}}_{B,\mathbf{v}}) \mid A, B \in \Sigma, \mathbf{v} \in \{1, \ldots, \mathsf{arity}(B) + n\}^{\mathsf{arity}(B)},$
$\mathbf{w} \in \{0, \ldots, \mathsf{arity}(B) + n\}^{\mathsf{arity}(A)}, A(*^{\mathbf{w}}_{B,\mathbf{v}}) \neq B(*^{\mathbf{v}}_{B,\mathbf{v}})\}$
- $\mathcal{D}_r^{\mathsf{f}_0} = \{\, B(*^{\mathbf{v}}_{B,\mathbf{v}}) \mid B \in \Sigma, \mathbf{v} \in \{1, \ldots, \mathsf{arity}(B)\}^{\mathsf{arity}(B)} \,\} \cup \{\bot\}$

We now show that the family Ψ_0^{f} is f-admissible.

Consider a datalog rule $r = \varphi \to \gamma \in \mathsf{rel}_{\mathsf{f}(\mathcal{O},\Sigma)}$. W.l.o.g. we can assume that $\gamma \notin \varphi$—otherwise r would be tautological and hence entailed by any ontology. Let σ be a substitution mapping all variables in r to distinct fresh constants. Since $\gamma \notin \varphi$ and σ is injective, we also have $\gamma\sigma \notin \varphi\sigma$. The fact $\gamma\sigma$ must be of the form $B(\mathbf{c})$ with $B \in \Sigma$ (if $B = \bot$ then it would be $\mathbf{c} = \emptyset$) and consider an injective substitution mapping all constants in $\mathbf{c} \cup \mathsf{Ct}(\mathcal{O})$ to $\{1, \ldots, \mathsf{arity}(B)+n\}$ and satisfying $c\eta = \mathsf{arity}(B) + i$ iff $c = c_i \in \mathsf{Ct}(\mathcal{O})$. Let τ be another substitution defined on all constants in $\mathsf{Sig}(\varphi\sigma \cup \gamma\sigma)$ and such that

$$c\tau = \begin{cases} *^{c\eta}_{B,\mathbf{c}\eta} & \text{if } c \in \mathbf{c} \cup \mathsf{Ct}(\mathcal{O}) \\ *^{0}_{B,\mathbf{c}\eta} & \text{otherwise} \end{cases}$$

Note that τ is compatible with θ^{f_0}.

It is easy to see that $(\gamma\sigma)\tau = B(*^{\mathbf{c}\eta}_{B,\mathbf{c}\eta}) \in \mathcal{D}_r^{\mathsf{f}_0}$ since $\mathbf{c}\eta \in \{1, \ldots, \mathsf{arity}(B)+n\}^{\mathsf{arity}(B)}$. On the other hand, each fact in $(\varphi\sigma)\tau$ must be of the form $A(*^{\mathbf{w}}_{B,\eta(\mathbf{c})})$ with $A \in \Sigma$ and $\mathbf{w} \in \{0, \ldots, \mathsf{arity}(B) + n\}^{\mathsf{arity}(A)}$. Because $\gamma\sigma \notin \varphi\sigma$ and η is injective, it is easy to see that $A(*^{\mathbf{w}}_{B,\mathbf{c}\eta}) \neq B(*^{\mathbf{c}\eta}_{B,\mathbf{c}\eta})$. Consequently, $(\varphi\sigma)\nu \subseteq \mathcal{D}_r^{\mathsf{f}_0}$. By Theorem 25, it follows that $\mathcal{O} \models r$ iff $\mathcal{M}^{\Psi_0^{\mathsf{f}}(\mathcal{O},\Sigma)} \models r$, hence Ψ_0^{f} is f-admissible.

Finally, consider $\mathcal{O} = \{A(x) \to B(x), B(x) \to A(x)\}$ and $\Sigma = \{A\}$. It is easy to see that $\mathcal{M}^{\Psi^{\mathsf{f}}(\mathcal{O},\Sigma)} = \mathcal{O} \not\subseteq \emptyset = \mathcal{M}^{\Psi_0^{\mathsf{f}}(\mathcal{O},\Sigma)}$, and thus Ψ^{f} is not f-optimal. □

Proposition 103. *The family Ψ^{q} is not q-optimal.*

Proof. Consider an arbitrary but fixed pair of $\mathcal{O}$ and Σ. Let $\mathsf{Ct}(\mathcal{O}) = \{c_1, \ldots, c_n\}$ and let $\{y_1, \ldots, y_m\}$ be the set of existentially quantified variables mentioned in $\mathcal{O}$, and $\{c_{y_1}, \ldots, c_{y_m}\}$ a corresponding set of fresh constants. Furthermore, for each predicate $B \in \Sigma$ and each array $\mathbf{v} \in \{1, \ldots, \mathsf{arity}(B) + n + m\}^{\mathsf{arity}(B)}$ consider a set of constants $\{ *^i_{B,\mathbf{v}} \mid 0 \leq i \leq \mathsf{arity}(B) + n + m \}$ such that

1. $*^0_{B,\mathbf{v}}, \ldots, *^{\mathsf{arity}}_{B,\mathbf{v}}(B)$ are fresh constants,
2. $*^{\mathsf{arity}}_{B,\mathbf{v}}(B), \ldots, *^{\mathsf{arity}(B)+n}_{B,\mathbf{v}}$ are such that $*^{\mathsf{arity}(B)+i}_{B,\mathbf{v}} = c_i \in \mathsf{Ct}(\mathcal{O})$ for each $1 \leq i \leq n$, and
3. $*^{\mathsf{arity}(B)+n+1}_{B,\mathbf{v}}, \ldots, *^{\mathsf{arity}(B)+n+m}_{B,\mathbf{v}}$ are such that $*^{\mathsf{arity}(B)+n+i}_{B,\mathbf{v}} = c_{y_i}$ for each $1 \leq i \leq m$

Let $*^{\mathbf{w}}_{B,\mathbf{v}}$ denote the array $(*^{w_1}_{B,\mathbf{v}}, \ldots, *^{w_n}_{B,\mathbf{v}})$ whenever $\mathbf{w} = (w_1, \ldots, w_n)$, and consider the uniform module setting family Ψ^{q}_0 such that $\Psi^{\mathsf{q}}_0(\mathcal{O}, \Sigma) = \langle \theta^{\mathsf{q}_0}, \mathcal{D}^{\mathsf{q}_0}_0, \mathcal{D}^{\mathsf{q}_0}_r \rangle$ is as follows:

- $\theta^{\mathsf{q}_0} = \{ y \mapsto c_y \mid y \text{ existentially quantified in } \mathcal{O} \} \cup \{ c \mapsto c \mid c \in \mathsf{Ct}(\mathcal{O}) \}$
- $\mathcal{D}^{\mathsf{q}_0}_0 = \{A(*^{\mathbf{w}}_{B,\mathbf{v}}) \mid A, B \in \Sigma, \mathbf{v} \in \{1, \ldots, \mathsf{arity}(B) + n + m\}^{\mathsf{arity}(B)}, \mathbf{w} \in \{0, 1, \ldots, \mathsf{arity}(B) + n\}^{\mathsf{arity}(A)}, A(*^{\mathbf{w}}_{B,\mathbf{v}}) \neq B(*^{\mathbf{v}}_{B,\mathbf{v}})\}$
- $\mathcal{D}^{\mathsf{q}_0}_r = \{ B(*^{\mathbf{v}}_{B,\mathbf{v}}) \mid B \in \Sigma, \mathbf{v} \in \{1, \ldots, \mathsf{arity}(B) + n + m\}^{\mathsf{arity}(B)} \} \cup \{\bot\}$

We now show that the family Ψ^{q}_0 is q-admissible.

Consider a rule $r = \varphi(\mathbf{x}) \to \exists \mathbf{y} \psi(\mathbf{x}, \mathbf{y}) \in \mathsf{rel}_{\mathsf{q}}(\mathcal{O}, \Sigma)$, a justification $\mathcal{O}'$ of r in $\mathcal{O}$ and a rule $s \in \mathcal{O}'$. To show that $\Psi^{\mathsf{q}}_0(\mathcal{O}, \Sigma)$ is q-admissible it suffices to show that $s \subseteq \mathcal{M}^{\Psi^{\mathsf{q}}_0(\mathcal{O},\Sigma)}$.

We can assume w.l.o.g. that $\bot \notin \varphi$ (otherwise r would be tautological) and $\psi = \bigvee_{i=1}^{n} \psi_i$ with $m > 0$ and each ψ_i a conjunction of atoms. Similarly to the proof of Theorem 25, consider a fresh predicate Q and the ontology

$$\mathcal{O}_Q = \{ \psi_i(\mathbf{x}, \mathbf{y}) \to Q(\mathbf{x}) \mid 1 \leq i \leq n \}$$

as well as a module setting $\chi_Q = \langle \theta^{\mathsf{q}_0}, \mathcal{D}_0^Q, \mathcal{D}_r^Q \rangle$ for $\mathcal{O} \cup \mathcal{O}_Q$ and Σ, satisfying in particular $\mathcal{P}^{\chi_Q} = \mathcal{P}^{\Psi_0^{\mathsf{q}}(\mathcal{O},\Sigma)} \cup \mathcal{O}_Q$. As argued in the proof of Theorem 25, given $s \in \mathcal{O}'$ and a substitution σ mapping variables in $\mathbf{x}$ to fresh distinct constants, there must be a proof $\rho = (T, \lambda)$ of $Q(\mathbf{x})\sigma$ in $\kappa(\mathcal{O} \cup \mathcal{O}_Q) \cup \varphi\sigma$ such that $\kappa(s) \cap \mathsf{supp}(\rho) \neq \emptyset$. Furthermore, thanks to $\mathcal{O}'$ being a justification, we can assume that ρ is laconic, and in particular for each leaf node $v \in T$ and each ancestor w of v in T it must be $\lambda(v) \not\subseteq \lambda(w)$.

By Corollary 23, either $s = \bot \to \Box$ or there exists a proof $\rho' = (T', \lambda')$ in $\mathcal{P}^{\Psi_0^{\mathsf{q}}(\mathcal{O},\Sigma)} \cup (\varphi\sigma)\theta^{\mathsf{q}_0}$ of either $Q(\mathbf{x})\sigma$ or $\bot$ that is embeddable into ρ modulo θ^{q_0} and such that $\mathsf{supp}(\rho') \cap \Xi^{\Psi_0^{\mathsf{q}}(\mathcal{O},\Sigma)}(s) \neq \emptyset$.

If $s = \bot \to \Box$ then there must be some rule s' in $\mathcal{O}'$ such that $\bot \in \mathsf{Sig}(\mathcal{O}')$. As we show next, when considering the case $s \neq \bot \to \Box$, it must be $s' \in \mathcal{M}^\chi$. Therefore $\bot \in \mathsf{Sig}(\mathcal{M}^\chi)$, and consequently $s = \bot \to \Box \in \mathcal{M}^\chi$.

Otherwise, if $\rho' = (T, \lambda)$ is a proof of $Q(\mathbf{x})\sigma$, as discussed in the proof of Theorem 25, a rule from $\Xi^{\Psi_0^{\mathsf{q}}(\mathcal{O},\Sigma)}(r)$ must be used in some subproof $\rho'' = (T'', \lambda'')$ of ρ' that is a proof of $(\gamma\sigma')\theta^{\mathsf{q}_0}$ for some $\gamma \in \bigcup_i \psi_i$ and some extension σ' of σ to $\mathbf{y}$. Let v be the root of T'' and $w_1, \ldots, w_n$ its leaves, by definition it must be $\lambda''(v) = (\gamma\sigma')\theta^{\mathsf{q}_0}$.

If v is also a leaf in T'' then there must be a rule of the form $(\ \to (\gamma\sigma')\theta^{\mathsf{q}_0})$ in $\Xi^{\Psi_0^{\mathsf{q}}(\mathcal{O},\Sigma)}(s)$, and all the terms in $(\gamma\sigma')\theta^{\mathsf{q}_0}$ must be constants from the domain of θ^{q_0}. This implies $(\gamma\sigma')\theta^{\mathsf{q}_0} \in \mathcal{D}_r^{\mathsf{q}_0}$, and consequently $(\ \to (\gamma\sigma')\theta^{\mathsf{q}_0}) \in \mathsf{supp}(\Psi_0^{\mathsf{q}}(\mathcal{O}, \Sigma))$ and $s \in \mathcal{M}^{\Psi_0^{\mathsf{q}}(\mathcal{O},\Sigma)}$.

Suppose v is not a leaf of T''. First of all, note that $(\varphi\sigma)\theta^{\mathsf{q}_0} = \varphi\sigma$ due to θ_0^{q} not modifying constants and $\varphi\sigma$ not using functional terms, and therefore $\lambda''(w_i) \in \varphi\sigma$ for each i. If there are functional terms in $\gamma\sigma'$ then $(\gamma\sigma')\theta^{\mathsf{q}_0} \notin \varphi\sigma$, since both the constants c_y and the constants in the range of σ were fresh by hypothesis, and therefore $\lambda''(w_i) \neq \lambda''(v)$ for each i. Suppose now that there are no functional terms in $\gamma\sigma'$. Then $\gamma\sigma'\theta^{\mathsf{q}_0} = \gamma\sigma'$. We know that ρ', and hence ρ'', is embeddable into ρ modulo

θ^{q_0}. It follows that there must exist some node $v' \in T$ that is not a leaf of T and such that $\lambda(v) \subseteq \lambda(v')\theta^{\mathsf{q}_0}$, and also a collection of leaves $w'_1, \ldots, w'_n$ of T such that $\lambda''(w_i) = \lambda(w'_i)\theta^{\mathsf{q}_0} \in \varphi\sigma\theta^{\mathsf{q}_0}$. Since neither $\gamma\sigma'$ nor $\varphi\sigma'$ use functional terms, and θ_0^{q} does not modify constants, it must in fact be $\lambda(v) \subseteq \lambda(v')$ and $\lambda''(w_i) = \lambda(w'_i)$ for each i. Furthermore, we had that $\lambda(w_i) \not\subseteq \lambda(v')$ for each i, which implies that, also in this case $\lambda''(w_i) \neq \lambda''(v)$ for each i. Therefore, regardless of $\gamma\sigma'$ mentioning functional terms, we have $\lambda''(w_i) \neq \lambda''(v)$ for each i. Now, the fact $\gamma\sigma'$ must be of the form $B(\mathbf{c})$ with $B \in \Sigma$ since $\gamma \in \bigcup_i \psi_i$ and by hypothesis $r \in \mathsf{rel}_{\mathsf{q}}(\mathcal{O}, \Sigma)$. Let η be an injective substitution mapping all constants in $\mathbf{c} \cup \mathsf{range}(\theta)$ to $\{1, \ldots, \mathsf{arity}(B)+n+m\}$ satisfying *(i)* $c\eta = \mathsf{arity}(B) + i$ iff $c = c_i \in \mathsf{Ct}(\mathcal{O})$ and *(ii)* $c\eta = \mathsf{arity}(B) + n + i$ iff $c = c_{y_i}$. Let τ be another substitution defined on all constants mentioned by ρ'' and such that

$$c\tau = \begin{cases} *^{c\eta}_{B,\mathbf{c}\eta} & \text{if } c \in \mathbf{c} \cup \mathsf{range}(\theta) \\ *^{0}_{B,\mathbf{c}\eta} & \text{otherwise} \end{cases}$$

Note that τ is compatible with θ. Since $\mathbf{c}\eta \in \{1, \ldots, \mathsf{arity}(B) + n + m\}^{\mathsf{arity}(B)}$, it is easy to see that $\lambda''(v)\tau = B(*^{\mathbf{c}\eta}_{B,\mathbf{c}\eta}) \in \mathcal{D}^{\mathsf{f}_0}_r$. On the other hand, as we have previously observed, no $\lambda''(w_i)$ mentions constants from the set $\{c_{y_1}, \ldots, c_{y_m}\}$, hence $\lambda''(w_i)\tau$ must be of the form $A(*^{\mathbf{w}}_{B,\eta(\mathbf{c})})$ with $A \in \Sigma$ and $\mathbf{w} \in \{0, \ldots, \mathsf{arity}(B) + n\}^{\mathsf{arity}(A)}$. Furthermore, because $\lambda''(w_i) \neq \lambda''(v)$ for each i, it follows that also $\lambda''(w_i)\tau \neq \lambda''(v)\tau$ and thus $\lambda''(w_i)\tau \in \mathcal{D}^{\mathsf{f}_0}_0$ for each i. The proof $\rho''\tau = (T'', \lambda''_\tau)$ such that $\lambda''_\tau(v) = (\lambda''(v))\tau$ is clearly a proof in $\mathcal{P}^{\Psi^{\mathsf{q}}_0(\mathcal{O},\Sigma)} \cup \mathcal{D}^{\mathsf{q}_0}_0$ of $\lambda''(v)\tau$ that has the same support as ρ'', therefore $s \in \mathcal{M}^{\Psi^{\mathsf{q}}_0(\mathcal{O},\Sigma)}$.

Finally, if ρ' is a proof of $\bot$ then it must be a proof in $\mathcal{P}^{\Psi^{\mathsf{q}}_0(\mathcal{O},\Sigma)}$. Since by assumption it is $\bot \notin \varphi$, the labels of all the leaves in ρ' must also be different from $\bot$. To check that also in this case $s \in \mathcal{M}^{\Psi^{\mathsf{q}}_0(\mathcal{O},\Sigma)}$, it suffices to follow a similar argument to the one above with a mapping τ' defined on all constants mentioned in ρ' and such

that

$$c\tau = \begin{cases} c & \text{if } c \in \mathsf{Ct}(\mathcal{O}) \cup \{c_{y_1}, \ldots, c_{y_m}\} \\ *^0_{B,\mathbf{c}\eta} & \text{otherwise} \end{cases}$$

We have proved that the family Ψ^{q}_0 is q-admissible. Now we will use it to show that Ψ^{q} is not q-optimal. As in the proof for Theorem 102, if we consider the ontology $\mathcal{O} = \{A(x) \to B(x), B(x) \to A(x)\}$ and the signature $\Sigma = \{A\}$, we can observe that $\mathcal{M}^{\Psi^{\mathsf{q}}(\mathcal{O},\Sigma)} = \mathcal{O} \not\subseteq \emptyset = \mathcal{M}^{\Psi^{\mathsf{q}}_0(\mathcal{O},\Sigma)}$, and thus Ψ^{q} is not q-optimal. □