UNIVERSITY OF LIVERPOOL

Modularity Through Inseparability : Algorithms, Extensions, and Evaluation

Thesis submitted in accordance with the requirements of the University of Liverpool for the degree of Doctor in Philosophy by

WILLIAM ANDREW JAMES GATENS

May 2017

Abstract

Module extraction is the task of computing, given a description logic ontology and a signature Σ of interest, a subset (called a module) such that for certain applications that only concern Σ, the ontology can be equivalently replaced by the module. In most applications of module extraction it is desirable to compute a module which is as small as possible, and where possible a minimal one.

In logic-based approaches to module extraction the most popular way to define modules is using inseparability relations, the strongest and most robust notion of this being model Σ-inseparability, where two ontologies are called Σ-inseparable iff the Σ-reducts of their models coincide. Then, a Σ-module is defined as a Σ-inseparable subset of the ontology.

Unfortunately deciding if a subset of an ontology is a minimal Σ-module, over ontologies formulated in even moderately expressive logics, is of perpetually high complexity and often undecidable, and for this reason approximation algorithms are required. Instead of computing a minimal Σ-module one computes some Σ-module and the main research task is to minimise the size of these modules — to compute an approximation of a minimal Σ-module.

This thesis considers research surrounding approximations based on the model Σ-inseparability relation including: improving and extending existing approximation algorithms, providing a highly-optimised implementations, and the introduction a new methodology to evaluate just how well approximations approximate minimal modules, all supported by a significant empirical investigation.

For my brother,

Robert Edward Harry Gatens

1992 – 2015

Contents

List of Figures

CHAPTER 1

Introduction

Ontologies in computer and information science are a means of knowledge representation used to specify and establish the vocabulary of a domain of interest in order to facilitate the exchange of information. Ontologies have the benefit of being able to represent information to be computer understandable, typically specified in a logical language to underpin the represented knowledge with unambiguous semantics. An important class of ontologies are based on Description Logics, a family of knowledge representation languages consisting of several decidable fragments of first-order logic which allows the formulation of ontologies from logical formulas known as *axioms* [Baa+03].

The introduction of the World Wide Web Consortium (W3C) endorsed ontology language OWL built on a description logic foundation, along with the research and development of a variety of tools to assist ontology engineering has lead the existence of ontologies covering a wide variety of domains, including: Medicine [Spa00; Gol+11], Biology [Whe+11], Chemistry [Deg+08], Law [Hoe+07] and Geography [HKH08].

1.1. Producing high quality ontologies

Once a domain's vocabulary is established, an ontology is constructed by translating the conceptual knowledge of the chosen domain into that of the chosen ontology language, giving meaning to the terms which make up the vocabulary. How this is achieved is particularly important; for ontologies to be useful they should to be of *high quality*, accurately capturing the knowledge of the agreed conceptualisation, providing *coverage* for the vocabulary – not being under-specified, whilst also not being over-specified. Part of achieving this is making the correct choice of underlying logic, a trade-off between expressive power and complexity of deriving information from the ontology. The chosen logic should be expressive enough to represent the knowledge correctly whilst providing appropriate performance for the ontology in the desired application.

These considerations make ontology engineering an intensive task, typically a collaborative effort between ontology engineers and domain experts, it is also an ongoing task, ontologies are constantly in need of maintenance, repair and extension. In order to assist this process, research has been focussed into the development of number of tools including, but not limited to:

- **Editors and management systems** – to assist modelling and to promote collaborative development, notable examples include Protégé [TNM08] and SWOOP [Kal+06b].

- **Reasoners** – to reveal and debug the information explicitly encoded in the ontology's axioms. As previously mentioned, the complexity of standard reasoning procedures is relative to the expressive power of the underlying logic ranging from PTIME [Cal+07; BBL05] for the least expressive to 2EXPTIME for the most [Kaz08] in the size of the input ontology. Even with high worst-case complexity considerations, highly optimised reasoners mean using very expressive logics is still often feasible, notable examples of reasoners include: Fact++ [TH06], Racer [Haa+12], HermiT [Cla10] and Pellet [Sir+07].

- **Repair tools** – to debug and fix incorrectly specified knowledge and modelling errors [LB10a; Kal+06a].

- **Design patterns** – for guidance and methodologies for creating high quality ontologies [LST13] and tools for evaluating the results [GF95; Ric+14].

The development of many large and complex ontologies which are of high quality has been possible with these various supporting tools in place. Important examples include: Systematized Nomenclature of Medicine – Clinical Terms (SNOMED CT) [Spa00] a large medical terminology used in the healthcare system of over 20 countries, the National Cancer Institute's Thesaurus and Ontology (NCI) [Gol+11] used to facilitate the use and standardisation of terminology across the domain of cancer related research and, the gene ontology (GO) [The12] providing a consistent description of gene products in terms of their associated biological processes.

1.2. Reuse of ontologies

The design of certain ontologies puts particular emphasis on reusability. Foundational or "upper" ontologies are generic ontologies which are applicable to many domains, prominent examples include: the Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) [Gan+02] and the Basic Formal Ontology (BFO) [GSG04]. Beyond this, there are many motivating reasons which makes reusing pre-existing knowledge highly desirable, especially considering the significant undertaking required to produce a high-quality ontology. Reusing an existing ontology not only saves time and effort, but allows a modeller to draw on knowledge over a vocabulary which they may not be an expert in.

In practice, reusing one ontology can often be achieved by simply importing it into another — the OWL standard even provides an annotation (`owl:imports`) to describe precisely this procedure [Hit+09]. However, in information intensive domains such as the Life Sciences, ontologies can be of considerable size, for example a 2015 version of NCI consists of over 250,000 axioms which define around 100,000 different terms. These axioms represent a vocabulary covering a wide number of topics, including: Genes, Diseases, Organisms and Processes. Attempting to reuse some information contained within such a large ontology by reusing the whole ontology comes with the inherent risk of greatly increasing the size, engineering effort, and complexity of the target application, along with importing information which may be irrelevant. In order to reduce the impact and required effort of reuse, the notion of *modularity* has been developed to facilitate the *partial reuse* of ontologies.

A *module* is a subset of an ontology, a set of axioms, providing coverage for a relevant part of the ontology's vocabulary. A user may only be interested in reusing the knowledge the ontology provides about certain terms, and can specify this by means of a *signature*, a subset of vocabulary, and a module will contain all the information relevant to the terms the signature contains. The purpose of this is to allow modules to act as an equivalent replacement for the whole ontology in applications which only consider only this restricted vocabulary.

For modules to be useful they should be as small as possible, allowing the reuse of relevant information without significant overhead. Ideally modules should only contain the axioms providing relevant knowledge, disregarding those which are deemed irrelevant.

1.2.1 Modules

The underlying motivation for modules creates the research task of deciding, given a signature, which axioms of the ontology the module should contain to provide all the relevant information – a task known as *module extraction*. For an ontology $\mathcal{O}$, the methodologies to extract a module $\mathcal{M} \subseteq \mathcal{O}$ for a signature Σ can be roughly divided into two categories:

- **Structural** – syntactically traverse through the axioms of the ontology using some heuristic to determine which axioms are relevant to the desired signature.

- **Logical** – extract modules satisfying desirable logical properties in order to preserve the knowledge contained within the axioms of an ontology.

Structural Approaches

Notable structural approaches include the PROMPT-FACTOR [NM03] tool, which given a signature Σ, goes through the axioms of the ontology $\mathcal{O}$ and adds to a module $\mathcal{M}$ any axiom which use a symbol from Σ. The signature is then expanded with any new symbols occurring in $\mathcal{M}$ and this process is repeated until a fixed point is reached. Another structural approach [SS09] used for the segmentation of the medical ontology GALEN, utilises a different heuristic to collect axioms that define signature symbols, pruning irrelevant axioms by traversing the hierarchical structure of the ontology "upwards" towards the most general symbols.

The inherent limitation of structural-based approaches is that they ignore the semantics of the ontology. Purely syntactical approaches can often collect axioms semantically irrelevant to the signature, or even worse, miss axioms which do convey relevant semantic information [Gra+07]. In this they have limited use as modules, they cannot reliably be used as a replacement for the entire ontology as they may represent different knowledge over the chosen signature.

Considering the effort to model ontologies to be of high quality, to accurately capture conceptual knowledge and to preserve this knowledge through re-use, logic-based module extraction approaches have become increasingly desirable.

Logical Approaches

The aim of a logic based module $\mathcal{M}$ is to preserve the semantic knowledge encoded in an ontology $\mathcal{O}$ with respect to a signature Σ.

One approach is to use Modular Ontology Languages (MOL), a family of formalisms which provide new syntax and semantics building ontologies which allows knowledge to be separated into distinct modules at development time. Notable examples include $\mathcal{E}$-Connections [Kut04; CPS06], Distributed Description Logics (DDL) [BS03] and Package-based Description Logics

(P-DL) [BCH06a; BCH06b]. With the boundaries of the knowledge clearly demarcated it is apparent where information to be re-used is located within the ontology. Unfortunately, the expressive powers of such ontologies are limited in order to maintain the modular structure of the knowledge, additionally, non-standard semantics means one may require non-standard tools to perform key tasks such as reasoning.

A preferred alternative is to maintain the standard ontology representation by building ontologies in a conventional way, using description logic axioms, and to extract modules from these ontologies directly to meet desirable logical properties in order to guarantee knowledge preservation. This typically amounts to the preservation of models or entailments over a given signature. With the desire for modules to be free of redundant information, an additional task is to produce a module which is small as possible, computing *minimal modules*. In addition to preserving knowledge, the underlying semantics of logic-based modules allows one to produce a module that will not cause unintended interactions in the common import scenario [Gra+08].

Several different notions for these logical modules exist, the majority of which are based around extracting a module which satisfies a so called *inseparability relation*, a family of equivalence relations which generalise conservative extensions [GLW06; LWW07]. If a module is inseparable for a signature Σ, it is logically indistinguishable from the original ontology over the vocabulary of Σ, and so comes with the guarantee to preserve all the knowledge over Σ. Several inseparability relations exists of varying strength depending on which logical properties need preserving based on the requirements of the desired application, the strongest is that of model-inseparability in which the extracted module is guaranteed to preserve every second-order entailment over Σ [Kon+09a]. In addition, the guarantee of knowledge preservation over a signature allows logical modules to be utilised in applications beyond the import scenario, this includes: reasoning [Gra+10; GPS12b; RGH12; TP12b], forgetting/hiding [KS13; LK13; LK14], logical difference [GPS12a; LK14], locating justifications [Sun+08; BS08], and matching [NK10].

Unfortunately the decision problems associated with logical modules based on inseparability relations, including that of model-inseparability, are perpetually of high complexity and often undecidable. As a result algorithms to produce minimal inseparability-based modules are only available for ontologies formulated in inexpressive description logics [Kon+09b; KWZ10; Kon+08a; Kon+13]. For more expressive logics, the undecidability considerations has driven research into production of practical algorithms that produce sound *approximations*, modules which still preserve the desired logical properties but which do not come with a guarantee of minimality. Nevertheless, one still wants to achieve a module which is as small as possible – to approximate minimal modules.

Different methodologies exist for computing approximations, including those based on model-theoretic notions of inseparability [Kon+13; Kon+08a], graph theory [NBM13], and datalog reasoning [Rom+14]. The most popular kind however, are those based on locality [Gra+08; SSZ09], applicable to very expressive logics and can produce a sound approximation of a module satisfying the model-inseparability relation using polynomial time algorithms. One limitation of the locality approach is often producing modules that are a lot larger than they need, containing many surplus axioms irrelevant to the specified signature, which may limit their usage in certain applications. As a result, there is still the need for the development of better approximations which contain fewer redundant axiom and which therefore better approximate minimal modules.

1.3. Contributions

The contributions presented in this thesis are motivated by the desire to understand how we can improve upon the approximations of minimal modules in expressive logics. To do this we will develop new approximation algorithms which can extract modules satisfying the model-inseparability relation in an attempt to improve on the corresponding approximations — those extracted for the same signature — produced by existing approaches. To measure the extent of the improvements our new approximations offer, we evaluate them against the following research questions which we will investigate over the course of this thesis.

Difference in module size

As we mentioned in the previous section, existing approaches can produce sound approximations which can be a lot larger than they need to be, and it is desirable to minimise these approximations. If also we produce a corresponding module which is a sound approximation but is also relatively smaller in size, we obtain the same logical guarantees of knowledge preservation but reduce the overhead of reusing the knowledge preserved within the module. This gives us the research question:

- How large and significant is the difference in size between the approximations we compute and the size of existing approximations?

Minimality

The main research task is still to achieve a module which is a small as possible, and ideally minimal, and although our approximations may be relatively smaller than the approximations produced by other approaches, they may still be considerably larger than the minimal modules they approximate. This leads to the next research question:

- How close in size are the approximations we compute to the minimal modules they approximate?

Performance

Finally, for our new approximations to be most useful, computing an approximation shouldn't come with significant overhead, especially considering the most popular existing approach based on locality is already known to be practically efficient [Del+12; Ves+13]. So our last research question is:

- Using our approaches, how much time does it take on average, and in the worst case, to compute an approximation?

1.3.1 Contributions of this Thesis

Towards answering the given research questions, this thesis makes new advances in the area of research surrounding the approximation of minimal modules in several ways:

Improving approximations. We present a new approximation algorithm which extends the model-theoretic approach for approximating minimal modules. This new algorithm offers notable improvements over its predecessor, namely it supports computing approximations from a wider range of ontologies by extending the model theoretic notions of modularity to more expressive description logics. It is also highly optimised, by examining and exploiting properties of the original approximation algorithm we obtain an algorithm that offers measurably better practical performance. More importantly, we also show this new algorithm can often produce modules that are significantly smaller than the corresponding approximations produced by competing approaches.

Combining approximations. As we discussed in the previous section there are several different notions which lead to the production of approximations for minimal modules defined over the model-inseparability relation which in turn can lead to the content of corresponding approximations produced by different approaches varying considerably. Each sound approximation contains the minimal module so it is the surplus axioms which do not preserve any knowledge over the specifying signature for which the approximations differ. From this observation we introduce a general way of combining two different approximations together into a single approximation algorithm which enables for the extraction of better approximations by discarding those axioms which are not considered relevant over the signature by *both* approximations. The modules produced this way are shown to be at least as small as the approximations produced by either approximation independently but may be even smaller.

Evaluating approximations. Evaluating the success of existing approximations is currently an open problem. If an approximation produced by one method is smaller than the corresponding approximation produced by another it is only comparatively closer to the minimal module it approximates, and may still be significantly larger. Nothing is currently known about how large and significant the difference in size is between approximations and minimal modules. In answer to this we introduce a novel approach which can estimate, for the first time, the difference in size between an approximation and its minimal module. This result being of particular importance to research surrounding approximations, as it can help identify those cases where approximations contain a large number of axioms which do not convey relevant knowledge about the specifying signature, so that they might be extended and improved.

We also present the results of large empirical investigation which not only shows the approximations which we produce are often smaller than rival approaches, but that they very successful approximations, and often coincide with the ideal minimal modules. In addition, we found each of our introduced approximations could be computed very efficiently in practice which we evaluated over a corpus of real-world ontologies. These results combined, we showed there is strong empirical evidence to prefer our new improved approximations in combination with existing approximations, rather than just existing approximations on their own.

1.3.2 Structure of this Thesis

The thesis is organised as follows:

- Chapter 2 is split into two parts. In the first part we introduce the syntax and semantics of relevant logical formalisms relevant to the thesis which includes Description Logics. The second part explores existing logic-based approaches to modularity which provides the setting for our new contributions.

- Chapter 3 is focussed on developing the improved module extraction algorithm which is called AMEX. Included is a small experimental evaluation to showcase the improved performance of the AMEX algorithm.

- In Chapter 4 we present a general methodology for combining approximation extraction procedures together. We also look at the specific case of combining the newly introduced AMEX algorithm with a locality-based notion of computing approximations.

- In Chapter 5 we introduce the methodology for evaluating the success of approximations — how significant the difference is between an approximation and its corresponding minimal module — and establish how this can be determined.

- Chapter 6 brings together all the results from the previous chapters into an extensive experimental evaluation in order to answer our proposed research questions. In this chapter we compare the approximation algorithms we have developed alongside the most popular rival approach across a corpus of real-world ontologies. We also use the methodology described in Chapter 5 to evaluate the success of each approximation.

- Finally, we have the Appendix which contains more detailed information about the ontologies our experimental evaluation and more in-depth breakdown of the results of the experiments.

Large portions of Chapter 3 along with some of the experimental results from Chapter 6 were published in the paper [GKW13] which was presented at the Workshop on Modular Ontologies (WOMO) 2013. These results were later extended in the paper [GKW14] which included the theoretical results from Chapter 5 and additional results from Chapter 6, this paper was presented at the European Conference on Artificial Intelligence (ECAI) 2014.

CHAPTER 2

Background

In this chapter we look at how ontologies can be formulated from one of the many logics that belong to the Description Logic family, outlining the syntax and semantics which allow conceptual knowledge to be described in an unambiguous way. Additionally, we describe how signatures are constructed to describe topics of interest from the global vocabulary of an ontology allowing the desired content of a module to be specified.

We will go on to explore current approaches to extracting logic-based modules from Description Logic ontologies which satisfy an *inseparability relation*, a class of equivalence relations which guarantees that a module preserves all the knowledge from an ontology over a given signature, distinguishing when minimal modules of these kinds are computable and when approximations are required.

2.1. Description Logics

Description Logics (DL) are decidable fragments of first-order logic (FOL) typically used to model a domain of interest by describing the relevant concepts of that domain, and the interrelations between them. The syntax of Description Logics is based on three countably infinite and disjoint sets of atomic elements:

- **individuals**, denoted $\mathsf{N_I}$, corresponding to FOL constants, used to denote single entities within the domain. Examples of individuals would be graham, liverpool or fido.
- **atomic concepts**, denoted, $\mathsf{N_C}$, also called *concept names*, correspond to

FOL unary predicates, used to describe the classes of individuals in the domain. Examples of atomic concepts would be, Otter, Person or Hospital

- **atomic roles**, denoted $\mathsf{N_R}$, also called *role names*, correspond to FOL binary predicates, used to relate concepts together. Examples of atomic roles would be hasPart, locatedIn, or eats.

The formal semantics of a DL is given by *interpretations* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where the *domain* $\Delta^{\mathcal{I}}$ is a non-empty set and $\cdot^{\mathcal{I}}$ is an *interpretation function* that maps each $A \in \mathsf{N_C}$ to a subset $A^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$ and each $r \in \mathsf{N_R}$ to a binary relation $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. The size of an interpretation $\mathcal{I}$ is given by the number of elements in its domain which is denoted by $\sharp\Delta^{\mathcal{I}}$. An interpretation is then called an n-element interpretation if $\sharp\Delta^{\mathcal{I}} = n$.

Family	Name	Syntax	Semantics
$\mathcal{EL}$	Top concept	$\top$	$\Delta^{\mathcal{I}}$
	Intersection	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
	Existential Restriction	$\exists r.C$	$\{d \in \Delta^{\mathcal{I}} \mid \exists e(d,e) \in r^{\mathcal{I}} \wedge e \in C^{\mathcal{I}}\}$
$\mathcal{AL}$	Top concept	$\top$	$\Delta^{\mathcal{I}}$
	Bottom concept	$\bot$	$\emptyset$
	Intersection	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
	Atomic Negation	$\neg A$	$\Delta^{\mathcal{I}} \setminus A^{\mathcal{I}}$
	Limited Existential Restriction	$\exists r.\top$	$\{d \in \Delta^{\mathcal{I}} \mid \exists e(d,e) \in r^{\mathcal{I}}\}$
	Universal Restriction	$\forall r.C$	$\{d \in \Delta^{\mathcal{I}} \mid \forall e(d,e) \in r^{\mathcal{I}} \rightarrow e \in C^{\mathcal{I}}\}$

Figure 2.1: The families $\mathcal{EL}$ and $\mathcal{AL}$

A DL also provides *constructors*, particular logical symbols which admit the inductive construction of *complex concepts* and *complex roles*. The expressive power or *expressitivity* of a DL is defined by the constructors which the logic permits. We consider two base description logics, $\mathcal{AL}$ and $\mathcal{EL}$ and their extensions throughout this thesis, the constructors available in these respective logics and the corresponding semantics can be found in Figure 2.1.

One may extend a base language by introducing additional constructors. Figure 2.2 lists commonly used DL constructors along with the semantics and corresponding symbol used to represent that constructor. The name of the resulting logic is obtained from name of the base logic, plus the symbols which represent any additional constructors used, for example, the logic $\mathcal{ALCI}$ is the language $\mathcal{AL}$ extended with both concept negation and inverse roles.

In more expressive languages, particularly those admitting concept negation, some expressions can serve as syntactic abbreviations for others (logically equivalent alternatives) notably: $\bot$ for $\neg\top$, $C \sqcup D$ for $\neg(\neg C \sqcap \neg D)$, $\forall r.C$ for $\neg\exists r.\neg C$. Should the language additionally permit unqualified number restrictions further abbreviations can be used, $(\geqslant 1\ r.C)$ for $\exists r.C$, $(\leqslant 0\ r.\neg C)$ for $\forall r.C$, $(\leqslant n\ r.C)$ for $\neg(\geq (n+1)\ r.C)$ and $(= n\ r.C)$ for $(\leqslant n\ r.C) \sqcap (\geqslant n\ r.C)$.

Symbol	Name	Syntax	Semantics
$\mathcal{U}$	Union	$C \sqcup D$	$C^\mathcal{I} \cup D^\mathcal{I}$
$\mathcal{C}$	Concept Negation	$\neg C$	$\Delta^\mathcal{I} \setminus C^\mathcal{I}$
$\mathcal{N}$	Unqualified Number Restriction	$(\geq n\ r.C)$	$\{d \in \Delta^\mathcal{I} \mid \lvert\{e \mid (d,e) \in r^\mathcal{I}\}\rvert \geq n\}$
		$(\leq n\ r.C)$	$\{d \in \Delta^\mathcal{I} \mid \lvert\{e \mid (d,e) \in r^\mathcal{I}\}\rvert \leq n\}$
		$(= n\ r.C)$	$\{d \in \Delta^\mathcal{I} \mid \lvert\{e \mid (d,e) \in r^\mathcal{I}\}\rvert = n\}$
$\mathcal{Q}$	Unqualified Number Restriction	$(\geq n\ r.C)$	$\{d \in C^\mathcal{I} \mid \lvert\{e \mid (d,e) \in r^\mathcal{I}\}\rvert \geq n\}$
		$(\leq n\ r.C)$	$\{d \in C^\mathcal{I} \mid \lvert\{e \mid (d,e) \in r^\mathcal{I}\}\rvert \leq n\}$
		$(= n\ r.C)$	$\{d \in C^\mathcal{I} \mid \lvert\{e \mid (d,e) \in r^\mathcal{I}\}\rvert = n\}$
$\mathcal{O}$	Nominals	$a^\mathcal{I}$	$\{a\}^\mathcal{I}$
$\mathcal{I}$	Inverse Role	r^-	$\{(e,d) \in \Delta^\mathcal{I} \times \Delta^\mathcal{I} \mid (d,e) \in r^\mathcal{I}\}$
	Where C, D are concepts, r, s are roles and a an individual		

Figure 2.2: Description Logic Constructors

A description logic *ontology* is a tuple $\mathcal{O} = (\mathcal{T}, \mathcal{R})$ consisting of **TBox** $\mathcal{T}$ describing the interrelations between (complex) concepts and an **RBox** $\mathcal{R}$ which does the same but for (complex) roles. These interrelations are modelled using a finite set of *axioms*, well-formed formulas constructed using special logical symbols. The syntax and semantics of TBox and RBox axioms relevant to the thesis can be found in Figure 2.3 and Figure 2.4 many of which also have a symbol representing if they are permitted in the chosen DL.

There are some exceptions to this naming scheme which describe particular

Name	Syntax	Semantics
Concept Inclusion	$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
Concept Equivalence	$C \equiv D$	$C^{\mathcal{I}} = D^{\mathcal{I}}$

Figure 2.3: TBox Axioms

Symbol	Name	Syntax	Semantics
$+$	Transitivity	$\mathsf{Trans}(r)$	$(\bigcup_{i\geq 1}(r^{\mathcal{I}})^i) = r^{\mathcal{I}}$
$\mathcal{F}$	Functionality	$\mathsf{Func}(r)$	$\forall d \in \Delta^{\mathcal{I}} \lvert\{e \in \Delta^{\mathcal{I}} \mid (a,b) \in r^{\mathcal{I}}\}\rvert \leq 1$
$\mathcal{H}$	Role Inclusion	$s \sqsubseteq t$	$s^{\mathcal{I}} \subseteq t^{\mathcal{I}}$
	Role Equivalence	$s \equiv t$	$s^{\mathcal{I}} = t^{\mathcal{I}}$
$\mathcal{R}$	Role Inclusion	$s \sqsubseteq t$	$s^{\mathcal{I}} \subseteq t^{\mathcal{I}}$
	Role Equivalence	$s \equiv t$	$s^{\mathcal{I}} = t^{\mathcal{I}}$
	Complex Role Inclusion	$r_1 \circ r_2 \sqsubseteq r$	$(r_1 \circ r_2)^{\mathcal{I}} \sqsubseteq r^{\mathcal{I}}$
	Role Disjointness	$\mathsf{Disj}(r_1, r_2)$	$r_1^{\mathcal{I}} \cap r_2^{\mathcal{I}} = \emptyset$
	Symmetric Role	$\mathsf{Symm}(r)$	$r^{\mathcal{I}} = (r^-)^{\mathcal{I}}$
	Asymmetric Role	$\mathsf{Asymm}(r)$	$r^{\mathcal{I}} \cap (r^-)^{\mathcal{I}} = \emptyset$
	Reflexive Role	$\mathsf{Refl}(r)$	$\forall d \in \Delta^{\mathcal{I}}, (d,d) \in r^{\mathcal{I}}$
	Irreflexive Role	$\mathsf{Irrefl}(r)$	$\forall d \in \Delta^{\mathcal{I}}, (d,d) \notin r^{\mathcal{I}}$
Where r and $r_1, r_2, \ldots, r_n$ are atomic roles and s and t are possibly complex roles			

Figure 2.4: RBox Axioms

families of languages, notable exceptions which have relevance to this thesis are: $\mathcal{S}$ – equivalent to $\mathcal{ALC}+$, which is often extended itself with further constructors, for example, $\mathcal{SHIQ}$ is equivalent to $\mathcal{ALCIQ}+$; DL-Lite – a syntactically restricted sub-language of $\mathcal{SHIF}(D)$; and $\mathcal{EL}++$ – equivalent to $\mathcal{ELOR}$.

Note in some cases an ontology may have an empty RBox in such a case we just refer to the ontology as a TBox. We also need to refer to an ontology containing no axioms, the empty TBox which we denote $\emptyset$.

The *expressitivity* of an ontology $\mathcal{O}$ is defined by the expressitivity of the description logic $\mathcal{L}$ needed to express all the axioms contained within $\mathcal{O}$ such that there is no logic $\mathcal{L}' \subseteq \mathcal{L}$ able to express all the axioms of $\mathcal{O}$, we call such an ontology an $\mathcal{L}$ ontology. Example 2.1.1 shows an $\mathcal{ALCQI}$ ontology with an accompanying natural language translation.

Example 2.1.1 (An $\mathcal{ALCQI}$ TBox describing bees)**.**

$$\mathcal{T} = \{\mathsf{Bee} \equiv \mathsf{Drone} \sqcup \mathsf{Worker} \sqcup \mathsf{Queen} \tag{2.1}$$

$$\mathsf{Bee} \sqsubseteq \forall \mathsf{eats.Honey} \sqcap \exists \mathsf{eats}^{-}.\mathsf{Bird} \tag{2.2}$$

$$\mathsf{Hive} \sqsubseteq (= 1 \ \mathsf{has.\ Queen})\} \tag{2.3}$$

A translation into natural language could be as follows – (2.1) a Bee *is defined as a* Drone *or a* Worker *or a* Queen*, (2.2) A Bee is a member of the sets of things which only* eats Honey AND *are* eaten by *a* Bird*, (2.3) a* Hive has exactly *one* Queen.

The satisfiability of ontologies is decided by means of interpretations, an interpretation $\mathcal{I}$ *satisfies* an axiom α if the formula $\alpha^{\mathcal{I}}$, obtained by mapping each entity in $\mathsf{sig}(\alpha)$ using the interpretation function $\cdot^{\mathcal{I}}$, is logically *true*. If $\mathcal{I}$ satisfies an axiom α we say $\mathcal{I}$ is a *model* of α, written $\mathcal{I} \models \alpha$. If $\mathcal{I}$ satisfies every axiom of an ontology $\mathcal{O}$ we say $\mathcal{I}$ is a model of $\mathcal{O}$, written $\mathcal{I} \models \mathcal{O}$. If every possible interpretation is a model of an axiom α (resp. ontology $\mathcal{O}$) we say α (resp. $\mathcal{O}$) is a *tautology*.

One of the most common reasoning tasks is to reveal what information can be inferred from an ontology even if it is not explicitly encoded in the form of axioms. If every model of an ontology $\mathcal{O}$ is also a model of some axiom α, we say α is an entailed from $\mathcal{O}$ or α follows from $\mathcal{O}$, if so we write $\mathcal{O} \models \alpha$. This simple example illustrates how entailments are not necessarily explicitly encoded in the axioms of an ontology but are derived from them.

Example 2.1.2 (Entailment)**.** *Consider the following ontology $\mathcal{O}$ consisting of the axioms $\mathcal{E}_1$ and $\mathcal{E}_2$.*

$$A \sqsubseteq B \tag{$\mathcal{E}_1$}$$

$$B \sqsubseteq \exists r.C \tag{$\mathcal{E}_2$}$$

First notice the axioms of an ontology are always entailments by definition, *clearly if for some interpretation $\mathcal{I}$ if it holds that $\mathcal{I} \models \mathcal{O}$ we must also have $\mathcal{I} \models \mathcal{E}_i$ for $i = 1, 2$. We can also derive implicit entailments from $\mathcal{O}$. Derived from $\mathcal{E}_2$ we can see it holds that $\mathcal{O} \models B \sqsubseteq \exists r.\top$, and from $\mathcal{E}_1$ and $\mathcal{E}_2$ together we can infer that $\mathcal{O} \models A \sqsubseteq \exists r.C$, both $B \sqsubseteq \exists r.\top$ and $A \sqsubseteq \exists r.C$ are also entailments of $\mathcal{O}$.*

Entailments that can be inferred from every ontology (i.e. tautologies) are known as trivial entailments for example: $A \sqsubseteq \top$, $A \sqsubseteq A$ or $\bot \sqsubseteq \top$.

Web Ontology Language - OWL

The Web Ontology Language (OWL) is a family of languages each built on a description logic foundation. The OWL family was designed to provide standardisation for authoring ontologies and is recommended by the World Wide Web Consortium (W3C). In addition to the logical underpinning, the OWL standard provide non-logical features such as support for annotations, describing common operations such as importing one ontology into another.

The current version, OWL 2, defines several profiles (or species), language variants which have been specially selected based on their levels of expressitivity and associated computational complexity, the idea being the ontology author selects the profile which best suits their intended application. The most important profiles, relevant to this thesis, in increasing order of expressitivity:

- OWL EL — based on $\mathcal{EL}++$ — an extension of $\mathcal{EL}$ for which standard reasoning tasks are still tractable [BBL05].
- OWL QL — based on DL-Lite — which offers low computational complexity for standard reasoning and query answering, particularly useful in applications using large amounts of data, where query answering is the most important task [Cal+07].
- OWL DL - based on $\mathcal{SROIQ}$ - designed to provide the highest expressitivity whilst maintaining computational completeness and decidabil-

ity [HKS06; Kaz08].

2.1.1 Signatures

In the Introduction we described the role of signatures to specify which subset of the ontology's vocabulary for which module should provide coverage, in practice this is simply a subset taken from the concept and role names from which the ontology is constructed.

A *signature* Σ is a finite subset of concept and role names i.e $\Sigma \subseteq (\mathsf{N_C} \cup \mathsf{N_R})$. The signature $\mathsf{sig}(C)$ ($\mathsf{sig}(\alpha), \mathsf{sig}(\mathcal{O})$) of a concept C (axiom α, ontology $\mathcal{O}$ resp.) is the set of concept and role names that occur in C ($\alpha, \mathcal{O}$, respectively), discarding any non-logical symbols such as concept or axiom constructors. If a $\mathsf{sig}(C) \subseteq \Sigma$ we call C a Σ-concept. We refer to the members of a signature as *entities* or *symbols*. As an example:

$$\mathsf{sig}(\mathsf{Hive} \sqsubseteq (= 1\mathsf{has}.\mathsf{Queen})) = \{\mathsf{Hive}, \mathsf{has}, \mathsf{Queen}\}$$

Sometimes it is useful to describe when two interpretation interpret signature symbols in an identical way. Given a signature Σ the Σ-reduct $\mathcal{I}|_\Sigma$ of an interpretation $\mathcal{I}$ is obtained from $\mathcal{I}$ by by setting $\Delta^{\mathcal{I}|_\Sigma} = \Delta^{\mathcal{I}}$ and $X^{\mathcal{I}|_\Sigma} = X^{\mathcal{I}}$ for all $X \in \Sigma$ and $X^{\mathcal{I}|_\Sigma} = \emptyset$ for all $X \notin \Sigma$. Two interpretations $\mathcal{I}$ and $\mathcal{J}$ are said to *coincide* on a signature Σ if $\mathcal{I}|_\Sigma = \mathcal{J}|_\Sigma$.

2.1.2 Acyclic Terminologies

Acyclic terminologies are a class of ontologies consisting of only a TBox which is constructed in a restricted way. The construction of acyclic terminologies limits the expressivity of the ontology as a whole but often offers better computational complexity than general (unrestricted) ontologies – a particular case pertaining to module extracting we will explore in Section 2.3. Furthermore it is often the case large parts of popular high quality ontologies conform to the restricted acyclic terminology specification, this includes both NCI and SNOMED CT.

Terminology

A TBox $\mathcal{T}$ is called a terminology if it satisfies the following two conditions:

1. All axioms of the TBox are of the form $A \equiv C$ or $A \sqsubseteq C$ where A is a concept name.
2. No concept name occurs more than once on the left-hand side of an axiom.

Any TBox that is *not* a terminology is referred to as a *general* TBox. When describing a terminology we often need to refer to the axioms it contains without distinguish between concept inclusions and equivalances, in this case we use the notation $A \bowtie C$ to describe an axiom which is of the form $A \equiv C$ or $A \sqsubseteq C$.

Acyclicity

To describe acyclicity in the context of terminologies we use the the depends relation $\prec_{\mathcal{T}} \subseteq \mathsf{N_C} \times (\mathsf{N_C} \cup \mathsf{N_R})$, which is defined by setting $A \prec_{\mathcal{T}} X$ if there exists an axiom $A \sqsubseteq C$ or $A \equiv C$ in $\mathcal{T}$ such that $X \in \mathsf{sig}(C)$. A terminology $\mathcal{T}$ is then called *acyclic* if the transitive closure $\prec_{\mathcal{T}}^{+}$ of $\prec_{\mathcal{T}}$ is irreflexive. Intuitively, a terminology is acyclic if it never defines any concept names in terms of themselves. Any terminology which is not acyclic is called *cyclic*.

Example 2.1.3. *(Acyclic Terminology)*

$$
\begin{aligned}
\mathcal{T} = \{\mathsf{Worker} &\sqsubseteq \forall \mathsf{eats.Honey} & \mathcal{T}' = \{\mathsf{Worker} &\sqsubseteq \forall \mathsf{eats.Honey} \\
\mathsf{Queen} &\sqsubseteq \exists \mathsf{hasParent}^{-}.\mathsf{Bee} & \mathsf{Worker} &\sqsubseteq \exists \mathsf{hasParent.Queen} \\
\mathsf{Bee} &\sqsubseteq \forall \mathsf{collects.Nectar}\} & \mathsf{Queen} &\sqsubseteq \exists \mathsf{hasParent}^{-}.\mathsf{Worker}\}
\end{aligned}
$$

$\mathcal{T}$ (left) is a valid acyclic terminology, $\mathcal{T}'$ (right) is a general cyclic TBox, the concept Worker *is repeated on the left hand side of the first two axioms and is also defined in terms of itself:* Worker $\prec_{\mathcal{T}}^{+}$ Worker.

2.2. Quantified Boolean Formulas

Several technical results of this thesis require a reduction to Quantified Boolean formulas (QBF) which are an extension of propositional formulas using quantifiers. We use the widely accepted syntax and semantics for QBF formulas, recounted here for completeness and to establish nomenclature for the sections that follow.

Propositional Formulas

Syntactically, formulas in propositional logic are built from *atoms* (also called *propositional variables*) of which we assume there are a countably infinite amount which we denote by letters such as p, q, r etc. which may be logically *true* or *false*. Atoms are combined using logical operators (also known as connectives) to produce formulas – an atom itself being a valid propositional formula – more complex formulas are inductively defined using the operators outlined in Figure 2.5.

Name	Syntax
Conjunction	$\alpha \wedge \beta$
Disjunction	$\alpha \vee \beta$
Negation	$\neg\alpha$

Figure 2.5: Propositional syntax

We use syntactic abbreviations for certain complex formulas: implication $\alpha \rightarrow \beta$ for $\neg\alpha \vee \beta$ and equivalence $\alpha \leftrightarrow \beta$ for $(\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha)$.

To provide a standard representation for propositional formulas it is useful to use a normal form such as Conjunctive Normal Form (CNF) which is built from *literals* – a literal being an atom p or its negation $\neg p$.

Definition 2.2.1 (Clause)**.** *A clause is a formula $\gamma = (L_1 \vee, \ldots, \vee L_n)$ where L_i ($1 \leq i \leq n$) is a literal.*

Definition 2.2.2 (Conjunctive Normal Form)**.** *A formula α is in conjunctive normal form (CNF) if and only if α is a conjunction of clauses $\alpha = \gamma_1 \wedge, \ldots, \wedge \gamma_n$ where γ_i ($1 \leq i \leq n$) is a clause.*

For every propositional formula there exists a logically equivalent one in CNF which can be achieved through the application of transformation rules which unfortunately can lead to exponential explosion of the formula [BL99a], for example the formula:

$$(x_1 \wedge y_1) \vee (x_2 \wedge y_2) \vee \cdots \vee (x_n \wedge y_n)$$

when transformed into CNF produces a formula with 2^n clauses. One way to avoid this is to use a short normal form translation such as the encoding described in [Tse68], by the introduction of fresh atoms one can produce a CNF formula which is not logically equivalent to the original formula but does preserve equisatisfiablity.

Quantified Boolean Formulas

Quantified Boolean formulas (QBF) are an extension to propositional formulas admitting universal ($\forall$) and existential ($\exists$) quantifiers. For the purposes of this thesis we only consider and describe QBF formulas in a normal form called *prenex form* which all QBF formula can be transformed into whilst maintaining logical equivalence [Bie+09].

Definition 2.2.3 (Prenex QBF Syntax)**.** *The syntax of well-formed QBF formulas is defined inductively:*

1. *A propositional formula is a well-formed QBF formula*
2. *If ψ is a well-formed QBF formula then for propositional variables x, y the formulas $\forall x \psi$ and $\exists y \psi$ are well-formed QBF formulas*
3. *Only formulas given by 1. and 2. are well-formed QBF formulas*

QBF formulas in prenex form consist of a sequence of quantifiers, called the *prefix* followed by a propositional formula called the *matrix*. A QBF formula Φ is called *closed* if every variable in the matrix of Φ also appears in the prefix of Φ. A variable occurring in the prefix is called *bound*. All other variables which occur in the matrix and not the prefix are called *free* i.e. all variables occurring in the matrix are either free or bound.

An example closed QBF formula in prenex form:

$$\forall p \exists q \exists r (p \vee q) \wedge r$$

If a quantifier is used more than once sequentially within a prefix it is often omitted to provide a compact representation e.g. $\forall p \exists q, r (p \vee q) \wedge r$ is equivalent to the example given above.

Definition 2.2.4 (QBF semantics). *A truth assignment* $\mathfrak{v}$ *given free variables* $\{y_1, \ldots, y_n\}$ *is a mapping:*

$$\mathfrak{v} : \{y_1, \ldots, y_n\} \rightarrow \{\textit{true}, \textit{false}\}$$

Which is extended to give a truth assignment of an arbitrary QBF formula Φ *defined inductively on the structure of* Φ*:*

$$\begin{aligned}
\Phi = y_i : \mathfrak{v}(\Phi) = \mathfrak{v}(y_i) \textit{ for } 0 \leq i \leq n \\
\Phi = \neg\Phi' : \quad & \mathfrak{v}(\neg\Phi') = \textit{true} \iff \mathfrak{v}(\Phi') = \textit{false} \\
\Phi = \Phi_1 \wedge \Phi_2 : \quad & \mathfrak{v}(\Phi_1 \wedge \Phi_2) = \textit{true} \iff \mathfrak{v}(\Phi_1) = \mathfrak{v}(\Phi_2) = \textit{true} \\
\Phi = \Phi_1 \vee \Phi_2 : \quad & \mathfrak{v}(\Phi_1 \vee \Phi_2) = \textit{true} \iff \mathfrak{v}(\Phi_1) = \textit{true or } \mathfrak{v}(\Phi_2) = \textit{true} \\
\Phi = \exists y \Phi' : \quad & \mathfrak{v}(\exists y \Phi') = \textit{true} \iff \mathfrak{v}(\Phi'[y/0]) = \textit{true or } \mathfrak{v}(\Phi'[y/1]) = \textit{true} \\
\Phi = \forall x \Phi' : \quad & \mathfrak{v}(\forall x \Phi') = \textit{true} \iff \mathfrak{v}(\Phi'[x/0]) = \mathfrak{v}(\Phi'[x/1]) = \textit{true}
\end{aligned}$$

Where $\Phi'[z/a]$ *denotes the substitution of free occurrences* z *by* a *in* Φ'*.*

We say a QBF formula Φ is *satisfiable* if and only if there exists a truth assignment $\mathfrak{v}$ where $\mathfrak{v}(\Phi) = 1$. If for all truth assignments $\mathfrak{v}(\Phi) = 0$ then we say Φ is *inconsistent*. If Φ is a closed formula, there exists precisely one truth assignment for Φ [Bie+09], in this case we say Φ is *true* if it is satisfiable or *false* when it is inconsistent.

2.3. Inseparability-based modules

As ontology is extended with new axioms it is often desirable to guarantee the modified version does not express any new information over a given signature, that the modifications do not imply any new unintended knowledge, this can be formally defined by means of conservative extensions. Inseparability relations further generalise conservative extensions which allows the definition of logic-based modules which guarantee the preservation of knowledge over a specified signature.

Definition 2.3.1 (Ghilardi et al. [GLW06] and Lutz et al. [LWW07]). *Let $\mathcal{M}, \mathcal{O}$ $\mathcal{M} \subseteq \mathcal{O}$ be $\mathcal{L}$ ontologies and Σ a signature*

- *$\mathcal{O}$ is a* deductive Σ-conservative extension *of $\mathcal{M}$ if for all entailments $\alpha \in \mathcal{L}$ with $\mathsf{sig}(\alpha) \subseteq \Sigma$ it holds that $\mathcal{M} \models \alpha$ if and only if $\mathcal{O} \models \alpha$*
- *$\mathcal{O}$ is a* model Σ-conservative extension *of $\mathcal{M}$ if*

$$\{\mathcal{I}|_\Sigma \mid \mathcal{I} \models \mathcal{M}\} = \{\mathcal{J}|_\Sigma \mid \mathcal{J} \models \mathcal{O}\}$$

- *$\mathcal{M}$ is a dCE Σ-module (mCE Σ-module) of $\mathcal{O}$ if $\mathcal{O}$ is a deductive Σ-conservative extension (model Σ-conservative extension) of $\mathcal{M}$.*

Deductive conservative extensions are defined in terms of a particular ontology language and there may exist an entailment constructed from a more expressive language than $\mathcal{L}$ which is entailed by $\mathcal{O}$ but not $\mathcal{M}$. In comparison, $\mathcal{O}$ being a model-conservative extension of $\mathcal{M}$ is a much stronger notion, and guarantees that every entailment over Σ is preserved irrespective of the DL in which a given entailment is formulated, and in fact is equivalent to $\mathcal{M}$ preserving every second-order formula over Σ [Kon+09a]. As a result, if $\mathcal{O}$ is a model Σ-conservative extension of $\mathcal{M}$ it is also a deductive Σ-conservative extension, and hence every mCE Σ-module is a dCE Σ-module, the converse however does not typically hold.

Inseparability relations are a family of equivalence relations between ontologies which generalise conservative extensions, applicable to arbitrary ontologies not only those defined over the subset relation.

Definition 2.3.2 (Inseparability relation). *Given two ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$, a signature Σ, and an inseparability relation $\mathbb{S}$, if $\mathcal{O}_1$ and $\mathcal{O}_2$ satisfy $\mathbb{S}$ w.r.t Σ we say they are $\mathbb{S}$ inseparable and write $\mathcal{O}_1 \equiv^{\mathbb{S}}_{\Sigma} \mathcal{O}_2$.*

Like conservative extensions, inseparability relations are parameterised by a signature, and if two ontologies satisfy an inseparability relation they are considered *inseparable* from each other – one cannot distinguished between them based on the information they provide over the signature — otherwise they are considered *separable*, some information over the signature can be derived from one ontology but not the other. For our purposes consider the specific relations which generalise the conservative extensions defined in Definition 2.3.1.

Definition 2.3.3 (Konev et al. [Kon+09a] and Sattler et al. [SSZ09]). *Given two $\mathcal{L}$ ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$ and a signature Σ*

- *$\mathcal{O}_1$ and $\mathcal{O}_2$ satisfy the subsumption inseparability relation $\mathbb{S} = \sqsubseteq$ if for every $\mathcal{L}$-entailment α such that $\mathsf{sig}(\alpha) \subseteq \Sigma$ we have $\mathcal{O}_1 \models \alpha$ if and only if $\mathcal{O}_2 \models \alpha$. If this is the case $\mathcal{O}_1$ and $\mathcal{O}_2$ are called subsumption inseparable and we write $\mathcal{O}_1 \equiv^{\sqsubseteq}_{\Sigma} \mathcal{O}_2$.*

- *$\mathcal{O}_1$ and $\mathcal{O}_2$ satisfy the model inseparability relation $\mathbb{S} = \mathsf{mod}$ if*

$$\{\mathcal{I}|_{\Sigma} \mid \mathcal{I} \models \mathcal{O}_1\} = \{\mathcal{J}|_{\Sigma} \mid \mathcal{J} \models \mathcal{O}_2\}$$

If this is the case $\mathcal{O}_1$ *and* $\mathcal{O}_2$ *are called* model inseparable *and we write* $\mathcal{O}_1 \equiv_{\Sigma}^{\mathsf{mod}} \mathcal{O}_2$.

Analogous to the relationship between the notions of conservative extensions, for each signature Σ we have $\equiv_{\Sigma}^{\mathsf{mod}} \subseteq \equiv_{\Sigma}^{\sqsubseteq}$. The flexibility of using inseparability relations allows for the definition of several types of modules:

Definition 2.3.4 (Konev et al. [Kon+08b; Kon+09a] and Grau et al. [Gra+07]). *Let* $\mathcal{M} \subseteq \mathcal{O}$ *be ontologies,* Σ *a signature and* $\mathbb{S}$ *an inseparability relation. Then* $\mathcal{M}$ *is called:*

- *A* plain Σ-module *if* $\mathcal{M} \equiv_{\Sigma}^{\mathbb{S}} \mathcal{O}$
- *A* self-contained Σ-module *if* $\mathcal{M} \equiv_{\Sigma \cup \mathsf{sig}(\mathcal{M})}^{\mathbb{S}} \mathcal{O}$
- *A* depleting Σ-module *if* $\mathcal{O} \setminus \mathcal{M} \equiv_{\Sigma \cup \mathsf{sig}(\mathcal{M})}^{\mathbb{S}} \emptyset$

A plain (self-contained, depleting) Σ*-module* $\mathcal{M}$ *is called* minimal *if there is no plain (self-contained, depleting)* Σ*-module* $\mathcal{N}$ *with* $\mathcal{N} \subset \mathcal{M}$.

A module is only useful if it is complete, in that it should provide coverage for Σ — any information over Σ provided by $\mathcal{O}$ should also be provided by $\mathcal{M}$. This can be considered a minimal requirement for a module, as a module can only reliably be used as an equivalent replacement for the entire ontology if this property holds. With this in mind, plain modules come as a natural definition for modules using inseparability relations. If $\mathcal{M}$ is a plain module it is indistinguishable from $\mathcal{O}$ as far as Σ is concerned, preserving all knowledge over Σ as to satisfy the chosen inseparability relation.

Consider the following ontology $\mathcal{O}$ consisting of the axioms $\{\alpha_1 - \alpha_4\}$:

$$A \sqsubseteq B \sqcap \exists r.C \tag{α_1}$$

$$A \sqsubseteq C \tag{α_2}$$

$$C \sqsubseteq B \tag{α_3}$$

$$B \sqsubseteq E \tag{α_4}$$

For the inseparability relation $\mathbb{S} = \sqsubseteq$ and signature $\Sigma = \{A, B\}$, the modules $\mathcal{M}_1 = \{\alpha_1\}$ and $\mathcal{M}_2 = \{\alpha_2, \alpha_3\}$ are both plain Σ-modules. It is easy to see, the only non-trivial inclusion entailed by $\mathcal{O}$ using only Σ-symbols is $\alpha = A \sqsubseteq B$, which follows from both $\mathcal{M}_1$ and $\mathcal{M}_2$.

One limitation of plain Σ-modules however, is they are not guaranteed to contain all the information about every symbol they utilise, but only those which have been specified by the chosen signature. As a result, in the common reusage scenario, one would have to keep track of the of the original specifying signature in order to know what information the module provides from the original ontology.

A stronger notion of modularity is that of self-contained Σ-modules, which have to be inseparable from the original ontology, not only for the specifying signature, but also the signature of the module itself. If we consider the plain Σ-module $\mathcal{M}_1$ above, we can see it utilises the symbol C in addition to the symbols in original signature Σ, yet it "misses" $\alpha_2 = A \sqsubseteq C$ which is entailed by $\mathcal{O}$, and therefore is not a self-contained Σ-module. Conversely, $\mathcal{M}_2$ *is* a self-contained module, any entailment over $\Sigma \cup \mathsf{sig}(\mathcal{M}_2)$ entailed by $\mathcal{O}$ is also entailed by $\mathcal{M}_2$.

Even the stronger notion of self-containment is not enough to capture *all* the relevant information described by an ontology — if $\mathcal{M}$ is a self-contained module, $\mathcal{O} \setminus \mathcal{M}$ may still entail information over $\Sigma \cup \mathsf{sig}(\mathcal{M})$ which is already entailed by $\mathcal{M}$. Stronger still, depleting Σ-modules ensure no information is "left" in the ontology, and require $\mathcal{O} \setminus \mathcal{M}$ to be inseparable from the empty ontology for $\Sigma \cup \mathsf{sig}(\mathcal{M})$ i.e. $\mathcal{O} \setminus \mathcal{M}$ must have no non-trivial entailments over $\Sigma \cup \mathsf{sig}(\mathcal{M})$. This means they have the benefit of preserving every justification for non-trivial entailments over $\Sigma \cup \mathsf{sig}(\mathcal{M})$, the minimal sets of axioms which are sufficient for each entailment to hold.

Consider the self-contained Σ-module $\mathcal{M}_2$ which is not a depleting Σ-module, the axiom α_1 causes $\mathcal{O} \setminus \mathcal{M}_2$ to entail $A \sqsubseteq B$, an entailment constructed solely from $\Sigma \cup \mathsf{sig}(\mathcal{M})$. A valid depleting Σ-module would consist of the axioms $\mathcal{M}_3 = \{\alpha_1, \alpha_2, \alpha_3\}$, where $\mathcal{O} \setminus \mathcal{M}_3$ consist of the single axiom α_4, which although uses the Σ-symbol B causes no additional entailments over $\Sigma \cup \mathsf{sig}(\mathcal{M})$.

With regards to the relative size of different modules, generally the stronger the module notion the larger the module; depleting modules are typically larger than self-contained modules which in turn are larger than plain modules. If one considers modules of the same type, stronger notions of inseparability typically lead to larger modules as a module must usually contain more axioms in order to preserve a stronger relation, e.g. a plain mCE Σ-module will typically be larger than a plain dCE Σ-module.

It is important to note, if a module is depleting it is not necessarily plain or self-contained. The relationship between the different types of modules described in Definition 2.3.4 strongly depends on so called *robustness properties* of the inseparability relations – how well the properties of an inseparability relation are preserved under the manipulation of the ontologies or signatures involved.

Example 2.3.1. *Consider the following* $\mathcal{ALC}$ *ontology* $\mathcal{O}$ *consisting of the axioms* $\{\beta_1 - \beta_3\}$ *as defined below:*

$$\mathsf{Student} \sqsubseteq \exists \mathsf{livesIn}.\mathsf{Flat} \quad (\beta_1)$$

$$\exists \mathsf{livesIn}^{-}.\top \sqsubseteq \mathsf{Property} \quad (\beta_2)$$

$$\mathsf{Property} \sqsubseteq \bot \quad (\beta_3)$$

Consider the module $\mathcal{M} = \{\beta_3\}$ *and the signature* $\Sigma = \{\mathsf{Student}, \mathsf{Property}\}$. *Observe that* $\mathcal{M}$ *is a depleting dCE* Σ*-module of* $\mathcal{O}$*, and there is no non-trivial* $\mathcal{ALC}$ *entailment* α *with* $\mathsf{sig}(\alpha) \subseteq \Sigma \cup \mathsf{sig}(\mathcal{M})$ *which follows from* $\mathcal{O} \setminus \mathcal{M}$*. But now consider the* $\mathcal{ALC}$ *inclusion* $\beta = \mathsf{Property} \sqsubseteq \bot$ *and notice that* $\mathcal{O} \models \beta$ *but also that* $\mathcal{M} \not\models \beta$*, and since* $\mathsf{sig}(\beta) \subseteq \Sigma$*,* $\mathcal{M}$ *is neither a plain nor self-contained dCE* Σ*-module.*

2.3.1 Robustness properties

Definition 2.3.5 (Kontchakov et al. [Kon+09b]). *Given ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$. An inseparability relation $\mathbb{S}$ is called* monotone *if it satisfies the following two conditions:*

($\mathcal{M}_\Sigma$) *If $\mathcal{O}_1 \equiv_\Sigma^\mathbb{S} \mathcal{O}_2$ then $\mathcal{O}_1 \equiv_{\Sigma'}^\mathbb{S} \mathcal{O}_2$ for all $\Sigma' \subseteq \Sigma$*

($\mathcal{M}_\mathcal{O}$) *If $\mathcal{O}_1 \subseteq \mathcal{O}_2 \subseteq \mathcal{O}_3$ and $\mathcal{O}_1 \equiv_\Sigma^\mathbb{S} \mathcal{O}_3$ then $\mathcal{O}_1 \equiv_\Sigma^\mathbb{S} \mathcal{O}_2$*

The first point implies the inseparability relation is preserved under shrinking of the signature, the second implies any ontology sandwiched between two inseparable ontologies should be inseparable from either.

Definition 2.3.6 (Kontchakov et al. [Kon+09b] and Grau et al. [Gra+08]). *Consider a description logic $\mathcal{L}$ and all $\mathcal{L}$-ontologies $\mathcal{O}'$, $\mathcal{O}_1$ and $\mathcal{O}_2$ and an inseparability relation $\mathbb{S}$, then $\mathbb{S}$ is called* robust under replacement *if for all signatures Σ we have $\mathcal{O}' \cup \mathcal{O}_1 \equiv_\Sigma^\mathbb{S} \mathcal{O}' \cup \mathcal{O}_2$ whenever $\mathcal{O}_1 \equiv_\Sigma^\mathbb{S} \mathcal{O}_2$ and $\mathsf{sig}(\mathcal{O}') \cap \mathsf{sig}(\mathcal{O}_1 \cup \mathcal{O}_2) \subseteq \Sigma$.*

If an inseparability relation $\mathbb{S}$ is robust under replacement and $\mathcal{M} \subseteq \mathcal{O}$ is a plain Σ-module ($\mathcal{M} \equiv_\Sigma^\mathbb{S} \mathcal{O}$) then, for any ontology $\mathcal{O}'$ such that $\mathsf{sig}(\mathcal{O}') \cap \mathsf{sig}(\mathcal{O}) \subseteq \Sigma$ it holds that $\mathcal{O}' \cup \mathcal{M} \equiv_\Sigma^\mathbb{S} \mathcal{O}' \cup \mathcal{O}$. That is, $\mathcal{M}$ can be used in place of $\mathcal{O}$ without losing any information over the signature Σ. Moreover this does not depend on a particular version of $\mathcal{O}'$ which may continue to evolve independently as long as it continues to only share Σ-symbols with $\mathcal{O}$. For these aforementioned reasons, robustness under replacement is considered critical for ontology reuse, a fundamental application of ontology modules, and is strongly preferable to consider modules based on inseparability relations satisfying this property.

Proposition 2.3.1 (Kontchakov et al. [Kon+09b]). *For an inseparability relation $\mathbb{S}$:*

- *If $\mathbb{S}$ satisfies $(\mathcal{M}_\Sigma)$ then every self-contained Σ-module is a plain Σ-module.*
- *If an $\mathbb{S}$ is robust under replacement then every depleting Σ-module is a self-contained Σ-module.*

Input: Ontology $\mathcal{O}$, Signature Σ, inseparability relation $\mathbb{S}$ which is monotone and robust under replacement
Output: Unique minimal depleting Σ-module $\mathcal{M}$ for $\mathbb{S}$

1 $\mathcal{M} ::= \emptyset, \mathcal{W} ::= \emptyset$
2 **while** $\mathcal{O} \setminus \mathcal{M} \neq \mathcal{W}$ **do**
3   choose $\alpha \in (\mathcal{O} \setminus \mathcal{M}) \setminus \mathcal{W}$
4     $\mathcal{W} ::= \mathcal{W} \cup \{\alpha\}$
5   **if** $\mathcal{W} \not\equiv^{\mathbb{S}}_{\Sigma\cup\mathsf{sig}(\mathcal{M})} \emptyset$ **then**
6         $\mathcal{M} ::= \mathcal{M} \cup \{\alpha\}$
7         $\mathcal{W} ::= \emptyset$
8   **end**
9 **end**
10 **return** $\mathcal{M}$

Figure 2.6: General depleting module extraction algorithm

The first point from Proposition 2.3.1 is simply a shrinking of the signature ($\mathcal{M}_\Sigma$), the second follows from if $\mathcal{T} \setminus \mathcal{M} \equiv^{\mathbb{S}}_{\Sigma\cup\mathsf{sig}(\mathcal{M})} \emptyset$ then $(\mathcal{T} \setminus \mathcal{M}) \cup \mathcal{M} \equiv^{\mathbb{S}}_{\Sigma\cup\mathsf{sig}(\mathcal{M})} \mathcal{M}$ and so $\mathcal{T} \equiv^{\mathbb{S}}_{\Sigma\cup\mathsf{sig}(\mathcal{M})} \mathcal{M}$. Assuming a chosen inseparability relation is both monotone and robust under replacement depleting Σ-modules become the most attractive type of modules to consider, as they come with the guarantee that they are additionally both self-contained and plain. Furthermore, there always exists a unique minimal depleting Σ-module.

Proposition 2.3.2 (Kontchakov et al. [Kon+09b]). *If $\mathbb{S}$ is monotone and robust under replacement there is a unique minimal depleting Σ-module and it is produced by the algorithm in Figure 2.6.*

Figure 2.6 is a general algorithm for the extraction of unique depleting Σ-modules, applicable to any inseparability relation both monotone and robust under replacement. It works by incrementally building a subset $\mathcal{W} \subseteq \mathcal{T} \setminus \mathcal{M}$ one axiom at a time (Lines 3–4). If this subset causes separability for $\mathbb{S}$ w.r.t $\Sigma \cup \mathsf{sig}(\mathcal{M})$ (Line 5) the last axioms added to $\mathcal{W}$ must be the cause, and must be relevant for $\Sigma \cup \mathsf{sig}(\mathcal{M})$, and is so added to the module $\mathcal{M}$. Following this $\mathcal{W}$ is reset (Line 7) and the process is restarted considering inseparability for the updated signature $\Sigma \cup \mathsf{sig}(\mathcal{M})$. This continues until every remaining axiom in $\mathcal{O} \setminus \mathcal{M}$ has been considered and no cause for separability has been found w.r.t

$\Sigma \cup \mathsf{sig}(\mathcal{M})$. The algorithm terminates returning the unique minimal depleting Σ-module of $\mathcal{O}$ for $\mathbb{S}$.

The algorithm runs in $O(|\mathcal{O}| + |\Sigma|)^2 \times T_c(\mathcal{O}, \Sigma)$ where $T_c(\mathcal{O}, \Sigma)$ is the time need to check for an ontology $\mathcal{O}$ whether $\mathcal{O} \equiv^{\mathbb{S}}_{\Sigma} \emptyset$. For specific logics and inseparability relations we will discuss the known decidability and complexity of deciding whether $\mathcal{O} \equiv^{\mathbb{S}}_{\Sigma} \emptyset$ in Section 2.4.

Proposition 2.3.3 (Konev et al. [Kon+09a]). *The model-inseparability relation* $\mathbb{S} = \mathsf{mod}$ *is both monotone and robust under replacement.*

As a result for each signature by Proposition 2.3.2 there exists a unique depleting mCE Σ-module which by Proposition 2.3.1 is also self-contained and plain. For deductive inseparability however, robustness under replacement often fails for many standard DLs. For example, any of the logics $\mathcal{ALC}$, $\mathcal{ALCI}$, $\mathcal{ALCQ}$, $\mathcal{ALCQI}$, $\mathcal{ALCO}$ or $\mathcal{ALCHO}$ are not robust under replacement for the inseparability relation $\mathbb{S} = \sqsubseteq$ [Kon+09a].

2.3.2 Safety

Along with containing all the information relevant to the signature, it is desirable that when reusing an ontology by importing a module that the module does not have any *unintended* interactions with the ontology that imports it. In a typical ontology re-use scenario a module may be utilised because the modeler is not an expert in the topic for which the module is relevant, and importing it should not change the information which the module provides. This notion has lead to the definition of safety.

Definition 2.3.7 (Grau et al. [Gra+08]). *Given an ontology $\mathcal{O}$, an inseparability relation $\mathbb{S}$ and signature Σ, $\mathcal{O}$ is safe for $\mathbb{S}$ if for every ontology $\mathcal{O}'$ such that* $\mathsf{sig}(\mathcal{O}) \cap \mathsf{sig}(\mathcal{O}') \subseteq \Sigma$ *it holds that* $\mathcal{O} \cup \mathcal{O}' \equiv^{\mathbb{S}}_{\mathsf{sig}(\mathcal{O}')} \mathcal{O}'$.

Typically importing a module $\mathcal{M}$ into an ontology $\mathcal{O}$ one would expected new information over $\mathsf{sig}(O)$ particularly those symbols from Σ which are reused from $\mathcal{M}$. If an ontology is safe it provides a safe interface for a signature

Σ, and any module describing Σ symbols may be imported with unintended interactions.

Theorem 2.3.1 (Grau et al. [Gra+08] and Konev et al. [Kon+13]). *Consider an ontology $\mathcal{O}$ and signature Σ. If $\mathcal{O} \equiv_{\Sigma}^{\mathsf{mod}} \emptyset$ then $\mathcal{O}$ is safe for Σ.*

Theorem 2.3.1 formulates safety such a way that quantification over the imported ontology as in Definition 2.3.7 is eliminated. Also, as a consequence, if $\mathcal{M}$ is a depleting Σ-module for the inseparability relation $\mathbb{S} = \mathsf{mod}$ then $\mathcal{O} \setminus \mathcal{M}$ is safe for $\Sigma \cup \mathsf{sig}(\mathcal{M})$, and so the module $\mathcal{M}$ can be maintained independently of $\mathcal{O}$.

2.3.3 Complexity and computability

We have seen strong arguments for inseparability-based modules particularly those defined by model-inseparability relations. Unfortunately, deciding inseparability-based modules suffers from high complexity and even undecidability in moderately expressive logics. Deciding if one ontology is a deductive conservative extension of another for a signature Σ, and hence deciding if $\mathcal{M} \subseteq \mathcal{O}$ is deductive Σ-module, is ExpTime-complete for general $\mathcal{EL}$ ontologies [LW10], for $\mathcal{ALC}$ and $\mathcal{ALCQI}$ is 2ExpTime-complete [GLW06; LWW07], for $\mathcal{ALCQIO}$ this problem becomes *undecidable* [LWW07]. Deciding model-conservative extensions is even harder, only for acyclic $\mathcal{EL}$ terminologies is deciding model-conservative extensions in PTime, for general $\mathcal{EL}$ and acyclic $\mathcal{ALC}$ the same problem is *undecidable* [LWW07]. One way to reduce the complexity is to only consider *concept signatures* – signatures consisting of concept names only – then, deciding if two ontologies are model-inseparable for a signature for both general $\mathcal{EL}$ and acyclic and general $\mathcal{ALCI}$ is coNExp$^{\text{NP}}$-complete. The special case for deciding if an ontology is model-inseparable from the empty ontology, for unrestricted signatures is still known to be undecidable for general $\mathcal{EL}$ and both acyclic and general $\mathcal{ALC}$, for concept signature it goes down to PTime for general $\mathcal{ELI}$ ontologies and Π_2^p for general and acyclic

$\mathcal{ALCI}$ [Kon+08a].

2.4. Modules for inseparability relations

As a consequence of the complexity results, two different approaches exist for computing logic-based modules. The first is to limit the expressivity of the logics, for example Konev et al. [Kon+08a] describe a method for computing *minimal* model Σ-modules acyclic $\mathcal{ELI}$ terminologies in polynomial time. The second, for more expressive logics, *approximations* are used. Approximations are modules that still satisfied the desired inseparability relation but are not necessarily *minimal*. That said, there is still a strong desire for modules to be as small as possible – to approximate minimal modules. The most popular kind of approximations are based on a notion called locality [Gra+08; SSZ09] which for general ontologies up to $\mathcal{SROIQ}$ in expressitivity, can compute an approximation of the minimal mCE Σ-module.

In the following section we will look in more detail at modules based on inseparability relations, noting when they are computable and when approximations are required. We focus in particular at those inseparability relations and modules that preserve the properties described in Section 2.3.1 which we have argued are critical for the common applications for which modules are relevant. As we have seen, preserving these properties guarantees that for each signature Σ, every depleting Σ-module is uniquely determined and both self-contained and plain. Furthermore, the relationship between inseparability relations guarantees the mCE Σ-modules we consider are also dCE Σ-modules, preserving both models and subsumptions over a signature.

2.4.1 Model inseparability modules for $\mathcal{ELI}$ and $\mathcal{ALCI}$

Konev et al. [Kon+08a; Kon+13] describe an approach for extracting plain and depleting modules for acyclic $\mathcal{ELI}$ terminologies – which they call weak and strong modules respectively. They show for the model-inseparability relation that separability from the empty TBox in acyclic $\mathcal{ELI}$ terminologies is caused by so called direct or indirect *dependencies* between signature symbols. A dependency is present if an axiom $A \sqsubseteq C \in \mathcal{T}$ with $A \in \Sigma$ uses another symbol $X \in \Sigma$ either directly or indirectly in its definition.

In order to extract a depleting Σ-module, their algorithm goes through the axioms of $\mathcal{T} \setminus \mathcal{M}$ and determines if any axiom causes a dependency between the symbols of $\Sigma \cup \mathsf{sig}(\mathcal{M})$. If such an axiom is located it is added to the module $\mathcal{M}$ and the signature of $\Sigma \cup \mathsf{sig}(\mathcal{M})$ is updated accordingly. Once no axiom in $\mathcal{T} \setminus \mathcal{M}$ contains a dependency between $\Sigma \cup \mathsf{sig}(\mathcal{M})$ symbols then the algorithm terminates guaranteeing $\mathcal{T} \setminus \mathcal{M} \equiv_{\Sigma}^{\mathsf{mod}} \emptyset$, which results in $\mathcal{M}$ being the unique minimal depleting Σ-module. Deciding if an $\mathcal{ELI}$ TBox contains dependencies, whether direct or indirect, can be done syntactically, and so the module extraction algorithm runs in polynomial time.

The authors also introduce an algorithm for extracting depleting mCE Σ-modules from acyclic $\mathcal{ALCI}$ terminologies. Following from the undecidability constraints of model-conservative extensions for $\mathcal{ALCI}$, they again utilise the idea of direct dependencies but to regain decidability, and they use a modified version of the algorithm in Figure 2.6 in order to compute an approximation of the minimal depleting mCE Σ-module. We will look at the algorithm in more detail in Chapter 3, where we will propose practical optimisations and improvements.

2.4.2 Concept and Query Inseparability for DL-Lite

Kontchakov et al. [Kon+09b] investigates the computation of minimal modules for several dialects of DL-Lite ontologies. They look at both deductive inseparability modules and additionally those based on the relation query inseparability $\equiv_{\Sigma}^{q}$. Two DL-Lite ontologies are considered query inseparable for a signature Σ if they always give the same answers to a query φ with $\mathsf{sig}(\varphi) \subseteq \Sigma$. They argue that query inseparability is the most appropriate inseparability relation for DL-Lite ontologies, typically providing an interface for querying instance data.

They also investigate the computation of plain dCE Σ-modules (MCM modules) which are shown to be minimal but not uniquely determined, and depleting query inseparability based modules (MDQM modules) which are both minimal and uniquely determined. The latter result coming as a direct consequence of Proposition 2.3.2 and the inherent robustness properties of query inseparability. Both MCM and MDQM modules can be computed by reducing the inseparability tests to the satisfiability of a class of QBF formulas which is Π_2^p-complete. Since query inseparability is decidable for DL-Lite, MDQM modules can be computed using the algorithm in Figure 2.6 with Line 5 deciding if $\mathcal{W} \not\equiv^q_{\Sigma\cup\mathsf{sig}(\mathcal{M})} \emptyset$.

2.4.3 Datalog Modules

Romero et al. [Rom+14] introduce a novel module extraction approach using a reduction to reasoning in the rule-based query language datalog [AHV95; Dan+01]. The intuition behind the approach is to convert an ontology $\mathcal{O}$ into a datalog program $\mathcal{P}$ by mapping every axiom to (possibly several) datalog rules. The resulting datalog program represents a strengthening of the original ontology, as the limited expressiveness of datalog means it is necessary to convert any disjunctions in the input ontology to conjunctions in the resulting datalog program.

The authors show that the proof that a Σ-formula φ ($\mathsf{sig}(\varphi) \subseteq \Sigma$) is entailed by $\mathcal{O}$ can be embedded in a proof ρ of the datalog program $\mathcal{P}$. This embedding has the key property that for any axiom in $\mathcal{O}$ needed to entail φ there will always exist at least one corresponding datalog rule in $\mathcal{P}$ which will be used in the proof of ρ, so it is sufficient to examine the rules used in the proof of ρ to identify those axioms of $\mathcal{O}$ which are necessary for the entailment to hold.

Utilising this property turns out to be a very flexible way to extract modules, as the construction of the datalog program and subsequent proofs can be tailored to ensure modules preserve different kinds of Σ-formulas in order to satisfy a chosen inseparability relation, which includes, but is not limited to, both deduction and model inseparability.

The modules extracted using the datalog approach are by default plain Σ-modules, but further modifications to the datalog programs are possible to additionally ensure each extracted module is also self-contained and depleting. The modules are also approximations, no minimality is guaranteed due to the fact that the datalog program $\mathcal{P}$ is a strengthening of the ontology $\mathcal{O}$, and there may be a proof of $\mathcal{P}$ corresponding to a Σ-formula which isn't actually an entailment of $\mathcal{O}$, which may then lead to unnecessary axioms appearing in an extracted module.

For all ontology languages that we have considered, the complexity of both producing a datalog program from an ontology and module extraction via datalog reasoning is tractable.

2.4.4 Locality based modules

Locality based modules (LBMs) are a family of approximations for minimal depleting mCE Σ-modules based on the locality of individual axioms. An axiom is considered "local" if given a signature Σ, it can always be satisfied independently of the symbols in Σ, but in a restricted way. The particular way in which the interpretation of an axiom is restricted is dependant on the variety of locality which is chosen, and there exists both semantic and syntactic methodologies for testing the locality of axioms, the former as hard as standard reasoning for the chosen logic, the latter can be performed in polynomial time. With the low complexity of syntactic locality, and the ability to extract modules from very expressive logics, LMBs are the most popular kind of approximation, and have found use in a variety of applications [Gra+10; KS13; LK14; Sun+08; NK10].

Semantic Locality

Semantic locality comes in different varieties depending on how the non-Σ symbols are interpreted. Satisfying every axiom α by interpreting every non-Σ symbol as the empty set is known as $\emptyset$-locality, whereas interpreted as the entire domain is known as Δ-locality.

Definition 2.4.1 (Grau et al. [Gra+08] and Sattler et al. [SSZ09]). *An axiom α is called $\emptyset$-local (Δ-local) w.r.t a signature Σ if, for each interpretation $\mathcal{I}$ there exists an interpretation $\mathcal{J}$ such that $\mathcal{I}|_\Sigma = \mathcal{J}|_\Sigma$ and $\mathcal{J} \models \alpha$ and for each $X \in \mathsf{sig}(\alpha) \setminus \Sigma, X^\mathcal{J} = \emptyset$ (for each $C \in \mathsf{sig}(\alpha) \setminus \Sigma, C^\mathcal{J} = \Delta^\mathcal{J}$, for each $R \in \mathsf{sig}(\alpha) \setminus \Sigma, R^\mathcal{J} = \Delta^\mathcal{J} \times \Delta^\mathcal{J}$).*

The intuition behind locality is to establish when an axiom says nothing about a signature Σ, if an axiom is local w.r.t a signature Σ, it is irrelevant for Σ, as for any interpretation $\mathcal{I}$ of Σ-symbols we can always find a model $\mathcal{J}$ of α which interprets the Σ-symbols in a uniform way, meaning the axiom is satisfiable however the Σ-symbols are interpreted.

Theorem 2.4.1 (Grau et al. [Gra+08]). *If all axioms in $\mathcal{O} \setminus \mathcal{M}$ are semantically local (either $\emptyset-$ or $\Delta-$ local) for $\Sigma \cup \mathsf{sig}(\mathcal{M})$ then $\mathcal{M}$ is a depleting mCE Σ-module.*

The converse of Theorem 2.4.1 typically does not hold. A minimal example from [Gra+08], the axiom $\alpha = A \equiv B$ and $\Sigma = \{A\}$ then α is neither $\emptyset$ or Δ local and will be included in the LBM for Σ but $\{A \equiv B\} \equiv_\Sigma^{\mathsf{mod}} \emptyset$. As a consequence of this locality based modules are approximations, depleting mCE Σ-modules but not necessarily minimal ones, as they may contain surplus axioms which are non-local but irrelevant for $\Sigma \cup \mathsf{sig}(\mathcal{M})$.

Testing the locality of single axioms and by Theorem 2.4.1 extracting LBMs can be achieved using off-the-shelf DL-reasoners, which makes it easier than deciding minimal logic-based modules (Section 2.3.3). However, standard reasoning in expressive description logics is still complex, for example $\mathcal{SROIQ}$ it is N2EXPTIME-complete [Kaz08]. In order to produce a tractable notion of locality and thereby a tractable module extraction procedure, a syntactic approximation to locality has been introduced.

Syntactic Locality

Grau et al. [Gra+08] introduce a syntactic approximation for both types of semantic locality for the description logic $\mathcal{SHIQ}$. The intuition behind syntactic locality is that it is often possible to determine if an axiom is local or not just by examining the syntax of that axiom. Jimenez-Ruiz et al. [Jim+08] later extended the notion to the more expressive $\mathcal{SROIQ}$.

Each type of semantic locality as in Definition 2.4.1 has a syntactic variant. $\top$-locality is analogous to Δ-locality whereas $\bot$-locality is to $\emptyset$-locality. Like semantic locality, syntactic locality is decided on a per axiom basis. Firstly one identifies those concept expressions which may be interpreted trivially whichever interpretation of Σ symbols is taken.

Definition 2.4.2 ([Gra+08]). *Given a signature Σ, concept name $A \notin \Sigma$, role name $r \notin \Sigma$, positive integer n and define two sets of concept expressions:* $\mathsf{Bot}(\Sigma)$ *and denote the members of this set $X^\bot$, and* $\mathsf{Top}(\Sigma)$ *whose members are denoted $X^\top$. Depending on the type of syntactic locality required, these sets of expressions are inductively defined as below.*

$(a)\bot$-*locality*
$\mathsf{Bot}(\Sigma) ::= A^\bot \mid \bot \mid \neg C^\top \mid C_1 \sqcap C_2^\bot \mid C_1^\bot \sqcap C_2 \mid \exists r.C^\bot \mid\ \geqslant n\ r.C^\bot \mid \exists r^\bot.C \mid\ \geqslant n\ r^\bot.C$
$\mathsf{Top}(\Sigma) ::= \top \mid \neg C^\bot \mid C_1^\top \sqcap C_2^\top \mid\ \geqslant 0\ r.C$

$(b)\top$-*locality*
$\mathsf{Bot}(\Sigma) ::= \bot \mid \neg C^\top \mid C_1 \sqcap C_2^\bot \mid C_1^\bot \sqcap C_2 \mid \exists r.C^\bot \mid\ \geqslant n\ r.C^\bot$
$\mathsf{Top}(\Sigma) ::= A^\top \mid \top \mid \neg C^\bot \mid C_1^\top \sqcap C_2^\top \mid \exists r^\top.C^\top \mid\ \geqslant n\ r^\top.C^\top \mid\ \geqslant 0\ r.C$

Intuitively each member of the set $\mathsf{Bot}(\Sigma)$ can be interpreted as the empty set, whereas those in $\mathsf{Top}(\Sigma)$ the whole domain without constraining the interpretation of Σ-symbols. Secondly, utilising this information of which expressions can be trivially interpreted one identifies those axioms which are syntactically local.

Definition 2.4.3. *Given an axiom α and a signature Σ, let C, D be arbitrary concept expressions, r, s be role names. Then α is:*

1. *Syntactically $\bot$-local if it is of the form*

   $$C^{\bot} \sqsubseteq D \mid C \sqsubseteq D^{\top} \mid C^{\bot} \equiv D^{\bot} \mid C^{\top} \equiv D^{\top} \mid r^{\bot} \sqsubseteq s \mid \mathsf{Trans}(r^{\bot}) \mid \mathsf{Func}(r^{\bot})$$

   For $C^{\bot}, D^{\bot}, r^{\bot} \in \mathsf{Bot}(\Sigma)$ and $C^{\top}, D^{\top} \in \mathsf{Top}(\Sigma)$

2. *Syntactically $\top$-local if it is of the form*

   $$C^{\bot} \sqsubseteq D \mid C \sqsubseteq D^{\top} \mid C^{\bot} \equiv D^{\bot} \mid C^{\top} \equiv D^{\top} \mid r \sqsubseteq s^{\top} \mid \mathsf{Trans}(s^{\top}) \mid \mathsf{Func}(s^{\top})$$

   For $C^{\bot}, D^{\bot} \in \mathsf{Bot}(\Sigma)$ and $C^{\top}, D^{\top}, s^{\top} \in \mathsf{Top}(\Sigma)$

Theorem 2.4.2 (Grau et al. [Gra+08] and Jimenez-Ruiz et al. [Jim+08])**.** *If all axioms in $\mathcal{O} \setminus \mathcal{M}$ are syntactically local (either $\top$- or $\bot$-local then $\mathcal{M}$ is a depleting mCE-module of $\mathcal{O}$.*

The procedures to check for syntactic locality are all syntactic and so the complexity of deciding the locality of a single axiom can be done in polynomial time.

Extracting and combining locality based modules

As locality based modules are decided on a per-axiom basis and not defined using an inseparability relation, a different algorithm is used, as described in Figure 2.7. One can observe that actually this algorithm is very similar to the one given for minimal depleting modules based on inseparability relations (Figure 2.6), but is described on a per-axiom basis. For some notion of locality $x \in \{\Delta, \emptyset, \top, \bot\}$, the algorithm goes through all the axioms of the ontology $\mathcal{O}$ repeatedly to ensure every axiom in $\mathcal{O} \setminus \mathcal{M}$ is x-local for $\Sigma \cup \mathsf{sig}(\mathcal{M})$. Then by Theorem 2.4.1 and Theorem 2.4.2 the "x-module" $\mathcal{M}$ is a depleting mCE-module of $\mathcal{O}$.

Input: Ontology $\mathcal{O}$, Signature Σ, $x \in \{\emptyset, \Delta, \top, \bot\}$
Output: x-module $\mathcal{M}$ of $\mathcal{O}$ w.r.t Σ

$\mathcal{M} ::= \emptyset, \mathcal{O}' ::= \mathcal{O}$
repeat
    changed = false
    foreach $\alpha \in \mathcal{O}'$ **do**
        if α *not* x*-local w.r.t* $\Sigma \cup \mathsf{sig}(\mathcal{M})$ **then**
            $\mathcal{M} ::= \mathcal{M} \cup \{\alpha\}$
            $\mathcal{O}' ::= \mathcal{O}' \setminus \{\alpha\}$
            changed = true
        end
    end
until *changed = false*
return $\mathcal{M}$

Figure 2.7: Extracting a locality based module

The basic algorithm runs in cubic time, needing to test for locality at most $(|\mathcal{O}| + |\Sigma|)^2$ times. Tsarkov [Tsa12] describes an optimised approach to reduce the complexity further to only require at most $(|\mathcal{O}| \times s)$ locality checks where $s = \mathsf{max}_{\alpha \in \mathcal{O}}(|\mathsf{sig}(\alpha)|)$.

Proposition 2.4.1 (Grau et al. [Gra+08]). *Given $\mathcal{M} \subseteq \mathcal{O}$ be ontologies and Σ a signature. If $\mathcal{M}$, is a $\bot$-module for Σ then $\mathcal{M}$ is a $\emptyset$-module for Σ. If $\mathcal{M}$ is a $\top$-module for Σ then $\mathcal{M}$ is a Δ-module for Σ.*

Proposition 2.4.1 implies that since each syntactically LBM contains its semantic counterpart, it is also a depleting mCE Σ-module. The converse of this proposition does not typically hold, there exist axioms which are syntactically local but not semantically for the analogous notion of locality. For example the axiom $\alpha = \exists r.\neg A \sqsubseteq \exists r.\neg B$ is $\emptyset$-local for $\Sigma = \{r\}$ since for any interpretation $\mathcal{I}$ of r, there always exists an interpretation $\mathcal{J}$ such that $r^{\mathcal{J}} = r^{\mathcal{I}}$ and $A^{\mathcal{J}} = B^{\mathcal{J}} = \emptyset$ which is a model of α. However, one can verify α is not $\bot$-local, it does not match the syntactic structure required by Definition 2.4.3.

For this reason, syntactic locality can be seen as an approximation for semantic locality which in turn is an approximation for the minimal depleting Σ modules. Syntactic locality is typically preferred due to the low computation complexity of testing for locality, combined with complexity of module extraction algorithm Figure 2.7 means syntactic LBMs can be extracted in polynomial time.

An additional property of locality based modules means different types of locality capture different relationships between the subclass/superclass relationship involving signature symbols. This can be described by the following proposition:

Proposition 2.4.2 (Grau et al. [Gra+08] and Jimenez-Ruiz et al. [Jim+08]). *Let $\mathcal{O}$ be an $\mathcal{SROIQ}$ ontology and Σ a signature then the following are equivalent:*

1. $\mathcal{O} \models A \sqsubseteq B$
2. *The $\bot$-module for $\Sigma = \{A\}$, $\mathcal{M}_\bot$, we have $\mathcal{M}_\bot \models A \sqsubseteq B$*
3. *The $\top$-module for $\Sigma = \{B\}$, $\mathcal{M}_\top$, we have $\mathcal{M}_\top \models A \sqsubseteq B$*

This property has been exploited for optimising ontology classification, computing the asserted hierarchy of concept names of the ontology [RGH12; TP12a; Gra+10]. Since every syntactic locality module is a semantic locality module by Proposition 2.4.2 also holds for Δ and $\emptyset$-modules.

Combining locality based modules

With the hope of producing smaller modules $\bot-$ and $\top$ locality ($\emptyset-$ and $\Delta-$ locality) can be iteratively nested until a fixed point, first extracting a $\top-$ ($\Delta-$) module and from the result extracting a $\bot-$ ($\emptyset-$) module for the same signature, until a fixpoint is reached, producing $\bot\top^*$-modules ($\emptyset\Delta^*$-modules) which are at least as small as the equivalent $\top-$ or $\bot-$ modules ($\Delta-$ or $\emptyset-$ modules). The number of iterations needed to reach a fixpoint is at most at large as the number of axioms in the ontology [SSZ09]. Such modules do not typically capture the subclass/superclass relationship involving signature terms (Proposition 2.4.2) but are still guaranteed to be depleting mCE Σ-modules [SSZ09].

The properties of depleting Σ-modules guarantee the sequence of nesting module extraction is not important, e.g. $\top\bot^*$ and $\bot\top^*$ modules are the same for a given input signature. And analogously to Proposition 2.4.1, each $\bot\top^*$ module is also a $\emptyset\Delta^*$ module extracted for a given signature. For LBMs the proof of this by Kazakov is published in [Ves13], whilst we prove the analogous case for depleting mCE Σ-modules generally in Chapter 4.

Syntactic vs. Semantic Locality

With the desire for small as possible approximations and with both syntactic and semantic notion of locality available a question is raised – "Is syntactic locality a good approximation?". Do we find the syntactic locality modules close in size to the corresponding semantic ones or should we prefer the more expensive semantic locality modules to obtain a better approximation of the minimal depleting mCE Σ-modules?

To answer this, Vescovo et al. [Ves+12; Ves+13] conducted a number of empirical studies over a wide range of real-world ontologies comparing semantic and syntactic locality on both a per-axiom and per-module basis. The conclusion of both studies is that statistically there no observable difference between any kind of semantic locality and its corresponding syntactic notion.

As $\bot\top^*$ modules not only contain the corresponding $\emptyset\Delta^*$ modules they can be computed efficiently and are at least as small as the equivalent $\top$ or $\bot$-modules they are the most attractive kind of locality based modules. We will henceforth refer to $\bot\top^*$-modules as STAR modules for ease of reference and pronunciation for the remainder of the thesis.

2.5. Success of approximations

When approximations are necessary, and minimal modules exist, as is the case for depleting mCE modules, it is still desirable to have modules which are as small as possible – to approximate minimal modules. How successful an approximation is can be measured by how close it is in size to the minimal module which it approximates. Currently there is limited support for this task and the only way to evaluate an approximation is to compare it to an extraction algorithm for inexpressive logics for which minimal modules can be automatically extracted.

$\bot$-modules are compared to modules produced by the MEX system for acyclic $\mathcal{ELI}$ in [Kon+08a]. Both approaches extract a depleting mCE Σ-modules, as a reminder, the MEX-modules are guaranteed to be minimal whereas the $\bot$-modules are an approximation. The experimental evaluation considered the large scale $\mathcal{EL}$ terminology SNOMED CT consisting of around 400,000 axioms. In the experimental evaluation, signatures of sizes 100 to 1000 consisting of random symbols were taken from the signature of the ontology and used to extract modules. On average the size of the MEX-modules ranged between 200 and 6000 axioms, the corresponding $\bot$-modules were up to around 4x as large.

A further comparison in [Kon+13], this time to STAR modules for different version of the SNOMED CT ontology both random and 159 specially tailored signatures representing various subsets of the ontology were chosen. For the random signatures the result was much the same, with STAR modules being up to 3.5x larger than the corresponding MEX module. For tailored signatures however, the STAR and MEX modules coincided in 83 of 159 cases, for the remaining cases the STAR module was up to 3x as large as the corresponding MEX module. The cases where MEX and STAR coincide is explained by proving that if an ontology only consists of $\mathcal{EL}$ concept inclusions (of the form $C \sqsubseteq D$) then MEX and STAR always coincide for any input signature, and hence the STAR module is the ideal minimal, which is what was observed in the axioms of the ontology relevant to the 83 tailored signatures.

STAR modules were also compared to MEX modules over a larger number of ontologies in [Ves+13] in which a corpus of 242 real-world ontologies of varying expressivity were selected. As MEX can only extract modules from acyclic $\mathcal{ELI}$ terminologies, $\mathcal{ELI}$ terminological versions of each ontology were created by removing axioms more expressive than $\mathcal{ELI}$ and breaking terminological cycles. Extracted from these preprocessed ontologies the sizes of STAR and MEX modules were compared using both random and axiom signatures as input. The results conclude that, for either random or axiom signatures, MEX modules were smaller than the corresponding STAR modules in around 27% of ontologies, the MEX module being between $0 - 26$ axioms smaller or relatively $0 - 80\%$ of STAR module size.

In summary, several experiments comparing LMBs to minimal depleting mCE Σ-modules for the inexpressive logic $\mathcal{ELI}$ reveal the approximation provided by locality-based approaches may sometimes coincide with the ideal minimal module but often may be significantly larger – the locality approximation is far from optimal in these cases – however, LBMs can be applied to much more expressive logics and nothing is yet known about the success of these approximations. In addition, nothing is known about the success of any other of the alternative approaches to producing approximations of depleting mCE Σ-modules, either the datalog approach or the model-theoretic approximation for $\mathcal{ALCI}$.

2.6. Summary

We have introduced the various logics relevant to this thesis including the family of Description Logics and have described how they are used to build ontologies in order to model a domain of interest. We also introduced inseparability relations which formally define when two ontologies are considered to represent the same knowledge over a given signature leading to the definition of several kinds of module which preserve specified knowledge from an ontology. We then surveyed the current approaches to extracting modules from ontologies, detailing when minimal modules can be automatically extracted, and when approximations are required. Finally, we commented on how the success of approximations can be evaluated, and currently how there is limited support for this task. In the next chapter we will look more on the model-theoretic approximation for $\mathcal{ALCI}$ and consider extensions and practical improvements to the originally proposed algorithm.

CHAPTER 3

Approximations for Acyclic Terminologies

In the previous chapter we described how the unique minimal depleting mCE modules for acyclic $\mathcal{ELI}$ terminologies can be extracted in PTIME, and when compared to corresponding STAR-module — extracted for the same signature — may sometimes coincide in size but are often significantly smaller. In more expressive logics, where approximations are necessary, it is currently unclear which approaches produce the most successful approximation for a given signature, that which is closest in size to the corresponding minimal module.

In this chapter we focus on extending and optimising the approximation algorithm for acyclic $\mathcal{ALCI}$ terminologies described by Konev et al. [Kon+08a; Kon+13], with the aim to develop a practically efficient algorithm which we may utilise in an series of experiments to evaluate the relative size of this approximation in comparison to others.

In Section 3.2 we look at what is already known from [Kon+08a; Kon+13], the theory which underpins an $\mathcal{ALCI}$ approximation algorithm, and how the decidability of depleting mCE Σ-module extraction can be regained by the introduction of an additional syntactic condition on the ontology. From Section 3.3 onward, we present our new contributions, using the $\mathcal{ALCI}$ approximation algorithm as a starting point, we propose a number of logical extensions to allow for the extraction of depleting modules from syntactic variants of acyclic terminologies, constructed from the more expressive $\mathcal{ALCQI}$. With these logical extensions in place, in Section 3.4 we exploit the properties the algorithm to propose a number of optimisations, which in turn lead to the development of a new rule-based algorithm AMEX. We show that AMEX produces an identical module to the unoptimised version, but offers measurably better practical performance. We confirm this, in Section 3.5, through a small experimental evaluation of module extraction applied to a real-world ontology. The AMEX algorithm will also become the basis for a larger, much more comprehensive investigation in Chapter 6, in which we provide a comparison between AMEX and several other approximations producing depleting mCE Σ-modules, including the locality based STAR module.

For the remainder of the thesis we mostly focus on mCE Σ-modules, and for simplicity, we simply refer to them as Σ-modules. Additionally, we drop the mod superscript and use $\equiv$ to represent the model-inseparability relation and make it explicitly clear if any modules are not Σ-modules or not based on the model-inseparability relation.

3.1. Model-inseparable modules

As we have seen, for two ontologies to be model-inseparable for a signature, for each model of one ontology there must exist a corresponding model of the other which interprets the signature symbols in the same way. This allows for the definition of modules — which are rarely logically equivalent to the ontology from which they are extracted — that are guaranteed to represent exactly the same information over a specified signature. Consider the following example of a plain module:

Example 3.1.1 (Plain Module)**.** *Let $\mathcal{O}$ be an ontology which consists of the following inclusions $\alpha_1 - \alpha_4$*

$$\mathsf{Dermal\_Neoplasm} \equiv \forall \mathsf{hasCell}.\mathsf{Neoplastic\_Cell} \sqcap \mathsf{Skin\_Neoplasm} \qquad (\alpha_1)$$

$$\mathsf{Dermal\_Neoplasm} \sqsubseteq \mathsf{Malignant\_Skin\_Neoplasm} \qquad (\alpha_2)$$

$$\mathsf{Malignant\_Skin\_Neoplasm} \sqsubseteq \exists \mathsf{hasCell}.\mathsf{Malignant\_Cell} \sqcap \mathsf{Skin\_Neoplasm} \qquad (\alpha_3)$$

$$\mathsf{Malignant\_Cell} \sqsubseteq \mathsf{Cell} \qquad (\alpha_4)$$

and let $\mathcal{M} = \{\alpha_1, \alpha_4\}$. Clearly $\mathcal{O}$ and $\mathcal{M}$ are not logically equivalent, but for the signature $\Sigma = \{\mathsf{Dermal\_Neoplasm}, \mathsf{Skin\_Neoplasm}\}$, $\mathcal{O} \equiv_\Sigma \mathcal{M}$, $\mathcal{M}$ is a plain module, and one can verify that $\{\mathcal{I}|_\Sigma \mid \mathcal{I} \models \mathcal{O}\} = \{\mathcal{J}|_\Sigma \mid \mathcal{J} \models \mathcal{M}\}$. In fact the information both the ontology and module provides over Σ can be expressed as a concept inclusion $\beta = \mathsf{Dermal\_Neoplasm} \sqsubseteq \mathsf{Skin\_Neoplasm}$, and it can be verified that $\mathcal{O} \equiv_\Sigma \mathcal{M} \equiv_\Sigma \{\beta\}$.

However, as we noted in the previous chapter, model-inseparability is strictly stronger than deduction-inseparability, and ontologies implying the same inclusions over a signature are not necessarily model-inseparable. Consider the signature $\Sigma' = \{\mathsf{Dermal\_Neoplasm}, \mathsf{Malignant\_Cell}\}$, there is no inclusion over Σ' which is implied by $\mathcal{O}$ but not $\mathcal{M}$ or vice versa. But, if we consider a model of $\mathcal{M}$, $\mathcal{I} = \{\Delta^\mathcal{I} = \{d\}$, $\mathsf{Dermal\_Neoplasm}^\mathcal{I} = \{d\}$ and $\mathsf{Malignant\_Cell}^\mathcal{I} = \emptyset, \cdots\}$, such a model exists, yet there is no model $\mathcal{J}$ of $\mathcal{O}$ such that $\mathcal{I}|_{\Sigma'} = \mathcal{J}|_{\Sigma'}$, and since $\Sigma' \subseteq \Sigma \cup \mathsf{sig}(\mathcal{M})$, $\mathcal{M}$ is not a self-contained module. Additionally, there exists no model of $\mathcal{O} \setminus \mathcal{M}$ which coincides with $\mathcal{I}$ on $\Sigma \cup \mathsf{sig}(\mathcal{M})$, so $\mathcal{M}$ is also not a depleting module.

Throughout this thesis we particularly focus on the stronger notion of depleting modules, which for the model-inseparability relation come with the guarantee of also being plain and self-contained modules by means of Proposition 2.3.1. Since every interpretation is a model of the empty ontology, in order to establish if some $\mathcal{M} \subseteq \mathcal{O}$ is a depleting module i.e. $\mathcal{O} \setminus \mathcal{M} \equiv_{\Sigma \cup \mathsf{sig}(\mathcal{M})} \emptyset$, we must ensure for every interpretation $\mathcal{I}$ there exists a model $\mathcal{J}$ of $\mathcal{O} \setminus \mathcal{M}$ which coincides with $\mathcal{I}$ on $\Sigma \cup \mathsf{sig}(\mathcal{M})$.

Consider the following example, where instead of describing the contents of a module itself, we describe the axioms left in an ontology after a module has been extracted, then by verifying if the residual axioms are inseparable from the empty ontology for $\Sigma \cup \mathsf{sig}(\mathcal{M})$, we can determine whether or not the extracted module is depleting.

Example 3.1.2 (Depleting Module)**.** *Let $\mathcal{O}$ be an ontology and $\mathcal{M}$ be module*

$\mathcal{M} \subseteq \mathcal{O}$ and let $\mathcal{O} \setminus \mathcal{M}$ consist of the axioms $\beta_1 - \beta_4$ below:

$$\mathsf{Lung\_Site} \equiv \mathsf{Lung} \sqcup \exists \mathsf{hasLocation}.\mathsf{Lung\_Tissue} \tag{β_1}$$

$$\mathsf{Membrane} \sqsubseteq \mathsf{Lung\_Tissue} \tag{β_2}$$

$$\mathsf{Pleura} \sqsubseteq \mathsf{Membrane} \tag{β_3}$$

$$\mathsf{Pleura} \sqsubseteq \forall \mathsf{hasLocation}.\mathsf{ThoracicCavity} \tag{β_4}$$

Then $\mathcal{M}$ is a depleting module where $\Sigma \cup \mathsf{sig}(\mathcal{M}) = \{\mathsf{Pleura}, \mathsf{hasLocation}\}$. To see this, let $\mathcal{I}$ be any *interpretation, and define $\mathcal{J}$ by setting $\Delta^{\mathcal{J}} = \Delta^{\mathcal{I}}$, $\mathsf{Pleura}^{\mathcal{J}} = \mathsf{Pleura}^{\mathcal{I}}$ and $\mathsf{hasLocation}^{\mathcal{J}} = \mathsf{hasLocation}^{\mathcal{I}}$, then select some $d \in \Delta^{\mathcal{J}}$ and interpret the remaining symbols $\mathcal{O} \setminus \mathcal{M}$ of in the following way:*

$$\begin{aligned}
\mathsf{Membrane}^{\mathcal{J}} &= \{d\} \cup \mathsf{Pleura}^{\mathcal{J}} \\
\mathsf{Lung_Tissue}^{\mathcal{J}} &= \{d\} \cup \mathsf{Pleura}^{\mathcal{J}} \\
\mathsf{Lung}^{\mathcal{J}} &= \{d\} \\
\mathsf{ThoracicCavity}^{\mathcal{J}} &= \{e \mid (d,e) \in \mathsf{hasLocation}^{\mathcal{J}}\}
\end{aligned}$$

and then interpret Lung_Site *by its definition*

$$\mathsf{Lung\_Site}^{\mathcal{J}} = (\mathsf{Lung} \sqcup \exists \mathsf{hasLocation}.\mathsf{Lung\_Tissue})^{\mathcal{J}}$$

one can verify that $\mathcal{J}$ is a model of $\mathcal{O} \setminus \mathcal{M}$ with $\mathcal{I}|_{\Sigma} = \mathcal{J}|_{\Sigma}$. As the choice of $\mathcal{I}$ was arbitrary, we have shown that for every interpretation $\mathcal{I}$ there always exists a model $\mathcal{J}$ of $\mathcal{O} \setminus \mathcal{M}$ that concides with $\mathcal{I}$ on $\Sigma \cup \mathsf{sig}(\mathcal{M})$ it follows that $\mathcal{M}$ is a depleting Σ-module by definition.

Now consider some subsets of $\mathcal{O}$ which are not *depleting modules, let $\mathcal{M}' \subseteq \mathcal{O}$ and $\mathcal{O} \setminus \mathcal{M}' = \{\beta_1 - \beta_4, \gamma\}$ where*

$$\gamma = \mathsf{Membrane} \sqsubseteq \neg \mathsf{Lung\_Site}$$

The axiom γ expresses the disjointness of the concepts Membrane *and* Lung_Site*,* *and although it contains no symbols from Σ itself, it does constrain the interpretation of Σ symbols. Consider the one-element interpretation $\mathcal{I}$ with $\Delta^{\mathcal{I}} = \{d\}$, $\mathsf{Pleura}^{\mathcal{I}} = \{d\}$ and $\mathsf{hasLocation}^{\mathcal{I}} = \{(d,d)\}$, then one can verify there is no model $\mathcal{J}$ of $\mathcal{O} \setminus \mathcal{M}'$ that coincides with $\mathcal{I}$ on $\Sigma \cup \mathsf{sig}(\mathcal{M})$, so $\mathcal{M}'$ is not a depleting module.*

Next, consider the subset $\mathcal{M}'' \subseteq \mathcal{O}$ with $\mathcal{O} \setminus \mathcal{M}'' = \{\beta_1 - \beta_4, \delta\}$ where

$$\delta = \mathsf{ThoracicCavity} \sqsubseteq \exists \mathsf{hasLocation}.\mathsf{Thorax}$$

Although for every one-element interpretation, we can always find an model $\mathcal{J}$ of $\mathcal{O} \setminus \mathcal{M}''$ that coincides on $\Sigma \cup \mathsf{sig}(\mathcal{M})$, for the two-element interpretation defined as $\Delta^{\mathcal{I}} = \{d, e\}, \mathsf{Pleura}^{\mathcal{I}} = \{d\}, \mathsf{hasLocation}^{\mathcal{I}} = \{(d, e)\}$ such a $\mathcal{J}$ doesn't exist, and then, again, $\mathcal{M}''$ is not a depleting module.

3.2. Acyclic $\mathcal{ALCI}$ Approximation

In the previous chapter we described how Konev et al. [Kon+08a; Kon+13] showed for acyclic $\mathcal{ELI}$ terminologies it is possible to produce the minimal depleting module in PTIME. For acyclic $\mathcal{ALCI}$ terminologies $\mathcal{T}$, deciding if some $\mathcal{M} \subseteq \mathcal{T}$ is a depleting module is undecidable, meaning automatic extraction of these modules is impossible. In answer to this, [Kon+13] introduced an approximation extraction algorithm, the theory that underpins this we recount in this section in order to establish what is already known, which in turn will provide a basis for several extensions and improvements of our own.

3.2.1 One-point criterion

The theory which motivates the $\mathcal{ALCI}$ approximation stems from that checking inseparability from the empty ontology for *concept* signatures ($\Sigma \subseteq \mathsf{sig}(\mathcal{O}) \cap \mathsf{N_C}$) is decidable. It turns out that in order to decide if an $\mathcal{ALCI}$ ontology is inseparable from the empty ontology for a signature, it is sufficient to decide inseparability considering only one-element interpretations.

Definition 3.2.1 (1-Σ-inseparability). *Let $\mathcal{O}_1$ and $\mathcal{O}_2$ be ontologies and Σ a signature. Then $\mathcal{O}_1$ and $\mathcal{O}_2$ are called 1-Σ-inseparable for Σ, in symbols $\mathcal{O}_1 \equiv^1_\Sigma \mathcal{O}_2$ if:*

$$\left\{\mathcal{I}|_\Sigma \,\middle|\, \sharp\Delta^{\mathcal{I}} = 1 \text{ and } \mathcal{I} \models \mathcal{O}_1\right\} = \left\{\mathcal{J}|_\Sigma \,\middle|\, \sharp\Delta^{\mathcal{J}} = 1 \text{ and } \mathcal{J} \models \mathcal{O}_2\right\}$$

Lemma 3.2.1 (One-Point Criterion – Konev et al. [Kon+13]). *Let $\mathcal{O}$ be a first-order ontology preserved under disjoint unions and Σ a* concept *signature. Then $\mathcal{O} \equiv_\Sigma \emptyset$ iff $\mathcal{O} \equiv^1_\Sigma \emptyset$.*

Theorem 3.2.1 (Konev et al. [Kon+13; Kon+08a]). *For an acyclic $\mathcal{ALCI}$ terminology $\mathcal{T}$ and a concept signature Σ, it is in Π^p_2 to decide whether $\mathcal{T} \equiv_\Sigma \emptyset$. The same problem is Π^p_2-hard for acyclic $\mathcal{ALC}$ TBoxes.*

The decidability of Theorem 3.2.1 follows from Lemma 3.2.1 in that to check for inseparability from the empty ontology one only needs to enumerate every one-element interpretation $\mathcal{I}$ — of which there is a finite amount — and verify if there exists an interpretation $\mathcal{J}$ such that $\mathcal{I}|_\Sigma = \mathcal{J}|_\Sigma$. Noting that when $\mathcal{I}$ is a one-element interpretation, and $\mathcal{I}|_\Sigma = \mathcal{J}|_\Sigma$ then $\mathcal{J}$ must also be a one-element interpretation, so all quantification is over objects of polynomial size.

We present here the proof of the Π^p_2 upper bound for $\mathcal{ALCI}$ taken from [Kon+13], which is achieved by reduction to $\forall\exists$-QBF. For an $\mathcal{ALCI}$ TBox $\mathcal{T}$ and concept signature Σ, we want to decide if $\mathcal{T} \equiv_\Sigma \emptyset$, the first step in the reduction is to convert $\mathcal{T}$ into a propositional formula, starting with the concepts which make up the axioms. Firstly taking a propositional variable p_A for each concept name $A \in \Sigma$ and a distinct concept name q_X for each symbol $X \in \mathsf{sig}(\mathcal{T}) \setminus \Sigma$. The translation is inductively extended to convert arbitrary complex concepts into propositional formulas $D^\dagger$:

$$
\begin{aligned}
A^\dagger &= p_A \quad \text{for all } A \in \Sigma \cap \mathsf{N_C} \\
A^\dagger &= q_A \quad \text{for all } A \in (\mathsf{sig}(\mathcal{T}) \setminus \Sigma) \cap \mathsf{N_C} \\
(D_1 \sqcap D_2)^\dagger &= D_1^\dagger \sqcap D_2^\dagger \\
(\neg D)^\dagger &= \neg D^\dagger \\
(\exists r.D)^\dagger = (\exists r^-.D)^\dagger &= q_r \wedge D^\dagger \qquad \text{for all } r \in \mathsf{sig}(\mathcal{T}) \cap \mathsf{N_R}
\end{aligned}
$$

So that for a one-element interpretation $\mathcal{I}$ with $\Delta^\mathcal{I} = \{d\}$ and $\mathfrak{v}$ a truth assignment that:

- $d \in A^\mathcal{I}$ iff $\mathfrak{v}(p_A) = 1$ for all $A \in \Sigma \cap \mathsf{N_C}$
- $d \in A^\mathcal{I}$ iff $\mathfrak{v}(q_A) = 1$ for all $A \in (\mathsf{sig}(\mathcal{T}) \setminus \Sigma) \cap \mathsf{N_C})$
- $(d, d) \in r^\mathcal{I}$ iff $\mathfrak{v}(q_r) = 1$ for all $A \in (\mathsf{sig}(\mathcal{T}) \cap \mathsf{N_R})$

Then $d \in D^\mathcal{I}$ iff $\mathfrak{v}(D^\dagger)$ for all $\mathcal{ALCI}$ concepts D over $\mathsf{sig}(\mathcal{T})$. The propositional translation is then extended to entire TBoxes

$$
\mathcal{T}^\dagger = \bigwedge_{C \sqsubseteq D} C^\dagger \rightarrow D^\dagger \wedge \bigwedge_{C \equiv D} C^\dagger \leftrightarrow D^\dagger
$$

to ensure that $\mathcal{I} \models \mathcal{T}$ iff $\mathfrak{v}(\mathcal{T}^\dagger) = \mathit{true}$. This can be used to verify for a particular one-element interpretation $\mathcal{I}$ whether it is a model of $\mathcal{T}$. To ensure that for every $\mathcal{I}$ there is a $\mathcal{J}$ such that $\mathcal{I}|_\Sigma = \mathcal{J}|_\Sigma$ and $\mathcal{J} \models \mathcal{T}$ we must ensure that however $\mathcal{I}$ interprets Σ symbols (concepts) some $\mathcal{J}$ interprets them identically, and that $\mathcal{J}$ is a model of $\mathcal{T}$. Let $\vec{p}$ denote the sequence of variables p_A for $A \in \Sigma$ and $\vec{p}$ denote those p_X with $X \in \mathsf{sig}(\mathcal{T}) \setminus \Sigma$ then the QBF formula

$$
\varphi_\mathcal{T} = \forall \vec{p} \exists \vec{q} \mathcal{T}^\dagger
$$

is logically true, by Lemma 3.2.1, iff $\mathcal{T} \equiv_\Sigma \emptyset$, and checking if a QBF formula of the form $\forall \vec{p} \exists \vec{q} \varphi$ is logically *true*, is well known to be Π_2^p-complete [Bie+09].

Although not explicitly proven in [Kon+13], the authors note that the upper-bound in the case of concept signatures is extremely robust under modifications of the description logic involved, with the Π_2^p lower bound already holding acyclic $\mathcal{ALC}$ TBoxes without role names, and the Π_2^p upper bound still holding for very expressive description logics such as $\mathcal{SHIQ}$ and first-order ontologies with models which are known to be preserved under disjoint unions [LPW11]. This motivates us to extend the upper-bound proof, in Section 3.3.2, to cover the more expressive logic $\mathcal{ALCQI}$.

The authors also note that Lemma 3.2.1 fails if one wants to decide model-inseparability between two TBoxes instead of just from the empty ontology, or when Σ contains a role name.

3.2.2 Unrestricted signatures

Konev et al. [Kon+08b; Kon+13] also identify a syntactic condition, known as direct Σ-dependencies, which can be used to identify those axioms of an acyclic terminology $\mathcal{T}$ which *may* cause it to be separable w.r.t a signature Σ from the empty ontology i.e. $\mathcal{T} \not\equiv_\Sigma \emptyset$. Their key observation is that if $\mathcal{T}$ is an acyclic $\mathcal{ALCI}$ terminology, and $\mathcal{T}$ contains no direct Σ-dependencies, then deciding inseparability for unrestricted signatures becomes equivalent to that of deciding inseparability for concept signatures, and hence decidable.

This property is particularly useful for deciding if $\mathcal{M} \subseteq \mathcal{T}$ is a depleting module. If all axioms causing a direct $\Sigma \cup \mathsf{sig}(\mathcal{M})$-dependency are moved straight to $\mathcal{M}$, then verifying if $\mathcal{T} \setminus \mathcal{M} \equiv_{\Sigma \cup \mathsf{sig}(\mathcal{M})} \emptyset$ becomes decidable. However, in the case of $\mathcal{ALCI}$, this syntactic condition is not entirely accurate, and may identify axioms which do not cause separability for $\Sigma \cup \mathsf{sig}(\mathcal{M})$, yet it is necessary to move them to $\mathcal{M}$ in order to regain decidability. The consequence of this is that $\mathcal{M}$ may not be the minimal depleting Σ-module but an approximation.

Direct Σ-dependencies

Firstly, using the depends relation $\prec_{\mathcal{T}}$ used to define acyclicity in Section 2.1.2, denote by $\prec_{\mathcal{T}}^{+}$ the transitive closure of the $\prec_{\mathcal{T}}$ relation, and set $\mathsf{depend}_{\mathcal{T}}(A) = \{X \mid A \prec_{\mathcal{T}}^{+} X\}$. Intuitively $\mathsf{depend}_{\mathcal{T}}(A)$ consists of all the symbols which are used in the definition of A in $\mathcal{T}$.

Definition 3.2.2 (Direct Σ-dependencies – Konev et al. [Kon+08a; Kon+13]). *Let $\mathcal{T}$ be an acyclic terminology, Σ a signature, and $A \in \Sigma$. We say that A* has a direct Σ-dependency in $\mathcal{T}$ *if* $\mathsf{depend}_{\mathcal{T}}(A) \cap \Sigma \neq \emptyset$. *We say that $\mathcal{T}$* contains an direct Σ-dependency *when there is an $A \in \Sigma$ that has an direct Σ-dependency in $\mathcal{T}$. We sometimes say A depends on a symbol X when $X \in \mathsf{depend}_{\mathcal{T}}(A) \cap \Sigma$.*

The notion of direct Σ-dependencies generalises the notion of acyclicity ($A \notin \mathsf{depend}_{\mathcal{T}}(A)$) to describe if within an acyclic terminology there exists a relationship between two symbols taken from a signature. It is known that the presence of a direct Σ-dependency in an acyclic $\mathcal{ELI}$ terminology implies that $\mathcal{T} \not\equiv_{\Sigma} \emptyset$ [Kon+13; Kon+08a].

Example 3.2.1. *Let $\mathcal{T}$ be the following $\mathcal{EL}$ acyclic terminology*
$\mathcal{T} = \{\mathsf{Dog} \sqsubseteq \exists \mathsf{hasOwner}.\mathsf{Human}\}$ *and* $\Sigma = \{\mathsf{Dog}, \mathsf{Human}\}$.

$\mathcal{T}$ contains a direct Σ-dependency, since $\mathsf{Dog} \in \Sigma$ and $\mathsf{depend}_{\mathcal{T}}(\mathsf{Dog}) \cap \Sigma = \{\mathsf{Human}\}$. To show that $\mathcal{T} \not\equiv_{\Sigma} \emptyset$, let $\mathcal{I}$ be the following interpretation $\Delta^{\mathcal{I}} = \{d\}$, $\mathsf{Dog}^{\mathcal{I}} = \{d\}$ and $\mathsf{Human}^{\mathcal{I}} = \mathsf{hasOwner}^{\mathcal{I}} = \emptyset$ then one can verify there is no model $\mathcal{J}$ of $\mathcal{T}$ such that $\mathcal{I}|_{\Sigma} = \mathcal{J}|_{\Sigma}$, one cannot find a model $\mathcal{J}$ of $\mathcal{T}$ without reinterpreting Σ-symbols from $\mathcal{I}$.

This property fails however when one considers, even simple ontologies, formulated in the more expressive $\mathcal{ALC}$.

Example 3.2.2. *Let $\mathcal{T}$ be the following $\mathcal{ALC}$ acyclic terminology*
$\mathcal{T} = \{\mathsf{Pet} \sqsubseteq \mathsf{Dog} \sqcup \mathsf{Cat}\}$ *and* $\Sigma = \{\mathsf{Pet}, \mathsf{Cat}\}$.

Clearly $\mathcal{T}$ contains a direct dependency (Pet depends on Cat), but notice that $\mathcal{T} \equiv_{\Sigma} \emptyset$, to see this, consider any interpretation $\mathcal{I}$ and the interpretation $\mathcal{J}$ which *is identical to $\mathcal{I}$ except for setting* $\mathsf{Dog}^{\mathcal{J}} = \mathsf{Pet}^{\mathcal{I}}$*, then* $\mathcal{J} \models \mathcal{T}$ *and* $\mathcal{I}|_\Sigma = \mathcal{J}|_\Sigma$*, and since* $\mathcal{I}$ *was arbitrary such a* $\mathcal{J}$ *always exists.*

Deciding inseparability from the empty ontology is then decidable if an acyclic $\mathcal{ALCI}$ ontology contains no direct Σ-dependencies, taken from the following lemma:

Lemma 3.2.2 ([Kon+13]). *Let* $\mathcal{T}$ *be an acyclic* $\mathcal{ALCI}$ *terminology* Σ *a signature and let:*

$$\mathsf{Lhs}_\Sigma(\mathcal{T}) = \{A \bowtie C \in \mathcal{T} \mid A \in \Sigma \textbf{ or } \exists X \in \Sigma\ (A \in \mathsf{depend}_\mathcal{T}(X))\}$$

then for every interpretation $\mathcal{I}$ *the following are equivalent:*

1. *there is a model* $\mathcal{J}$ *of* $\mathcal{T}$ *with* $\mathcal{J}|_\Sigma = \mathcal{I}|_\Sigma$
2. *there is a model* $\mathcal{J}$ *of* $\mathsf{Lhs}_\Sigma(\mathcal{T})$ *with* $\mathcal{J}|_\Sigma = \mathcal{I}|_\Sigma$

Intuitively, the set $\mathsf{Lhs}_\Sigma(\mathcal{T})$ contains all axioms of $\mathcal{T}$ that are influenced by Σ. The key observation in regaining decidability is that if an acyclic $\mathcal{ALCI}$ terminology $\mathcal{T}$ contains no direct Σ-dependencies then $\mathsf{Lhs}_\Sigma(\mathcal{T})$ contains no role name from Σ. Given an acyclic $\mathcal{ALCI}$ terminology $\mathcal{T}$ such that $\mathcal{T}$ contains no direct Σ-dependencies, by Lemma 3.2.2, $\mathcal{T} \equiv_\Sigma \emptyset$ iff $\mathsf{Lhs}_\Sigma(\mathcal{T}) \equiv_\Sigma \emptyset$ but since $\mathcal{T}$ contains no direct Σ-dependencies then $\mathsf{Lhs}_\Sigma(\mathcal{T})$ contains no role name from Σ, and since model-inseparability is monotone by Proposition 2.3.3 then $\mathsf{Lhs}_\Sigma(\mathcal{T}) \equiv_\Sigma \emptyset$ iff $\mathsf{Lhs}_\Sigma(\mathcal{T}) \equiv_{\Sigma \cap \mathsf{N_C}} \emptyset$ which by Theorem 3.2.2 can be decided in Π_2^p.

Theorem 3.2.2 (Konev et al. [Kon+13]). *Given an acyclic* $\mathcal{ALCI}$ *terminology* $\mathcal{T}$ *and signature* Σ*, such that* $\mathcal{T}$ *contains no direct* Σ*-dependencies, it is in* Π_2^p *to decide whether* $\mathcal{T} \equiv_\Sigma \emptyset$*.*

3.2.3 Approximation Extraction Algorithm

A consequence of Theorem 3.2.2 enables a modified version of the general algorithm producing the unique minimal depleting module for an inseparability relation (Figure 2.6), which can be applied to acyclic $\mathcal{ALCI}$ terminologies. The result of the modification is shown in Figure 3.1. The modification simply ensures that on each iteration of the algorithm, any subset $\mathcal{W} \subseteq \mathcal{T} \setminus \mathcal{M}$ does not contain a direct $\Sigma \cup \mathsf{sig}(\mathcal{M})$ dependency, guaranteeing that $\mathcal{W} \not\equiv_{\Sigma \cup \mathsf{sig}(\mathcal{M})} \emptyset$ is decidable (Line 6).

Theorem 3.2.3 (Konev et al. [Kon+13]). *Let $\mathcal{T}$ be an acyclic $\mathcal{ALCI}$-TBox and Σ a signature. Then the algorithm given in Figure 3.1 computes the unique minimal depleting Σ-module $\mathcal{M}$ such that $\mathcal{T} \setminus \mathcal{M}$ does not have any direct $(\Sigma \cup \mathsf{sig}(\mathcal{M}))$-dependencies.*

Input: Acyclic $\mathcal{ALCI}$ TBox $\mathcal{T}$, Signature Σ
Output: Minimal depleting module $\mathcal{M}$ such that $\mathcal{T} \setminus \mathcal{M}$ has no direct $\Sigma \cup \mathsf{sig}(\mathcal{M})$-dependency

1 $\mathcal{M} ::= \emptyset$
2 $\mathcal{W} ::= \emptyset$
3 **while** $\mathcal{T} \setminus \mathcal{M} \neq \mathcal{W}$ **do**
4   choose $\alpha \in (\mathcal{T} \setminus \mathcal{M}) \setminus \mathcal{W}$
5     $\mathcal{W} ::= \mathcal{W} \cup \{\alpha\}$
6   **if** *$\mathcal{W}$ contains a direct $(\Sigma \cup \mathsf{sig}(\mathcal{M}))$-dependency* ***or*** $\mathcal{W} \not\equiv_{\Sigma \cup \mathsf{sig}(\mathcal{M})} \emptyset$ **then**
7         $\mathcal{M} ::= \mathcal{M} \cup \{\alpha\}$
8         $\mathcal{W} ::= \emptyset$
9   **end**
10 **end**
11 **return** $\mathcal{M}$

Figure 3.1: Original module extraction algorithm

It is possible the syntactic check for direct $\Sigma \cup \mathsf{sig}(\mathcal{M})$-dependencies may capture axioms which do not cause separability in $\mathcal{T} \setminus \mathcal{M}$— shown by the simple example Example 3.2.2 — i.e. $\mathcal{M}$ is still a depleting module without these axioms included, yet they need to be removed from $\mathcal{T} \setminus \mathcal{M}$ to ensure we can decide that $\mathcal{M}$ is a depleting module. The module produced by Figure 3.1 is therefore an approximation and may not coincide with the unique minimal depleting module, which is known to exist by Proposition 2.3.2, but the decidability constraints mean it cannot be automatically extracted.

As for the runtime of the algorithm, recall in the general case is in $O((|\mathcal{O}| +$

$|\Sigma|)^2 \times T_c(\mathcal{O}, \Sigma))$, where $T_c(\mathcal{O}, \Sigma)$ is the time needed to check for an ontology $\mathcal{O}$ and signature Σ whether $\mathcal{O} \equiv_\Sigma \emptyset$. In the case of acyclic $\mathcal{ALCI}$ terminologies $\mathcal{T}$, checking for direct Σ-dependencies can be achieved by simple reachability analysis which is in $O(|\mathcal{T}|)$, and deciding if $\mathcal{T} \equiv_\Sigma \emptyset$ where $\mathcal{T}$ contains no direct Σ-dependencies is in Π_2^p.

3.3. Logical extensions

Now we have established what is already known about the model-theoretic based $\mathcal{ALCI}$ approximation algorithm, we next consider extending the theorems in the previous section to facilitate depleting module extraction variant of acyclic terminologies that admit *repeated concept inclusions*, which, in addition, may be constructed using the more expressive logic $\mathcal{ALCQI}$. We note the new extensions we present are strictly generalisations and are still applicable to acyclic $\mathcal{ALCI}$ terminologies.

3.3.1 Terminologies with repeated concept inclusions

A variant of acyclic terminologies are those with repeated concept inclusions. These ontologies satisfy all the conditions for being an acyclic terminology with the exception that they contain repeated concept inclusions (RCIs) of the form $A \sqsubseteq C_1, \ldots, A \sqsubseteq C_n$ for some concept name A which we call a *repeated concept name*. A terminology containing at least one repeated concept name we call *acyclic terminology with RCIs*. Real-world ontologies which contain RCIs include the important NCI ontology, the 08.09d version of which contains 14,326 repeated concept names with up to 31 RCIs for a single repeated name.

One can convert such a ontology into a logically equivalent acyclic terminology by replacing each RCI of the form $A \sqsubseteq C_1, \ldots, A \sqsubseteq C_n$ with a fresh axiom $A \sqsubseteq C_1 \sqcap \cdots \sqcap C_n$. However, such an explicit conversion is an unattractive solution when targeting an ontology for depleting module extraction, because if the fresh axiom is added to a module, the signature of the module now contains every symbol from $\mathsf{sig}(C_i)$ for $1 \leq i \leq n$. As extracting depleting modules considers $\Sigma \cup \mathsf{sig}(\mathcal{M})$, a larger amount of symbols within $\mathsf{sig}(\mathcal{M})$ comes with the inherent risk of increasing the size of the resulting module considerably.

Example 3.3.1 (Repeated Concept Inclusions). *Consider the logically equivalent $\mathcal{EL}$ terminologies $\mathcal{T}_1 = \{\alpha_1 - \alpha_5\}$ and $\mathcal{T}_2 = \{\beta_1 - \beta_3\}$, and notice $\mathcal{T}_2$ is identical to $\mathcal{T}_1$ apart from the RCIs for the concept name* Insulin *are joined to produce the single concept inclusion β_1.*

$$\begin{aligned}
\mathcal{T}_1 = \{\mathsf{Insulin} &\sqsubseteq \mathsf{Pancreatic_Product} && (\alpha_1)\\
\mathsf{Insulin} &\sqsubseteq \mathsf{Hormone} && (\alpha_2)\\
\mathsf{Insulin} &\sqsubseteq \mathsf{Biological_Product} && (\alpha_3)\\
\mathsf{Pancreatic_Product} &\sqsubseteq \mathsf{Hormone} && (\alpha_4)\\
\mathsf{Hormone} &\sqsubseteq \mathsf{Biological_Product}\} && (\alpha_5)
\end{aligned}$$

$$\begin{aligned}
\mathcal{T}_2 = \{\mathsf{Insulin} &\sqsubseteq \mathsf{Pancreatic_Product} \sqcap \mathsf{Biological_Product} \sqcap \mathsf{Hormone} && (\beta_1)\\
\mathsf{Pancreatic_Product} &\sqsubseteq \mathsf{Hormone} && (\beta_2)\\
\mathsf{Hormone} &\sqsubseteq \mathsf{Biological_Product}\} && (\beta_3)
\end{aligned}$$

For the signature $\Sigma = \{\mathsf{Insulin}, \mathsf{Pancreatic\_Product}\}$ consider the unique minimal depleting modules extracted from each ontology. The module $\mathcal{M}_1$ extracted from $\mathcal{T}_1$ consists of a single axiom $\mathcal{M}_1 = \{\alpha_1\}$, whereas the module $\mathcal{M}_2$ for $\mathcal{T}_2$ consists of the whole ontology $\mathcal{M}_2 = \mathcal{T}_2$.

This is because $\mathcal{M}_2$ must contain the axiom β_1 as it expresses non-trivial information over Σ which resulting in us having

$$\{\mathsf{Pancreatic\_Product}, \mathsf{Biological\_Product}, \mathsf{Hormone}\} \subseteq (\Sigma \cup \mathsf{sig}(\mathcal{M}_2))$$

which then "pulls in" the axioms β_2 and β_3 to ensure $\mathcal{T}_2 \setminus \mathcal{M}_2 \equiv_{\Sigma \cup \mathsf{sig}(\mathcal{M}_2)} \emptyset$.

Conversely, $\mathcal{M}_1$ only contains single axiom α_1 which we can show to be a depleting module of $\mathcal{T}_1$ for Σ. Let $\mathcal{I}$ be any interpretation, and con-

struct a model $\mathcal{J}$ *of* $\mathcal{T}_1 \setminus \mathcal{M}_1$ *as follows: set* $\Delta^{\mathcal{J}} = \Delta^{\mathcal{I}}$, $\mathsf{Insulin}^{\mathcal{J}} = \mathsf{Insulin}^{\mathcal{I}}$ *and* $\mathsf{Pancreatic\_Product}^{\mathcal{J}} = \mathsf{Pancreatic\_Product}^{\mathcal{I}}$ *and* $\mathsf{Biological\_Product}^{\mathcal{J}} = \mathsf{Hormone}^{\mathcal{J}} = (\mathsf{Insulin}^{\mathcal{J}} \cup \mathsf{Pancreatic\_Product}^{\mathcal{J}})$*, then* $\mathcal{I}|_{\Sigma \cup \mathsf{sig}(\mathcal{M}_1)} = \mathcal{J}|_{\Sigma \cup \mathsf{sig}(\mathcal{M}_1)}$ *and* $\mathcal{J} \models (\mathcal{T}_1 \setminus \mathcal{M}_1)$ *and so* $\mathcal{T}_1 \setminus \mathcal{M}_1 \equiv_{\Sigma \cup \mathsf{sig}(\mathcal{M}_1)} \emptyset$.

By not combining every RCI into a single axiom we obtain a module which is a much smaller percentage of the original ontology. This module can also be considered a better approximation, as it only contains axioms which represent knowledge over the input signature rather than those which are collected as a side-effect of converting our ontology into an valid acyclic terminology.

To avoid the situation described in Example 3.3.1, we take a more considered approach, extending the existing notions for depleting module extraction for acyclic terminologies to those which *may* additionally contain RCIs. This enables us to identify those RCIs which are relevant to $\Sigma \cup \mathsf{sig}(\mathcal{M})$ so that we may only target them for extraction into a module, preventing irrelevant axioms unnecessarily appearing in our extracted modules.

Per-axiom direct Σ-dependencies

We first extend the notion of direct Σ-dependencies defined on on a per-axiom basis, starting with the generation of dependencies from a terminology with RCIs.

Definition 3.3.1. *(Per-axiom dependencies) Given an acyclic terminology with RCIs* $\mathcal{T}_{\mathsf{RCI}}$*, define* $\mathsf{depend}_{\mathcal{T}_{\mathsf{RCI}}}(A \bowtie C)$ *for an axiom* $A \bowtie C \in \mathcal{T}_{\mathsf{RCI}}$

$$\mathsf{depend}_{\mathcal{T}_{\mathsf{RCI}}}(A \bowtie C) = \bigcup_{X \in \mathsf{sig}(C) \cap \mathsf{N_C}} \mathsf{depend}_{\mathcal{T}_{\mathsf{RCI}}}(X)$$

Intuitively, the set $\mathsf{depend}_{\mathcal{T}_{\mathsf{RCI}}}$ defines for an axiom, the set of symbols that an *axiom* uses in its definition, even indirectly, allowing us to establish which axioms use a particular symbol from $\mathsf{sig}(\mathcal{T})$.

Example 3.3.2. *Let $\mathcal{T}$ be the following acyclic terminology with RCIs:*

$$\mathsf{Cat} \sqsubseteq \mathsf{Pet} \tag{3.1}$$

$$\mathsf{Dog} \sqsubseteq \mathsf{Pet} \tag{3.2}$$

$$\mathsf{Dog} \sqsubseteq \forall \mathsf{eats}.\mathsf{Meat} \tag{3.3}$$

$$\mathsf{Pet} \equiv \mathsf{Animal} \tag{3.4}$$

By Definition 3.3.1: $\mathsf{depend}_{\mathcal{T}_{\mathsf{RCI}}}(\mathsf{Pet} \equiv \mathsf{Animal}) = \{\mathsf{Animal}\}$ *and* $\mathsf{depend}_{\mathcal{T}_{\mathsf{RCI}}}(\mathsf{Dog} \sqsubseteq \forall \mathsf{eats}.\mathsf{Meat}) = \{\mathsf{eats}, \mathsf{Meat}\}$ *as they only contain undefined concepts and role names. Now by the transitive nature of the* depends *relation* $\mathsf{depend}_{\mathcal{T}_{\mathsf{RCI}}}(\mathsf{Dog} \sqsubseteq \mathsf{Pet}) = \mathsf{depend}_{\mathcal{T}_{\mathsf{RCI}}}(\mathsf{Cat} \sqsubseteq \mathsf{Pet}) = \{\mathsf{Pet}, \mathsf{Animal}\}$.

Notice that if a concept name is not repeated in $\mathcal{T}$ then $\mathsf{depend}_{\mathcal{T}}(A) = \mathsf{depend}_{\mathcal{T}_{\mathsf{RCI}}}(A \bowtie C)$ for the single axiom $A \bowtie C \in \mathcal{T}$. That is, per-axiom dependencies are still general enough to apply to those TBoxes without RCIs. This property also allows us to generalise the definition of acyclicity to those terminologies with RCIs.

Definition 3.3.2 (Acyclic Terminology). *Given a terminology with RCIs $\mathcal{T}_{\mathsf{RCI}}$, we say it is* acyclic *if for all axioms $A \bowtie C \in \mathcal{T}$ we have $A \notin \mathsf{depend}_{\mathcal{T}_{\mathsf{RCI}}}(A \bowtie C)$, otherwise it is called* cyclic.

The notion of direct Σ-dependencies (Definition 3.2.2) links the dependencies between symbols used in the definition of a concept, to that of Σ-inseparability, a direct Σ-dependency present in a $\mathcal{ALCI}$ acyclic terminology may imply separability from the empty ontology. The following notion is analogous to direct Σ-dependencies but defined over axioms.

Definition 3.3.3 (Axiom dependencies). *An* axiom $A \bowtie C$ *causes an axiom Σ-dependency in an acyclic terminology with RCIs $\mathcal{T}_{\mathsf{RCI}}$, for a signature Σ, if $A \in \Sigma$ and* $\mathsf{depend}_{\mathcal{T}_{\mathsf{RCI}}}(A \bowtie C) \cap \Sigma \neq \emptyset$. *We say $\mathcal{T}$ contains an axiom Σ-dependency if there is axiom $\alpha \in \mathcal{T}$ which has an axiom Σ-dependency.*

More importantly, by Theorem 3.2.2, for a signature Σ, if no direct Σ-dependencies are present in an acyclic $\mathcal{ALCI}$ terminology $\mathcal{T}$ then it is decidable to verify if $\mathcal{T} \equiv_\Sigma \emptyset$, which is essential for computing depleting modules. Motivated by this idea, in the next section we will show decidability for the same problem is also possible for $\mathcal{ALCQI}$ terminologies with RCIs, as long as no axiom Σ-dependencies are present.

3.3.2 Deciding inseparability for acyclic $\mathcal{ALCQI}$ with RCIs

Concept signatures for acyclic $\mathcal{ALCQI}$

To extract a depleting module an acyclic $\mathcal{ALCQI}$ ontology with RCIs, we must be able to decide inseparability from the empty ontology for unrestricted signature. Towards this result we first we prove that deciding if $\mathcal{T}_{\mathsf{RCI}} \equiv_\Sigma \emptyset$ is still decidable where $\mathcal{T}_{\mathsf{RCI}}$ is an acyclic $\mathcal{ALCQI}$ terminology with RCIs and Σ is a concept signature. We achieve this extending the reduction provided by Lemma 3.2.1.

$$(\geq n\, r.C)^\dagger = \begin{cases} n = 0 & \top \\ n = 1 & q_r \wedge C^\dagger \\ n > 1 & \bot \end{cases}$$

$$(\leq n\, r.C)^\dagger = \begin{cases} n = 0 & \neg q_r \vee \neg C^\dagger \\ n = 1 & \top \\ n > 1 & \top \end{cases}$$

$$(= n\, r.C)^\dagger = (\leq n\, r.C)^\dagger \wedge (\geq n\, r.C)^\dagger$$

Figure 3.2: Translation of cardinality restrictions into propositional formulas

To extend the reduction, we consider how arbitrary cardinality restrictions can be translated into propositional formulas in order to determine the validity of acyclic $\mathcal{ALCQI}$ terminologies with RCIs under one-element interpretations. The translation of a cardinality restriction D to a propositional formula $D^\dagger$ is shown in Figure 3.2, where q_r is a propositional variable associated with the role name r.

With the extended translation in place, the translation of $\mathcal{ALCQI}$ axioms to propositional formulas, and construction of the QBF formula is achieved in the same fashion described in the proof of Theorem 3.2.1, which leads to the following lemma:

Lemma 3.3.1. *For an acyclic $\mathcal{ALCQI}$ terminology with RCIs $\mathcal{T}_{\mathsf{RCI}}$ and a concept signature Σ, it is in Π_2^p to decide whether $\mathcal{T}_{\mathsf{RCI}} \equiv_\Sigma \emptyset$.*

Proof. To decide if $\mathcal{T}_{\mathsf{RCI}} \equiv_\Sigma \emptyset$ it is sufficient to decide if $\mathcal{T}_{\mathsf{RCI}} \equiv^1_\Sigma \emptyset$ by Lemma 3.2.1, as $\mathcal{ALCQI}$ has models preserved under disjoint unions [LPW11]. The bound for deciding if $\mathcal{T}_{\mathsf{RCI}} \equiv^1_\Sigma \emptyset$ is given using the same reduction to QBF used in the proof of Lemma 3.2.1, extending the result from $\mathcal{ALCI}$ to $\mathcal{ALCQI}$ by adding the translation of arbitrary cardinality restrictions to propositional formulas as described in Figure 3.2. This still amounts to deciding the validity of a $\forall\exists$-QBF formula, which is in Π_2^p.

$\square$

In Chapter 5 we establish the correctness of the reductions used in both Lemma 3.2.1 and Lemma 3.3.1 where we examine exactly n-Σ-inseparability for ontologies up to $\mathcal{SHIQ}$ in expressitivity, which we will go on to utilise in evaluating the success of approximations. Exactly n-Σ-inseparability is defined as inseparability from the empty ontology if considering only interpretations of size n for some $0 < n$, generalising the case of 1-Σ-inseparability, and like 1-Σ-inseparability is reducible to $\forall\exists$-QBF.

Regaining decidability for unrestricted signatures

To show decidability, in the Π_2^p bound, for unrestricted signatures, we provide a proof analogous to Lemma 3.2.2 which applies to acyclic $\mathcal{ALCQI}$ terminologies with RCIs.

First we introduce a notion of definitional depth, which is useful for ordering the axioms for acyclic terminologies. We use a definition of this notion to include RCIs.

Definition 3.3.4 (Definitorial depth $d_{\mathcal{T}_{\mathsf{RCI}}}$)**.** *For an acyclic terminology with RCIs $\mathcal{T}_{\mathsf{RCI}}$, set $d_{\mathcal{T}_{\mathsf{RCI}}}(A) = 0$ if there is no $A \bowtie C \in \mathcal{T}_{\mathsf{RCI}}$.* If $A \bowtie C \in \mathcal{T}_{\mathsf{RCI}}$ *is not repeated for a concept name A set $d_{\mathcal{T}_{\mathsf{RCI}}}(A) = 1 + max\{d_{\mathcal{T}_{\mathsf{RCI}}}(B) \mid$ B occurs in C$\}$ for some concept name B. For a repeated concept name A with axioms $A \sqsubseteq C_1, A \sqsubseteq C_2, \ldots, A \sqsubseteq C_n \in \mathcal{T}_{\mathsf{RCI}}$ set $d_{\mathcal{T}_{\mathsf{RCI}}}(A) = 1 + max\{d_{\mathcal{T}_{\mathsf{RCI}}}(B) \mid B$ occurs in any $C_i,\ 0 < i \leq n\}$.*

The "defined" concepts of $\mathcal{T}$ are defined as $\mathsf{Def}(\mathcal{T}_{\mathsf{RCI}}) = \{A \mid A \equiv C \in \mathcal{T}\}$, the undefined ones $\mathsf{Undef}(\mathcal{T}_{\mathsf{RCI}}) = \{A \mid A \bowtie C \notin \mathcal{T}\}$ i.e. concepts with $d_{\mathcal{T}_{\mathsf{RCI}}}(A) = 0$. The ordering asserted by definitorial guarantees that for any axiom $A \bowtie C \in \mathcal{T}$ that if $B \in \mathsf{sig}(C)$ then $d_{\mathcal{T}_{\mathsf{RCI}}}(B) < d_{\mathcal{T}_{\mathsf{RCI}}}(A)$.

Example 3.3.3. *Consider the following acyclic terminology with RCIs:*

$$
\begin{aligned}
\mathcal{T} = \{\mathsf{Animal} &\sqsubseteq \exists \mathsf{eats}.\mathsf{Meat} \\
\mathsf{Dog} &\sqsubseteq \mathsf{Animal} \\
\mathsf{Dog} &\sqsubseteq \forall \mathsf{eats}.\mathsf{Meat} \\
\mathsf{Meat} &\sqsubseteq \mathsf{Food}\}
\end{aligned}
$$

Since $d_{\mathcal{T}_{\mathsf{RCI}}}(\mathsf{Food}) = 0$, Food *is undefined, then, as* Food *appears in the definition of* Meat *it follows that $d_{\mathcal{T}_{\mathsf{RCI}}}(\mathsf{Meat}) = 1$ similarly as* Meat *appears in the definition of* Animal*, $d_{\mathcal{T}_{\mathsf{RCI}}}(\mathsf{Animal}) = 2$. Now, the repeated concept name* Dog *uses both* Animal *and* Meat *in its definition, taking the maximal values of these concepts results in $d_{\mathcal{T}_{\mathsf{RCI}}}(\mathsf{Dog}) = 3$.*

Lemma 3.3.2. *Let $\mathcal{T}_{\mathsf{RCI}}$ be an acyclic $\mathcal{ALCQI}$ terminology with RCIs and Σ a signature and let:*

$$\mathsf{Lhs}_{\Sigma}(\mathcal{T}_{\mathsf{RCI}}) = \{A \bowtie C \in \mathcal{T}_{\mathsf{RCI}} \mid A \in \Sigma \textbf{ or } \exists X \in \Sigma\ (A \in \mathsf{depend}_{\mathcal{T}_{\mathsf{RCI}}}(X))\}$$

For every interpretation $\mathcal{I}$ the following are equivalent:

1. *there is a model $\mathcal{J}$ of $\mathcal{T}_{\mathsf{RCI}}$ with $\mathcal{J}|_{\Sigma} = \mathcal{I}|_{\Sigma}$*

2. *there is a model $\mathcal{J}$ of* $\mathsf{Lhs}_\Sigma(\mathcal{T}_{\mathsf{RCI}})$ *with* $\mathcal{J}|_\Sigma = \mathcal{I}|_\Sigma$

Proof. From (1) $\Rightarrow$ (2) is immediate. For the proof from (2) $\Rightarrow$ (1). Let $\mathcal{J}$ be a model of $\mathsf{Lhs}_\Sigma(\mathcal{T}_{\mathsf{RCI}})$ such that $\mathcal{I}|_\Sigma = \mathcal{J}|_\Sigma$. Let $\Sigma' = \mathsf{sig}(\mathcal{T}_{\mathsf{RCI}}) \setminus \mathsf{sig}(\mathsf{Lhs}_\Sigma(\mathcal{T}_{\mathsf{RCI}}))$ Obtain an interpretation $\mathcal{J}'$ by setting $\Delta^{\mathcal{J}'} = \Delta^{\mathcal{J}}$ and:

- $X^{\mathcal{J}'} = X^{\mathcal{J}}$ for all $X \in (\mathsf{N_C} \cup \mathsf{N_R}) \setminus \Sigma'$
- $r^{\mathcal{J}'} = r^{\mathcal{J}}$ for all role names $r \in \Sigma'$
- For concept names $A \in \Sigma'$ the definition of $A^{\mathcal{J}'}$ is by induction on the definitorial depth of A. Set $A^{\mathcal{J}'} = A^{\mathcal{J}}$ with $d_{\mathcal{T}_{\mathsf{RCI}}}(A) = 0$. Assume $B^{\mathcal{J}'}$ has been defined for all B with $d_{\mathcal{T}_{\mathsf{RCI}}}(B) = n$. Let $A \in \Sigma'$ with $d_{\mathcal{T}_{\mathsf{RCI}}}(A) = n + 1$. If $A \notin \mathsf{Def}(\mathcal{T}_{\mathsf{RCI}})$ then set $A^{\mathcal{J}'} = \emptyset$ otherwise there is a unique concept definition $A \equiv C$ such that each $B^{\mathcal{J}'}$ is defined for all $B \in \mathsf{sig}(C)$. Set $A^{\mathcal{J}'} = C^{\mathcal{J}'}$

Since $X^{\mathcal{J}'} = X^{\mathcal{J}}$ for all $X \in (\mathsf{N_C} \cup \mathsf{N_R}) \setminus \Sigma'$ and additionally for all role names $r \in \mathsf{sig}(\mathcal{T}_{\mathsf{RCI}})$ we have $r^{\mathcal{J}'} = r^{\mathcal{J}}$ and for all concepts $A \in \mathsf{Undef}(\mathcal{T}_{\mathsf{RCI}})$ we have $A^{\mathcal{J}'} = A^{\mathcal{J}}$ then for all symbols $X \in \Sigma$ we have $X^{\mathcal{J}'} = X^{\mathcal{J}}$ and since $\mathcal{J}|_\Sigma = \mathcal{I}|_\Sigma$ then $\mathcal{J}'|_\Sigma = \mathcal{I}|_\Sigma$ as required.

We now show that $\mathcal{J}'$ is a model of $\mathcal{T}_{\mathsf{RCI}}$. Since $\mathcal{J}$ coincides with $\mathcal{J}$' on $\mathsf{sig}(\mathsf{Lhs}_\Sigma(\mathcal{T}_{\mathsf{RCI}}))$ $\mathcal{J}'$ is a model of $\mathsf{Lhs}_\Sigma(\mathcal{T}_{\mathsf{RCI}})$. Now let $A \bowtie C \in \mathcal{T}_{\mathsf{RCI}} \setminus \mathsf{Lhs}_\Sigma(\mathcal{T}_{\mathsf{RCI}})$. By definition of $\mathsf{Lhs}_\Sigma(\mathcal{T}_{\mathsf{RCI}})$ we have $A \in \mathsf{sig}(\mathcal{T}_{\mathsf{RCI}}) \setminus \mathsf{sig}(\mathsf{Lhs}_\Sigma(\mathcal{T}_{\mathsf{RCI}}))$. We distinguish two cases: First, let $A \bowtie C$ be of the form $A \sqsubseteq C$, by definition of $\mathsf{Lhs}_\Sigma(\mathcal{T}_{\mathsf{RCI}})$ if $A \sqsubseteq C \in \mathcal{T}_{\mathsf{RCI}} \setminus \mathsf{Lhs}_\Sigma(\mathcal{T}_{\mathsf{RCI}})$ then for all other repeated concept-inclusions $A \sqsubseteq D$ we have $A \sqsubseteq D \in \mathcal{T}_{\mathsf{RCI}} \setminus \mathsf{Lhs}_\Sigma(\mathcal{T}_{\mathsf{RCI}})$. Since $A^{\mathcal{J}'} = \emptyset$ then $\mathcal{J}'$ satisfies every, possibly repeated, concept-inclusion. Second, let $A \bowtie C$ be of the form $A \equiv C$, then $A^{\mathcal{J}'} = C^{\mathcal{J}'}$ and so $\mathcal{J}'$ satisfies $A \equiv C$ as required.

□

Lemma 3.3.3. *Given a signature Σ and an acyclic terminology with RCIs $\mathcal{T}_{\mathsf{RCI}}$,* *if* $\mathcal{T}_{\mathsf{RCI}}$ *contains no axiom* Σ*-dependencies, then* $\mathsf{Lhs}_\Sigma(\mathcal{T}_{\mathsf{RCI}})$ *contains no role name from* Σ.

Proof. Let Σ be a signature and $\mathcal{T}_{\mathsf{RCI}}$ an acyclic terminology with RCIs. For a proof by contradiction assume there exists some axiom $B \bowtie D \in \mathsf{Lhs}_\Sigma(\mathcal{T}_{\mathsf{RCI}})$ such that for some $r \in \Sigma$ we have $r \in \mathsf{sig}(D)$. Now since $B \bowtie D \in \mathsf{Lhs}_\Sigma(\mathcal{T}_{\mathsf{RCI}})$, one of the following conditions hold:

1. $B \in \Sigma$. Since $r \in \mathsf{sig}(D)$ we have $r \in \mathsf{depend}_{\mathcal{T}_{\mathsf{RCI}}}(B \bowtie C) \cap \Sigma$, and $\mathcal{T}_{\mathsf{RCI}}$ contains an axiom Σ-dependency contradicting our original assumption.

2. There exists some $X \in \Sigma$ such that $B \in \mathsf{depend}_{\mathcal{T}_{\mathsf{RCI}}}(X)$. Then there must an axiom $X \bowtie D'$ such $B \in \mathsf{depend}_{\mathcal{T}_{\mathsf{RCI}}}(X \bowtie D')$ caused by the axiom $B \bowtie D$ and since $r \in \mathsf{sig}(D)$ we also have $r \in \mathsf{depend}_{\mathcal{T}}(X \bowtie D') \cap \Sigma$ and so $\mathcal{T}_{\mathsf{RCI}}$ contains an axiom dependency, again contradicting the initial assumption.

□

Theorem 3.3.1. *Given an acyclic* $\mathcal{ALCQI}$ *terminology with RCIs* $\mathcal{T}_{\mathsf{RCI}}$*, and signature* Σ*, such that* $\mathcal{T}_{\mathsf{RCI}}$ *contains no axiom* Σ*-dependencies, it is in* Π_2^p *to decide whether* $\mathcal{T}_{\mathsf{RCI}} \equiv_\Sigma \emptyset$.

Proof. By Lemma 3.3.2 $\mathcal{T}_{\mathsf{RCI}} \equiv_\Sigma \emptyset$ iff $\mathsf{Lhs}_\Sigma(\mathcal{T}_{\mathsf{RCI}}) \equiv_\Sigma \emptyset$ but since $\mathcal{T}_{\mathsf{RCI}}$ contains no axiom Σ-dependencies then by Lemma 3.3.3 $\mathsf{Lhs}_\Sigma(\mathcal{T}_{\mathsf{RCI}})$ contains no role name from Σ, and since model inseparability is known to be monotone by Proposition 2.3.3, it follows that $\mathsf{Lhs}_\Sigma(\mathcal{T}_{\mathsf{RCI}}) \equiv_\Sigma \emptyset$ iff $\mathsf{Lhs}_\Sigma(\mathcal{T}_{\mathsf{RCI}}) \equiv_{\Sigma \cap \mathsf{N_C}} \emptyset$ which by Lemma 3.3.1 can be decided in Π_2^p. □

Modified Algorithm

Theorem 3.3.1 suggests a modification of the original $\mathcal{ALCI}$ extraction algorithm (Figure 3.1) which can be applied to acyclic $\mathcal{ALCQI}$ with RCIs to produced what will be referred to as a dependency-free depleting Σ-module.

Input: Acyclic $\mathcal{ALCQI}$ terminology with optional RCIs $\mathcal{T}_{\mathsf{RCI}}$, Signature Σ
Output: Minimal dependency-free depleting Σ-module of $\mathcal{T}_{\mathsf{RCI}}$

1 $\mathcal{M} ::= \emptyset$
2 $\mathcal{W} ::= \emptyset$
3 **while** $\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M} \neq \mathcal{W}$ **do**
4   choose $\alpha \in (\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}) \setminus \mathcal{W}$
5     $\mathcal{W} ::= \mathcal{W} \cup \{\alpha\}$
6   **if** $\mathcal{W}$ *contains an axiom* $(\Sigma \cup \mathsf{sig}(\mathcal{M}))$*-dependency* **or** $\mathcal{W} \not\equiv_{\Sigma \cup \mathsf{sig}(\mathcal{M})} \emptyset$ **then**
7         $\mathcal{M} ::= \mathcal{M} \cup \{\alpha\}$
8         $\mathcal{W} ::= \emptyset$
9   **end**
10 **end**
11 **return** $\mathcal{M}$

Figure 3.3: Extracting minimal dependency-free Σ-modules from acyclic $\mathcal{ALCQI}$ terminologies with RCIs

Definition 3.3.5 (Dependency-free depleting Σ-module)**.** *Let $\mathcal{T}_{\mathsf{RCI}}$ be an acyclic $\mathcal{ALCQI}$ terminology with RCIs, and a signature Σ, then $\mathcal{M} \subseteq \mathcal{T}_{\mathsf{RCI}}$ is a called a dependency-free depleting Σ-module of $\mathcal{T}_{\mathsf{RCI}}$ if $\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}$ contains no axiom $\Sigma \cup \mathsf{sig}(\mathcal{M})$-dependencies* and *$\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M} \equiv_{\Sigma \cup \mathsf{sig}(\mathcal{M})} \emptyset$.*

Figure 3.3 shows the result of this modification, and to produce a dependency-free depleting Σ-module it can be seen that direct $\Sigma \cup \mathsf{sig}(\mathcal{M})$-dependencies are replaced with axiom $\Sigma \cup \mathsf{sig}(\mathcal{M})$-dependencies to ensure $\mathcal{W} \not\equiv_{\Sigma \cup \mathsf{sig}(\mathcal{M})} \emptyset$ for some $\mathcal{W} \subseteq \mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}$ (Line 6) is decidable by Theorem 3.3.1. The algorithm terminates, returning $\mathcal{M}$, when $\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}$ contains no axiom $\Sigma \cup \mathsf{sig}(\mathcal{M})$-dependencies and $\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M} \equiv_{\Sigma \cup \mathsf{sig}(\mathcal{M})} \emptyset$, which ensures $\mathcal{M}$ is a depleting Σ-module. But again an approximation of the unique minimal depleting Σ-module, as like direct Σ-dependencies, axiom Σ-dependencies may still capture axioms irrelevant for $\Sigma \cup \mathsf{sig}(\mathcal{M})$. However, it is also possible to show the module produced in Figure 3.3 is the unique minimal dependency free Σ-module using the following lemma, which is a direct generalisation of a claim used in the proof of Proposition 2.3.2 taken from [KWZ10].

Lemma 3.3.4. *Let $\mathcal{T}_{\mathsf{RCI}}$ be an acyclic $\mathcal{ALCQI}$ terminology with RCIs, Σ a signature, and $\mathcal{M} \subseteq \mathcal{T}_{\mathsf{RCI}}$ be a dependency-free depleting Σ-module of $\mathcal{T}_{\mathsf{RCI}}$. Suppose* *there exists a Σ' such that $\Sigma \subseteq \Sigma' \subseteq (\Sigma \cup \mathsf{sig}(\mathcal{M}))$ and let $\mathcal{W} \subseteq \mathcal{T}_{\mathsf{RCI}}$ be a* minimal *set such that either $\mathcal{W}$ contains an axiom Σ'-dependency or $\mathcal{W} \not\equiv_{\Sigma'} \emptyset$. Then $\mathcal{W} \subseteq \mathcal{M}$.*

Proof. Suppose the lemma does not hold, that $\mathcal{W} \not\subseteq \mathcal{M}$, we show that $\mathcal{W}$ contains no axiom Σ'-dependency and that $\mathcal{W} \equiv_{\Sigma'} \emptyset$, contradicting the assumptions of the lemma. Let $\mathcal{X} = \mathcal{W} \cap \mathcal{M}$ and observe that $\mathcal{X}$ can neither contain an axiom Σ'-dependency nor can we have $\mathcal{X} \not\equiv_{\Sigma'} \emptyset$, as either of which would be contrary to the minimality of $\mathcal{W}$.

First we prove that $\mathcal{W}$ contains no axiom Σ-dependency. To show this we demonstrate that $(\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}) \cup \mathcal{X} \supseteq \mathcal{W}$ contains no axiom Σ'-dependency. If this is not the case then there exists a pair of symbols $\{A, X\} \subseteq \Sigma'$ with an axiom $A \bowtie C \in (\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}) \cup \mathcal{X}$ and where the set $\mathsf{chain}^X_{(\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}) \cup \mathcal{X}}(A \bowtie B)$ is non-empty. Let γ be a dependency chain $\gamma = A_1 \bowtie C_1, A_2 \bowtie C_2, \ldots, A_n \bowtie C_n \in \mathsf{chain}^X_{(\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}) \cup \mathcal{X}}(A \bowtie B)$. Since $(\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}) \cup \mathcal{X}$ is an acyclic terminology with RCIs, each $A_i \bowtie C_i$ appears in $\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}$ or in $\mathcal{X}$. Observe both $\gamma \notin \mathsf{chain}^X_{\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}}(A_1 \bowtie C_1)$ and $\gamma \notin \mathsf{chain}^X_{\mathcal{X}}(A_1 \bowtie C_1)$ as either would cause $\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}$ or $\mathcal{X}$ to contain an axiom Σ'-dependency. As the definition of dependency chains implies each A_{i+1} appears in $\mathsf{sig}(C_i)$, then either there must exist an $A_i \in \mathsf{sig}(\mathcal{X})$ such that $A_i \in \mathsf{depend}_{\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}}(A_1 \bowtie C_1)$ *or* an $A_j \in \mathsf{sig}(\mathcal{X})$ with $X \in \mathsf{depend}_{\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}}(A_j \bowtie C_j)$ *or* an $\{A_i, A_j\} \subseteq \mathsf{sig}(\mathcal{X})$ such that $A_j \in \mathsf{depend}_{\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}}(A_i \bowtie C_i)$ where $1 < i \leq n$ and $i < j$. In each of these cases, as $(\Sigma' \cup \mathsf{sig}(\mathcal{X})) \subseteq (\Sigma \cup \mathsf{sig}(\mathcal{M}))$, $\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}$ contains an axiom $\Sigma \cup \mathsf{sig}(\mathcal{M})$-dependency, contradicting the assumption that $\mathcal{M}$ is a dependency-free depleting Σ-module.

Now we show that $\mathcal{W} \equiv_{\Sigma'} \emptyset$. As $\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M} \equiv_{\Sigma \cup \mathsf{sig}(\mathcal{M})} \emptyset$ by the definition of a dependency free depleting Σ-module and the fact by Proposition 2.3.3 inseparability is robust under replacement we obtain $(\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}) \cup \mathcal{X} \equiv_{\Sigma \cup \mathsf{sig}(\mathcal{M})} \mathcal{X}$. Then using $\Sigma' \subseteq \Sigma \cup \mathsf{sig}(\mathcal{M})$ and the fact the inseparability is both a transitive and monotone relation, we conclude from $X \equiv_{\Sigma'} \emptyset$ that $(\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}) \cup \mathcal{X} \equiv_{\Sigma'} \emptyset$. As $\emptyset \subseteq \mathcal{W} \subseteq (\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}) \cup \mathcal{X}$ we obtain $\mathcal{W} \equiv_{\Sigma'} \emptyset$ as required.

We have now shown that $\mathcal{W}$ does not contain an axiom Σ'-dependency and that $\mathcal{W} \equiv_{\Sigma'} \emptyset$ which contradicts the original assumptions of the lemma. $\square$

Now for the minimality claim. In the proof of the following theorem, and in general, we refer to one execution of the main **while** loop of the Figure 3.3 as an iteration, and denote a particular iteration with an index i, the state of the sets $\mathcal{W}$ and $\mathcal{M}$ on this iteration are denoted $\mathcal{W}_i$ and $\mathcal{M}_i$ respectively.

Theorem 3.3.2. *The algorithm in Figure 3.3 produces the unique minimal dependency-free depleting Σ-module.*

Proof. Analogously to the algorithm for $\mathcal{ALCI}$ (Figure 3.1), we partition our ontology into two sets $\mathcal{W}$ and $\mathcal{M}$, where $\mathcal{W}$ has no axiom $\Sigma \cup \mathsf{sig}(\mathcal{M})$-dependencies and $\mathcal{W} \equiv_{\Sigma \cup \mathsf{sig}(\mathcal{M})} \emptyset$. The algorithm terminates when we have $\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M} = \mathcal{W}$ resulting in an $\mathcal{M}$ which is a depleting module of $\mathcal{T}_{\mathsf{RCI}}$ such that $\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}$ contains no axiom $\Sigma \cup \mathsf{sig}(\mathcal{M})$-dependency, a dependency-free depleting Σ-module by definition.

For the uniqueness and minimality claim, let $\mathcal{M}_0 \subseteq \mathcal{T}_{\mathsf{RCI}}$ be a dependency-free depleting Σ-module of $\mathcal{T}_{\mathsf{RCI}}$. We prove by induction on the iterations of the main *while* loop of the algorithm that the $\mathcal{M}$ produced is contained in $\mathcal{M}_0$.

Base case: On iteration 1 we have $\mathcal{M}_1 = \emptyset \subseteq \mathcal{M}_0$. **Inductive Step:** Assume for all iterations $m < l$ that $\mathcal{M}_m \subseteq \mathcal{M}_0$. Consider iteration l and assume w.l.o.g it is the first iteration such that $\mathcal{W}_{l-1}$ does not have an axiom dependency or that $\mathcal{W}_{l-1} \equiv_{\Sigma \cup \mathsf{sig}(\mathcal{M}_{l-1})} \emptyset$ but by choosing an axiom α (Line 4) and setting $\mathcal{W}_l = \mathcal{W}_{l-1} \cup \{\alpha\}$ that $\mathcal{W}_l$ contains an axiom $\Sigma \cup \mathsf{sig}(\mathcal{M}_l)$-dependency or that $\mathcal{W}_l \not\equiv_{\Sigma \cup \mathsf{sig}(\mathcal{M}_l)} \emptyset$. In either case we obtain $\alpha \in \mathcal{M}_l$. Line 7.

Now to prove that $\mathcal{M}_l \subseteq \mathcal{M}_0$ we must also show that $\alpha \in \mathcal{W}_0$. To this end again consider $\mathcal{W}_l$ and notice there must exist a minimal $\mathcal{W}_0$ with $\mathcal{W}_0 \subseteq \mathcal{W}_l$ for which we have $\alpha \in \mathcal{W}_0$. Since by the induction hypothesis $\mathsf{sig}(\mathcal{M}_l) \subseteq \mathsf{sig}(\mathcal{M}_0)$ we can conclude $\Sigma \cup \mathsf{sig}(\mathcal{M}_l) \subseteq \Sigma \cup \mathsf{sig}(\mathcal{M}_0)$ and can now apply Lemma 3.3.4 (with $\mathcal{W} = \mathcal{W}_0$ and $\mathcal{M} = \mathcal{M}_0$) and conclude $\mathcal{W}_0 \subseteq \mathcal{M}_0$ which implies $\alpha \in \mathcal{M}_0$ as required. $\square$

As an example, consider an application of the extraction algorithm in Figure 3.3 to the following acyclic terminology with RCIs:

Example 3.3.4. *Let $\mathcal{T}_{\mathsf{RCI}} = \alpha_1 - \alpha_5$ be the following acyclic $\mathcal{ALCQ}$ terminology with RCIs, and let $\Sigma = \{\mathsf{Animal\_Group}, \mathsf{Lion}\}$.*

$$\mathsf{Animal\_Group} \sqsubseteq (\geq 2\ \mathsf{has.Animal}) \quad (\alpha_1)$$

$$\mathsf{Lion} \sqsubseteq \mathsf{Mammal} \quad (\alpha_2)$$

$$\mathsf{Lion} \sqsubseteq \mathsf{Cat} \quad (\alpha_3)$$

$$\mathsf{Mammal} \sqsubseteq \mathsf{Animal} \quad (\alpha_4)$$

$$\mathsf{Mammal} \sqsubseteq \forall \mathsf{has.WarmBlood} \quad (\alpha_5)$$

Assuming the algorithm chooses the axioms (Line 4) in the given order, it can be verified that $\mathcal{W}$ contains no axiom $\Sigma \cup \mathsf{sig}(\mathcal{M})$-dependencies and that $\mathcal{W} \equiv_{\Sigma\cup\mathsf{sig}(\mathcal{M})} \emptyset$ until $\mathcal{W} = \{\alpha_1\}$ when we find, $\mathcal{W}$ contains no axiom $\Sigma \cup \mathsf{sig}(\mathcal{M})$-dependency but $\mathcal{W} \not\equiv_{\Sigma\cup\mathsf{sig}(\mathcal{M})} \emptyset$, and we set $\mathcal{M} = \mathcal{M} \cup \{\alpha_1\}$ and then $\mathcal{W}$ is reset (Line 8).

$\Sigma \cup \mathsf{sig}(\mathcal{M}) = \{\mathsf{Animal\_Group}, \mathsf{Lion}, \mathsf{has}, \mathsf{Animal}\}$ and we take action next when $\mathcal{W} = \{\alpha_2, \alpha_3, \alpha_4\}$ where we find $\mathcal{W}$ contains an axiom $\Sigma \cup \mathsf{sig}(\mathcal{M})$-dependency, as $\mathsf{Lion} \in \Sigma \cup \mathsf{sig}(\mathcal{M})$ and $\mathsf{Animal} \in \mathsf{depend}_{\mathcal{W}}(\alpha_2) \cap (\Sigma \cup \mathsf{sig}(\mathcal{M}))$ resulting in $\mathcal{M} = \mathcal{M} \cup \{\alpha_4\}$, and $\mathcal{W}$ is again reset.

$\Sigma \cup \mathsf{sig}(\mathcal{M}) = \{\mathsf{Animal\_Group}, \mathsf{Lion}, \mathsf{has}, \mathsf{Animal}, \mathsf{Mammal}\}$, we next find an axiom $\Sigma \cup \mathsf{sig}(\mathcal{M})$-dependency when $\mathcal{W} = \{\alpha_2\}$ as $\mathsf{Lion} \in \Sigma \cup \mathsf{sig}(\mathcal{M})$ and $\mathsf{Mammal} \in \mathsf{depend}_{\mathcal{W}}(\alpha_2) \cap (\Sigma \cup \mathsf{sig}(\mathcal{M}))$ and set $\mathcal{M} = \mathcal{M} \cup \{\alpha_2\}$, $\mathcal{W}$ is reset, $\Sigma \cup \mathsf{sig}(\mathcal{M})$ is unchanged.

Finally when $\mathcal{W} = \{\alpha_3, \alpha_5\}$, $\mathcal{W}$ contains an axiom $\Sigma \cup \mathsf{sig}(\mathcal{M})$-dependency as $\mathsf{Mammal} \in \Sigma \cup \mathsf{sig}(\mathcal{M})$ and $\mathsf{has} \in \mathsf{depend}_{\mathcal{W}}(\alpha_5) \cap (\Sigma \cup \mathsf{sig}(\mathcal{M}))$ and we set $\mathcal{M} = \mathcal{M} \cup \{\alpha_5\}$. At this point it can be verified that there there is no $\mathcal{W} \subseteq \mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}$ that contains a axiom $\Sigma \cup \mathsf{sig}(\mathcal{M})$-dependency or that $\mathcal{W} \not\equiv_{\Sigma\cup\mathsf{sig}(\mathcal{M})} \emptyset$, so the algorithm terminates with $\mathcal{M} = \{\alpha_1, \alpha_2, \alpha_4, \alpha_5\}$.

There are a couple of interesting observations about the module produced in

Example 3.3.4. Firstly, the module does not contain every RCI for the repeated name Lion (α_3 does not belong to $\mathcal{M}$). This is achieved by having dependencies work on a per-axiom basis, which is beneficial for keeping the module as small as possible, especially when a repeated name corresponds to several RCIs.

Secondly, the RCI α_5 is captured by an axiom Σ-dependency but is not contained in the minimal depleting module for Σ. The minimal is in fact $\mathcal{M}' = \{\alpha_1, \alpha_2, \alpha_4\}$, the module produced is an approximation. To see this, let $\mathcal{I}$ be an arbitrary interpretation, to construct a model $\mathcal{J}$ of $\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}'$, first set $X^{\mathcal{J}} = X^{\mathcal{I}}$ for all symbols $X \in \{\mathsf{Lion}, \mathsf{Mammal}, \mathsf{has}\}$ to ensure $\mathcal{I}|_{\Sigma \cup \mathsf{sig}(\mathcal{M}')} = \mathcal{J}|_{\Sigma \cup \mathsf{sig}(\mathcal{M}')}$, finally set $\mathsf{Cat}^{\mathcal{J}} = \mathsf{Lion}^{\mathcal{J}}$ and $\mathsf{WarmBlood}^{\mathcal{J}} = \{e \mid (d, e) \in \mathsf{has}^{\mathcal{J}}\}$, then one can verify $\mathcal{J} \models \mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}'$ and so $\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}' \equiv_{\Sigma \cup \mathsf{sig}(\mathcal{M}')} \emptyset$, $\mathcal{M}'$ is a depleting module for $\mathcal{T}_{\mathsf{RCI}}$ for Σ with $\mathcal{M}' \subseteq \mathcal{M}$.

3.4. Improving practical performance

As module extraction has many practical applications, it is certainly desirable to have an algorithm that performs well practically, especially as we are interested in providing an implementation for comparative purposes.

The algorithms for both original approximation for acyclic $\mathcal{ALCI}$, and our newly proposed modification stem naturally from their associated theorems, ensuring a depleting module is produced from the supplied ontology and signature. However, by exploring the theory supporting the algorithms, it is possible to identify several details that may be modified to provide a computational saving.

In the sections that follow we will work towards developing a new algorithm which produces an identical module but offers measurably better practical performance. We achieve this using the module extraction algorithm for acyclic $\mathcal{ALCQI}$ with RCIs (Figure 3.3) as our starting point, which we refer to, for ease of reference, as the *iterative extraction algorithm*.

3.4.1 Detecting axiom dependencies

The "black box" nature of the iterative algorithm, each iteration incrementally building a subset $\mathcal{W} \subseteq \mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}$ one axiom at a time, has the limitation that the detection of an axiom $\Sigma \cup \mathsf{sig}(\mathcal{M})$-dependency in $\mathcal{W}$ results in only a single axiom being added to the module $\mathcal{M}$. We observe a relation between axioms, allowing us to identify some cases where more than one axiom may be added to $\mathcal{M}$ on a single iteration of the algorithm.

Additionally, to determine if an axiom $\Sigma \cup \mathsf{sig}(\mathcal{M})$-dependency is present on a particular iteration, we must compute the set $\mathsf{depend}_{\mathcal{W}}$ for the current $\mathcal{W}$, as it may change each time an axiom is added. We show it is possible to avoid this re-computation entirely, the set $\mathsf{depend}_{\mathcal{T}_{\mathsf{RCI}}}$ can be used for deciding if a dependency is present for any $\mathcal{W} \subseteq \mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}$, and only needs to computed once.

Axiom Chains

Definition 3.4.1 (Axiom Chains). *An axiom chain is an ordered set $A_1 \bowtie C_1, A_2 \bowtie C_2, \ldots, A_n \bowtie C_n$ with each $A_{i+1} \in \mathsf{sig}(C_i)$ for $0 < i \leq n$. The "head" of the chain is the first axiom in the chain, i.e. $A_1 \bowtie C_1$. The length of a chain γ is defined number of axioms in the chain.*

Definition 3.4.2 (Dependency Chain). *For an acyclic terminology with RCIs $\mathcal{T}_{\mathsf{RCI}}$, let $\mathsf{chain}^{X}_{\mathcal{T}_{\mathsf{RCI}}}(A_1 \bowtie C_1)$ be the set of all axiom chains $\gamma = A_1 \bowtie C_1, A_2 \bowtie C_2 \ldots A_n \bowtie C_n$ with $X \in \mathsf{sig}(C_n)$ and $\gamma \subseteq \mathcal{T}_{\mathsf{RCI}}$.*

A dependency chain represents the sequence of axioms which induces a particular symbol in $\mathsf{depend}_{\mathcal{T}_{\mathsf{RCI}}}$, formalised by the following lemma:

Lemma 3.4.1. *Given an acyclic terminology with RCIs $\mathcal{T}_{\mathsf{RCI}}$, a symbol $D \in \mathsf{depend}_{\mathcal{T}_{\mathsf{RCI}}}(A \bowtie C)$ iff there exists a dependency chain $\gamma \in \mathsf{chain}^{D}_{\mathcal{T}_{\mathsf{RCI}}}(A \bowtie C)$.*

Proof. First consider the set of dependencies $\mathsf{depend}_{\gamma}(A \bowtie C)$ induced by a chain $\gamma \in \mathsf{chain}^{D}_{\mathcal{T}_{\mathsf{RCI}}}(A \bowtie C)$, clearly $D \in \mathsf{depend}_{\gamma}(A \bowtie C)$ and since $\gamma \subseteq \mathcal{T}_{\mathsf{RCI}}$ it follows that $\mathsf{depend}_\gamma(A \bowtie C) \subseteq \mathsf{depend}_{\mathcal{T}_{\mathsf{RCI}}}(A \bowtie D)$ and so $D \in \mathsf{depend}_{\mathcal{T}_{\mathsf{RCI}}}(A \bowtie C)$.

Now assume the exists an arbitrary $D \in \mathsf{depend}_{\mathcal{T}_{\mathsf{RCI}}}(A \bowtie C)$. By definition there exists some $X_1 \in (\mathsf{sig}(C) \cap \mathsf{N_C})$ such that $D \in \mathsf{depend}_{\mathcal{T}}(X_1)$ and so there exists a sequence of depends relations $X_1 \prec_{\mathcal{T}_{\mathsf{RCI}}} X_2 \prec_{\mathcal{T}_{\mathsf{RCI}}} \ldots \prec_{\mathcal{T}_{\mathsf{RCI}}} X_m \prec_{\mathcal{T}_{\mathsf{RCI}}} D$ which must be induced by sequence of axioms $\gamma = A \bowtie C, X_1 \bowtie C_1, \ldots, X_m \bowtie C_m$ with $\gamma \subseteq \mathcal{T}_{\mathsf{RCI}}$ and $X_1 \in \mathsf{sig}(C)$ and both $X_{i+1} \in C_i$ for $0 < i \leq m$ and $D \in \mathsf{sig}(C_m)$, γ conforms to the definition of a chain $\gamma \in \mathsf{chain}^D_{\mathcal{T}_{\mathsf{RCI}}}(A \bowtie C)$. Since our choice of symbol D was arbitrary such a dependency chain always exists. □

Chains as part of a module

We now look at the relationship between dependency chains and the depleting module produced for a signature Σ when the iterative algorithm is applied to an acyclic terminology with RCIs.

Theorem 3.4.1. *Let $\mathcal{T}_{\mathsf{RCI}}$ be an acyclic TBox with RCIs and Σ a signature. Let $\mathcal{M}_{min} \subseteq \mathcal{T}_{\mathsf{RCI}}$ be the minimal dependency-free depleting Σ-module of $\mathcal{T}_{\mathsf{RCI}}$. Suppose that for some run of the algorithm in Figure 3.3 at some iteration i, for some axiom chain $\gamma \in \mathsf{chain}^X_{\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}_i}(A_1 \bowtie C_1)$ we have $\{A_1, X\} \subseteq \Sigma \cup \mathsf{sig}(\mathcal{M}_i)$. Then $\gamma \subseteq \mathcal{M}_{min}$.*

Proof. Let i be an iteration and γ be a chain $\gamma \in \mathsf{chain}^X_{\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}_i}(A_1 \bowtie C_1)$, and assume that $\{A_1, X\} \subseteq \Sigma \cup \mathsf{sig}(\mathcal{M}_i)$. We prove the theorem by induction on the length of γ.

Base case: For a chain length 1, the chain γ consists of the single axiom $A_1 \bowtie C_1$. Consider a run of the algorithm in Figure 3.3 which selects $\alpha = A_1 \bowtie C_1$ in the next iteration and sets $\mathcal{W} = \{\alpha\}$. Since $\gamma \in \mathsf{chain}^X_{\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}_i}(A_1 \bowtie C_1)$ we have $X \in \mathsf{sig}(C_1)$, and by the original assumption we have $\{A_1, X\} \subseteq \Sigma \cup \mathsf{sig}(\mathcal{M}_n)$, so clearly $A_1 \bowtie C_1$ causes an axiom $\Sigma \cup \mathsf{sig}(\mathcal{M}_n)$-dependency in $\mathcal{W}$, and set $\mathcal{M}_{i+1} = \mathcal{M}_i \cup \{\alpha\}$ (Line 7). As the algorithm in Figure 3.3 always outputs $\mathcal{M}_{min}$ regardless of the choice of α, $\gamma \subseteq \mathcal{M}_{min}$.

Inductive step:

Assume for all runs of the algorithm and all iterations i, for all chains γ of length $m < l$, with $\gamma \in \mathsf{chain}^{X}_{\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}_i}(A_1 \bowtie C_1)$ and $\{A, X\} \subseteq \Sigma \cup \mathsf{sig}(\mathcal{M}_i)$ that $\gamma \subseteq \mathcal{M}_{min}$.

Let the length of γ be l, so $\gamma = A_1 \bowtie C_1, A_2 \bowtie C_2, \ldots, A_l \bowtie C_l$.

Let $j : 1 \leq j \leq l$ be the smallest number such that for $\mathcal{W} = \{A_1 \bowtie C_1, A_2 \bowtie C_2, \ldots, A_j \bowtie C_j\}$ either $\mathcal{W}$ contains an axiom $(\Sigma \cup \mathsf{sig}(\mathcal{M}_i))$-dependency or $\mathcal{W} \not\equiv_{\Sigma \cup \mathsf{sig}(\mathcal{M}_i)} \emptyset$, notice that such a j always exists as the set γ contains an axiom $\Sigma \cup \mathsf{sig}(\mathcal{M}_i)$ dependency itself by the original assumption. As a result, there exists an iteration i of the algorithm in Figure 3.3, which adds $A_j \bowtie C_j$ to $\mathcal{M}_i$.

Now consider the state of the algorithm on iteration $i + 1$ and notice the remaining axioms of γ' in $\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}_i$ consist of two chains $\gamma_1 = A_1 \bowtie C_1, \ldots, A_{j-1} \bowtie C_{j-1}$ and $\gamma_2 = A_{j+1} \bowtie C_{j+1}, A_2 \bowtie C_2, \ldots, A_l \bowtie C_l$ which both satisfy the conditions of the theorem. To see this, distinguish three cases based on the position of the axiom $A_j \bowtie C_j$ within γ which was added to $\mathcal{M}_i$:

1. $A_j \bowtie C_j = A_1 \bowtie C_1$.

   γ_1 is empty. For γ_2, by the original assumption we have $X \in \mathsf{sig}(C_n)$ so $\gamma_2 \in \mathsf{chain}^{X}_{\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}_i}(A_2 \bowtie C_2)$, also by the original assumption $X \in \Sigma \cup \mathsf{sig}(\mathcal{M}_i)$ and since $A_1 \bowtie C_1 \in \mathcal{M}_i$ with $A_2 \in \mathsf{sig}(C_1)$ we have $\{A_2, X\} \subseteq \Sigma \cup \mathsf{sig}(\mathcal{M}_i)$.

2. $A_j \bowtie C_j = A_k \bowtie C_k$ for $1 < k < l$.

   - γ_1 : Since $A_k \in \mathsf{sig}(C_{k-1})$ we have $\gamma_1 \in \mathsf{chain}^{A_k}_{\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}_i}(A_1 \bowtie C_1)$, and since $A_k \bowtie C_k \in \mathcal{M}_i$, and by the original assumption $A_1 \in \Sigma \cup \mathsf{sig}(\mathcal{M}_i)$, we have $\{A_1, A_k\} \subseteq \Sigma \cup \mathsf{sig}(\mathcal{M}_i)$.

- γ_2 : By the original assumption we have $X \in \mathsf{sig}(C_n)$ so $\gamma_2 \in \mathsf{chain}^X_{\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}_i}(A_{k+1} \bowtie C_{k+1})$, and since $A_k \bowtie C_k \in \mathcal{M}_i$ and $A_{k+1} \in \mathsf{sig}(C_k)$ we have $A_k \in \Sigma \cup \mathsf{sig}(\mathcal{M}_i)$ and by the original assumption $X \in \Sigma \cup \mathsf{sig}(\mathcal{M}_i)$ we have $\{A_k, X\} \subseteq \Sigma \cup \mathsf{sig}(\mathcal{M}_i)$

3. $A_j \bowtie C_j = A_n \bowtie C_n$

   γ_2 is empty. For γ_1, since $A_n \bowtie C_n \in \mathcal{M}_i$ with $A_n \in \mathsf{sig}(C_{i-1})$, we have both $\gamma_1 \in \mathsf{chain}^{A_n}_{\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}_i}(A_1 \bowtie C_1)$ and $A_n \in \Sigma \cup \mathsf{sig}(\mathcal{M}_i)$, also by the original assumption $A_1 \in \Sigma \cup \mathsf{sig}(\mathcal{M}_i)$ so $\{A_1, A_n\} \in \Sigma \cup \mathsf{sig}(\mathcal{M}_i)$

Since both γ_1 and γ_2 are less than l in length, then by the induction hypothesis we have $\gamma_l \subseteq \mathcal{M}_{min}$, for $l = 1, 2$, so $\gamma \subseteq \mathcal{M}_{min}$.

□

Clearly Theorem 3.4.1 is only applicable if on some iteration i, we have an axiom $\Sigma \cup \mathsf{sig}(\mathcal{M}_i)$-dependency in $\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}_i$. As a consequence of this theorem, locating an axiom $\Sigma \cup \mathsf{sig}(\mathcal{M})$-dependency on some iteration, means we are able to identify possibly several dependency chains which may be immediately added to the module, as we can be assured they will be added on some future iteration. We will utilise property in producing a more efficient module extraction algorithm in Section 3.4.3.

Detecting dependencies in $\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}$

Lemma 3.4.2. *Let $\mathcal{T}_{\mathsf{RCI}}$ and $\mathcal{M} \subseteq \mathcal{T}_{\mathsf{RCI}}$ be acyclic terminologies with RCIs, and Σ a signature, and let there exist an axiom $A_1 \bowtie C_1 \in \mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}$, then: $A_1 \bowtie C_1$ causes an axiom $\Sigma \cup \mathsf{sig}(\mathcal{M})$-dependency in $\mathcal{T}_{\mathsf{RCI}}$ if and only if $A_1 \bowtie C_1$ causes an axiom $\Sigma \cup \mathsf{sig}(\mathcal{M})$-dependency in $\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}$.*

Proof. $\Leftarrow$ The proof in this direction is trivial. Since $\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M} \subseteq \mathcal{T}_{\mathsf{RCI}}$ the set $\mathsf{depend}_{\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}}(A_1 \bowtie C_1) \subseteq \mathsf{depend}_{\mathcal{T}_{\mathsf{RCI}}}(A_1 \bowtie C_1)$ for all axioms $A_1 \bowtie C_1 \in (\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M})$ hence, an axiom causing a dependency in $\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}$ implies it causes a dependency in $\mathcal{T}_{\mathsf{RCI}}$.

$\Rightarrow$ Assume $A_1 \bowtie C_1$ causes an axiom $\Sigma \cup \mathsf{sig}(\mathcal{M})$-dependency in $\mathcal{T}_{\mathsf{RCI}}$, by definition $A \in \Sigma \cup \mathsf{sig}(\mathcal{M})$ and some symbol $X \in \mathsf{depend}_{\mathcal{T}_{\mathsf{RCI}}}(A_1 \bowtie C_1) \cap (\Sigma \cup \mathsf{sig}(\mathcal{M}))$, then by Lemma 3.4.1 there exists a chain of axioms $\gamma = A_1 \bowtie C_1, A_2 \bowtie C_2, \ldots, A_n \bowtie C_n \in \mathsf{chain}^X_{\mathcal{T}_{\mathsf{RCI}}}(A)$. By the original assumption we know $A_1 \bowtie C_1 \in \mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}$ but for the other axioms in the chain γ this may not still be the case, so we distinguish two cases:

1. $\gamma \subseteq \mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}$. We have $A_1 \in \Sigma \cup \mathsf{sig}(\mathcal{M})$ by the initial assumption, and since $\gamma \in \mathsf{chain}^X_{\mathcal{T}_{\mathsf{RCI}}}(A)$ by Lemma 3.4.1 it still holds that $X \in \mathsf{depend}_{\mathcal{T}_{\mathsf{RCI}}}(A_1 \bowtie C_1) \cap (\Sigma \cup \mathsf{sig}(\mathcal{M}))$, therefore $A_1 \bowtie C_1$ causes a $\Sigma \cup \mathsf{sig}(\mathcal{M})$-dependency in $\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}$.

2. $\gamma \not\subseteq \mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}$. The chain must now stop at some axiom $A_j \bowtie C_j$, for $1 < j < n$, which implies $A_{j+1} \bowtie C_{j+1} \in \mathcal{M}$, so $A_{j+1} \in \Sigma \cup \mathsf{sig}(\mathcal{M})$. From the definition of a axiom chain we have $A_{j+1} \in \mathsf{sig}(C_j)$ and since $A_1 \in \Sigma \cup \mathsf{sig}(\mathcal{M})$ by the initial assumption, and $A_{j+1} \in \mathsf{depend}_{\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}}(A_1 \bowtie C_1) \cap (\Sigma \cup \mathsf{sig}(\mathcal{M}))$, $A_1 \bowtie C_1$ causes an axiom $\Sigma \cup \mathsf{sig}(\mathcal{M})$-dependency in $\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}$.

$\square$

The immediate consequence of Lemma 3.4.2 is that to determine if some axiom $A \bowtie C$ causes a $\Sigma \cup \mathsf{sig}(\mathcal{M})$-dependency in $\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}$, instead of building up a set $\mathcal{W} \subseteq \mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}$ incrementally – as in the iterative algorithm – we can simply check if $A \bowtie C$ causes one in $\mathcal{T}_{\mathsf{RCI}}$. This can be achieved utilising $\mathsf{depend}_{\mathcal{T}_{\mathsf{RCI}}}$ which unlike $\mathsf{depend}_{\mathcal{W}}$, which may change on each iteration, is fixed and only needs to be computed once, and can be retained for the purpose of checking the presence of any axiom $\Sigma \cup \mathsf{sig}(\mathcal{M})$-dependency. The benefit of this is a computational saving over every iteration of the algorithm.

3.4.2 Deciding inseparability

Recall that in the iterative algorithm, we must establish for some $\mathcal{W} \subseteq (\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M})$ if $\mathcal{W} \not\equiv_{\Sigma \cup \mathsf{sig}(\mathcal{M})} \emptyset$, which is decidable when $\mathcal{W}$ contains no axiom $\Sigma \cup \mathsf{sig}(\mathcal{M})$-dependencies by Theorem 3.3.1. If we find that $\mathcal{W} \not\equiv_{\Sigma \cup \mathsf{sig}(\mathcal{M})} \emptyset$ we identify the last axiom α added to $\mathcal{W}$ as "separability causing" and set $\mathcal{M} = \mathcal{M} \cup \{\alpha\}$. This ensures when the algorithm terminates, $\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M} \equiv_{\Sigma \cup \mathsf{sig}(\mathcal{M})} \emptyset$, $\mathcal{M}$ being a depleting module.

Locating Separability Causing Axioms

Instead of building up subsets $\mathcal{W} \subseteq \mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}$ one axiom at a time until a separability causing axiom is located, we take a more goal-orientated approach. First we establish formally exactly what a separability causing axiom is:

Definition 3.4.3 (Separability Causing Axiom). *We call an axiom $A \bowtie C \in \mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}$ "separability causing" if there exists a $\mathcal{W} \subseteq \mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}$ such that:*

- $A \bowtie C \in \mathcal{W}$
- $\mathcal{W} \setminus \{A \bowtie C\} \equiv_{\Sigma \cup \mathsf{sig}(\mathcal{M})} \emptyset$
- $\mathcal{W} \not\equiv_{\Sigma \cup \mathsf{sig}(\mathcal{M})} \emptyset$

Theorem 3.4.2. *$\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M} \not\equiv_{\Sigma \cup \mathsf{sig}(\mathcal{M})} \emptyset$ iff $\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}$ contains a separability causing axiom.*

Proof. First assume $\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M} \not\equiv_{\Sigma \cup \mathsf{sig}(\mathcal{M})} \emptyset$, then there must exist a non-empty $\mathcal{W} \subseteq \mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}$ such that $\mathcal{W} \not\equiv_{\Sigma \cup \mathsf{sig}(\mathcal{M})} \emptyset$. Now let $\mathcal{W}' \subseteq \mathcal{W}$ be the smallest non-empty subset such that $\mathcal{W}' \not\equiv_{\Sigma \cup \mathsf{sig}(\mathcal{M})} \emptyset$, and choose some $\alpha \in \mathcal{W}'$, but since $\mathcal{W}'$ is the smallest of its kind then $\mathcal{W}' \setminus \{\alpha\} \equiv_{\Sigma \cup \mathsf{sig}(\mathcal{M})} \emptyset$, α is a separability causing axiom by definition.

Now assume $\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}$ contains a separability causing axiom, then there must exist some $\mathcal{W} \subseteq \mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}$ such that $\mathcal{W} \not\equiv_{\Sigma \cup \mathsf{sig}(\mathcal{M})} \emptyset$, and then clearly $\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M} \not\equiv_{\Sigma \cup \mathsf{sig}(\mathcal{M})} \emptyset$.

$\square$

Input: Acyclic terminology with optional RCIs $\mathcal{T}_{\mathsf{RCI}}$, Module $\mathcal{M}$ (possibly empty) and Signature Σ such that $\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}$ contains no axiom $\Sigma \cup \mathsf{sig}(\mathcal{M})$-dependencies and $\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M} \not\equiv_{\Sigma \cup \mathsf{sig}(\mathcal{M})} \emptyset$
Output: Separability causing axiom α

1 $\mathcal{W} := \mathsf{lastAdded} := top\_half(\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M})$
2 $\mathsf{lastRemoved} := bottom\_half(\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M})$
3 **while** $\mathsf{lastAdded} \neq \emptyset$ **do**
4   **if** $\mathsf{Lhs}_{\Sigma}(\mathcal{W}) \equiv_{\Sigma} \emptyset$ **then**
5         $\mathsf{lastAdded} := top\_half(\mathsf{lastRemoved})$
6         $\mathcal{W} := \mathcal{W} \cup \mathsf{lastAdded}$
7         $\mathsf{lastRemoved} := \mathsf{lastRemoved} \setminus \mathsf{lastAdded}$
8   **else**
9         $\mathsf{lastRemoved} := bottom\_half(\mathsf{lastAdded})$
10         $\mathcal{W} := \mathcal{W} \setminus \mathsf{lastRemoved}$
11         $\mathsf{lastAdded} := \mathsf{lastAdded} \setminus \mathsf{lastRemoved}$
12   **end**
13 **end**
14 **return** *the last axiom of* $\mathcal{W}$
15 **function** *top_half(W)* := **return** $\{\mathcal{W}[0], \ldots, \mathcal{W}[\lfloor \frac{|\mathcal{W}|}{2} \rfloor - 1]\}$
16 **function** *bottom_half(W)* := **return** $\{\mathcal{W}[\lfloor \frac{|\mathcal{W}|}{2} \rfloor], \ldots, \mathcal{W}[|\mathcal{W}| - 1]\}$

Figure 3.4: Locating a separability causing axiom

Given an acyclic $\mathcal{ALCQI}$ terminology with RCIs such that $\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M} \not\equiv_{\Sigma \cup \mathsf{sig}(\mathcal{M})} \emptyset$, we can use the algorithm described in Figure 3.4 to locate a separability causing axiom. The algorithm works much like a binary search, treating $\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}$ as an ordered set. First considering $\mathcal{W}$ equal to the top half of $\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}$, we check if $\mathcal{W} \equiv_{\Sigma \cup \mathsf{sig}(\mathcal{M})} \emptyset$, if this is indeed the case, we grow $\mathcal{W}$ from the bottom, if not we half it again, each time verifying if $\mathcal{W} \equiv_{\Sigma \cup \mathsf{sig}(\mathcal{M})} \emptyset$. This process is repeated until we have $\mathcal{W}$ such that $\mathcal{W} \not\equiv_{\Sigma \cup \mathsf{sig}(\mathcal{M})} \emptyset$ where by shrinking $\mathcal{W}$ by a single axiom would result in $\mathcal{W} \equiv_{\Sigma \cup \mathsf{sig}(\mathcal{M})} \emptyset$, the last axiom in $\mathcal{W}$ is therefore separability causing.

Another improvement this algorithm offers is the utilisation of $\mathsf{Lhs}_{\Sigma \cup \mathsf{sig}(\mathcal{M})}(\mathcal{W})$ for deciding inseparability (Line 4), which can be used as an equivalent for $\mathcal{W}$ as a result of Lemma 3.3.2. $\mathsf{Lhs}_{\Sigma}(\mathcal{W})$ may contain significantly fewer axioms than the whole of $\mathcal{W}$, especially for a large $\mathcal{W}$, a smaller input offering a potential reduction in the amount of computation necessary to decide if $\mathcal{W} \equiv_{\Sigma\cup\mathsf{sig}(\mathcal{M})} \emptyset$.

Locating separability causing axioms using the algorithm in Figure 3.4 has a clear benefit over the iterative algorithm. In the worst case, the iterative algorithm requires $|\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}|$ inseparability checks to find a single separability causing axiom, in contrast the binary search algorithm in needs just $log_2(|\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}|)$.

3.4.3 Introducing AMEX

Based on our observations about the enhancements in the methods for detecting axiom dependencies and deciding inseparability, we propose a new algorithm, called AMEX, for the extraction of dependency-free depleting Σ-modules from acyclic $\mathcal{ALCQI}$ ontologies with RCIs.

Input: Acyclic $\mathcal{ALCQI}$ terminology with RCIs $\mathcal{T}_{\mathsf{RCI}}$, signature Σ
Initialise: $\mathcal{M} = \emptyset$.
Apply rules **R1** and **R2** exhaustively, preferring **R1**.
Output: $\mathcal{M}$ the unique minimal dependency-free Σ-module of $\mathcal{T}_{\mathsf{RCI}}$

(R1) If an axiom $A \bowtie C \in \mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}$ causes an axiom $\Sigma \cup \mathsf{sig}(\mathcal{M})$-dependency in $\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}$, then for each $X \in \mathsf{depend}_{\mathcal{T}_{\mathsf{RCI}}}(A \bowtie C) \cap \Sigma \cup \mathsf{sig}(\mathcal{M})$ set $\mathcal{M} := \mathcal{M} \bigcup_{\gamma \in \mathsf{chain}^X_{\mathcal{T}_{\mathsf{RCI}}}(A \bowtie C)} \gamma$

(R2) If $\mathsf{Lhs}_{\Sigma\cup\mathsf{sig}(\mathcal{M})}(\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}) \not\equiv_{\Sigma\cup\mathsf{sig}(\mathcal{M})} \emptyset$, then locate the first separability causing axiom $A \bowtie C \in \mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}$ and set $\mathcal{M} := \mathcal{M} \cup \{A \bowtie C\}$

Figure 3.5: AMEX module extraction algorithm

Figure 3.5 shows the proposed new algorithm, the extraction of depleting modules broken into two rules, moving away from the "black box" nature of the iterative algorithm. The first rule **R1** represents the syntactic detection of axiom $\Sigma \cup \mathsf{sig}(\mathcal{M})$-dependencies and the second **R2** the semantic condition to ensure $\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M} \equiv_{\Sigma\cup\mathsf{sig}(\mathcal{M})} \emptyset$.

Theorem 3.4.3. *Given an acyclic $\mathcal{ALCQI}$ terminology with RCIs $\mathcal{T}_{\mathsf{RCI}}$ and signature Σ, the algorithm in Figure 3.5 produces a dependency-free depleting Σ-module of $\mathcal{T}_{\mathsf{RCI}}$.*

Proof. By the definition of a dependency-free depleting Σ-module, we must show that $\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}$ contains no axiom $\Sigma \cup \mathsf{sig}(\mathcal{M})$-dependency and that $\mathcal{M}$ is a depleting Σ-module i.e. $\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M} \equiv_{\Sigma \cup \mathsf{sig}(\mathcal{M})} \emptyset$.

If **R1** is applied then some axiom $A \bowtie C \in \mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}$ causes a axiom $\Sigma \cup \mathsf{sig}(\mathcal{M})$-dependency for the current $\Sigma \cup \mathsf{sig}(\mathcal{M})$ and we add the corresponding dependency chains to $\mathcal{M}$. It is sufficient to add every $\gamma \in \mathsf{chain}^{X}_{\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}}(A \bowtie C)$ for every $X \in \mathsf{depend}_{\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}}(A \bowtie C) \cap \Sigma \cup \mathsf{sig}(\mathcal{M})$ to $\mathcal{M}$ to ensure $A \bowtie C$ causes no axiom $\Sigma \cup \mathsf{sig}(\mathcal{M})$-dependency in $\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}$, as by Lemma 3.4.1 if $\mathsf{chain}^{X}_{\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}}(A \bowtie C) = \emptyset$ then $X \notin \mathsf{depend}_{\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}}(A \bowtie C)$, so $A \bowtie C$ no longer causes an axiom dependency in $\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}$.

Now if **R2** is applicable then $\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}$ contains no axiom $\Sigma \cup \mathsf{sig}(\mathcal{M})$-dependency, and it is decidable to verify if $\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M} \equiv_{\Sigma \cup \mathsf{sig}(\mathcal{M})} \emptyset$ by Theorem 3.3.1. If we find $\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M} \not\equiv_{\Sigma \cup \mathsf{sig}(\mathcal{M})} \emptyset$ then $\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}$ must contain at least one separability causing axiom by Theorem 3.4.2, one of which can be, one of which can be located and added to $\mathcal{M}$ using the algorithm in Figure 3.4.

Since both rules are applied exhaustively, when **R1** is not applicable then $\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}$ contains no axiom $\Sigma \cup \mathsf{sig}(\mathcal{M})$-dependency, and when **R2** is not applicable $\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}$ contains no separability causing axiom, and then $\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M} \equiv_{\Sigma \cup \mathsf{sig}(\mathcal{M})} \emptyset$ by Theorem 3.4.2, and therefore $\mathcal{M}$ is a dependency-free depleting Σ-module as required. □

Theorem 3.4.4. *Let $\mathcal{T}_{\mathsf{RCI}}$ be an acyclic terminology with RCIs and Σ a signature. Then the AMEX algorithm in Figure 3.5 produces the unique minimal dependency-free depleting Σ-module of $\mathcal{T}_{\mathsf{RCI}}$.*

Proof. AMEX produces dependency-free depleting Σ-module by Theorem 3.4.3. We just need to prove the minimality claim.

To do this, let $\mathcal{M}_0$ be the unique minimal dependency-free depleting Σ-module of $\mathcal{T}_{\mathsf{RCI}}$. For some run of the algorithm, we prove by induction on the number of rule applications that $\mathcal{M} \subseteq \mathcal{M}_0$ by showing every axiom added to $\mathcal{M}$ by either rule is also contained in $\mathcal{M}_0$.

Base case: $\mathcal{M} = \emptyset \subseteq \mathcal{M}_0$

Inductive step: Assume for all $m < l$ rule applications we have $\mathcal{M} \subseteq \mathcal{M}_0$. Consider rule application l and distinguish between which rule is applicable:

- **R1** is applicable. Then there exists an axiom $A \bowtie C \in \mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}$ which causes an axiom $\Sigma \cup \mathsf{sig}(\mathcal{M})$-dependency in $\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}$. So we have $A \in \Sigma \cup \mathsf{sig}(\mathcal{M})$ for which $\mathsf{depend}_{\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}}(A \bowtie C) \cap (\Sigma \cup \mathsf{sig}(\mathcal{M})) \neq \emptyset$. Then for each $X \in \mathsf{depend}_{\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}}(A \bowtie C) \cap (\Sigma \cup \mathsf{sig}(\mathcal{M}))$ by Lemma 3.4.1 there exists at least one chain $\gamma \in \mathsf{chain}^X_{\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}}(A \bowtie C)$. But now consider output $\mathcal{M}_{iter}$ of the iterative algorithm for the same signature, since we have $\{A, X\} \subseteq \Sigma \cup \mathsf{sig}(\mathcal{M})$ and $\gamma \in \mathsf{chain}^X_{\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}}(A \bowtie C)$, by Theorem 3.4.1 we have $\gamma \subseteq \mathcal{M}_{iter}$, but by Lemma 3.3.4 $\mathcal{M}_{iter} = \mathcal{M}_0$, so for each $X \in \mathsf{depend}_{\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}}(A \bowtie C) \cap (\Sigma \cup \mathsf{sig}(\mathcal{M}))$ and each chain $\gamma \in \mathsf{chain}^X_{\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}}(A \bowtie C)$ we have $\gamma \subseteq \mathcal{M}_0$ as required.

- **R2** is applicable. Then $\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M} \not\equiv_{\Sigma \cup \mathsf{sig}(\mathcal{M})} \emptyset$ and there must exist a separability causing axiom by Theorem 3.4.2. Locating a separability causing axiom using the procedure in Figure 3.4 locates a $\mathcal{W} \subseteq \mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}$ such that $\mathcal{W} \not\equiv_{\Sigma \cup \mathsf{sig}(\mathcal{M})} \emptyset$, but where removing a single axiom α from $\mathcal{W}$ we have $\mathcal{W} \setminus \{\alpha\} \equiv_{\Sigma \cup \mathsf{sig}(\mathcal{M})} \emptyset$, the axiom α is "separability causing". It should be clear there must exists some minimal subset $\mathcal{W}' \subseteq \mathcal{W}$ such that $\mathcal{W}' \not\equiv_{\Sigma \cup \mathsf{sig}(\mathcal{M})} \emptyset$ with $\alpha \in \mathcal{W}'$ but since by the induction hypothesis we have $\mathsf{sig}(\mathcal{M}) \subseteq \mathsf{sig}(\mathcal{M}_0)$ then $\Sigma \cup \mathsf{sig}(\mathcal{M}) \subseteq \Sigma \cup \mathsf{sig}(\mathcal{M}_0)$ but now we can apply Lemma 3.3.4 (with $\mathcal{W} = \mathcal{W}'$ and $\mathcal{M}_0 = \mathcal{M}$) and conclude $\alpha \in \mathcal{M}_0$ as required.

$\square$

Deciding if **R1** is applicable — if an axiom $A \bowtie C$ causes an axiom $\Sigma \cup \mathsf{sig}(\mathcal{M})$-dependency in $\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}$— can be achieved in practice by deciding if an axiom $A \bowtie C$ causes an axiom $\Sigma \cup \mathsf{sig}(\mathcal{M})$-dependency in $\mathcal{T}_{\mathsf{RCI}}$, by Lemma 3.4.2. After a dependency is detected, locating the dependency chains which cause the dependency can be achieved using reachability analysis, following the sequence of symbols used in the dependency causing axiom.

To decide if **R2** is applicable, we first verify if $\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M} \not\equiv_{\Sigma \cup \mathsf{sig}(\mathcal{M})} \emptyset$, then we only need search for a separability causing axiom if one exists, only a single inseparability check is required to verify that $\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M} \equiv_{\Sigma \cup \mathsf{sig}(\mathcal{M})} \emptyset$. In comparison to the iterative algorithm, the we are required to perform an inseparability check each time we add an axiom to some $\mathcal{W}$ that does not cause an axiom $\Sigma \cup \mathsf{sig}(\mathcal{M})$-dependency, failure to do so could miss the detection of a separability causing axiom.

The application of both rules is achieved utilising the whole of $\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}$, it is not necessary to build up a subset $\mathcal{W} \subseteq \mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}$ as in the iterative algorithm which reduces the space requirement compared to the original iterative algorithm.

The worst case complexity for AMEX is the same as the iterative algorithm, which is $O(|\mathcal{T}_{\mathsf{RCI}}| + |\Sigma|) \times \Pi_2^p$. Even considering this, we show in the next section and through our extensive experimental evaluation in Chapter 6 that the optimisations we have introduced means extracting a module using AMEX can be done extremely efficiently in practice.

Example 3.4.1. *As a comparison we extract a module using the same ontology and signature as the example for the iterative approach Example 3.3.4.*

Starting with the ontology $\mathcal{T}_{\mathsf{RCI}} = \{\alpha_1 - \alpha_5\}$ from Example 3.3.4 and $\Sigma = \{\mathsf{Animal\_Group}, \mathsf{Lion}\}$.

*First **R1** is not applicable but **R2** does apply as it can be verified that $\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M} \not\equiv_{\Sigma \cup \mathsf{sig}(\mathcal{M})} \emptyset$, so we locate the first separability causing axiom using the procedure in Figure 3.4. First we consider $\mathcal{W}$ to be the top half of $\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}$, $\mathcal{W} = \{\alpha_1, \alpha_2\}$*

and we find $\mathcal{W} \not\equiv_{\Sigma \cup \mathsf{sig}(\mathcal{M})} \emptyset$, so we shrink $\mathcal{W}$ by half, resulting in $\mathcal{W} = \{\alpha_1\}$, and still can verify that $\mathcal{W} \not\equiv_{\Sigma \cup \mathsf{sig}(\mathcal{M})} \emptyset$ and since $\mathcal{W}$ was shrunk by a single axiom, we locate α_1 as the separability causing axiom and set $\mathcal{M} = \mathcal{M} \cup \{\alpha_1\}$, and then $\Sigma \cup \mathsf{sig}(\mathcal{M}) = \{\mathsf{Animal\_Group}, \mathsf{Lion}, \mathsf{has}, \mathsf{Animal}\}$.

*Next, we find that **R1** is applicable, that α_2 causes an axiom $\Sigma \cup \mathsf{sig}(\mathcal{M})$-dependency in $\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}$, as we have* $\mathsf{Lion} \in \Sigma \cup \mathsf{sig}(\mathcal{M})$ *and* $\{\mathsf{Animal}, \mathsf{has}\} \subseteq \mathsf{depend}_{\mathcal{T}_{\mathsf{RCI}}}(\alpha_2) \cap (\Sigma \cup \mathsf{sig}(\mathcal{M}))$. *And we find two chains* $\gamma_1 = \{\alpha_2, \alpha_4\} \in \mathsf{chain}^{\mathsf{Animal}}_{\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}}(\alpha_2)$ *and* $\gamma_2 = \{\alpha_2, \alpha_5\} \in \mathsf{chain}^{\mathsf{has}}_{\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}}(\alpha_2)$ *and set* $\mathcal{M} = \mathcal{M} \cup \gamma_i$ *for* $i = 1, 2$.

No rule is now applicable so we are done. The result is an identical module as the iterative algorithm example, $\mathcal{M} = \{\alpha_1, \alpha_2, \alpha_4, \alpha_5\}$.

If we compare the number of operations needed to extract the module in Example 3.4.1 compared to iterative algorithm in Example 3.3.4 assuming the axioms are processed in the order specified by input ontology. We find AMEX does *slightly* less computation to both detect axiom $\Sigma \cup \mathsf{sig}(\mathcal{M})$-dependencies in $\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}$ and to locate separability causing axioms to establish that $\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M} \equiv_{\Sigma \cup \mathsf{sig}(\mathcal{M})} \emptyset$. This is expected over such a small ontology, the computational saving AMEX provides comes from the efficient collection of dependency chains and separability causing axioms, which is most effective over a larger search space.

For a clearer picture of the potential performance improvements AMEX can provide, we now compare the iterative algorithm to AMEX for a range of signatures by extracting modules from a larger real-world ontology.

3.5. Comparing performance

We now provide a comparison between newly introduced AMEX (Figure 3.5 and the iterative algorithm (Figure 3.3). It is important to note, comparing the highly optimised AMEX to a naive implementation of the iterative algorithm is hardly a fair one, therefore we present the result of these experiments to give the reader some idea of the potential improvements AMEX offers.

We have implemented both algorithms in Java using the library for ontology manipulation OWL-API [HB11], and use the QBF solver sKizzo [Ben04] for deciding inseparability. As we expect the iterative algorithm to perform poorly in practice, as input to our experiments, we use the small ontology Lipid Ontology (LiPrO) taken from the bio-medical BioPortal repository [Whe+11]. The terminological part of LiPrO is an $\mathcal{ALCIN}$ acyclic terminology with RCIs, consisting of 776 axioms whose signature consists of 764 symbols. Our input signatures are 400 signatures taken at random from the signature of LiPrO, they consist of four sets of 100 signatures containing 25, 50, 75 or 100 symbols.

By Theorem 3.3.2 and Theorem 3.4.4, for a given signature, both algorithms produce the same module as output — the unique minimal dependency-free depleting Σ-module. We examine the number of total operations each algorithm performs as a metric for comparison. Figure 3.6 shows a table summarising the results of our experimental evaluation, achieved by extracting modules for each of the 400 signatures from the LiPrO ontology. We distinguish and count each type of "check" each algorithm makes, whether it be dependency check (Dep.), deciding if $\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}$ contains any axiom $\Sigma \cup \mathsf{sig}(\mathcal{M})$-dependencies, or an inseparability check (Insep.) deciding if $\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M} \equiv_{\Sigma \cup \mathsf{sig}(\mathcal{M})} \emptyset$, both algorithms utilising checks in a different manner towards the same result.

		Iterative algorithm Avg.			AMEX Avg.		
$\lvert\Sigma\rvert$	$\lvert\mathcal{M}\rvert$	Dep.	Insep.	Total	Dep.	Insep.	Total
25	82.99	18,132	18,051	36,183	7,728	1.2	7,729.2
50	136.50	29,236	29,105	58,362	14,382	1	14,382
75	196.30	36,726	36,529	73,256	19,788	1	19,789
100	234.00	42,109	41,859	83,970	24,463	1	24,464

Figure 3.6: Comparison of "checks" between old and new algorithms

Evaluating the results, it is obvious AMEX performs considerably fewer checks of either kind than the iterative algorithm for the extracted signatures. Notice, even in the worst case we use 41.45% fewer dependency checks on average for any input signature, going up to 57.19% in the best case. This can be explained by the collection of dependency chains in AMEX, a single dependency check identifying several axioms to be added to the module. Figure 3.7 shows the average length of the dependency chains detected, and the average frequency of occurrence of chains greater than one axiom in length. We observe a larger signature correlates to a higher frequency of chains, but chains which tend to be shorter in length.

$\lvert\Sigma\rvert$	Length Avg.	Freq. length > 1 Avg.
25	15.35	19.65
50	13.88	40.81
75	12.90	66.00
100	12.55	88.53

Figure 3.7: Chain metrics for AMEX

The largest difference, however, comes from the number of inseparability checks which are performed in each case. Notice for all of the signatures of size 50, 75 or 100, AMEX performs only a single inseparability check on average per extraction. This implies that in these cases, moving every axiom which causes an axiom $\Sigma \cup \mathsf{sig}(\mathcal{M})$-dependency to $\mathcal{M}$ is sufficient to capture all separability causing axioms, the single inseparability check is simply used to verify that $\mathcal{M}$ is indeed a depleting module. Even in the case of signatures size 25, all but two of the extractions required a single inseparability check, the remaining two used 11 inseparability checks each to locate two inseparability causing axioms, resulting in an average of 1.2 checks per extraction.

To summarise, for the LiPrO ontology, the total checks over both deciding the presence of axiom dependency and deciding inseparability are significantly reduced. This is particularly notable in the case of deciding inseparability, which is the most computationally expensive procedure in the algorithm (in Π_2^p). This suggests module extraction with AMEX would perform generally better than the iterative algorithm when applied to *any* ontology. We perform an extensive experimental evaluation in Chapter 6, in which we will observe how well AMEX performs over a much larger ontology and signature selection.

3.6. Conclusion

In this chapter we began by exploring what is already known from [Kon+08a; Kon+13]. The authors show for an acyclic $\mathcal{ALCI}$ terminology $\mathcal{T}$, deciding if $\mathcal{T} \equiv_\Sigma \emptyset$ is decidable in Π_2^p, on the condition that Σ consists only of concept names, and that same problem for unrestricted signatures is also decidable in Π_2^p if an additional syntactic condition is imposed on the ontology, namely it being free of direct Σ-dependencies. This observation then lead to the development of an algorithm which allows, for an acyclic $\mathcal{ALCI}$ terminology $\mathcal{T}$, and a signature Σ, the extraction of the unique minimal depleting module $\mathcal{M}$ such that $\mathcal{T} \setminus \mathcal{M}$ contains no direct $\Sigma \cup \mathsf{sig}(\mathcal{M})$-dependencies, an approximation of the ideal minimal module.

For the remainder of the chapter we presented our new contributions. We started by extending the theory which underpins the $\mathcal{ALCI}$ approximation, in order to support depleting module extraction from acyclic $\mathcal{ALCQI}$ terminologies, which may additional contain repeated concept names. To this end, by Lemma 3.3.1, we showed that deciding if $\mathcal{T}_{\mathsf{RCI}} \equiv_\Sigma \emptyset$ where $\mathcal{T}_{\mathsf{RCI}}$ is an acyclic $\mathcal{ALCQI}$ ontology with RCIs and Σ is a concept signature is also decidable in Π_2^p. The case for unrestricted signatures, by Theorem 3.3.1, we showed was also decidable in Π_2^p, if like the $\mathcal{ALCI}$ case, we impose an additional syntactic condition on the ontology. For this we used axiom Σ-dependencies, which generalise the notion of direct Σ-dependencies in order to support acyclic terminologies with RCIs. With these theoretical results established, we showed a modification of the $\mathcal{ALCI}$ extraction algorithm can be used to extract a depleting module, which, by Theorem 3.3.2, we proved is the unique minimal dependency-free depleting Σ-module (minimal depleting Σ-module such that $\mathcal{T}_{\mathsf{RCI}} \setminus \mathcal{M}$ contains no axiom Σ-dependency) which can be extracted from an acyclic $\mathcal{ALCQI}$ terminology with RCIs.

Finally, using the $\mathcal{ALCQI}$ extraction algorithm as a starting point, we suggest a number of optimisations which led to the production of a new rule-based algorithm AMEX, which produces an identical module but offers measurably better practical performance. The potential performance gain we verified by comparing AMEX to the unoptimised algorithm by means of a small experimental evaluation in which we extracted modules from the real-world ontology LiPr0.

CHAPTER 4

Hybrid Module Extraction

We have now considered several approaches which produce approximations, necessary under the undecidability constraints of computing minimal depleting Σ-modules in even moderately expressive logics. These approximations range from the specialised AMEX procedure which we introduced in the previous chapter, only applicable to acyclic $\mathcal{ALCQI}$ terminologies which may optionally contain RCIs, to the very general locality-based ones such as STAR ($\top\bot^*$-locality), applicable to general, cyclic $\mathcal{SROIQ}$ ontologies.

The benefit of using a general approach means a module can be extracted from a wider number of ontologies, but a procedure with such a broad scope can lead to approximations which are much larger than the minimal depleting Σ-modules they approximate, containing many surplus axioms which do not belong to the minimal Σ-module. We show exactly this in Chapter 6, where our experimental analysis reveals that the modules produced by the STAR approach can be significantly larger than the corresponding ones produced by AMEX.

With this in mind, in this chapter we aim to combine several extraction procedures together, utilising the often more successful specialised approximations to remove unnecessary axioms from our modules, whilst still maintaining the inclusive nature of the general approximations. The approach we take is to generalise the already successful STAR approximation which combines two different locality notions together into a single extraction procedure.

In our effort to combine extraction procedures together we will be drawing on the specifics of the locality-based STAR-modules. As a reminder for the reader we recounted the details of locality-based modules including STAR modules in Section 2.4.4.

4.1. Combining depleting modules

As we described in Section 2.4.4, the syntactic locality-based modules (LBMs) $\top-$ and $\bot$-modules are both known to be depleting Σ-modules, but the nature of each locality variant means they may capture different axioms in the modules they produce. The observation about the types of axiom each variant captures has been exploited to particular success in helping to improve ontology classification [Cla10].

Beyond this, the main research task surrounding modularity is to minimise the size of approximations. To this end, with the aim of reducing the size of the locality approximations STAR-modules ($\bot\top^*$- modules) were developed, produced by iteratively extracting $\top-$ and $\bot$-modules from one from the other until a fixpoint is reached. STAR modules have also been shown to be depleting Σ-modules, and are at least as small as the corresponding $\bot$ or $\top$ Σ-modules [SSZ09].

It turns out this property is not only limited LBMs, and in fact we can show that any two procedures which can extract a depleting Σ-module from an ontology can be combined together to produce a depleting Σ-module which, under some natural conditions, is no larger than the corresponding modules extracted by either procedure.

Definition 4.1.1 (Module extraction procedure). *Let $\mathcal{O}$ be an ontology, and x a procedure. If x can extract a depleting Σ-module from $\mathcal{O}$ for a signature Σ, we call x a depleting module extraction procedure for $\mathcal{O}$.*

Definition 4.1.2 (x-module). *Let $\mathcal{O}$ be an ontology and x be a depleting module extraction procedure for $\mathcal{O}$. If $\mathcal{M}$ is a Σ-module extracted from $\mathcal{O}$ using x for a signature Σ, we call $\mathcal{M}$ an x-module and we write $\mathcal{M} = x\text{-mod}(\mathcal{O}, \Sigma)$, e.g. $\mathcal{M} = \mathit{STAR}\text{-mod}(\mathcal{O}, \Sigma)$ represents the module $\mathcal{M}$ extracted by the STAR procedure from $\mathcal{O}$ for Σ.*

First we show that by nesting the extraction of depleting Σ-modules we obtain a depleting Σ-module.

Theorem 4.1.1. *Let $\mathcal{M} \subseteq \mathcal{M}' \subseteq \mathcal{O}$ be ontologies and Σ a signature such that $\mathcal{M}'$ is a depleting Σ-module of $\mathcal{O}$ and $\mathcal{M}$ is a depleting Σ-module of $\mathcal{M}'$. Then $\mathcal{M}$ is a depleting Σ-module of $\mathcal{O}$.*

Proof. Assume that $\mathcal{M} \subseteq \mathcal{M}' \subseteq \mathcal{O}$ and Σ is a signature such that $\mathcal{M}'$ is a depleting Σ-module of $\mathcal{O}$ and $\mathcal{M}$ is a depleting Σ-module of $\mathcal{M}'$ To prove that $\mathcal{M}$ is a depleting Σ-module of $\mathcal{O}$, consider an interpretation $\mathcal{I}$. We have to show that there exists a model $\mathcal{J}$ of $\mathcal{O} \setminus \mathcal{M}$ such that $\mathcal{J}|_{\Sigma \cup \mathsf{sig}(\mathcal{M})} = \mathcal{I}|_{\Sigma \cup \mathsf{sig}(\mathcal{M})}$. As $\mathcal{M}$ is a depleting Σ-module of $\mathcal{M}'$, there exists an interpretation $\mathcal{J}'$ such that $\mathcal{J}'|_{\Sigma \cup \mathsf{sig}(\mathcal{M})} = \mathcal{I}|_{\Sigma \cup \mathsf{sig}(\mathcal{M})}$ and $\mathcal{J}' \models (\mathcal{M}' \setminus \mathcal{M})$. Similarly, as $\mathcal{M}'$ is a depleting Σ-module of $\mathcal{O}$, there exists an interpretation $\mathcal{J}$ such that $\mathcal{J}|_{\Sigma \cup \mathsf{sig}(\mathcal{M}')} = \mathcal{J}'|_{\Sigma \cup \mathsf{sig}(\mathcal{M}')}$ and $\mathcal{J} \models (\mathcal{O} \setminus \mathcal{M}')$. As $\mathsf{sig}(\mathcal{M}) \subseteq \mathsf{sig}(\mathcal{M}') \subseteq \mathsf{sig}(\mathcal{O})$ we have $\mathcal{J}|_{\Sigma \cup \mathsf{sig}(\mathcal{M})} = \mathcal{I}|_{\Sigma \cup \mathsf{sig}(\mathcal{M})}$ and $\mathcal{J} \models (\mathcal{M}' \setminus \mathcal{M})$. But then $\mathcal{J} \models ((\mathcal{O} \setminus \mathcal{M})$ and so $\mathcal{J}$ is as required. $\square$

Corollary 4.1.1. *Let $\mathcal{O}$ be an ontology, Σ a signature and x and y be depleting module extraction procedures for $\mathcal{O}$. Then both xy-mod$(\mathcal{O}, \Sigma)$ = x-mod$(y$-mod$(\mathcal{O}, \Sigma), \Sigma)$ and yx-mod$(\mathcal{O}, \Sigma)$ = y-mod$(x$-mod$(\mathcal{O}, \Sigma), \Sigma)$ are depleting Σ-modules.*

Proof. Since both x and y extract depleting modules from $\mathcal{O}$, it is an immediate consequence of Theorem 4.1.1. $\square$

Corollary 4.1.1 allows us to extract depleting Σ-modules by nesting procedures together and the approximations we produce as a result are still guaranteed to be depleting Σ-modules for the input ontology. This allows us to potentially produce better approximations; the different notions used to produce approximations by extraction procedures means that for a signature Σ different axioms may be considered irrelevant — not contained in the minimal depleting Σ-module — and by nesting procedures these axioms can be discarded, resulting in a module which may be closer the ideal minimal.

We can improve on this further by generalising the STAR procedure into what will be known as the hybrid extraction procedure, which works by extracting a sequence of nested modules in an iterative fashion until a fixpoint is reached. We can achieve this using the algorithm in Figure 4.1.

Input: Ontology $\mathcal{O}$, signature Σ, depleting module extraction procedures for $\mathcal{O}$ x and y

Output: Depleting Σ-module $\mathcal{M}$ of $\mathcal{O}$

1 $\mathcal{M} ::= x\text{-mod}(\mathcal{O}, \Sigma)$
2 **repeat**
3 $\quad \mathcal{M}_{\mathsf{prev}} ::= \mathcal{M}$
4 $\quad \mathcal{M} ::= y\text{-mod}(\mathcal{M}_{\mathsf{prev}}, \Sigma)$
5 $\quad$ **if** $|\mathcal{M}| < |\mathcal{M}_{\mathsf{prev}}|$ **then**
6 $\quad\quad \mathcal{M}_{prev} ::= \mathcal{M}$
7 $\quad\quad \mathcal{M} ::= x\text{-mod}(\mathcal{M}_{\mathsf{prev}}, \Sigma)$
8 $\quad$ **end**
9 **until** $|\mathcal{M}| = |\mathcal{M}_{\mathsf{prev}}|$
10 **return** $\mathcal{M}$

Figure 4.1: Hybrid extraction algorithm

Lemma 4.1.1. *Given an ontology $\mathcal{O}$ and signature Σ, the output of the algorithm in Figure 4.1 is a depleting Σ-module of $\mathcal{O}$.*

Proof. We prove that the output of the algorithm is depleting Σ-module by showing that on each iteration of the **repeat** loop of the algorithm the invariant that $\mathcal{M}$ (extracted on Line 4 and Line 7) is a depleting Σ-module of $\mathcal{O}$, so when the fixpoint is reached and the algorithm terminates the $\mathcal{M}$ produced as output must also be depleting Σ-module of $\mathcal{O}$.

For a proof by induction on the iterations of the **repeat** loop of the algorithm.

Base case: On iteration 1. Since $\mathcal{M}$ is a depleting Σ-module extracted by x from $\mathcal{O}$ on Line 1, the module $\mathcal{M}_{\mathsf{prev}}$ on Line 3 is a depleting Σ-module of $\mathcal{O}$, and since the $\mathcal{M}$ on Line 4 extracted by y is a depleting Σ-module of $\mathcal{M}_{\text{prev}}$, $\mathcal{M}$ is a depleting Σ-module of $\mathcal{O}$ by Theorem 4.1.1.

If the condition on line Line 5 applies, since $\mathcal{M}$ on line Line 4 is a depleting Σ-module of $\mathcal{O}$, it follow that $\mathcal{M}_{\text{prev}}$ on Line 6 is also one, and since $\mathcal{M}$ on Line 7 is depleting Σ-module of $\mathcal{M}_{\text{prev}}$ extracted by x it follows that $\mathcal{M}$ is a depleting Σ-module of $\mathcal{O}$ by Theorem 4.1.1.

Inductive step: Assume for all iterations $m < l$ that the $\mathcal{M}$ produced by the algorithm is a depleting Σ-module of $\mathcal{O}$. Consider iteration l. The proof is the same as in the base case, except the proof that the initial $\mathcal{M}_{\text{prev}}$ on Line 3 is a depleting Σ-module of $\mathcal{O}$, comes from previous iteration by means of the induction hypothesis instead of Line 1.

□

Similarly to the definition of STAR modules in [SSZ09], the module produced by the hybrid procedure is the least fixpoint in the sequence $\{\mathcal{M}_i\}_{i\geq 1}$ where $\mathcal{M}_1$ is x-mod$(\mathcal{O}, \Sigma)$ and for $i \geq 2$

$$\mathcal{M}_i = \begin{cases} x\text{-mod}(\mathcal{M}_{i-1}, \Sigma) & \text{if } i \text{ is even;} \\ y\text{-mod}(\mathcal{M}_{i-1}, \Sigma) & \text{otherwise.} \end{cases}$$

The output of the algorithm is $\mathcal{M}_n$, where n is the smallest $n > 0$ with $\mathcal{M}_n = \mathcal{M}_{n+1}$. Observe that the sequence of modules is strictly decreasing in size as otherwise we have reached the fixpoint. This means the algorithm will eventually terminate, the value for n being bounded by the number of axioms in $\mathcal{O}$.

The condition on Line 5 ensures we only do the computation necessary to reach the fixpoint. If we do find that $|\mathcal{M}| = |\mathcal{M}_{\text{prev}}|$ then the extraction of the y-module on Line 4 must have left the previously extracted x-module unchanged, so extracting another x-module for the same signature won't make any difference to the result.

Definition 4.1.3 (Hybrid module)**.** *For an ontology $\mathcal{O}$ and signature Σ. A run of* *the algorithm in Figure 4.1 starting by extracting a x-module then an y-module until a fixpoint is reached we call the hybrid yx-module of $\mathcal{O}$ written $\mathcal{M} = \mathsf{Hyb}_{[y,x]}\text{-}mod(\mathcal{O}, \Sigma)$.*

It is easy to see, for a run of the algorithm with an ontology $\mathcal{O}$, signature Σ, and using the $\top$-locality $\bot$-locality procedures, by definition we have a $\mathsf{Hyb}_{[\bot,\top]}\text{-mod}(\mathcal{O}, \Sigma) = \mathsf{STAR}\text{-mod}(\mathcal{O}, \Sigma)$.

Lemma 4.1.2. *Let $\mathcal{O}_1 \subseteq \mathcal{O}_2$ be $\mathcal{SROIQ}$ ontologies and Σ a signature, then $\mathcal{M}_1 \subseteq \mathcal{M}_2$ where $\mathcal{M}_i$ is the minimal depleting Σ-module for $\mathcal{O}_i$ for $i = 1, 2$.*

Proof. This proof follows directly from the monotonicity of the model-inseparability relation (Proposition 2.3.3) and underlying logic.

Notice that $\mathcal{M}_2' = \mathcal{M}_2 \cap \mathcal{O}_1$ is *a* depleting module of $\mathcal{O}_1$. Indeed, since $\mathcal{M}_2$ is depleting Σ-module of $\mathcal{O}_2$, by definition we have $\mathcal{O}_2 \setminus \mathcal{M}_2 \equiv_{\Sigma \cup \mathsf{sig}(\mathcal{M}_2)} \emptyset$ and since $\mathcal{O}_1 \setminus \mathcal{M}_2' \subseteq \mathcal{O}_2 \setminus \mathcal{M}_2$ it holds that $\mathcal{O}_1 \setminus \mathcal{M}_2' \equiv_{\Sigma \cup \mathsf{sig}(\mathcal{M}_2)} \emptyset$ and then, since $(\Sigma \cup \mathsf{sig}(\mathcal{M}_2')) \subseteq (\Sigma \cup \mathsf{sig}(\mathcal{M}_2))$ by monotonicity of the model inseparability relation we have $\mathcal{O}_1 \setminus \mathcal{M}_2' \equiv_{\Sigma \cup \mathsf{sig}(\mathcal{M}_2')} \emptyset$ and $\mathcal{M}_2'$ is a depleting Σ-module of $\mathcal{O}_1$ as claimed. Now since $\mathcal{M}_1$ is the *minimal* depleting Σ-module of $\mathcal{O}_1$ we have $\mathcal{M}_1 \subseteq \mathcal{M}_2'$ but then by definition $\mathcal{M}_2' \subseteq \mathcal{M}_2$ so we can conclude that $\mathcal{M}_1 \subseteq \mathcal{M}_2$ as required. □

Definition 4.1.4 (Subset preserving). *Let x and be a module extraction procedure for an ontology $\mathcal{O}_2$. It is called subset preserving if $\mathcal{O}_1 \subseteq \mathcal{O}_2$ implies x-mod$(\mathcal{O}_1, \Sigma) \subseteq x$-mod$(\mathcal{O}_2, \Sigma)$.*

The STAR extraction procedure is known to be subset preserving from [SSZ09] and the following lemma shows that AMEX is also subset preserving over ontologies for which it is an extraction procedure.

Lemma 4.1.3 (AMEX subset preservation). *Let $\mathcal{T}_1 \subseteq \mathcal{T}_2$ be acyclic $\mathcal{ALCQI}$ terminologies with RCIs and Σ a signature. Let $\mathcal{M}_1$ = AMEX-mod$(\mathcal{T}_1, \Sigma)$ and $\mathcal{M}_2$ = AMEX-mod$(\mathcal{T}_2, \Sigma)$. Then $\mathcal{M}_1 \subseteq \mathcal{M}_2$.*

Proof. Let $\mathcal{T}_1 \subseteq \mathcal{T}_2$ be acyclic $\mathcal{ALCQI}$ terminologies and Σ a signature and assume $\mathcal{M}_1 = \text{AMEX-mod}(\mathcal{T}_1, \Sigma)$ and $\mathcal{M}_2 = \text{AMEX-mod}(\mathcal{T}_2, \Sigma)$.

Since by Theorem 3.4.4 $\mathcal{M}_i$ is the unique minimal dependency-free depleting Σ-module of $\mathcal{T}_i$, for $i = 1, 2$, we can prove the claim using a similar argument to the one used in Lemma 4.1.2, namely to show that $\mathcal{M}_2' = \mathcal{M}_2 \cap \mathcal{T}_1$ is a dependency-free depleting Σ-module of $\mathcal{T}_1$ and therefore $\mathcal{M}_1 \subseteq \mathcal{M}_2$.

To this end, we must show that $\mathcal{M}_2'$ is both a depleting Σ-module of $\mathcal{T}_1$ and that it contains no axiom $\Sigma \cup \mathsf{sig}(\mathcal{M}_2')$-dependency. By definition of $\mathcal{M}_2$ we have $\mathcal{T}_2 \setminus \mathcal{M}_2 \equiv_{\Sigma \cup \mathsf{sig}(\mathcal{M}_2)} \emptyset$ and since $\mathcal{T}_1 \setminus \mathcal{M}_2' \subseteq \mathcal{T}_2 \setminus \mathcal{M}_2$ it follows by the monotonicity of model inseparability that $\mathcal{T}_1 \setminus \mathcal{M}_2 \equiv_{\Sigma \cup \mathsf{sig}(\mathcal{M}_2')} \emptyset$, that is $\mathcal{M}_2'$ is a depleting Σ-module of $\mathcal{T}_1$. What remains to be shown is that $\mathcal{T}_1 \setminus \mathcal{M}_2'$ contains no axiom $\Sigma \cup \mathsf{sig}(\mathcal{M}_2')$-dependency.

For a proof by contradiction assume $\mathcal{T}_1 \setminus \mathcal{M}_2'$ does contain an axiom $\Sigma \cup \mathsf{sig}(\mathcal{M}_2')$-dependency, that there is an axiom $\alpha \in \mathcal{T}_1 \setminus \mathcal{M}_2'$ such that there exists a symbol $A \in \Sigma \cup \mathsf{sig}(\mathcal{M}_2')$ with a symbol $X \in \mathsf{depend}_{\mathcal{T}_1 \setminus \mathcal{M}_2'}(\alpha) \cap \Sigma \cup \mathsf{sig}(\mathcal{M}_2')$. But then immediately since $\mathcal{T}_1 \setminus \mathcal{M}_2' \subseteq \mathcal{T}_2 \setminus \mathcal{M}_2$ and $\Sigma \cup \mathsf{sig}(\mathcal{M}_2') \subseteq \Sigma \cup \mathsf{sig}(\mathcal{M}_2)$ that $\mathcal{T}_2 \setminus \mathcal{M}_2$ also contains an axiom $\Sigma \cup \mathsf{sig}(\mathcal{M}_2)$-dependency, which is contradiction to the original assumption.

Now since $\mathcal{M}_1$ is the unique minimal dependency-free depleting Σ-module of $\mathcal{T}_1$ and $\mathcal{M}_2'$ is *a* dependency-free depleting Σ-module of $\mathcal{T}_1$ it follows that $\mathcal{M}_1 \subseteq \mathcal{M}_2'$ and since $\mathcal{M}_2' \subseteq \mathcal{M}_2$ by definition it follows that $\mathcal{M}_1 \subseteq \mathcal{M}_2$ as required.

□

Also notice by the transitivity of the subset relation if we have two extraction procedures x and y which are both subset preserving it follows that the hybrid procedure is also subset preserving.

We can also show, generalising the same result for STAR-modules in [SSZ09], that the order in which subset preserving module extraction procedures are combined makes no difference to the module which is produced.

Lemma 4.1.4. *Let x and y be subset preserving depleting module extraction procedures, $\mathcal{O}$ an ontology, and Σ a signature. Then* $\mathsf{Hyb}_{[x,y]}$*-mod*$(\mathcal{O},\Sigma) = \mathsf{Hyb}_{[y,x]}$*-mod*$(\mathcal{O},\Sigma)$.

Proof. This proof is a generalisation of the same result for locality modules by Kazakov, which was published in [Ves13].

Since by definition a hybrid xy-module starts by extracting a y-module from $\mathcal{O}$, then an x-module, and so on until a fixpoint is reached, we have:

$$\mathsf{Hyb}_{[x,y]}\text{-mod}(\mathcal{O},\Sigma) = (\mathsf{Hyb}_{[y,x]})y\text{-mod}(\mathcal{O},\Sigma) = \mathsf{Hyb}_{[y,x]}\text{-mod}(y\text{-mod}(\mathcal{O},\Sigma),\Sigma)$$

But then since by y-mod$(\mathcal{O},\Sigma) \subseteq \mathcal{O}$ and the fact that $\mathsf{Hyb}_{[y,x]}$-mod is a subset preserving procedure we can conclude that:

$$\begin{array}{c} \mathsf{Hyb}_{[y,x]}\text{-mod}(y\text{-mod}(\mathcal{O},\Sigma),\Sigma) \subseteq \mathsf{Hyb}_{[y,x]}\text{-mod}(\mathcal{O},\Sigma) \\ \parallel \\ \mathsf{Hyb}_{[x,y]}\text{-mod}(\mathcal{O},\Sigma) \end{array}$$

This proves that $\mathsf{Hyb}_{[x,y]}$-mod$(\mathcal{O},\Sigma) \subseteq \mathsf{Hyb}_{[y,x]}$-mod$(\mathcal{O},\Sigma)$. The converse direction can be proven in the same way. □

The following lemma also follows immediately from the property of subset preservation by extraction procedures. It is guaranteed the module produced by the hybrid procedure is no smaller than the module produced by either of the input extraction procedures.

Lemma 4.1.5. *Let x and y be subset preserving depleting module extraction procedures for an ontology $\mathcal{O}$, and Σ a signature. Then x-mod$(\mathcal{O},\Sigma) \subseteq \mathsf{Hyb}_{[x,y]}$-mod$(\mathcal{O},\Sigma)$ and y-mod$(\mathcal{O},\Sigma) \subseteq \mathsf{Hyb}_{[x,y]}$-mod$(\mathcal{O},\Sigma)$.*

4.2. Combining STAR and AMEX

Part of our motivation for combining extraction procedures together is to be able to use the AMEX procedure along with a more general procedure such as STAR to hopefully produce a more successful approximation for a given input signature.

However, if we combine AMEX and STAR naïvely using the algorithm Figure 4.1 we are still limited to extracting modules from acyclic $\mathcal{ALCQI}$ ontologies with RCIs. Although the STAR is an extraction procedure for ontologies up to $\mathcal{SROIQ}$ in expressivity, there is no guarantee AMEX is an extraction procedure for the output of the STAR approach, so we must limit our input ontologies to the less expressive ontologies AMEX supports in order for the hybrid approach to work. To tackle this limitation, we observe that in *principle*, there is a way for AMEX to extract modules from general ontologies.

Lemma 4.2.1. *Let $\mathcal{O}$ be an ontology, let $\mathcal{O}_1 \subseteq \mathcal{O}$ and $\mathcal{O}_2 := \mathcal{O} \setminus \mathcal{O}_1$. If $\mathcal{M}$ is a depleting $\Sigma \cup \mathsf{sig}(\mathcal{O}_2)$-module of $\mathcal{O}_1$, then $\mathcal{M} \cup \mathcal{O}_2$ is a depleting Σ-module of $\mathcal{O}$.*

Proof. Since $\mathcal{M}$ is a depleting $\Sigma \cup \mathsf{sig}(\mathcal{O}_2)$-module of $\mathcal{O}_1$ we have $\mathcal{O}_1 \setminus \mathcal{M} \equiv_{(\Sigma \cup \mathsf{sig}(\mathcal{O}_2)) \cup \mathsf{sig}(\mathcal{M})} \emptyset$. Notice that $\mathcal{O} \setminus (\mathcal{M} \cup \mathcal{O}_2) = \mathcal{O}_1 \setminus \mathcal{M}$ so we have $\mathcal{O} \setminus (\mathcal{M} \cup \mathcal{O}_2) \equiv_{(\Sigma \cup \mathsf{sig}(\mathcal{O}_2)) \cup \mathsf{sig}(\mathcal{M})} \emptyset$, so $\mathcal{M} \cup \mathcal{O}_2$ is a depleting Σ-module of $\mathcal{O}$ as required. $\square$

An immediate consequence of Lemma 4.2.1 enables us to to extract a module using AMEX from a general ontology $\mathcal{O}$. We can split $\mathcal{O}$ into two parts $\mathcal{O}_{\mathcal{ALCQI}}$ and $\mathcal{O}_{\mathsf{rest}}$, where $\mathcal{O}_{\mathcal{ALCQI}}$ is an acyclic $\mathcal{ALCQI}$ terminology which is a subset of $\mathcal{O}$ (ideally as large as possible) and $\mathcal{O}_{\mathsf{rest}} := \mathcal{O} \setminus \mathcal{O}_{\mathcal{ALCQI}}$. Then we can extract an AMEX-module $\mathcal{M}$ from $\mathcal{O}_{\mathcal{ALCQI}}$ for $\Sigma \cup \mathsf{sig}(\mathcal{O}_{\mathsf{rest}})$ to produce a module $\mathcal{M} \cup \mathcal{O}_{\mathsf{rest}}$ which is a depleting module of $\mathcal{O}$. Extracting an AMEX-module from a general ontology using this methodology directly is unlikely to compute small modules, especially if the size of $\mathcal{O}_{\mathsf{rest}}$ is unavoidably large, but does enable us to combine AMEX with STAR in a specialised version of the hybrid algorithm from Figure 4.1.

Input: $\mathcal{SROIQ}$ Ontology $\mathcal{O}$, signature Σ
Output: Depleting module $\mathcal{M}$ of $\mathcal{O}$ w.r.t Σ

$\mathcal{M} ::= \mathsf{STAR}(\mathcal{O}, \Sigma)$
repeat
    $\mathcal{M}_{\mathsf{prev}} ::= \mathcal{M}$
    $\mathcal{M}_{\mathcal{ALCQI}} ::= \mathsf{get\_acyclic\_alcqi\_subset}(\mathcal{M}_{\mathsf{prev}})$
    $\mathcal{M}_{\mathsf{rest}} ::= \mathcal{M}_{\mathsf{prev}} \setminus \mathcal{M}_{\mathcal{ALCQI}}$
    $\mathcal{M} ::= \mathsf{AMEX}(\mathcal{M}_{\mathcal{ALCQI}}, \Sigma \cup \mathsf{sig}(\mathcal{M}_{\mathsf{rest}})) \cup \mathcal{M}_{\mathsf{rest}}$
    if $|\mathcal{M}| < |\mathcal{M}_{\mathsf{prev}}|$ **then**
        $\mathcal{M}_{\mathsf{prev}} ::= \mathcal{M}$
        $\mathcal{M} ::= \mathsf{STAR}(\mathcal{M}_{\mathsf{prev}}, \Sigma)$
    end
until $|\mathcal{M}| = |\mathcal{M}_{\mathsf{prev}}|$
return $\mathcal{M}$

Figure 4.2: STAR-AMEX extraction algorithm

In the algorithm in Figure 4.2 we assume the function get_acyclic_alcqi_subset takes an ontology $\mathcal{O}$ as a parameter and returns a subset of $\mathcal{O}$ which is an $\mathcal{ALCQI}$ acyclic terminology with RCIs (the empty ontology being a valid acyclic $\mathcal{ALCQI}$ terminology with RCIs) — exactly how this might be achieved we consider in Section 4.3.

Definition 4.2.1 (Hybrid STAR-AMEX module)**.** *Let $\mathcal{O}$ be a $\mathcal{SROIQ}$ ontology, Σ a signature. If the algorithm in Figure 4.2 takes $\mathcal{O}$ and Σ as input and outputs a module $\mathcal{M}$. We call $\mathcal{M}$ the hybrid STAR-AMEX-module $\mathcal{O}$ w.r.t Σ, written $\mathcal{M} = {}^{\mathsf{STAR}}_{\mathsf{AMEX}}\text{-}mod(\mathcal{O}, \Sigma)$.*

Theorem 4.2.1. *Let $\mathcal{O}$ be a $\mathcal{SROIQ}$ ontology, Σ a signature. The module $\mathcal{M} = {}^{\mathsf{STAR}}_{\mathsf{AMEX}}\text{-}mod(\mathcal{O}, \Sigma)$ is a depleting Σ-module of $\mathcal{O}$.*

Proof. We can use a modified version of the inductive proof of the general hybrid algorithm (Figure 4.1) to show that on every iteration we maintain the invariant that $\mathcal{M}$ is a depleting Σ-module of $\mathcal{O}$.

For a proof by induction on the iterations of the **repeat** loop of the algorithm.

Base case: On iteration 1, for $\mathcal{M}$ on Line 1, since $\mathcal{O}$ is a $\mathcal{SROIQ}$ ontology and STAR is a depleting module procedure for $\mathcal{O}$, it follows that $\mathcal{M}$ is a depleting Σ-module of $\mathcal{O}$.

For $\mathcal{M}$ on Line 6, since $\mathcal{M}$ on Line 1 is a depleting Σ-module of $\mathcal{O}$ it follows that $\mathcal{M}_{\mathsf{prev}} = \mathcal{M}_{\mathcal{ALCQI}} \cup \mathcal{M}_{\mathsf{rest}}$ on Line 3 is a depleting Σ-module of $\mathcal{O}$. Then by Theorem 3.4.3 $\mathsf{AMEX}(\mathcal{M}_{\mathcal{ALCQI}}, \Sigma \cup \mathsf{sig}(\mathcal{M}_{\mathsf{rest}}))$ extracts a depleting $\Sigma \cup \mathsf{sig}(\mathcal{M}_{\mathsf{rest}})$-module of $\mathcal{M}_{\mathcal{ALCQI}}$, it follows by Lemma 4.2.1 that $\mathcal{M} = \mathsf{AMEX}(\mathcal{M}_{\mathcal{ALCQI}}, \Sigma \cup \mathsf{sig}(\mathcal{M}_{\mathsf{rest}})) \cup \mathcal{M}_{\mathsf{rest}}$ is a Σ-module of $\mathcal{M}_{\mathsf{rest}}$, but then by Theorem 4.1.1 since $\mathcal{M}_{\mathsf{rest}}$ is a depleting Σ-module of $\mathcal{O}$, it follows that $\mathcal{M}$ is a depleting Σ-module of $\mathcal{O}$ as required.

Finally, $\mathcal{M}$ on Line 9, since $\mathcal{M}$ on Line 6 is a depleting Σ-module of $\mathcal{O}$, $\mathcal{M}_{\mathsf{prev}}$ on Line 8 is also one, and since $\mathcal{M}$ extracted by STAR is depleting Σ-module of $\mathcal{M}_{\mathsf{prev}}$, it is a depleting Σ-module of $\mathcal{O}$ by Theorem 4.1.1.

Inductive step: Assume for all iterations $m < l$ that the $\mathcal{M}$ produced by the algorithm is a depleting module of $\mathcal{O}$ w.r.t Σ. Consider iteration l. The proof again is the same as in the base case, except the proof that the intital $\mathcal{M}_{\mathsf{prev}}$ on Line 3 is a depleting module of $\mathcal{O}$ w.r.t Σ, comes from previous iteration by means of the induction hypothesis instead of Line 1.

□

As an example we examine a run of the algorithm in Figure 4.2 in which the produced hybrid STAR-AMEX-module is smaller than corresponding the STAR-module on its own.

Example 4.2.1. *Consider the general $\mathcal{ALCN}$ ontology Food, consisting of the axioms $\{\mathcal{F}_1 - \mathcal{F}_6\}$, which is a subset of an example OWL-DL ontology from the*

World Wide Web Consortium's (W3C) website [Gro04].

$$\mathsf{DessertCourse} \equiv \mathsf{MealCourse} \sqcap (\forall \mathsf{hasFood}.\mathsf{Dessert}) \qquad (\mathcal{F}_1)$$

$$\mathsf{SeafoodCourse} \equiv \mathsf{MealCourse} \sqcap (\forall \mathsf{hasFood}.\mathsf{Seafood}) \qquad (\mathcal{F}_2)$$

$$\mathsf{MealCourse} \sqsubseteq \forall \mathsf{hasFood}.\mathsf{EdibleThing} \qquad (\mathcal{F}_3)$$

$$\exists \mathsf{hasFood}.\top \sqsubseteq \mathsf{MealCourse} \qquad (\mathcal{F}_4)$$

$$\mathsf{EdibleThing} \sqcap \mathsf{MealCourse} \sqsubseteq \bot \qquad (\mathcal{F}_5)$$

$$\mathsf{Dessert} \sqcap \mathsf{Seafood} \sqsubseteq \bot \qquad (\mathcal{F}_6)$$

For the signature $\Sigma = \{\mathsf{hasFood}\}$ *we extract the module* $\mathcal{M} = {}^{STAR}_{AMEX}\text{-}mod(\mathsf{Food}, \Sigma)$

To begin with (on Line 1 of the algorithm) we extract the module $\mathcal{M} = STAR(\mathsf{Food}, \Sigma)$ *which can be verified to contain the entire ontology* $\mathcal{M} = \mathsf{Food}$.

- *Iteration 1:*
  - *AMEX (Line 6):*

    We split $\mathcal{M}_{\mathsf{prev}} = \mathsf{Food}$ *(Line 3) to obtain* $\mathcal{M}_{\mathcal{ALCQI}} = \{\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3\}$ *and* $\mathcal{M}_{\mathsf{rest}} = \{\mathcal{F}_4, \mathcal{F}_5, \mathcal{F}_6\}$ *and extract the module* $\mathcal{M}' = AMEX(\mathcal{M}_{\mathcal{ALCQI}}, \Sigma \cup \mathsf{sig}(\mathcal{M}_{\mathsf{rest}}))$ *to obtain the module* $\mathcal{M} = \mathcal{M}' \cup \mathcal{M}_{\mathsf{rest}}$.

    To do this we begin with $\mathcal{M}' = \emptyset$ *and find that **R1** of AMEX applies, that* $\mathcal{F}_3$ *causes a direct* $\Sigma \cup \mathsf{sig}(\mathcal{M}') \cup \mathsf{sig}(\mathcal{M}_{\mathsf{rest}})$*-dependency in* $\mathcal{M}_{\mathcal{ALCQI}} \setminus \mathcal{M}'$*, we have* $\mathsf{MealCourse} \in \Sigma \cup \mathsf{sig}(\mathcal{M}') \cup \mathsf{sig}(\mathcal{M}_{\mathsf{rest}})$ *and* $\mathsf{hasFood} \in \mathsf{depend}_{\mathcal{M}_{\mathcal{ALCQI}} \setminus \mathcal{M}'}(\mathcal{F}_3) \cap (\Sigma \cup \mathsf{sig}(\mathcal{M}') \cup \mathsf{sig}(\mathcal{M}_{\mathsf{rest}}))$ *so we locate the dependency chain* $\{\mathcal{F}_3\} \in \mathsf{chain}^{\mathsf{hasFood}}_{\mathcal{M}_{\mathcal{ALCQI}} \setminus \mathcal{M}'}(\mathcal{F}_3)$ *and set* $\mathcal{M}' = \mathcal{M}' \cup \{\mathcal{F}_3\}$*. At this point we find neither **R1** or **R2** are applicable and therefore* $\mathcal{M}'$ *is a depleting* $\Sigma \cup \mathsf{sig}(\mathcal{M}_{\mathsf{rest}})$ *module of* $\mathcal{M}_{\mathcal{ALCQI}}$.

    Rejoining the result with $\mathcal{M}_{\mathsf{rest}}$ *we obtain the module* $\mathcal{M} = \mathcal{M}' \cup \mathcal{M}_{\mathsf{rest}} = \{\mathcal{F}_3, \mathcal{F}_4, \mathcal{F}_5, \mathcal{F}_6\}$.
  - *STAR (Line 9):*

    Beginning with the module $\mathcal{M}_{\mathsf{prev}} = \{\mathcal{F}_3, \mathcal{F}_4, \mathcal{F}_5, \mathcal{F}_6\}$ *(Line 8) we extract the module* $\mathcal{M} = STAR(\mathcal{M}_{\mathsf{prev}}, \Sigma)$ *beginning by extracting a* $\bot$-

module.

  * $\mathcal{M}_\bot = \bot$-*mod*$(\mathcal{M}_{\mathsf{prev}}, \Sigma)$ *initially the only non-local axiom w.r.t* $\Sigma \cup \mathsf{sig}(\mathcal{M}_\bot)$ *is* $\mathcal{F}_3$ *so we set* $\mathcal{M}_\bot = \{\mathcal{F}_3\}$. *Now* $\Sigma \cup \mathsf{sig}(\mathcal{M}_\bot) = \{\mathsf{hasFood}, \mathsf{MealCourse}, \mathsf{EdibleThing}\}$ *which results in first* $\mathcal{F}_4$ *then* $\mathcal{F}_5$ *becoming non-local for* $\Sigma \cup \mathsf{sig}(\mathcal{M}_\bot)$ *so we set* $\mathcal{M}_\bot \cup \{\mathcal{F}_4, \mathcal{F}_5\}$. $\Sigma \cup \mathsf{sig}(\mathcal{M}_\bot)$ *is unchanged and no more axioms in* $\mathcal{M}_{\mathsf{prev}} \setminus \mathcal{M}_\bot$ *are detected as being non-local w.r.t* $\Sigma \cup \mathsf{sig}(\mathcal{M}_\bot)$ *and we obtain the module* $\mathcal{M}_\bot = \{\mathcal{F}_3, \mathcal{F}_4, \mathcal{F}_5\}$
  * $\mathcal{M}_\top = \top$-*mod*$(\mathcal{M}_\bot, \Sigma)$ *- does not improve any further on* $\mathcal{M}_\bot$*, one can verify that* $\mathcal{M}_\top = \mathcal{M}_\bot$.

Since the STAR procedure has reached a fixpoint we obtain the module $\mathcal{M} = \{\mathcal{F}_3, \mathcal{F}_4, \mathcal{F}_5\}$.

- *Iteration 2:*

  – *AMEX (Line 6): We begin with* $\mathcal{M}_{\mathsf{prev}} = \{\mathcal{F}_3, \mathcal{F}_4, \mathcal{F}_5\}$*, but cannot reduce the size of* $\mathcal{M}$ *any further by using AMEX. We find splitting* $\mathcal{M}_{\mathsf{prev}}$ *into* $\mathcal{M}_{\mathcal{ALCQI}}$ *and* $\mathcal{M}_{\mathsf{rest}}$ *that* $\mathcal{M}_{\mathcal{ALCQI}}$ *only contains a single axiom* $\mathcal{F}_3$ *and extracting* $\mathcal{M} = \mathit{AMEX}(\mathcal{M}_{\mathcal{ALCQI}}, \Sigma \cup \mathsf{sig}(\mathcal{M}_{\mathsf{rest}}))$ *we immediately find* $\mathcal{F}_3$ *has* $\Sigma \cup \mathsf{sig}(\mathcal{M}') \cup \mathsf{sig}(\mathcal{M}_{\mathsf{rest}})$*-dependency so must belong to* $\mathcal{M}'$*, and so obtain the module* $\mathcal{M}_{\mathit{AMEX}} = \mathcal{M}' \cup \mathcal{M}_{\mathsf{rest}} = \{\mathcal{F}_3, \mathcal{F}_4, \mathcal{F}_5\}$.

The hybrid STAR-AMEX procedure has then reached a fixpoint, so we are done, and obtain the depleting Σ*-module of* Food*:* $\mathcal{M} = {}^{\mathit{STAR}}_{\mathit{AMEX}}$*-mod*$(\mathcal{O}, \Sigma) = \{\mathcal{F}_3, \mathcal{F}_4, \mathcal{F}_6\}$*, which in this case coincides with the minimal depleting* Σ*-module of* Food*. Notice that the corresponding STAR-module for the is equivalent to initial STAR-module extracted on Line 1 and contains the entire ontology: STAR*$(\mathsf{Food}, \Sigma) = \mathsf{Food}$.

4.3. Splitting ontologies for AMEX

For the hybrid STAR-AMEX procedure to work in practice we must be able to split a STAR-module $\mathcal{M}$, which is an ontology up to $\mathcal{SROIQ}$ in expressivity, into two parts $\mathcal{M} = \mathcal{M}_{\mathcal{ALCQI}} \cup \mathcal{M}_{\mathsf{rest}}$ where $\mathcal{M}_{\mathcal{ALCQI}}$ is an acyclic $\mathcal{ALCQI}$ terminology with RCIs, as only then are we able to apply the AMEX procedure.

The approach we take to split the module is to begin with $\mathcal{M}_{\mathsf{rest}} = \emptyset$ then move axioms to $\mathcal{M}_{\mathsf{rest}}$ until $\mathcal{M}_{\mathcal{ALCQI}} = \mathcal{M} \setminus \mathcal{M}_{\mathsf{rest}}$ is an acyclic $\mathcal{ALCQI}$ terminology with RCIs. To achieve this we use the following heuristic:

Definition 4.3.1. *Move axioms from $\mathcal{M}$ to $\mathcal{M}_{\mathsf{rest}}$ in the following order:*

(M1) Move axioms to $\mathcal{M}_{\mathsf{rest}}$ until every axiom in $\mathcal{M}_{\mathcal{ALCQI}}$ is expressed in a logic $\mathcal{L} \subseteq \mathcal{ALCQI}$ and of the form $A \bowtie C$ where A a concept name

(M2) Move axioms to $\mathcal{M}_{\mathsf{rest}}$ until $\mathcal{M}_{\mathcal{ALCQI}}$ is a terminology with RCIs

(M3) Move axioms to $\mathcal{M}_{\mathsf{rest}}$ until $\mathcal{M}_{\mathcal{ALCQI}}$ is acyclic

Clearly if each of these points are achieved, $\mathcal{M}_{\mathcal{ALCQI}} = \mathcal{M} \setminus \mathcal{M}_{\mathsf{rest}}$ will be an acyclic $\mathcal{ALCQI}$ terminology with RCIs. To achieve (M1) it is always necessary to move axioms more expressive than $\mathcal{ALCQI}$ to $\mathcal{M}_{\mathsf{rest}}$, which can be achieved trivially by simply inspecting the structure and constructors an axiom utilises. (M2) and (M3) however require a bit more thought, as often we will be faced with a choice of axiom(s) to move which will affect how we split the original module.

For the AMEX procedure to be most effective, the size of $\mathcal{M}_{\mathcal{ALCQI}}$ should be as large as possible: firstly giving the specialised procedure a greater chance at identify axioms which do not belong to the minimal depleting module, secondly, as AMEX-modules must be depleting modules for the extended signature $\Sigma \cup \mathsf{sig}(\mathcal{M}_{\mathsf{rest}})$, if $\mathcal{M}_{\mathsf{rest}}$ is particularly large it becomes less likely for AMEX to discard axioms as being semantically irrelevant.

4.3.1 Moving non-terminological axioms

After (M1) is achieved, all axioms in $\mathcal{M}_{\mathcal{ALCQI}}$ must be of the form $A \bowtie C$ with A a concept name, however $\mathcal{M}_{\mathcal{ALCQI}}$ may still not be logically equivalent to terminology (a terminology with RCIs) which at most can contain one equivalence per concept name which unlike concept inclusions are not allowed to be repeated. An ontology which violates this condition is said to contain at least one *shared name*.

Definition 4.3.2 (Shared name)**.** *Let $\mathcal{O}$ be an ontology where every axiom is of the form $A \bowtie C$ with A a concept name $A \in \mathsf{N_C} \cap \mathsf{sig}(\mathcal{O})$. If there exists an equivalence $B \equiv D \in \mathcal{O}$ and also a distinct axiom $B \bowtie D' \in \mathcal{O}$ for some concept name B, B is called a* shared name*.*

If $\mathcal{M}$ contains shared names we have potentially two options of which axioms to move to $\mathcal{M}_{\mathsf{rest}}$ to ensure $\mathcal{M}_{\mathcal{ALCQI}} = \mathcal{M} \setminus \mathcal{M}_{\mathsf{rest}}$ is a terminology with RCIs. For each shared name A in $\mathcal{M}$, our first option is leave a single equivalence $A \equiv C$ in $\mathcal{M}_{\mathcal{ALCQI}}$ and move every other equivalence and concept inclusion sharing the same name to $\mathcal{M}_{\mathsf{rest}}$, second option is to leave every (possibly repeated) concept inclusion $A \bowtie C_1, A \bowtie C_2, \ldots, A \bowtie C_n$ in $\mathcal{M}_{\mathcal{ALCQI}}$ and move every equivalence sharing the same name to $\mathcal{M}_{\mathsf{rest}}$.

With the aim of keeping $\mathcal{M}_{\mathsf{rest}}$ as small as possible, our decision is to take the latter option, which is likely to be much more preferable in general. This decision follows from an analysis of the corpus of real-world ontologies we use in our experimental evaluation in Chapter 6, 82 of which contain at least one shared name after (M1) has been achieved. What we revealed is in 97.38% of cases where a concept name is shared, there exists just a single equivalence which shares a name with several concept inclusions. As a result, retaining the concept inclusions would in most cases moving just one equivalence to $\mathcal{M}_{\mathsf{rest}}$ for each shared name. The alternative, retaining the single equivalence, we found on average would consist of moving 23.5 concept inclusions to $\mathcal{M}_{\mathsf{rest}}$ for each shared name, but up to 368 in the worst case.

An exception we make to this decision is when we have an $\mathcal{M}$ in which a shared name is only shared between concept equivalences, then it is unnecessary to move every equivalence to $\mathcal{M}_{\mathsf{rest}}$ and we can retain one of these equivalences whilst still maintaining that $\mathcal{M}_{\mathcal{ALCQI}}$ as an $\mathcal{ALCQI}$ terminology with RCIs.

It is of course possible to find an example where by retaining a single equivalence rather than multiple concept inclusions for a shared name leads to smaller modules produced by AMEX (depending on the signatures of the axioms involved) but generally speaking, our choice of retaining the inclusions over that of the equivalences will help keep the size of $\mathcal{M}_{\mathsf{rest}}$ to a minimum.

4.3.2 Breaking terminological cycles

Once (M1) and (M2) are achieved $\mathcal{M}_{\mathcal{ALCQI}} = \mathcal{M} \setminus \mathcal{M}_{\mathsf{rest}}$ must be a terminology with RCIs which is at most $\mathcal{ALCQI}$ in expressitivity but still may contain cycles. There are in principle several ways to go about removing axioms to break cycles in a terminology, one such as applying the heuristic from [Ves+13]. However, under the scope of hybrid module extraction, we observe that regardless of how we break cycles within the terminology, every axiom which contributes to that cycle still ends up in the module we extract using AMEX.

To see why, consider this example of breaking a cycle in STAR-module for use with AMEX in the hybrid STAR-AMEX extraction procedure:

Example 4.3.1. *Given a STAR-module $\mathcal{M} = \{\mathcal{C}_1 - \mathcal{C}_4\}$*

$$D \sqsubseteq A \tag{$\mathcal{C}_1$}$$

$$A \sqsubseteq B \tag{$\mathcal{C}_2$}$$

$$B \sqsubseteq C \tag{$\mathcal{C}_3$}$$

$$C \sqsubseteq D \tag{$\mathcal{C}_4$}$$

Observe that $\mathcal{M}$ is cyclic, in this case for each axiom $X \bowtie Y \in \mathcal{T}_{\mathsf{RCI}}$, we have $X \in \mathsf{depend}_{\mathcal{T}_{\mathsf{RCI}}}(X \bowtie Y)$. Now we can break this cycle, making $\mathcal{M}_{\mathcal{ALCQI}} = \mathcal{M} \setminus \mathcal{M}_{\mathsf{rest}}$ acyclic, by moving at least one axiom involved in the cycle to $\mathcal{M}_{\mathsf{rest}}$. If we do this, as an example by moving $\mathcal{C}_3$ to $\mathcal{M}_{\mathsf{rest}}$, and then extract the module $\mathcal{M}' = \mathsf{AMEX\text{-}mod}(\mathcal{M}_{\mathcal{ALCQI}}, \Sigma \cup \mathsf{sig}(\mathcal{M}_{\mathsf{rest}}))$. Starting with $\mathcal{M}' = \emptyset$, immediately

R1 of AMEX applies since $C \in (\Sigma \cup \mathsf{sig}(\mathcal{M}_{\mathsf{rest}}))$ and $B \in \mathsf{depend}_{\mathcal{M}_{\mathcal{ALCQI}}}(\mathcal{C}_4) \cap (\Sigma \cup \mathsf{sig}(\mathcal{M}_{\mathsf{rest}}))$, and then we can identify the dependency chain $\gamma = \{\mathcal{C}_4, \mathcal{C}_1, \mathcal{C}_2\} \in \mathsf{chain}^B_{\mathcal{M}_{\mathcal{ALCQI}}}(\mathcal{C}_4)$ (the remaining axioms of the cycle), and so we set $\mathcal{M}' = \mathcal{M}' \cup \{\gamma\}$. Now to produce depleting Σ-module $\mathcal{M}$ of the original ontology we rejoin our module with $\mathcal{M}_{\mathsf{rest}}$, $\mathcal{M} = \mathcal{M}' \cup \mathcal{M}_{\mathsf{rest}}$, which now contains the entire cycle.

It can be verified that whichever combination of axioms are taken to break the cycle in this example, every axioms of the cycle will eventually end up in the extracted AMEX-module. This can also be observed in the general case, the relationship between the signature of axioms involved in a cycle, as well as the fact that we extract a module using AMEX for $\Sigma \cup \mathsf{sig}(\mathcal{M}_{\mathsf{rest}})$ means that however a cycle is broken we cannot avoid every axiom involved being pulled into our extracted module.

Identify cycle causing axioms

Rather than attempting to break cycles, we shift our task towards locating all axioms which belong to a cycle within a terminology so we can move them directly to $\mathcal{M}_{\mathsf{rest}}$. The most natural way of achieving this is to use the field of graph theory for which the presence and detection of cycles is a well understood.

We use the standard graph theoretical terminology, such as that which described in [BM07; GY05]. A directed graph is $G = (V, E)$ is a finite set of V vertices, and finite set of ordered pairs $E \subseteq V \times V$ of edges. A loop is an edge in a directed graph that joins a vertex to itself. A path is a finite sequence $\langle v_0, e_1, v_1, e_2, v_2, \ldots, v_{k-1}, e_{k-1}, v_k\rangle$ whose terms alternate vertices and edges such that for $1 \leq i \leq k$ the edge $e_i = (v_{i-1}, v_i)$, and where each internal vertex v_j, $1 \leq j \leq k-1$ is distinct. A vertex v has a cycle if there exists a non-trivial path (has at least one edge) from a v to itself. A directed graph which contains no cycles is called a directed acyclic graph (DAG). A vertex x is said to be reachable from a vertex y if there exists a path from x to y. A directed graph is called strongly connected if every vertex is reachable from every other vertex. The strongly connected components of a graph is the partition of a graph into its subgraphs induced by the equivalence relation of being strongly connected.

Input: Terminology with RCIs $\mathcal{T}_{\mathsf{RCI}}$

Output: Dependency graph G

$G = (V, E)$
`// Map each concept name to a vertex`
foreach $A \in (\mathsf{sig}(\mathcal{T}_{\mathsf{RCI}}) \cap \mathsf{N_C})$ **do**
    $V ::= V \cup \{v_A\}$
end
`// Directed edge between corresponding vertices if` $A \prec_{\mathcal{T}_{\mathsf{RCI}}} B$
foreach $A \bowtie C \in \mathcal{T}_{\mathsf{RCI}}$ **do**
    foreach $B \in \mathsf{sig}(C)$ **do**
        if $\{(v_A, v_B)\} \notin E$ **then**
            $E ::= E \cup \{(v_A, v_B)\}$
        end
    end
end
return G

Figure 4.3: Computing dependency graph

The relationship between the cycles of a terminology and the cycles of a graph can be observed by the inspection of the graph induced by $\prec_{\mathcal{T}_{\mathsf{RCI}}}$ (depends) relation between symbols of the terminology. The dependency graph of a terminology with RCIs is computed by associating each concept name in with a vertex, add adding an edge between vertices if $\prec_{\mathcal{T}_{\mathsf{RCI}}}$ relates their associated concept names together. Based on this definition we can compute a dependency graph using the algorithm in Figure 4.3.

Now given the dependency graph $G_{\mathcal{T}_{\mathsf{RCI}}}$ of a terminology $\mathcal{T}_{\mathsf{RCI}}$, notice that since the set $\mathsf{depend}_{\mathcal{T}_{\mathsf{RCI}}}$ is given by the transitive closure of the $\prec_{\mathcal{T}_{\mathsf{RCI}}}$ relation, for an axiom $A \bowtie C \in \mathcal{T}_{\mathsf{RCI}}$ it holds that $A \in \mathsf{depend}_{\mathcal{T}_{\mathsf{RCI}}}(A \bowtie C)$ if and only if the vertex v_A associated with A has a cycle in $G_{\mathcal{T}_{\mathsf{RCI}}}$. So towards being able to locate the axioms which cause a cycle in $\mathcal{T}_{\mathsf{RCI}}$ we first locate those vertices which have a cycle in $G_{\mathcal{T}_{\mathsf{RCI}}}$.

We can achieve this by observing a graph G only has a cycle if: 1. G contains a strongly connected component consisting of more than one vertex. 2. G contains a loop. If this is the case then there exists two distinct vertices which are mutually reachable from each other, and/or a vertex v with an edge, and therefore a path, which joins v to itself. It is also not hard to see the converse is also true, if neither of these conditions hold then G cannot contain a cycle. The axioms belonging to cycles in $\mathcal{T}_{\mathsf{RCI}}$ are then those axioms which introduce concepts for which the corresponding vertices in $G_{\mathcal{T}_{\mathsf{RCI}}}$ leads to either of these conditions holding.

In practice, the strongly connected components of a graph can be computed in linear time by one of several algorithms [Sha81; Tar72; Dij+76], and if we compute the strongly connected components of $G_{\mathcal{T}_{\mathsf{RCI}}}$ and consider each strongly connected component $\mathsf{SCC} \in G_{\mathcal{T}_{\mathsf{RCI}}}$ which contains more than one vertex, then by considering each vertex $v_A \in \mathsf{SCC}$ it is those axioms $A \bowtie C \in \mathcal{T}_{\mathsf{RCI}}$ with $B \in \mathsf{sig}(C)$ such that $A \neq B$ and $v_B \in \mathsf{SCC}$ which belong to a cycle in $\mathcal{T}_{\mathsf{RCI}}$. Why it is necessary to establish if we have $v_B \in \mathsf{SCC}$ in addition to v_A is that $\mathcal{T}_{\mathsf{RCI}}$ may contain RCIs, and it is possible that some inclusions for a shared name contribute to a cycle whilst others do not, and in the latter case these axioms should not be identified as belonging a cycle. Those axioms which lead to vertices with loops in the induced dependency graph are more trivial to find, one simply needs to locate those axioms $A \bowtie C \in \mathcal{T}_{\mathsf{RCI}}$ for which we have $A \in \mathsf{sig}(C)$.

As an example, consider a cyclic terminology and its induced dependency graph $G_{\mathcal{T}_{\mathsf{RCI}}}$ both of which are shown in Figure 4.4. On the dependency graph we have also highlighted all strongly connected components with more than one vertex, and those vertices which have loops i.e. those vertices which have cycles in $G_{\mathcal{T}_{\mathsf{RCI}}}$.

$$
\begin{array}{ll}
A \sqsubseteq B & (\mathcal{D}_1) \\
A \sqsubseteq \exists r.E & (\mathcal{D}_2) \\
B \sqsubseteq C \sqcap E & (\mathcal{D}_3) \\
C \equiv F \sqcap D & (\mathcal{D}_4) \\
D \sqsubseteq \forall r.A & (\mathcal{D}_5) \\
E \sqsubseteq G & (\mathcal{D}_6) \\
G \sqsubseteq H \sqcap I & (\mathcal{D}_7) \\
H \sqsubseteq \exists r.G & (\mathcal{D}_8) \\
X \sqsubseteq Y \sqcup Z & (\mathcal{D}_9) \\
W \equiv Z \sqcap \forall r.W & (\mathcal{D}_{10})
\end{array}
$$

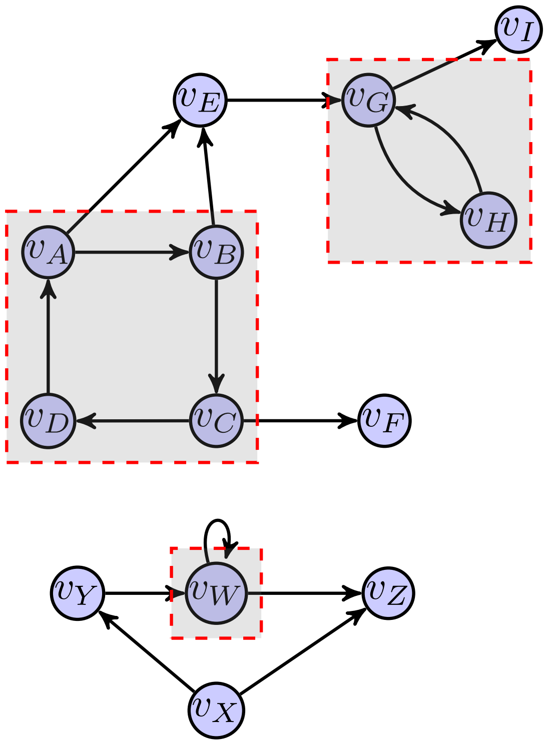


Figure 4.4: Detecting cycles of a terminology using a dependency graph

Now to identify those axioms with a cycle in $\mathcal{T}_{\mathsf{RCI}}$ we can examine highlighted vertices. From the strongly connected component consisting of the vertices $C_1 = \{v_A, v_B, v_C, v_D\}$, we can locate the axioms $\mathcal{D}_1, \mathcal{D}_3, \mathcal{D}_4$ and $\mathcal{D}_5$ as belonging to a cycle in $\mathcal{T}_{\mathsf{RCI}}$, that is for each axiom $A' \bowtie C' \in \{\mathcal{D}_1, \mathcal{D}_3, \mathcal{D}_4, \mathcal{D}_5\}$, there is a vertex $v_{A'} \in C_1$ and a distinct symbol $B' \in \mathsf{sig}(C')$ such that $v_{B'} \in C_1$. Notice this is not the case for $\mathcal{D}_2$, although it is an RCI for the repeated name A it does not belong to a cycle.

In a similar way to the first component, we find a second $C_2 = \{v_G, v_H\}$ and then can identify the two axioms $\mathcal{D}_7$ and $\mathcal{D}_8$ as belonging to a cycle. Finally, the vertex v_W has a loop in $G_{\mathcal{T}_{\mathsf{RCI}}}$, so we identify the axiom $\mathcal{D}_{10}$ as belonging to a cycle, which should be clear as we have $\mathsf{D}_{10} = W \equiv Z \sqcap \forall r.W$ and $W \in \mathsf{sig}(Z \sqcap \forall r.W)$. We have now considered all vertices with a cycle in $G_{\mathcal{T}_{\mathsf{RCI}}}$, and identified all axioms in $\mathcal{T}_{\mathsf{RCI}}$ which are responsible for them occurring, and can then verify this is all of the axioms which contribute to a cycle in $\mathcal{T}_{\mathsf{RCI}}$, indeed, the axioms $\{\mathcal{D}_2, \mathcal{D}_6, \mathcal{D}_{10}\}$ constitute an acyclic terminology.

Going back to the context of hybrid module extraction, we can verify whether or not $\mathcal{M}_{\mathcal{ALCQI}} = \mathcal{M} \setminus \mathcal{M}_{\mathsf{rest}}$ is cyclic by inspecting its associated dependency graph, and if we do find it is cyclic we can use the described methodology to locate all axioms involved in a cycle to move to $\mathcal{M}_{\mathsf{rest}}$, which in turn ensures that $\mathcal{M}_{\mathcal{ALCQI}}$ is acyclic for use with the AMEX procedure.

4.4. Conclusion

In this chapter we introduced a hybrid module extraction algorithm which exploits the methodologies different module depleting Σ-module extraction algorithms use, with the aim of minimising the size of our approximations. The hybrid algorithm extracts a depleting Σ-module from an ontology by generalising the approach the locality-based STAR approach uses, facilitating the iterative nested extraction of two input procedures until a fixpoint is reached, which results in those axioms which either input procedure deem semantically irrelevant over the input signature being discarded. The module produced using this method, under mild conditions, is at least as small as the corresponding module produced independently by either of the input procedures, but may be be even smaller.

Next we considered a modification of the hybrid algorithm to enable the combination of AMEX and STAR extraction algorithms without losing the generality that the STAR procedure provides in terms of ontologies it accepts as input. This still amounted to iterative nested extraction, but with an additional step which involved filtering axioms, allowing AMEX to work, and then rejoining the filtered axioms back on afterwards. This resulted in the production of the STAR-AMEX hybrid extraction procedure which is guaranteed to extract a module at least as small as the corresponding STAR module, but with the aim of using the specialised nature of the AMEX procedure to further reduce the size of the extracted modules. We will go on to evaluate modules produced by the hybrid STAR-AMEX extraction algorithm in Chapter 6.

In the next chapter, to estimate the success of a given approximation, we will be considering a methodology which allows us to estimate the difference in size between an approximation and the minimal module it approximates.

CHAPTER 5

How Good is an Approximation?

Driven by the undecidability of computing minimal modules in moderately expressive logics, we have introduced two new approximation procedures to compete with the already successful locality based modules: AMEX which can extract a depleting module from acyclic $\mathcal{ALCQI}$ terminologies with RCIs, and the hybrid STAR-AMEX procedure, introduce the previous chapter, and able to extract depleting modules from general cyclic $\mathcal{SROIQ}$ ontologies (the most expressive logic STAR itself supports). Each of these were developed with the aim of minimising the size of the module which is extracted, to better approximate minimal modules, and how successful an approximation is can be considered how close in size it comes to the corresponding minimal module.

The success of any approximation is currently an open problem; although the size of locality based modules, especially STAR modules, have been systematically analysed in great detail [Ves+12; Ves+13], and the size of extracted modules may be improved upon by utilising AMEX or the hybrid STAR-AMEX procedure, nothing is yet known about how large and significant the difference between a given approximation and a minimal depleting module actually is, so it is not known how well one can approximate minimal modules.

In this chapter we introduce a methodology which can help close this gap, so that for the first time we are able to evaluate how well one can approximate minimal modules. Those modules which are guaranteed to be depleting modules we can consider an *upper approximation*, the unique minimal depleting module is always contained within each of these modules. In the sections that follow, we also introduce a *lower approximation*, a module that is always contained in the minimal depleting module but is not guaranteed to be a depleting module itself. As minimal modules lie between the upper and lower approximations, we know that if the upper and lower approximation coincide (or come very close together) the upper approximation coincides (or comes close to) the minimal depleting module. We go on to consider how we might compute lower approximations which we achieve using a reduction to QBF.

5.1. Upper and lower approximations

Upper approximation

Any of the modules produced by the STAR, AMEX or hybrid STAR-AMEX procedures can be used as an upper approximation, each of which is guaranteed to extract a depleting Σ-module by Theorem 2.4.1, Theorem 3.4.3 and Theorem 4.2.1 respectively.

Lower approximation

For the lower approximation, we introduce a new depleting module notion based on a weakening of the model inseparability relation, where two ontologies are considered inseparable for a signature Σ if their models coincide on Σ for models constructed using only a chosen number of domain elements rather than requiring them to coincide on all models.

Definition 5.1.1 (n-inseparability). *Assume $\mathcal{T}_1$ and $\mathcal{T}_2$ are TBoxes. Then $\mathcal{T}_1$ and $\mathcal{T}_2$ are n-inseparable in symbols $\mathcal{T}_1 \equiv_{\Sigma}^{\leqslant n} \mathcal{T}_2$ if:*

$$\{\mathcal{I}|_\Sigma \mid \sharp\Delta^\mathcal{I} = 1 \ldots n \textit{ and } \mathcal{I} \models \mathcal{T}_1\} \;=\; \{\mathcal{I}|_\Sigma \mid \sharp\Delta^\mathcal{I} = 1 \ldots n \textit{ and } \mathcal{I} \models \mathcal{T}_2\}.$$

Recall we have already visited the specific case of 1-inseparability in Chapter 3 which by Theorem 3.3.1 gives Π_2^p upper bound of deciding if $\mathcal{T}_{\mathsf{RCI}} \equiv_\Sigma \emptyset$ where $\mathcal{T}_{\mathsf{RCI}}$ is an acyclic terminology with RCIs which is free of direct Σ-dependencies, which in turn enables the second rule of the AMEX procedure to decide if a subset of such a terminology is a depleting Σ-module.

Intuitively, two ontologies are inseparable over Σ, then they are n-inseparable over Σ. The following example shows the converse does not hold:

Example 5.1.1. *Given the signature* $\Sigma = \{\mathsf{PleuralTissue}, \mathsf{hasLocation}\}$ *and the TBox* $\mathcal{T} = \{\mathsf{PleuralTissue} \sqsubseteq \forall\mathsf{hasLocation}.\mathsf{ThoracicCavity}, \mathsf{ThoracicCavity} \sqsubseteq \exists\mathsf{hasLocation}.\mathsf{Thorax}\}$ *then* $\mathcal{T} \equiv_{\Sigma}^{\leqslant 1} \emptyset$ *but* $\mathcal{T} \not\equiv_{\Sigma} \emptyset$.

- ***Claim:*** $\mathcal{T} \equiv_{\Sigma}^{\leqslant 1} \emptyset$.
  Let $\mathcal{I}$ be any interpretation with $\Delta^{\mathcal{I}} = \{d_1\}$. We need to show there exists a model $\mathcal{J}$ of $\mathcal{T}$ such that $\mathcal{I}|_{\Sigma} = \mathcal{J}|_{\Sigma}$. To construct such a model $\mathcal{J}$, let $\Delta^{\mathcal{J}} = \Delta^{\mathcal{I}}$ and $\mathsf{PleuralTissue}^{\mathcal{J}} = \mathsf{PleuralTissue}^{\mathcal{I}}$ *and* $\mathsf{hasLocation}^{\mathcal{J}} = \mathsf{hasLocation}^{\mathcal{I}}$ *to ensure $\mathcal{I}|_{\Sigma} = \mathcal{J}|_{\Sigma}$. For the remaining symbols if* $\mathsf{PleuralTissue}^{\mathcal{J}} = \emptyset$ *then let* $\mathsf{ThoracicCavity}^{\mathcal{J}} = \mathsf{Thorax}^{\mathcal{J}} = \emptyset$. *If* $\mathsf{PleuralTissue}^{\mathcal{J}} = \{d\}$ *and* $\mathsf{hasLocation}^{\mathcal{J}} = \{(d, d)\}$ *set* $\mathsf{ThoracicCavity}^{\mathcal{J}} = \mathsf{Thorax}^{\mathcal{J}} = \{d\}$*, otherwise set* $\mathsf{ThoracicCavity}^{\mathcal{J}} = \mathsf{Thorax}^{\mathcal{J}} = \emptyset$. *One can verify $\mathcal{J}$ is a model of $\mathcal{T}$ and hence $\mathcal{T} \equiv_{\Sigma}^{1} \emptyset$*

- ***Claim:*** $\mathcal{T} \not\equiv_{\Sigma} \emptyset$.
  Let $\mathcal{I}$ be the following 2 element interpretation: $\Delta^{\mathcal{I}} = \{d_1, d_2\}$, $\mathsf{PleuralTissue}^{\mathcal{I}} = \{d_1\}$*,* $\mathsf{hasLocation}^{\mathcal{I}} = (d_1, d_2)$*,* $\mathsf{ThoracicCavity}^{\mathcal{I}} = \{d_2\}$. *We show there is no model $\mathcal{J}$ of $\mathcal{T}$ such that $\mathcal{I}|_{\Sigma} = \mathcal{J}|_{\Sigma}$. To try and construct $\mathcal{J}$. Let* $\mathsf{PleuralTissue}^{\mathcal{J}} = \mathsf{PleuralTissue}^{\mathcal{I}}$ *and* $\mathsf{hasLocation}^{\mathcal{J}} = \mathsf{hasLocation}^{\mathcal{I}}$ *to ensure $\mathcal{I}|_{\Sigma} = \mathcal{J}|_{\Sigma}$. For the remaining symbols, as* $d_1 \in \mathsf{PleuralTissue}^{\mathcal{J}}$ *for $\mathcal{J}$ to be a model of $\mathcal{T}$ we must have* $d_1 \in (\forall\mathsf{hasLocation}.\mathsf{ThoracicCavity})^{\mathcal{J}}$ *to achieve this let* $\mathsf{ThoracicCavity}^{\mathcal{J}} = \{d_2\}$. *But then we must have* $d_2 \in (\exists\mathsf{hasLocation}.\mathsf{Thorax})^{\mathcal{J}}$ *but since* $\mathsf{hasLocation}^{\mathcal{J}} = \{(d_1, d_2)\}$ *there is no interpretation of* $\mathsf{Thorax}^{\mathcal{J}}$ *which makes this possible with reinterpreting* hasLocation. *It should now be clear there is no model $\mathcal{J}$ of $\mathcal{T}$ such that $\mathcal{I}|_{\Sigma} = \mathcal{J}|_{\Sigma}$ and so $\mathcal{T} \not\equiv_{\Sigma} \emptyset$.*

Definition 5.1.2 (n-depleting Σ-module)**.** *Let $\mathcal{M}_{\leq n} \subseteq \mathcal{O}$ be ontologies,* *and n a positive integer. Then $\mathcal{M}_{\leq n}$ is an n-depleting Σ-module of $\mathcal{O}$ if $\mathcal{O} \setminus \mathcal{M}_{\leq n} \equiv^{\leq n}_{\Sigma \cup \mathsf{sig}(\mathcal{M}_{\leq n})} \emptyset$.*

The following theorem proves the unique minimal n-depleting Σ-module is always contained in the minimal depleting Σ-module and therefore can be used as a lower approximation. The existence of an unique minimal n-depleting module itself is proven later by means of Lemma 5.2.4.

Theorem 5.1.1. *Let $\mathcal{O}$ be an ontology, Σ a signature, and n a positive integer. Let $\mathcal{M}$ be the unique minimal depleting Σ-module of $\mathcal{T}$, and let $\mathcal{M}_{\leq n}$ be the unique minimal n-depleting Σ-module of $\mathcal{O}$. Then $\mathcal{M}_{\leq n} \subseteq \mathcal{M}$.*

Proof. Notice since $\mathcal{M}$ is a depleting Σ-module of $\mathcal{O}$, it is also an n-depleting Σ-module of $\mathcal{O}$. Indeed, since $\mathcal{O} \setminus \mathcal{M} \equiv_{\Sigma \cup \mathsf{sig}(\mathcal{M})} \emptyset$, for *all* interpretations $\mathcal{I}$ there exists a model $\mathcal{J}$ of $\mathcal{O} \setminus \mathcal{M}$ such that $\mathcal{I}|_{\Sigma \cup \mathsf{sig}(\mathcal{M})} = \mathcal{J}|_{\Sigma \cup \mathsf{sig}(\mathcal{M})}$ which includes each $\mathcal{I}$ of size 1 through n. Now since $\mathcal{M}_{\leq n}$ is the unique minimal n-depleting Σ-module, $\mathcal{M}_{\leq n} \subseteq \mathcal{M}$ immediately follows. $\square$

Ideally one wants to compute an n-depleting module which is an accurate approximation, coinciding or being very close in size to the minimal depleting module. This means one must select an appropriate value for n for which to compute an n-depleting module. Increasing the value for n *may* increase the size of an n-depleting module, as $\mathcal{M}_{\leq n}$ may need to contain more axioms in order for $\mathcal{O} \setminus \mathcal{M}_{\leq n}$ to be inseparable from the empty ontology over an increased number of interpretations. For example take the TBox and signature from Example 5.1.1, the minimal 1-depleting module is empty, whereas the minimal 2-depleting module contains the entire TBox and therefore coincides with the minimal depleting Σ-module. We also note that increasing the value for n cannot result in the lower approximation becoming smaller, as by definition each minimal n-depleting module is also an m-depleting module for $1 \leq m < n$, so minimal n-depleting modules for increasing values of n is a sequence $\mathcal{M} \supseteq \mathcal{M}_n \supseteq \mathcal{M}_{n-1} \supseteq \cdots \supseteq \mathcal{M}_2 \supseteq \mathcal{M}_1$, where $\mathcal{M}$ is the minimal depleting Σ-module.

Even for increasingly large values for n there is no guarantee an n-depleting module will eventually coincide with the minimal Σ-module $\mathcal{M}$, as there may exist a larger finite or even infinite interpretation for which there is no model of $\mathcal{O} \setminus \mathcal{M}$ which coincides on $\Sigma \cup \mathsf{sig}(\mathcal{M})$. Moreover, we cannot even decide if an n-depleting module coincides with the minimal. For logics where approximations are required, this would amount to checking if some subset of an ontology is a minimal depleting Σ-module which is of course undecidable. Indeed, the only time we can know for certain if an n-depleting module coincides with the minimal depleting Σ-module — apart from checking "by hand" – is when the upper and lower approximations coincide. Even this considered, the size of the gap between the lower and upper approximations is still a strong indication to the success of an upper approximation.

5.2. Computing the lower approximation

Rather than choosing a fixed value for n, we can compute the lower approximation incrementally, by first computing a 1-depleting module then a 2-depleting module and so on until we have considered interpretations up to a certain size or the lower and upper approximations come sufficiently close together to give a good estimation of the success of the upper approximation. This also makes sense algorithmically, and can avoid wasted computation. If we compute a minimal n-depleting module incrementally and find some value of n where the upper and lower approximations coincide, it would be unnecessary to considering any subsequent values of n.

Definition 5.2.1 (Exactly n-inseparability). *Assume $\mathcal{O}_1$ and $\mathcal{O}_2$ are TBoxes. Then $\mathcal{O}_1$ and $\mathcal{O}_2$ are* exactly *n-Σ-inseparable in symbols $\mathcal{O}_1 \equiv^n_\Sigma \mathcal{O}_2$ if:*

$$\{\mathcal{I}|_\Sigma \mid \sharp\Delta^\mathcal{I} = n \text{ and } \mathcal{I} \models \mathcal{O}_1\} \;=\; \{\mathcal{I}|_\Sigma \mid \sharp\Delta^\mathcal{I} = n \text{ and } \mathcal{I} \models \mathcal{O}_2\}.$$

Exactly n-inseparability represents a further weakening of model inseparability relation in which the conditions for two ontologies to be inseparable only requires their n-element models to coincide on Σ, rather than models *up to* n in size as is the case for n-inseparability. As a result of this weakening we find that, unlike n-inseparability, when two ontologies are exactly n-inseparable they are not necessarily exactly m-inseparable for $1 \leq m < n$. To see this consider the following example:

Example 5.2.1. *Given the TBox* $\mathcal{T} = \{\mathsf{Human} \sqsubseteq \exists \mathsf{eats}.\mathsf{Meat}, \mathsf{Human} \sqsubseteq \exists \mathsf{eats}.\neg \mathsf{Meat}\}$ *and the signature* $\Sigma = \{\mathsf{Human}\}$ *we have* $\mathcal{T} \equiv^2_\Sigma \emptyset$ *but* $\mathcal{T} \not\equiv^1_\Sigma \emptyset$

- ***Claim:*** $\mathcal{T} \equiv^2_\Sigma \emptyset$.

  To see this let $\mathcal{I}$ *be an arbitrary interpretation with* $\Delta^\mathcal{I} = \{d_1, d_2\}$ *and construct a interpretation* $\mathcal{J}$ *of* $\mathcal{T}$ *as follows: To ensure* $\mathcal{I}|_\Sigma = \mathcal{J}|_\Sigma$ *let* $\Delta^\mathcal{J} = \Delta^\mathcal{I}$ *and* $\mathsf{Human}^\mathcal{J} = \mathsf{Human}^\mathcal{I}$*. For the remaining symbols let* $\mathsf{eats}^\mathcal{J} = \{(d_1, d_1), (d_2, d_1), (d_1, d_2), (d_2, d_2)\}$ *and* $\mathsf{Meat}^\mathcal{J} = \{d_1\}$ *then* $\Delta^\mathcal{J} = (\exists \mathsf{eats}.\mathsf{Meat})^\mathcal{J} = (\exists \mathsf{eats}.\neg \mathsf{Meat})^\mathcal{J}$*. One can now verify* $\mathcal{J}$ *is a model of* $\mathcal{T}$ *and so* $\mathcal{T} \equiv^2_\Sigma \emptyset$.

- ***Claim:*** $\mathcal{T} \not\equiv^1_\Sigma \emptyset$.

  To see this let $\mathcal{I}$ *be the following* 1 *element interpretation:* $\Delta^\mathcal{I} = \{d_1\}$*,* $\mathsf{Human}^\mathcal{I} = \{d_1\}$*,* $\mathsf{eats}^\mathcal{I} = \{(d_1, d_1)\}$ *and* $\mathsf{Meat}^\mathcal{I} = \{d_1\}$*. We show there is no model* $\mathcal{J}$ *of* $\mathcal{T}$ *such that* $\mathcal{I}|_\Sigma = \mathcal{J}|_\Sigma$*. To try and construct* $\mathcal{J}$*, let* $\Delta^\mathcal{J} = \Delta^\mathcal{I}$ *and* $\mathsf{Human}^\mathcal{J} = \mathsf{Human}^\mathcal{J}$ *to ensure* $\mathcal{I}|_\Sigma = \mathcal{J}|_\Sigma$*, but for* $\mathcal{J}$ *to be a model of* $\mathcal{T}$ *we must have* $d_1 \in (\exists \mathsf{eats}.\mathsf{Meat})^\mathcal{J}$ *so let* $\mathsf{eats}^\mathcal{J} = \{(d_1, d_1)\}$ *and* $\mathsf{Meat} = \{d_1\}$ *but now we must also have* $d_1 \in \neg \mathsf{Meat}$ *which is impossible. It should now be obvious that there is no* 1*-element model* $\mathcal{J}$ *of* $\mathcal{T}$ *such that* $\mathcal{I}|_\Sigma = \mathcal{J}|_\Sigma$ *and so* $\mathcal{T} \not\equiv^1_\Sigma \emptyset$.

Lemma 5.2.1. *Exactly* n*-inseparability is a monotone separability relation*

Proof. By Definition 2.3.5 it suffices to show the following:

- $(\mathcal{M}_\Sigma)$ If $\mathcal{O}_1 \equiv^n_\Sigma \mathcal{O}_2$ then $\mathcal{O}_1 \equiv^n_{\Sigma'} \mathcal{O}_2$ for all $\Sigma' \subseteq \Sigma$
- $(\mathcal{M}_\mathcal{O})$ If $\mathcal{O}_1 \subseteq \mathcal{O}_2 \subseteq \mathcal{O}_3$ and $\mathcal{O}_1 \equiv^n_\Sigma \mathcal{O}_3$ then $\mathcal{O}_1 \equiv^n_\Sigma \mathcal{O}_2$

1. $(\mathcal{M}_\Sigma)$. Let $\mathcal{O}_1$ and $\mathcal{O}_2$ be ontologies and $\Sigma' \subseteq \Sigma$ be signatures and assume $\mathcal{O}_1 \equiv^n_\Sigma \mathcal{O}_2$ for a positive integer n. To show $\mathcal{O}_1 \equiv^n_{\Sigma'} \mathcal{O}_2$ let $\mathcal{I}$ be an n-element model of $\mathcal{O}_1$, we have to show there is an n-element model $\mathcal{J}$ of $\mathcal{O}_2$ such that $\mathcal{I}|_{\Sigma'} = \mathcal{J}|_{\Sigma'}$. Since $\mathcal{O}_1 \equiv^n_\Sigma \mathcal{O}_2$ there is an n-element model $\mathcal{J}$ of $\mathcal{O}_2$ such that $\mathcal{I}|_\Sigma = \mathcal{J}|_\Sigma$, and since $\Sigma' \subseteq \Sigma$, $\mathcal{J}$ is an also n-element model of $\mathcal{O}_2$ that coincides with $\mathcal{I}$ on Σ'.

   □

2. $(\mathcal{M}_\mathcal{O})$. Let $\mathcal{O}_1 \subseteq \mathcal{O}_2 \subseteq \mathcal{O}_3$ be ontologies and Σ a signature. Assume that $\mathcal{O}_1 \equiv^n_\Sigma \mathcal{O}_3$. To show that $\mathcal{O}_1 \equiv^n_\Sigma \mathcal{O}_2$, let $\mathcal{I}$ be an n-element model of $\mathcal{O}_1$ we need to show there exists an n-element model $\mathcal{J}$ of $\mathcal{O}_2$ such that $\mathcal{I}|_\Sigma = \mathcal{J}|_\Sigma$. By $\mathcal{O}_1 \equiv^n_\Sigma \mathcal{O}_3$ there exists an n-element model $\mathcal{J}$ of $\mathcal{O}_3$ such that $\mathcal{I}|_\Sigma = \mathcal{J}|_\Sigma$ but since $\mathcal{O}_2 \subseteq \mathcal{O}_3$ it holds that $\mathcal{J}$ is model of $\mathcal{O}_2$ with $\mathcal{I}|_\Sigma = \mathcal{J}|_\Sigma$.

   □

Lemma 5.2.2. *Exactly n-inseparability is robust under replacements.*

Proof. Let $\mathcal{O}$, $\mathcal{O}_1$ and $\mathcal{O}_2$ be ontologies and Σ a signature, and n a positive integer. By Definition 2.3.6 it suffices to show the following: If $\mathcal{O}_1 \equiv^n_\Sigma \mathcal{O}_2$ and $\mathsf{sig}(\mathcal{O}) \cap \mathsf{sig}(\mathcal{O}_1 \cup \mathcal{O}_2) \subseteq \Sigma$ then $\mathcal{O}_1 \cup \mathcal{O} \equiv^n_\Sigma \mathcal{O}_2 \cup \mathcal{O}$

Assume that $\mathcal{O}_1 \equiv^n_\Sigma \mathcal{O}_2$ and $\mathsf{sig}(\mathcal{O}) \cap \mathsf{sig}(\mathcal{O}_1 \cup \mathcal{O}_2) \subseteq \Sigma$. To show that $\mathcal{O}_1 \cup \mathcal{O} \equiv^n_\Sigma \mathcal{O}_2 \cup \mathcal{O}$, let $\mathcal{I}$ be a model of $\mathcal{O}_1 \cup \mathcal{O}$ with $|\Delta^\mathcal{I}| = n$. We need to show there is a model $\mathcal{J}$ of $\mathcal{O}_2 \cup \mathcal{O}$ such that $\mathcal{I}|_\Sigma = \mathcal{J}|_\Sigma$. Since by the original assumption $\mathcal{O}_1 \equiv^n_\Sigma \mathcal{O}_2$ there exists a model $\mathcal{J}$ of $\mathcal{O}_2$ with $|\Delta^\mathcal{J}| = n$ such that $\mathcal{I}|_\Sigma = \mathcal{J}|_\Sigma$. Now we can assume w.l.o.g that additionally $\mathcal{J}$ coincides with $\mathcal{I}$ on all symbols that are not in $\mathsf{sig}(\mathcal{O}_2)$, and since $\mathsf{sig}(\mathcal{O}) \cap \mathsf{sig}(\mathcal{O}_2) \subseteq \Sigma$, $\mathcal{J}$ coincides on with $\mathcal{I}$ on all symbols in $\mathsf{sig}(\mathcal{O})$. It then follows that $\mathcal{J}$ is also a model of $\mathcal{O}$, as required. □

Lemma 5.2.3. *There exists a unique minimal exactly n-depleting Σ-module.*

Proof. Since by Lemma 5.2.2 and Lemma 5.2.1 exactly n-inseparability is a monotone relation which is robust under replacements the proof immediately follows from Proposition 2.3.2. □

Definition 5.2.2 (exactly n-depleting Σ-module). *Let $\mathcal{M}_n \subseteq \mathcal{O}$ be ontologies then $\mathcal{M}_n$ is an exactly n-depleting Σ-module of $\mathcal{O}$ if $\mathcal{O} \setminus \mathcal{M}_n \equiv^n_{\Sigma \cup \mathsf{sig}(M)} \emptyset$.*

Exactly n-depleting Σ-modules are simply depleting modules defined over the exactly n-inseparability relation. Unlike minimal n-depleting modules, minimal exactly n-depleting modules are not guaranteed to be subsets of each other, following exactly n-inseparability does not guaranteed m-inseparability where $m \neq n$. For example take the TBox and signature from Example 5.2.1, one can verify the minimal exactly 2-depleting module is empty, whereas the minimal exactly 1-depleting module contains the entire TBox.

Lemma 5.2.4. *Let $\mathcal{O}$ be an ontology and Σ a signature. Then $\mathcal{M}_{\leq n} = \bigcup\limits_{i}^{n} \mathcal{M}_i$ is the unique minimal n-depleting module of $\mathcal{O}$ where $\mathcal{M}_i$ is the unique minimal exactly i-depleting module of $\mathcal{O}$.*

Proof. By definition $\mathcal{M}_{\leq n}$ is comprised of unique and minimal exactly depleting modules so it follow that $\mathcal{M}_{\leq n}$ itself is minimal and uniquely determined.

It remains to be shown that $\mathcal{M}_{\leq n}$ is an n-depleting module. For a proof by contradiction assume $\mathcal{M}_{\leq n}$ is *not* an n-depleting Σ-module. Then there must exist an $\alpha \in \mathcal{O} \setminus \mathcal{M}_{\leq n}$ such that $(\mathcal{O} \setminus \mathcal{M}_{\leq n}) \cup \{\alpha\} \not\equiv^{\leq n}_{\Sigma \cup \mathsf{sig}(\mathcal{M})} \emptyset$, which by definition means there exists a j for $1 \leq j \leq n$ for which $(\mathcal{O} \setminus \mathcal{M}_{\leq n}) \cup \{\alpha\} \not\equiv^{j}_{\Sigma \cup \mathsf{sig}(\mathcal{M})} \emptyset$. Now consider $\mathcal{M}_j$ the minimal exactly j-depleting Σ-module of $\mathcal{O}$, which by definition of $\mathcal{M}_{\leq n}$ we have $\alpha \notin \mathcal{M}_j$ but then by monotonicity $(\mathcal{O} \setminus \mathcal{M}_j) \cup \{\alpha\} \equiv^{j}_{\Sigma \cup \mathsf{sig}(\mathcal{M})} \emptyset$ and then since $\mathcal{M}_j \subseteq \mathcal{M}_{\leq n}$ it follows that $(\mathcal{O} \setminus \mathcal{M}_{\leq n}) \cup \{\alpha\} \equiv^{j}_{\Sigma \cup \mathsf{sig}(\mathcal{M})} \emptyset$ which is a contradiction.

□

Lemma 5.2.4 gives us a way of achieving the incremental computation of the lower approximation. Beginning by computing the minimal 1-depleting module (equivalently minimal exactly 1-depleting module) we can compute the minimal 2-depleting module by combining this result with the minimal 2-depleting module, and can continue in this fashion for successive values of n for as long as desired to produce the minimal n-depleting module.

5.3. Deciding exactly n-inseparability from the empty ontology

In order to compute exactly n-depleting Σ-modules, towards being able to compute the lower approximation, we must be able to decide if for an ontology $\mathcal{O}$, signature Σ, and positive integer n if $\mathcal{O} \equiv^n_\Sigma \emptyset$. In contrast to deciding Σ-inseparability, in logics where approximations are required, checking if $\mathcal{O} \equiv^n_\Sigma \emptyset$ for a fixed n is decidable, as one can enumerate every interpretation which has a finite domain. Our approach is to generalise the reduction to QBF for deciding if $\mathcal{T} \equiv^1_\Sigma \emptyset$ from [Kon+13] where $\mathcal{T}$ is an acyclic $\mathcal{ALCI}$ terminology, extended to acyclic $\mathcal{ALCQI}$ with RCIs in Lemma 3.3.1, which in both cases gave a Π^p_2 upper bound of deciding 1-inseparability from the empty ontology.

The reduction covers ontologies up to $\mathcal{SRIQ}$ in expressitivity and entails constructing a closed QBF formula which is logically true iff $\mathcal{O} \equiv^n_\Sigma \emptyset$. To begin, we construct a propositional formula from the axioms of $\mathcal{O}$ so that the formula's possible truth assignments mirror the possible interpretations of $\mathcal{O}$ in an n-element interpretation. We will make a number of claims about how the construction of this propositional formula mirrors the possible assignments of an n-element interpretation, all of which are proven through Theorem 5.3.1.

Let $\mathcal{O}$ be a $\mathcal{SRIQ}$ ontology, Σ a signature, and consider an arbitrary n-element interpretation $\mathcal{I}_n = \{d_1, d_2, \ldots, d_n\}$ and a truth assignment $\mathfrak{v}$. Propositional atoms are used to represent when a domain element is interpreted as belonging to a concept or role name under $\mathcal{I}_n$. For every domain element $d \in \Delta^{\mathcal{I}_n}$ we introduce a fresh propositional variable P_{A_d} where $\mathfrak{v}(P_{A_d}) = \mathit{true}$ exactly when $d \in A^{\mathcal{I}_n}$ for every $A \in \Sigma \cap \mathsf{N_C}$, and a distinct fresh variable Q_{A_d}

for those concept names $A \in (\mathsf{sig}(\mathcal{O}) \setminus \Sigma) \cap \mathsf{N_C}$. Similarly for role names, for all $(d,e) \in \Delta^{\mathcal{I}_n} \times \Delta^{\mathcal{I}_n}$ we use a fresh variable $P_{r_{(d,e)}}$ where $\mathfrak{v}(P_{r_{(d,e)}}) = true$ exactly when $(d,e) \in r^{\mathcal{I}_n}$ for $r \in \Sigma \cap \mathsf{N_R}$ and for those role names $r \in (\mathsf{sig}(\mathcal{T}) \setminus \Sigma) \cap \mathsf{N_R}$ a distinct variable $Q_{r_{(d,e)}}$ is used.

$$
\begin{array}{rcl}
[r]_{(d,e)} & = & P_{r_{(d,e)}} \text{ for all } r \in \Sigma \cap \mathsf{N_R} \\
{[r]}_{(d,e)} & = & Q_{r_{(d,e)}} \text{ for all } r \in (\mathsf{sig}(\mathcal{T}) \setminus\ \Sigma) \cap \mathsf{N_R} \\
{[r^-]}_{(d,e)} & = & [r]_{(e,d)} \\
{[r \circ s]}_{(d,e)} & = & \bigvee\limits_{f \in \Delta^{\mathcal{J}_n}} [r]_{(d,f)} \wedge [s]_{(f,e)}
\end{array}
$$

Figure 5.1: Translation of roles to propositional formulas

$$
\begin{array}{rcl}
[A]_d & = & P_{A_d} \text{ for all } A \in \mathsf{N_C} \cap \Sigma \\
{[A]}_d & = & Q_{A_d} \text{ for all } A \in (\mathsf{sig}(\mathcal{T}) \setminus\ \Sigma) \cap \mathsf{N_C} \\
{[\top]}_d & = & true \\
{[\bot]}_d & = & false \\
{[C_1 \sqcap C_2]}_d & = & [C_1]_d \wedge [C_2]_d \\
{[\neg C]}_d & = & \neg([C]_d) \\
{[(\geq m\ r.C)]}_d & = & \begin{cases} m = 0 & true \\ n \geq m > 0 & \bigvee\limits_{s \in \mathsf{Rsets}(m)} \bigwedge\limits_{e \in s} [r]_{(d,e)} \wedge [C]_e \\ m > n & false \end{cases}
\end{array}
$$

Where $\mathsf{Rsets}(m)$ is the set of all subset size m taken from elements of $\Delta^{\mathcal{J}_n}$
Example for Δ_3 the set

$$\mathsf{Rsets}(2) = \{\{d_1, d_2\}\{d_1, d_3\}\{d_2, d_3\}\}$$

Figure 5.2: Translation of concepts to propositional formulas

We inductively extend this notion in order to represent complex role expressions R by propositional formulas. We use the notation $[R]_{(d,e)}$ to represent the evaluation of R under domain elements $(d,e) \in \Delta^{\mathcal{I}_n} \times \Delta^{\mathcal{I}_n}$ where $\mathfrak{v}([R]_{(d,e)}) = true$ exactly when $(d,e) \in R^{\mathcal{I}_n}$. For evaluating an arbitrary $\mathcal{SRIQ}$ role expression R under arbitrary domain elements $(d,e) \in \Delta^{\mathcal{I}_n} \times \Delta^{\mathcal{I}_n}$ we use the translation shown in Figure 5.1.

By utilising the translation of role expressions we further extend this notion to translation arbitrary concept expressions C. We use the notation $[C]_d$ to represent the evaluation of C under an arbitrary domain element $d \in \Delta^{\mathcal{I}_n}$, where $\mathfrak{v}([C]_d) = true$ exactly when $d \in C^{\mathcal{I}_n}$. The translation for evaluating arbitrary $\mathcal{SRIQ}$ concept expressions under arbitrary domain elements $d \in \Delta^{\mathcal{I}_n}$ is shown in Figure 5.2.

With this in place, we can reduce satisfiability of arbitrary $\mathcal{SRIQ}$ axioms under $\mathcal{I}_n$ to the validity of propositional formulas. We define a mapping:

$$\dagger : \mathcal{SRIQ} \text{ axiom} \rightarrow \text{propositional formula}$$

Where given $\mathcal{I}_n$ and an axiom $\alpha \in \mathcal{O}$ it holds that $\mathfrak{v}(\alpha^\dagger) = true$ iff $\mathcal{I}_n \models \alpha$. The proposed translation of axioms for a $\mathcal{SRIQ}$ TBox and RBox can be found in Figure 5.3 and Figure 5.4 respectively.

$$\begin{aligned}
(C \sqsubseteq D)^\dagger &= \bigwedge_{d \in \Delta^{\mathcal{I}_n}} [C]_d \rightarrow [D]_d \\
(C \equiv D)^\dagger &= (C \sqsubseteq D)^\dagger \wedge (D \sqsubseteq C)^\dagger
\end{aligned}$$

Figure 5.3: Translation of TBox axioms into propositional formulas

We can now establish satisfiability of an axiom under an arbitrary n-element interpretation by checking if its propositional translation is valid under some valuation assignment $\mathfrak{v}$. But to be able to decide if for an $\mathcal{SRIQ}$ ontology $\mathcal{O}$ and signature Σ if $\mathcal{O} \equiv^n_\Sigma \emptyset$, by definition we must be able to decide if for *all* n-element interpretations $\mathcal{I}_n$ there exists an interpretation $\mathcal{J}_n$ such that $\mathcal{I}_n|_\Sigma = \mathcal{J}_n|_\Sigma$ where $\mathcal{J}_n \models \mathcal{O}$. To do this construct a QBF formula

$$\psi_\mathcal{O} := \forall \vec{p} \exists \vec{q} \bigwedge_{\alpha \in \mathcal{O}} \alpha^\dagger$$

where $\vec{p} = \{[P]_d \mid d \in \Delta^{\mathcal{I}_n} \text{ and } P \in \Sigma\}$ and $\vec{q} = \{[Q]_d \mid d \in \Delta^{\mathcal{I}_n}$ and $Q \in \mathsf{sig}(\mathcal{O}) \setminus \Sigma\}$. Intuitively, we construct $\vec{p}$ to consist of atomic formulas that

$$
\begin{array}{rcl}
(r \sqsubseteq s)^\dagger & = & \bigwedge\limits_{(d,e)\in\Delta^{\mathcal{J}_n}\times\Delta^{\mathcal{J}_n}} [r]_{(d,e)} \rightarrow [s]_{(d,e)} \\
(r \equiv s)^\dagger & = & (r \sqsubseteq s)^\dagger \wedge (s \sqsubseteq r)^\dagger \\
(r_1 \circ r_2 \circ \cdots \circ r_n \sqsubseteq r) & = & \bigwedge\limits_{(d,e)\in\Delta^{\mathcal{J}_n}\times\Delta^{\mathcal{J}_n}} [r_1 \circ r_2 \circ \cdots \circ r_n]_{(d,e)} \rightarrow [r]_{(d,e)} \\
\mathsf{Disj}(r,s)^\dagger & = & \bigwedge\limits_{(d,e)\in\Delta^{\mathcal{J}_n}\times\Delta^{\mathcal{J}_n}} [r]_{(d,e)} \rightarrow \neg[s]_{(d,e)} \\
\mathsf{Symm}(r)^\dagger & = & \bigwedge\limits_{\substack{(d,e)\in\Delta^{\mathcal{J}_n}\times\Delta^{\mathcal{J}_n}\\ d\neq e}} [r]_{(d,e)} \leftrightarrow [r^-]_{(d,e)} \\
\mathsf{Asymm}(r)^\dagger & = & \bigwedge\limits_{(d,e)\in\Delta^{\mathcal{J}_n}\times\Delta^{\mathcal{J}_n}} \begin{cases} d = e & \neg[r]_{(d,e)} \\ d \neq e & [r]_{(d,e)} \leftrightarrow \neg[r^-]_{(d,e)} \end{cases} \\
\mathsf{Refl}(r)^\dagger & = & \bigwedge\limits_{d\in\Delta^{\mathcal{J}_n}} [r]_{(d,d)} \\
\mathsf{Irrefl}(r)^\dagger & = & \bigwedge\limits_{d\in\Delta^{\mathcal{J}_n}} \neg[r]_{(d,d)} \\
\mathsf{Trans}(r)^\dagger & = & \bigwedge\limits_{\substack{(d,e)\in\Delta^{\mathcal{J}}\times\Delta^{\mathcal{J}}\\ (e,e')\in\Delta^{\mathcal{J}}\times\Delta^{\mathcal{J}}\\ d\neq e, e\neq e'}} [r]_{(d,e)} \wedge [r]_{(e,e')} \rightarrow [r]_{(d,e')}
\end{array}
$$

Figure 5.4: Translation of RBox axioms into propositional formulas

represent all possible ways of interpreting a symbol from Σ under an n-element interpretation, and similarly for $\vec{q}$ but for those symbols which do not belong to Σ. Notice that by quantifying over all signature and non-signature symbols that $\psi_{\mathcal{O}}$ a closed formula and has precisely one truth assignment and it is $true$ if satisfiable under this assignment and $false$ if it is not.

Theorem 5.3.1. *Let $\mathcal{O}$ be a $\mathcal{SRIQ}$ ontology and Σ a signature. Then $\mathcal{O} \equiv^n_\Sigma \emptyset$ iff the QBF formula $\psi_{\mathcal{O}} := \forall\vec{p}\exists\vec{q} \bigwedge_{\alpha\in\mathcal{O}} \alpha^\dagger$ is true.*

Proof. Suppose that $\mathcal{O} \equiv^n_\Sigma \emptyset$. We now show that $\psi_{\mathcal{O}}$ is true. Consider an arbitrary assignment I of truth values to propositions in $\vec{p}$. We now must show there exists an assignment J to the truth values in $\vec{q}$ such that the propositional formula $\mathcal{O}^\dagger$ is *true* under the assignment $I \cup J$.

Define a n-element interpretation $\mathcal{I}_n$ as follows:

- $\Delta^{\mathcal{I}_n} = \{d_1 \ldots d_n\}$

- $A^{\mathcal{I}_n} = \{d \mid I$ assigns P_{A_d} to true $\}$
- $r^{\mathcal{I}_n} = \{(d, e) \mid I$ assigns $P_{r_{(d,e)}}$ to true $\}$

Since $\mathcal{O} \equiv_{\Sigma}^{n} \emptyset$ there exists an interpretation $\mathcal{J}_n$ such that $\mathcal{I}_n|_{\Sigma} = \mathcal{J}_n|_{\Sigma}$ and $\mathcal{J}_n \models \mathcal{O}$. We define the assignment J by setting:

- J assigns *true* to P_{A_d} if $d \in A^{\mathcal{J}_n}$ otherwise assigns *false* to P_{A_d} for all $A \in \mathsf{sig}(\mathcal{O}) \setminus \Sigma$
- J assigns *true* to $P_{r_{(d,e)}}$ if $(d, e) \in r^{\mathcal{J}_n}$ otherwise assigns *false* to $P_{r_{(d,e)}}$ for all $r \in \mathsf{sig}(\mathcal{O}) \setminus \Sigma$

Now what remains to be shown is the proof of the following claim which is proven by means of Lemma 5.3.4.

Claim 5.3.1. *$\psi_{\mathcal{O}}$ is true under the assignment $I \cup J$.*

Conversely assume that $\psi_{\mathcal{O}}$ is true. We now show that $\mathcal{O} \equiv_{\Sigma}^{n} \emptyset$. Given an arbitrary interpretation with domain $\Delta^{\mathcal{I}_n} = \{d_1 \dots d_n\}$ we have to show there exists a model $\mathcal{J}_n$ of $\mathcal{O}$ such that $\mathcal{I}_n|_{\Sigma} = \mathcal{J}_n|_{\Sigma}$.

Define a truth assignment I for the propositions of $\vec{p}$ as follows:

- For $A \in \Sigma$, I assigns *true* to P_{A_d} if $d \in A^{\mathcal{I}_n}$ otherwise assigns *false* to P_{A_d}.
- For $r \in \Sigma$, I assigns *true* to $P_{r_{(d,e)}}$ if $(d, e) \in r^{\mathcal{I}_n}$ otherwise assigns *false* to $P_{r_{(d,e)}}$

Since $\psi_{\mathcal{O}}$ is true, there exists a truth assignment J for the propositions in $\vec{q}$ such that $\mathcal{O}^{\dagger}$ is true under $I \cup J$.

Now define $\mathcal{J}_n$ as an extension of $\mathcal{I}_n$ as follows:

- For $A \in \mathsf{sig}(\mathcal{O}) \setminus \Sigma$ set $A^{\mathcal{J}_n} = \{d \mid J$ assigns *true* to $q_{A_d}\}$
- For $r \in \mathsf{sig}(\mathcal{O}) \setminus \Sigma$ set $r^{\mathcal{J}_n} = \{(d, e) \mid J$ assigns *true* to $q_{r_{(d,e)}}\}$

Now since $\mathcal{J}_n$ interprets the symbols from Σ in the same way as $\mathcal{I}_n$ we have $\mathcal{I}_n|_\Sigma = \mathcal{J}_n|_\Sigma$. What remains to be shown is the following claim which is proven by means of Lemma 5.3.3.

Claim 5.3.2. $\mathcal{J}_n \models \mathcal{O}$.

$\square$

Towards the proof of these claims we need some intermediate lemmas.

Lemma 5.3.1. *Let $\mathcal{O}$ be the ontology, $\psi_\mathcal{O}$ the QBF formula, $\mathcal{J}_n$ the interpretation and $I \cup J$ the valuation from Theorem 5.3.1. For an arbitrary role expression R in $\mathcal{O}$ and its translation $([R]_{(d,e)})$ in $\psi_\mathcal{O}$ under arbitrary domain elements $d, e \in \Delta^{\mathcal{J}_n}$ using the translation in Figure 5.1. It holds that $(d, e) \in R^{\mathcal{J}_n} \Leftrightarrow \mathfrak{v}([R]_{(d,e)}) = true$ under $I \cup J$.*

Proof. For a proof by structural induction on role expressions R:

- **Base case:** For an atomic role r, $(d, e) \in r^{\mathcal{J}_n} \Leftrightarrow \mathfrak{v}([r]_{(d,e)}) = true$ follows from the construction of $I \cup J$ and $\mathcal{J}_n$.
- **Induction:** Complex role expression r
  - Let $R = r^-$ where r is atomic.

    Now $(d, e) \in (r^-)^{\mathcal{J}_n} \Leftrightarrow (e, d) \in r^{\mathcal{J}_n}$. By the induction hypothesis for arbitrary domain elements d, e we have $(e, d) \in r^{\mathcal{J}_n} \Leftrightarrow \mathfrak{v}([r]_{(e,d)}) = true$ and therefore $(d, e) \in (r^-)^{\mathcal{J}_n} \Leftrightarrow \mathfrak{v}([r]_{(e,d)}) = true$
  - Let $R = r \circ s$ for atomic roles r, s

    Now $(d, e) \in (r \circ s)^{\mathcal{J}_n} \Leftrightarrow \exists f \in \Delta^{\mathcal{J}_n}$ such that $(d, f) \in r^{\mathcal{J}_n}$ and $(f, e) \in s^{\mathcal{J}_n}$. By the induction hypothesis $(d, f) \in r^{\mathcal{J}_n} \Leftrightarrow \mathfrak{v}([r]_{(d,f)}) = true$ and $(f, e) \in s^{\mathcal{J}_n} \Leftrightarrow \mathfrak{v}([s]_{(f,e)}) = true$. Since the domain is finite, the existence of any $f \in \Delta^{\mathcal{J}_n}$ can be expressed as a disjunction, and so, $(d, e) \in (r \circ s)^{\mathcal{J}_n} \Leftrightarrow \mathfrak{v}(\bigvee_{f \in \Delta^{\mathcal{J}_n}} [r]_{(d,f)} \wedge [s]_{(f,e)}) = true$

$\square$

Lemma 5.3.2. *Let $\mathcal{O}$ be the ontology, $\psi_{\mathcal{O}}$ the QBF formula, $\mathcal{J}_n$ the interpretation and $I \cup J$ the valuation from Theorem 5.3.1. For an arbitrary concept C in $\mathcal{O}$ and its translation ($[C]_d$) in $\psi_{\mathcal{O}}$ under an arbitrary domain element $d \in \Delta^{\mathcal{J}_n}$ using the translation in Figure 5.2. It holds that $d \in C^{\mathcal{J}_n} \Leftrightarrow \mathfrak{v}([C]_d) = true$ under $I \cup J$.*

Proof. For a proof by structural induction on concepts C.

- **Base case:** $d \in A^{\mathcal{J}_n} \Leftrightarrow \mathfrak{v}([A]_d) = true$ follows immediately from the construction of $I \cup J$ and $\mathcal{J}_n$.
- **Induction:** Complex concept C
  - Let $C = C_1 \sqcap C_2$
    By the induction hypothesis $d \in {C_i}^{\mathcal{J}_n} \Leftrightarrow \mathfrak{v}([C_i]_d) = true$ for $i \in \{1, 2\}$
    Now $d \in C^{\mathcal{J}_n} \Leftrightarrow d \in {C_1}^{\mathcal{J}_n}$ and $d \in {C_2}^{\mathcal{J}_n} \Leftrightarrow \mathfrak{v}([C_1]_d) = true$ and $\mathfrak{v}([C_2]_d) = true$ and therefore $d \in C^{\mathcal{J}_n} \Leftrightarrow \mathfrak{v}([C_1]_d \wedge [C_2]_d) = true$
  - Let $C = \neg D$
    By the induction hypothesis $d \notin D^{\mathcal{I}} \Leftrightarrow \mathfrak{v}([D]) = false$
    Now $d \in C^{\mathcal{J}_n} \Leftrightarrow d \in \Delta^{\mathcal{J}_n} \setminus D^{\mathcal{J}_n} \Leftrightarrow d \notin D^{\mathcal{J}_n}$ and therefore $d \in C^{\mathcal{J}_n} \Leftrightarrow \mathfrak{v}(\neg[D]_d) = true$
  - Let $C = (\geq m\ r.D)$ where r is atomic or r^-
    Now $d \in C^{\mathcal{J}_n} \Leftrightarrow |\{(d, e) \in r^{\mathcal{J}_n} \wedge e \in D^{\mathcal{J}_n}\}| \geq m$ i.e. there are at least m r-successors in D. Distinguish three cases to match those in the given translation:
    1. $m = 0$
       Such a cardinality restriction it is trivial to satisfy as a negative number of r-successors are not possible, and so $d \in C \Leftrightarrow true$
    2. $|\Delta^{\mathcal{J}_n}| \geq m > 0$
       By Lemma 5.3.1 for arbitrary domain elements $d, e \in \Delta^{\mathcal{J}_n}$ $(d, e) \in r^{\mathcal{J}_n} \Leftrightarrow \mathfrak{v}([r]_{(d,e)}) = true$ and by the induction hypothesis $e \in D^{\mathcal{J}_n} \Leftrightarrow \mathfrak{v}([D]_e) = true$

Then r has m successors in D iff for some $e_1, \dots e_m \in \Delta^{\mathcal{J}_n}$ $\mathfrak{v}(\bigwedge_{e_i}[r]_{(d,e)} \wedge [C]_e) = true$ for $m \geq i > 0$. Therefore this holds for any combination size m iff $\mathfrak{v}(\bigvee_{s \in \mathsf{Rsets}(m)} \bigwedge_{e \in s}[r]_{(d,e)} \wedge [C]_e) = true$

3. $m > |\Delta^{\mathcal{J}_n}|$

   Given a domain size n it is not possible for a role r to have m r-successors. Therefore $d \in C^{\mathcal{J}_n} \Leftrightarrow false$

□

Lemma 5.3.3. *Proof of Claim 5.3.2.* $\mathcal{J}_n \models \mathcal{O}$

Proof. Distinguish different axiom types $\alpha \in \mathcal{O}$. We use the notation ψ_α to refer to the part of the formula $\psi_\mathcal{O}$ due to $\alpha \in \mathcal{O}$.

1. $\alpha = C \sqsubseteq D$

   For an arbitrary $d \in \Delta^{\mathcal{J}_n}$ assume $d \in C^{\mathcal{J}_n}$. We need to show that $d \in D^{\mathcal{J}_n}$. By Lemma 5.3.2 $\mathfrak{v}([C]_d) = true$ under $I \cup J$. Now the translated formula $\psi_\alpha = \bigwedge_{d \in \Delta^{\mathcal{J}_n}} [C]_d \rightarrow [D]_d$ is part of the true formula $\psi_\mathcal{O}$ and so $\mathfrak{v}([D]_d) = true$ but then by Lemma 5.3.2 it holds that $d \in D^{\mathcal{J}_n}$.

2. $\alpha = r \sqsubseteq s$

   Proof uses the same argument as the previous but Lemma 5.3.1 applies instead of Lemma 5.3.2.

3. $\alpha = r_1 \circ r_2 \circ \dots \circ r_n \sqsubseteq r$

   Assume for some arbitrary domain elements $(d, e) \in \Delta^{\mathcal{J}_n} \times \Delta^{\mathcal{J}_n}$ we have $(d, e) \in (r_1 \circ r_2 \circ \dots \circ r_n)^{\mathcal{J}_n}$. We need to show that $(d, e) \in r^{\mathcal{J}_n}$. From Lemma 5.3.1 it holds that $\mathfrak{v}([r_1 \circ r_2 \circ \dots \circ r_n]_{(d,e)}) = true$ under $I \cup J$. Now as the subformula $\psi_\alpha = \bigwedge_{(d,e) \in \Delta^{\mathcal{J}_n} \times \Delta^{\mathcal{J}_n}} [r_1 \circ r_2 \circ \dots \circ r_n]_{(d,e)} \rightarrow [r]_{(d,e)}$ considers all combinations of domain elements *and* is part of the true formula $\psi_\mathcal{O}$ if follows that $\mathfrak{v}([r]_{(d,e)}) = true$ it follows from Lemma 5.3.1 that $(d, e) \in r^{\mathcal{J}_n}$.

4. $\alpha = \mathsf{Disj}(r, s)$

Assume that for arbitrary elements $(d, e) \in \Delta^{\mathcal{J}_n} \times \Delta^{\mathcal{J}_n}$ that $(d, e) \in r^{\mathcal{J}_n}$. We need to show $(d, e) \notin s^{\mathcal{J}_n}$. By Lemma 5.3.1 $\mathfrak{v}([r]_{(d,e)}) = true$ under $I \cup J$. Since the translated formula $\psi_\alpha = \bigwedge_{(d,e)\in\Delta^{\mathcal{J}_n}\times\Delta^{\mathcal{J}_n}} [r]_{(d,e)} \to \neg[s]_{(d,e)}$ is part of the true formula $\psi_{\mathcal{O}}$ and hence $\mathfrak{v}(\neg[s]_{(d,e)}) = true$ and so $\mathfrak{v}([s]_{(d,e)}) = 0$ and by Lemma 5.3.1 $(d, e) \notin s^{\mathcal{J}_n}$

5. $\alpha = \mathsf{Symm}(r)$

   Assume for arbitrary elements $d, e \in \Delta^{\mathcal{J}_n}$ that $(d, e) \in r^{\mathcal{J}_n}$. We need this show this assumption holds iff $(d, e) \in (r^-)^{\mathcal{J}_n}$. The case where $d = e$ holds trivially. For the remaining cases by Lemma 5.3.1 $\mathfrak{v}([r]_{(d,e)}) = true$ under $I \cup J$. Now translated formula $\psi_\alpha = \bigwedge_{\substack{(d,e)\in\Delta^{\mathcal{J}_n}\times\Delta^{\mathcal{J}_n}\\ d\neq e}} [r]_{(d,e)} \leftrightarrow [r^-]_{(d,e)}$ which considers the translation under all other domain elements, is part of the true formula $\psi_{\mathcal{O}}$. So $\mathfrak{v}([r]_{(d,e)})) = true$ iff $\mathfrak{v}([r^-]_{(d,e)}) = true$ and therefore by Lemma 5.3.1 $(d, e) \in r^{\mathcal{J}_n}$ iff $(d, e) \in (r^-)^{\mathcal{J}_n}$.

6. $\alpha = \mathsf{Asymm}(r)$

   Assume for arbitrary elements $(d, e) \in \Delta^{\mathcal{J}_n} \times \Delta^{\mathcal{J}_n}$ we have $(d, e) \in r^{\mathcal{J}_n}$. We prove this assumption holds iff $(d, e) \in \neg(r^-)^{\mathcal{J}_n}$. Distinguish two cases to match those in the given translation:

$$\psi_\alpha = \bigwedge_{(d,e)\in\Delta^{\mathcal{J}_n}\times\Delta^{\mathcal{J}_n}} \begin{cases} d = e & \neg[r]_{(d,e)} \\ d \neq e & [r]_{(d,e)} \leftrightarrow \neg[r^-]_{(d,e)} \end{cases}$$

   (a) $d = e$

   Now since the sub-formula $\neg[r]_{(d,e)}$ is part of the true formula ψ_α. By Lemma 5.3.1 $\mathfrak{v}(\neg[r]_{(d,e)}) = 1 \Leftrightarrow (d, e) \in \neg r^{\mathcal{J}_n}$. But then $(d, e) \notin r^{\mathcal{J}_n}$ in contradiction to the original assumption.

   (b) $d \neq e$

   By Lemma 5.3.1 $\mathfrak{v}([r]_{(d,e)}) = true$ under $I \cup J$. Now since the sub-formula $[r]_{(d,e)} \leftrightarrow \neg[r^-]_{(d,e)}$ is part of the true formula ψ it holds that $\mathfrak{v}([r]_{(d,e)}) = true$ iff $\mathfrak{v}(\neg[r^-]_{(d,e)}) = true$. Then by Lemma 5.3.1 it holds that $(d,e) \in \neg(r^-)^{\mathcal{J}_n}$ and therefore $(d,e) \in r^{\mathcal{J}_n}$ iff $(d,e) \in \neg(r^-)^{\mathcal{J}_n}$ as required.

7. $\alpha = \mathsf{Refl}(r)$

   We show for an arbitrary element $d \in \Delta^{\mathcal{J}_n}$ we have $(d,d) \in r^{\mathcal{J}_n}$. As $\psi_\alpha = \bigwedge_{d \in \Delta^{\mathcal{J}_n}} [r]_{(d,d)}$ is part of the true formula $\psi_{\mathcal{O}}$. It holds that $\mathfrak{v}([r]_{(d,d)}) = true$ for all $d \in \Delta^{\mathcal{J}_n}$ under $I \cup J$. Therefore, by Lemma 5.3.1 it holds that $(d,d) \in r^{\mathcal{J}_n}$.

8. $\alpha = \mathsf{Irrefl}(r)$

   We show for an arbitrary element $d \in \Delta^{\mathcal{J}_n}$ we have $(d,d) \notin r^{\mathcal{J}_n}$. As $\psi_\alpha = \bigwedge_{d \in \Delta^{\mathcal{J}_n}} \neg[r]_{(d,d)}$ is part of the true formula $\psi_{\mathcal{O}}$. It holds that $\mathfrak{v}(\neg[r]_{(d,d)}) = true$ under $I \cup J$ for all $d \in \Delta^{\mathcal{J}_n}$ and so $\mathfrak{v}([r]_{(d,d)}) = false$. Therefore by Lemma 5.3.1 it holds that $(d,d) \notin r^{\mathcal{J}_n}$.

9. $\alpha = \mathsf{Trans}(r)$

   First assume for some arbitrary domain elements $d, e, e' \in \Delta^{\mathcal{J}_n}$ we have both $(d,e) \in r^{\mathcal{J}_n}$ and $(e,e') \in r^{\mathcal{J}_n}$ we show that we also have $(d,e') \in r^{\mathcal{J}_n}$ and thus satisfying the transitive relation. The case in which $d = e$ or $e = e'$ holds trivially. For the remaining cases by Lemma 5.3.1 it follows that $\mathfrak{v}([r]_{(d,e)}) = true$ and $\mathfrak{v}([r]_{(e,e')}) = true$ and so $\mathfrak{v}([r]_{(d,e)} \wedge [r]_{(e,e')}) = 1$.

   Now the subformula $\psi_\alpha = \bigwedge_{\substack{(d,e) \in \Delta^{\mathcal{J}_n} \times \Delta^{\mathcal{J}_n} \\ (e,e') \in \Delta^{\mathcal{J}_n} \times \Delta^{\mathcal{J}_n} \\ d \neq e, e \neq e'}} [r]_{(d,e)} \wedge [r]_{(e,e')} \to [r]_{(d,e')}$ which considers the translation under all other domain elements, is part of the true formula $\psi_{\mathcal{O}}$. It follows that $\mathfrak{v}([r]_{(d,e')}) = true$ and so by Lemma 5.3.1 it follows that $(d,e') \in r^{\mathcal{J}_n}$.

Lemma 5.3.4. *Proof of Claim 5.3.1. $\psi_{\mathcal{O}}$ is true under the assignment $I \cup J$.*

1. $\psi_\alpha = \bigwedge_{d \in \Delta^{\mathcal{J}_n}} [C]_d \to [D]_d$

   For an arbitrary $d \in \Delta^{\mathcal{J}_n}$ assume $\mathfrak{v}([C]_d) = true$. We need to show that $\mathfrak{v}([D]_d) = true$. Since $\mathfrak{v}([C]_d) = true$ it holds by Lemma 5.3.2 that $d \in$

$C^{\mathcal{J}_n}$. Now as $\psi_\alpha = C \sqsubseteq D^\dagger$ and $\mathcal{J}_n \models C \sqsubseteq D$ we have $d \in D^{\mathcal{J}_n}$ so by Lemma 5.3.2 it follows that $\mathfrak{v}([D])_d) = true$.

2. $\bigwedge_{(d,e)\in\Delta^{\mathcal{J}_n}\times\Delta^{\mathcal{J}_n}} [r]_{(d,e)} \rightarrow [s]_{(d,e)}$

   Proof uses the same argument as the previous case but Lemma 5.3.1 applies instead of Lemma 5.3.2.

3. $\bigwedge_{(d,e)\in\Delta^{\mathcal{J}_n}\times\Delta^{\mathcal{J}_n}} [r_1 \circ r_2 \circ \cdots \circ r_n]_{(d,e)} \rightarrow [r]_{(d,e)}$

   Assume the formula $\mathfrak{v}([r_1 \circ r_2 \circ \cdots \circ r_n]_{(d,e)}) = true$ for an arbitrary elements $(d,e) \in \Delta^{\mathcal{J}_n} \times \Delta^{\mathcal{J}_n}$. We show $\mathfrak{v}([r]_{(d,e)}) = true$. Under the original assumption, by Lemma 5.3.1 it follows that $(d,e) \in (r_1 \circ r_2 \circ \cdots \circ r_n)^{\mathcal{J}_n}$. Now as $\psi_\alpha = (r_1 \circ r_2 \circ \cdots \circ r_n \sqsubseteq r)^\dagger$ and $\mathcal{J}_n \models (r_1 \circ r_2 \circ \cdots \circ r_n \sqsubseteq r)$ it follows that $(d,e) \in r^{\mathcal{J}_n}$ and therefore, by Lemma 5.3.1 $\mathfrak{v}([r]_{(d,e)}) = true$.

4. $\psi_\alpha = \bigwedge_{(d,e)\in\Delta^{\mathcal{J}_n}\times\Delta^{\mathcal{J}_n}} [r]_{(d,e)} \rightarrow \neg[s]_{(d,e)}$

   For an arbitrary $(d,e) \in \Delta^{\mathcal{J}_n} \times \Delta^{\mathcal{J}_n}$ assume $\mathfrak{v}([r]_{(d,e)})) = true$. We need to show that $\mathfrak{v}(\neg[s]_{(d,e)}) = true$. By Lemma 5.3.1 we have $(d,e) \in r^{\mathcal{J}_n}$. Now since $\psi_\alpha = \mathsf{Disj}(r,s)^\dagger$ and $\mathcal{J}_n \models \mathsf{Disj}(r,s)$ it follows that $(d,e) \notin s^{\mathcal{J}_n}$ and so by Lemma 5.3.1 $\mathfrak{v}([s]_{(d,e)}) = false$ and therefore $\mathfrak{v}(\neg[s]_{(d,e)}) = true$.

5. $\psi_\alpha = \bigwedge_{\substack{(d,e)\in\Delta^{\mathcal{J}_n}\times\Delta^{\mathcal{J}_n} \\ d\neq e}} [r]_{(d,e)} \leftrightarrow [r^-]_{(d,e)}$

   For an arbitrary $d,e \in \Delta^{\mathcal{J}_n}$ assume $\mathfrak{v}([r]_{(d,e)}) = true$ we show that this holds iff $\mathfrak{v}([r^-]_{(d,e)}) = true$. The case where $d = e$ holds trivially as $[r]_{(d,e)} = [r^-]_{(d,e)}$. For the remaining cases by Lemma 5.3.1 we have $(d,e) \in r^{\mathcal{J}_n}$. Now since $\psi_\alpha = \mathsf{Symm}(r)^\dagger$ and $\mathcal{J}_n \models \mathsf{Symm}(r)$ it follows that $(d,e) \in r^{\mathcal{J}_n}$ iff $(d,e) \in (r^-)^{\mathcal{J}_n}$. Then by Lemma 5.3.1 $\mathfrak{v}([r]_{(d,e)}) = true$ iff $\mathfrak{v}([r^-]_{(d,e)}) = true$ as required.

6. $\psi_\alpha = \bigwedge_{(d,e)\in\Delta^{\mathcal{J}_n}\times\Delta^{\mathcal{J}_n}} \begin{cases} d = e & \neg[r]_{(d,e)} \\ d \neq e & [r]_{(d,e)} \leftrightarrow \neg[r^-]_{(d,e)} \end{cases}$

   For arbitrary elements $d,e \in \Delta^{\mathcal{J}_n}$. Distinguish two cases to match the given translation:

- $d = e$

  Since $\psi_\alpha = \mathsf{Asymm}(r)^\dagger$ and $\mathcal{J}_n \models \mathsf{Asymm}(r)$ we must have $(d, e) \notin r^{\mathcal{J}_n}$, and then by Lemma 5.3.1 it holds that $\mathfrak{v}(\neg[r]_{(d,e)}) = true$

- $d \neq e$

  Assume that $\mathfrak{v}([r]_{(d,e)}) = true$. We have to show that $\mathfrak{v}([r])_{(d,e)}) = true$. Since $\mathfrak{v}([r]_{(d,e)}) = true$, by Lemma 5.3.1 we have $(d, e) \in r^{\mathcal{J}_n}$, and since $\alpha = \mathsf{Asymm}(r)^\dagger$ and $\mathcal{J}_n \models \mathsf{Asymm}(r)$ it holds that $(d, e) \in r^{\mathcal{J}_n}$ iff $(d, e) \in \neg(r^-)^{\mathcal{J}_n}$ then by Lemma 5.3.1 $\mathfrak{v}([r]_{(d,e)}) = true$ iff $\mathfrak{v}([\neg r^-]_{(d,e)}) = true$ as required.

7. $\psi_\alpha = \bigwedge_{d \in \Delta^{\mathcal{J}_n}} [r]_{(d,d)}$

   For an arbitrary $d \in \Delta^{\mathcal{J}_n}$ we have to show that $\mathfrak{v}([r]_{(d,d)}) = true$. As $\alpha = \mathsf{Refl}(r)^\dagger$ and $\mathcal{J}_n \models \mathsf{Refl}(r)$. It follows that for all $d \in \Delta^{\mathcal{J}_n}$ we have $(d, d) \in r^{\mathcal{J}_n}$ and then, by Lemma 5.3.1 $\mathfrak{v}([r]_{(d,d)}) = true$ as required.

8. $\psi_\alpha = \bigwedge_{d \in \Delta^{\mathcal{J}_n}} \neg[r]_{(d,d)}$

   For an arbitrary $d \in \Delta^{\mathcal{J}_n}$ we need to show that $\mathfrak{v}(\neg[r]_{(d,d)}) = true$. As $\psi_\alpha = \mathsf{Irrefl}(r)^\dagger$ and $\mathcal{J}_n \models \mathsf{Irrefl}(r)$. It follows that for all $d \in \Delta^{\mathcal{J}_n}$ we have $(d, d) \notin r^{\mathcal{J}_n}$ and then, by Lemma 5.3.1 $\mathfrak{v}([r]_{(d,d)}) = false$ and therefore $\mathfrak{v}(\neg[r]_{(d,d)}) = true$ as required.

9. $\psi_\alpha = \bigwedge_{\substack{(d,e) \in \Delta^{\mathcal{J}_n} \times \Delta^{\mathcal{J}_n} \\ (e,e') \in \Delta^{\mathcal{J}_n} \times \Delta^{\mathcal{J}_n} \\ d \neq e, e \neq e'}} [r]_{(d,e)} \wedge [r]_{(e,e')} \rightarrow [r]_{(d,e')}$

   First assume for arbitrary domain elements $d, e, e' \in \Delta^{\mathcal{J}_n}$ it holds that $\mathfrak{v}([r]_{(d,e)}) = true$ and $\mathfrak{v}([r]_{(e,e')}) = true$. We need to show it also holds that $\mathfrak{v}([r]_{(d,e')}) = true$. From the initial assumption it holds by Lemma 5.3.1 that $(d, e) \in r^{\mathcal{J}_n}$ and $(e, e') \in r^{\mathcal{J}_n}$ Now $\psi_\alpha = \mathsf{Trans}(r)^\dagger$ and $\mathcal{J}_n \models \mathsf{Trans}(r)$ it holds that $(d, e') \in r^{\mathcal{J}_n}$. Therefore, by Lemma 5.3.1 $\mathfrak{v}([r]_{(d,e')})$ as required.

□

Theorem 5.3.2. *Let $\mathcal{O}$ be a $\mathcal{SRIQ}$ ontology, Σ as signature, and n a positive integer. Then it is in Π_2^p to decide if $\mathcal{O} \equiv_\Sigma^n \emptyset$.*

Proof. The proof is an immediate consequence of Theorem 5.3.1 which deciding if $\mathcal{O} \equiv_{\Sigma}^{n} \emptyset$ is reduced the satisfiability of a $\forall\exists$-QBF formula, a problem which is well known to be in Π_2^p [Bie+09]. $\square$

This result also gives a proof of the Π_2^p upper bound for deciding Σ-inseparability the empty ontology for $\mathcal{SHIQ}$ ontologies and concept signatures.

Lemma 5.3.5. *For a $\mathcal{SHIQ}$ ontology $\mathcal{O}$ and a concept signature Σ, it is in Π_2^p to decide whether $\mathcal{O} \equiv_{\Sigma} \emptyset$.*

Proof. Since $\mathcal{SHIQ}$ is preserved under disjoint unions and Σ is a concept signature, it follows that $\mathcal{O} \equiv_{\Sigma} \emptyset$ iff $\mathcal{O} \equiv_{\Sigma}^{1} \emptyset$ by Lemma 3.2.1 (One-point criterion) and since $\mathcal{SHIQ}$ is a sub-language of $\mathcal{SRIQ}$ deciding if $\mathcal{O} \equiv_{\Sigma}^{1} \emptyset$ can be reduced to the satisfiability of a $\forall\exists$-QBF formula by Theorem 5.3.1 so is in Π_2^p.

$\square$

5.3.1 Nominals

Although our reduction of exactly n-inseparability does not permit nominals, in theory they can be supported. This is only possible if unique name assumption (UNA) is not made — a standard assumption for most Description Logics. The UNA is the assumption that for an interpretation $\mathcal{I}$ of an ontology $\mathcal{O}$ the interpretation function maps different individual names to different elements of the domain (i.e, $a^{\mathcal{I}} \neq b^{\mathcal{I}}$ for all $a, b \in \mathsf{sig}(\mathcal{O})$ such that $a \neq b$) [HT02].

In the context of deciding n-inseparability, it may not be possible to interpret an ontology under an n-element interpretation without violating the UNA. For example, if an ontology $\mathcal{O}$ contains the axiom $A \sqsubseteq \{a\} \sqcap \{b\}$ and we consider the case of 1-inseparability, we clearly cannot interpret $\mathcal{O}$ under an interpretation $\mathcal{I}$ with $\sharp\Delta^{\mathcal{I}} = 1$ and still maintain the individuals $a^{\mathcal{I}}$ and $b^{\mathcal{I}}$ are mapped to different domain elements. In the general case, if the UNA is made and $\mathcal{O}$ is an ontology with nominals that uses more than n distinct individual names, it is not possible to decide n-inseparability whilst still preserving the UNA, an arbitrary n-element interpretation is not a valid interpretation of such an ontology.

Assuming the UNA is not made, there is a way of deciding n-inseparability for ontologies containing nominals. First we would need to extend our notion of signatures to include individual names e.g. $\mathsf{sig}(\mathsf{Liverpool\_Student} \sqsubseteq \exists\mathsf{studiesAt}.\{\mathsf{liverpool}\}) = \{\mathsf{Liverpool\_Student}, \mathsf{studiesAt}, \mathsf{liverpool}\}$. Then if we want to decide if for a $\mathcal{SROIQ}$ ontology $\mathcal{O}$ if $\mathcal{O} \equiv^n_\Sigma \emptyset$ can extend the QBF reduction as used in the $\mathcal{SRIQ}$ case.

Then we need to extend the translation of concepts (Figure 5.2) to include nominals which is shown in Figure 5.5 we do this in order to evaluate the possible ways ontologies containing nominals can be evaluated under some n-element interpretation $\mathcal{I}_n$. For every element $d \in \Delta^\mathcal{I}$ we introduce a fresh propositional atom P_{i_d} for those individuals $i \in \Sigma \cap \mathsf{N_I}$ where $\mathfrak{v}(P_{i_d}) = \mathit{true}$ exactly when $i^{\mathcal{J}_n} = \{d\}$, similarly a fresh and distinct variable Q_{i_d} is used for those individuals not in the signature.

$$\begin{aligned} [i]_d &= P_{i_d} \text{ for all } i \in \Sigma \cap \mathsf{N_I} \\ [i]_d &= Q_{i_d} \text{ for all } i \in (\mathsf{sig}(\mathcal{O}) \setminus \Sigma) \cap \mathsf{N_I} \end{aligned}$$

Figure 5.5: Translation of nominals to propositional atoms

In addition, we need to introduce a global constraint to ensure that we can only interpret nominals in a valid way. A constraint for an individual i can be constructed by interpreting i under the domain elements taken from $\Delta^{\mathcal{I}_n} = \{d_1, d_2, \ldots, d_n\}$ as follows:

$$\mathsf{const}(i) = ([i]_{d_1} \vee [i]_{d_2} \vee \cdots \vee [i]_{d_n}) \wedge \bigwedge_{j=1}^{n-1} \bigwedge_{k=j+1}^{n} (\neg[i]_{d_j} \vee \neg[i]_{d_k})$$

The first half of the constraint ensures that a nominal is always assigned to an element of $\Delta^{\mathcal{I}_n}$, the second half ensures that an individual cannot be interpreted as two distinct domain elements at the same time e.g. we can never have both $i^{\mathcal{I}_n} = \{d_1\}$ and $i^{\mathcal{I}_n} = \{d_2\}$ in a given interpretation.

Then for a $\mathcal{SROIQ}$ ontology and signature Σ and positive integer n it can be verified that $\mathcal{O} \equiv^n_\Sigma \emptyset$ iff the formula

$$\psi_{\mathcal{O}} := \forall \vec{p} \exists \vec{q} \bigwedge_{i \in \Sigma \cap \mathsf{N_I}} \mathsf{const}(i) \rightarrow \bigwedge_{j \in (\mathsf{sig}(\mathcal{O}) \setminus \Sigma) \cap \mathsf{N_I}} \mathsf{const}(j) \bigwedge_{\alpha \in \mathcal{O}} \alpha^\dagger$$

is logically true.

The construction of $\psi_{\mathcal{O}}$ is very similar to the $\mathcal{SRIQ}$ case, the sequences $\vec{p}$ and $\vec{q}$ are the same as in the original reduction containing all atoms which correspond to signature and non-signature symbols respectively (now including individuals), and the axioms of $\mathcal{O}$ are translated by the original translation but now including the translation of nominals from Figure 5.5. The additional parts of $\psi_{\mathcal{O}}$ are simply to ensure the constraints for individuals hold as described, so that the possible truth valuations in which $\psi_{\mathcal{O}}$ is interpreted always corresponds to a valid n-element interpretation of $\mathcal{O}$.

5.3.2 Extracting exactly n-depleting modules

As it is decidable to verify if $\mathcal{O} \equiv^n_{\Sigma \cup \mathsf{sig}(\mathcal{M})} \emptyset$ by Theorem 5.3.2, and that exactly n-inseparability is both monotone and robust under replacements, the general depleting Σ-module extraction algorithm (Figure 2.6) can be used to extract the unique minimal exactly n-depleting Σ-module from a $\mathcal{SRIQ}$ ontology.

We can do some optimisations from the general algorithm by defining n-separability causing axioms which are analogous to separability causing axioms which we defined for the AMEX procedure, but defined instead over exactly n-inseparability.

Definition 5.3.1 (Exactly n-separability causing axiom)**.** *Let $\mathcal{M} \subseteq \mathcal{O}$ be TBoxes then an axiom $\alpha \in \mathcal{O} \setminus \mathcal{M}$ is called exactly n-separability causing if for a subset $\mathcal{W} \subseteq \mathcal{O} \setminus \mathcal{M}$:*

$$\alpha \in \mathcal{W}; \quad (\mathcal{W} \setminus \{\alpha\}) \equiv^n_{\Sigma \cup \mathsf{sig}(\mathcal{M})} \emptyset; \quad \mathcal{W} \not\equiv^n_{\Sigma \cup \mathsf{sig}(\mathcal{M})} \emptyset.$$

Then one can show, similarly to how the rule based AMEX produces the same module as the iterative extraction algorithm, that the algorithm in Figure 5.6 computes the same unique minimal module as the general algorithm.

Input: $\mathcal{SRIQ}$ Ontology $\mathcal{O}$, Signature Σ, positive integer n
Initialise: $\mathcal{M} := \emptyset$
Output: Minimal exactly n-depleting module $\mathcal{M}$ of $\mathcal{O}$

Apply rule $\mathsf{R}_{\text{n-sep}}$ exhaustively.

($\mathsf{R}_{\text{n-sep}}$) If $\mathcal{O} \setminus \mathcal{M} \not\equiv^n_{\Sigma \cup \mathsf{sig}(\mathcal{M})} \emptyset$, then locate the first exactly n-separability causing axiom $A \bowtie C \in \mathcal{O} \setminus \mathcal{M}$ and set $\mathcal{M} := \mathcal{M} \cup \{A \bowtie C\}$

Figure 5.6: Exactly n-depleting module extraction algorithm

This allows the optimisation of locating exactly n-separability causing axioms using a binary search procedure, similar to the one used for locating separability causing axioms in AMEX (Figure 3.4). This improves on the general algorithm which requires $|\mathcal{O} \setminus \mathcal{M}|$ exactly n-inseparability checks to locate a single exactly n-inseparability causing axiom, in comparison to just $log_2(|\mathcal{O} \setminus \mathcal{M}|)$ if using the binary search procedure.

Now we can compute the lower approximation, and extract an n-depleting module from an arbitrary $\mathcal{SRIQ}$ ontology using the algorithm in Figure 5.6 by taking the union of all exactly n-depleting modules up from exactly 1-depleting to exactly n-depleting which is then the minimal n-depleting module as a result of Lemma 5.2.4.

5.4. Conclusion

In this chapter we have introduced a methodology which allows for the first time to estimate the success of an approximation of the minimal depleting Σ-module. This was achieved by treating each sound approximation procedure, those that are guaranteed to extract a depleting Σ-module, as an upper approximation. We then introduced a lower approximation by means of n-depleting modules, which are not necessarily depleting Σ-modules themselves but are always contained in the minimal depleting Σ-module. The difference in size between the upper and lower approximations then giving an estimation into the success of an upper approximation.

We saw that n-depleting modules are depleting modules defined over the n-inseparability relation in which inseparability is decided over interpretations whose domain contains at most n domain elements. We went on to prove that minimal n-depleting modules were the union of all minimal exactly n-depleting modules from exactly 1-depleting up to exactly n-depleting where exactly n-inseparability is inseparability decided over interpretations that have *exactly* n domain elements.

We then proved that exactly n-inseparability was a monotone relation that is robust under replacements and that deciding exactly n-inseparability from empty ontology for a $\mathcal{SRIQ}$ ontology was decidable (in Π_2^p) by a reduction to $\forall\exists$ QBF. This meant we could use an optimisation of the general depleting module extraction algorithm to compute the unique minimal exactly n-depleting module and in turn the unique minimal n-depleting module.

In the next chapter we will be bringing the results from the previous chapters together in an extensive experimental evaluation in which we evaluate relative sizes of the STAR, AMEX and hybrid STAR-AMEX approximations and also the success of these approximations by the computation of minimal n-depleting modules.

CHAPTER 6

Experimental Evaluation

For approximations of minimal depleting Σ-modules in expressive logics only locality-based modules such STAR have been systematically analysed in great detail, both in terms of module sizes and performance [Ves+12; Ves+13]. What is not currently known is how large and significant the difference between STAR-modules and the minimal modules they approximate, and how far one can improve on the approximations produced by the STAR extraction procedure.

In this chapter we present an empirical investigation into the approximation of minimal depleting Σ-modules in expressive logics. Particularly we are interested in seeing how far we can improve on the STAR approximation with both AMEX for terminologies up to acyclic $\mathcal{ALCQI}$ in expressivity, and with the hybrid STAR-AMEX extraction procedure for general $\mathcal{SROIQ}$ ontologies.

This investigation involves comparing the sizes of the upper approximations produced by the STAR, AMEX and hybrid STAR-AMEX procedures across a corpus of real-world ontologies to evaluate if there is any significant difference in the size of modules extracted for the same signature. Alongside, we also consider the computation n-depleting modules (up to 2-depleting), for use as a lower approximation, in order to evaluate how successful each upper approximation is — exactly how well they approximate their minimal modules.

6.1. Research questions

To investigate how we have improved the approximation of minimal modules, we look to the research questions that we proposed in Section 1.3 and aim to answer these questions over the course of the experimental investigation that follows. As we have introduced several different types of approximations, we restate specific versions of each research question so that we can consider answering research questions about different notations of approximation independently.

Difference in module size

As the STAR, AMEX, and hybrid STAR-AMEX procedures each produce an approximation which contains the unique minimal depleting Σ-module, the comparative size of modules extracted for the same signature is the main metric in evaluating different approximations.

The AMEX procedure alone has limited applications, restricted by the ontologies for which it is an extraction procedure which motivated the development of the hybrid procedure, an extraction procedure for very expressive ontologies which is at least as small as a corresponding STAR-module. This opens the question on how well they can compete with the already successful STAR procedure over different types of ontology, and our specific research questions over the size of modules can be formulated as follows:

- How often and how significant is the difference in size between STAR-modules and the corresponding AMEX approximations?

- How often and significant is the difference in size between STAR-modules and the corresponding hybrid STAR-AMEX approximations?

Minimality

The size of extracted modules gives a hint to the success of an approximation — how well they approximate the minimal depleting Σ-module. An approximation which is comparatively smaller than another must be closer in size to the ideal minimal, but may still be significantly larger. With the introduction of n-depleting modules in the previous chapter we have a way of estimating the difference in size between an approximation and the minimal module it approximates, leading to the specific research question:

- How close in size are the STAR, AMEX and hybrid STAR-AMEX approximations to the minimal modules they approximate?

Performance

The STAR extraction procedure is achieved by purely syntactic means and is known to be incredibly efficient at extracting modules from a wide range of ontologies [Ves+13]. In addition to how well the AMEX and the hybrid STAR-AMEX procedures compete with STAR over module in size, we also want to know how well they compete in terms of performance, an extraction procedure is most useful if it does not come with significant overhead.

In addition, we want to explore how feasible it is to extract an n-depleting module to compute the lower approximation. The specific research questions surrounding performance:

- How long does it take on average, and in the worst case, to extract an AMEX-module, hybrid STAR-AMEX-module compared to a STAR-module

- How long does it take on average, and in the worst case, to extract a n-depleting module?

6.2. Experimental Setting

We outline here the implementation details of our extraction algorithms along with our ontology and signature selection over which our research questions will be evaluated. We start with with the details of our algorithm implementations which we used to perform the evaluation.

Extraction procedures

We implemented the AMEX (Figure 3.5) procedure in the Java programming language aided by the OWL-API library for ontology manipulation [HB11]. **R1** of AMEX for detecting and removing axiom dependency chains is implemented in pure Java, and the application of **R2** for deciding Σ-inseparability and subsequently locating separability causing axioms is achieved by the reduction to QBF using off-the-shelf QBF solvers, the specifics of which we will come on to in a moment. For comparative purposes we use the implementation of the STAR extraction algorithm as implemented in the OWL-API library version 3.2.4.1806. The hybrid STAR-AMEX extraction procedure (Figure 4.2) was then easily implemented, also in Java, by combining the AMEX and STAR procedures together.

Extracting a minimal n-depleting Σ module was achieved by taking the union of minimal exactly n-depleting Σ-modules from 1-depleting up to n-depleting which is the minimal n-depleting module by Lemma 5.2.4. Before we extracted an n-depleting module from an ontology we first performed a preprocessing step by extracting a hybrid STAR-AMEX module. This optimisation is possible as each hybrid STAR-AMEX-module is a depleting Σ-module which by Theorem 5.1.1 the minimal n-depleting Σ-module is always contained. For the extraction of minimal exactly n-depleting Σ-modules (Figure 5.6) we have also produced an implementation which is assisted by the OWL-API. However, the majority of the workload involves deciding exactly n-inseparability and locating exactly n-separability causing axioms, so for this we have also used off-the-shelf

QBF solvers to perform the reduction we discussed in the previous chapter.

Solving QBF instances

To solve particular instances of QBF reductions we used 3 different QBF solvers: quantor [EB05] , sKizzo [Ben04], and depQBF [LB10b]. Each of these solvers required the input formula to be converted into CNF (Conjunctive Normal Form) which can cause exponential growth of the original formula [BL99b]. To combat this, for our reductions we used the translation described in [Tse68] which allows, by the introduction of fresh propositional variables, a linear encoding of an arbitrary propositional formula into conjunctive normal form whilst maintaining equisatisfiablity.

Some preliminary experiments in which we tested solving QBF reductions for deciding inseparability showed that performance of specific QBF solvers varied considerably depending on the input ontologies and signatures we considered, and as a result there was no one solver which performed optimally over all QBF instances we tested. For this reason, to do the actual solving we fed each QBF instance generated by one of our algorithms to each QBF solver in turn with a timeout of 10 seconds. If one QBF solver could not solve the problem in the given 10 seconds we simply moved onto the next one, and then if no solver could solve the problem without timing out we simply went back to first solver and waited until a result was returned. The order each QBF solver was considered was the same as when we listed them above: quantor, sKizzo, then depQBF.

Hardware

For all our experiments were carried out on PC with an Intel Intel i7-2600 CPU @ 3.40GHz with 4GB of heap space allocated for use by our Java programs.

6.2.1 Ontology selection

For our experiments to be meaningful it is important to use ontologies which are desirable for modularisation and reuse. For this reason we have selected our ontologies from well known and publicly accessible corpora from across the web. Each ontology was obtained in the form of a source .owl, a file which specifies each ontology according to the OWL standard, which means it may contain many non-logical constructs such as annotations and labels which do convey any semantic meaning. For our purposes we only consider the logical axioms of the each ontology — those which correspond to description logic axioms which we described in Section 2.1.

NCI

We put particular focus on the National Cancer Institute (NCI) ontology [1] due to its importance, size, expressivity, and the fact that it is high-quality ontology, all of which make it ideal for applications surrounding module extraction. The NCI ontology itself is actively maintained by a team of ontology engineers and domain experts, covering the domain of cancer and general health care research, a new version is released at a rate of once a month [Gol+11].

For our experimental evaluation we use the NCI Thesaurus version 08.09d which is a $\mathcal{ALCH}$ ontology containing 98,752 axioms. The majority of the axioms which make up the ontology form an acyclic terminology with RCIs, 87,934 axioms are concept inclusions of the form $A \sqsubseteq C$ and 10,366 are equivalences of the form $A \equiv C$. Most of these axioms (all but 4588) are $\mathcal{EL}$ inclusions, the non-$\mathcal{EL}$ axioms contain a total of 7806 occurrences of the universal restriction constructor ($\forall$). The remaining 452 axioms are a combination of role inclusions, of the form $r \sqsubseteq s$, and domain and range restrictions, non-terminological axioms of the form $\top \sqsubseteq \forall r.C$ or $\exists r.\top \sqsubseteq C$. The signature of this version of NCI contains 68,862 concept names and 88 role names.

[1] https://ncit.nci.nih.gov/

Experimental Corpus

In addition to NCI, in order to give a more comprehensive experimental evaluation, we also consider a large corpus of ontologies which varies considerably in both size and expressivity. To build this corpus we began by taking all ontologies from the Tones [2] and the NCBO Bioportal [3] ontology repositories, both of which include ontologies covering a variety of domains, the latter focussing mainly on the biomedical sector. With this initial corpus of ontologies in place, we removed any ontologies which were unable to parse with the OWL-API. In addition, under consideration that the AMEX and hybrid STAR-AMEX procedures as well as the lower approximation all use reductions to QBF (with a Π_2^p worst case) which may take a considerable amount of time compute, we also removed any ontology that contained over $10,000$ axioms, which enables us to evaluate a broader range of ontologies over a reasonable time frame.

We next removed any ontology where over 95% of axioms corresponded to an acyclic $\mathcal{ELI}$ terminology which consists of only of concept inclusions. This is motivated by Proposition 38 from [Kon+13] where it is shown that if $\mathcal{T}$ is an acyclic $\mathcal{ELI}$ terminology containing no concept equivalences, both the STAR-module and module produced by the original acyclic $\mathcal{ALCI}$ extraction algorithm (for which AMEX produces an identical module) coincide with the minimal Σ-module of $\mathcal{T}$, for every signature Σ. For this reason comparing the sizes of modules produced by the STAR, AMEX or Hybrid STAR-AMEX procedures is unlikely to produce any significant differences in module size. Moreover, as the modules produced by these procedures are likely to coincide with the minimal depleting Σ-modules, computing a lower approximation for these ontologies is unlikely to reveal any meaningful results.

After filtering ontologies from the original corpus as described we obtain 172 ontologies, which notably includes several ontologies which have previously been studied in work surrounding modularity [Ves+13; Ves+10; Gra+08]: uni-

[2] http://owl.cs.manchester.ac.uk/repository
[3] https://bioportal.bioontology.org/

versity, People, miniTambis, Tambis and Galen. One final step was performed, as we are particularly interested in evaluating the success of our upper approximations by computing a lower approximation — which only supports ontologies up to $\mathcal{SRIQ}$ in expressivity — we modified 36 of the ontologies to remove any axioms containing nominals.

The 172 ontologies in our experimental corpus represent a diverse range over both size and expressivity. The size of our ontologies, measured in total number of axioms each ontology contains, ranges from 50 to 9645 axioms. To give an idea of the distribution of expressivity over all ontologies we can divide them into bins based on the family of description logics each ontology belongs. This is similar to the approach described in [WPH06; Ves13]. A natural way to do this is to consider which of the three OWL profiles (as described in Section 2.1) each ontology fits into. As a natural starting point we have three bins: $\mathcal{EL}$ which forms the basis for OWL EL, $\mathcal{SHIF}$ which forms the basis of OWL QL, and finally $\mathcal{SROIQ}$ which forms the basis of OWL DL.

Further refining this idea, we use a $\mathcal{SRIQ}$ bin instead of a $\mathcal{SROIQ}$ bin, as all axioms containing nominals have already been removed from our ontologies, so $\mathcal{SRIQ}$ is the most expressive logic in which any ontology can be formulated. Additionally, as both $\mathcal{SHIF}$ and $\mathcal{SRIQ}$ have a large range of sub-languages we introduce two additional bins $\mathcal{S}$ and $\mathcal{SHIN}$ to give a greater idea of the granularity of expressiveness of our selected ontologies. This gives us 5 bins of increasing expressivity, from $\mathcal{EL}$ to $\mathcal{SRIQ}$. Observe that the bins are strictly increasing in expressivity — the logics to which the bins correspond form a total order $\mathcal{EL} \subsetneq \mathcal{S} \subsetneq \mathcal{SHIF} \subsetneq \mathcal{SHIN} \subsetneq \mathcal{SRIQ}$. The result from dividing the 173 ontologies into the described bins can be seen in Figure 6.1.

Bin	$\mathcal{EL}$	$\mathcal{S}$	$\mathcal{SHIF}$	$\mathcal{SHIN}$	$\mathcal{SRIQ}$
Count	9	17	75	35	36

Figure 6.1: Expressivity distribution for experimental corpus

We observe that to produce the corpus, we are left with only a few very inexpressive ontologies with the majority (146 out of 173) being at least $\mathcal{SHIF}$ in expressivity. We also observe that there is a fairly even split of ontologies which contain terminological cycles. We evaluated this using the heuristic from Section 4.3.2, which is used for detecting cycles in the hybrid STAR-AMEX extraction procedure, and found 92 of the ontologies in the corpus contain terminological cycles and 81 did not.

In addition to this summary we also provide a more comprehensive breakdown of the expressivity and size for each ontology of the experimental corpus which be found in Appendix A. In the presentation of the results we use abbreviations for ontology names to provide a compact representation. The full ontology name to which each abbreviation corresponds can be also found in this appendix.

6.2.2 Signature selection

There is no well defined way of selecting signatures for modularisation, neither is it feasible to consider all possible signatures of an ontology, for which there are exponentially many. Furthermore, the number of possible depleting Σ-modules of an ontology is, in general, exponential in the size of the ontology, and often different signatures can lead to the same module being extracted, which makes it hard to even extract *different* modules from an ontology [Ves+10]. With this in mind we consider two different types of signature:

Axiom Signatures – All depleting Σ-modules of an ontology are known to be composed from a set of disjoint modules which cannot be further decomposed themselves, so called “genuine modules” [Ves+10]. For an ontology $\mathcal{O}$ there are linearly many genuine modules in the size of $\mathcal{O}$, which correspond to the depleting Σ-module extracted using the signature $\mathsf{sig}(\alpha)$ for each axiom $\alpha \in \mathcal{O}$.

Random signatures – The signatures of all of our experimental ontologies typically contain very few role names in comparison to the amount of concept names, so taking a signature size n at random from the signature of an ontology is not likely to represent many role names. To combat this, we consider a random concept signature *plus* a percentage of role names e.g a random signature size 200 of concepts from $\mathsf{sig}(\mathcal{O}) \cap \mathsf{N_C}$ and 50% of the role names chosen at random from $\mathsf{sig}(\mathcal{O}) \cap \mathsf{N_R}$.

Only linearly many axioms signatures makes computing all genuine modules for an ontology feasible in most cases. An axiom signature is typically small in size compared to the signature of the whole ontology, but extracting modules using them gives a lot of insight to the composition of all modules of an ontology. We also note axiom signatures have been extensively considered when comparing semantic and syntactic locality [Ves+13].

6.3. Experiments on NCI

In this chapter we perform experiments on the NCI ontology version 08.09d which we previously described. As this is the only version we are considering we refer to it as simply NCI.

6.3.1 Fragments of NCI

As AMEX is an extraction procedure for acyclic terminologies with RCIs up to $\mathcal{ALCQI}$ in expressivity, it cannot be used to extract modules from NCI ontology. However, as previously mentioned the majority of NCI (all but 452 axioms) corresponds to an acyclic terminology with RCIs. So for the evaluation the AMEX extraction procedure we consider three fragments of the NCI ontology all of which are acyclic $\mathcal{ALC}$ terminologies (with RCIs) which makes them all suitable for use with AMEX:

- NCI*, consisting of all 87,934 concept inclusions, and all 10,366 equivalences — 98,300 axioms in total
- NCI$^\star(\sqsubseteq)$, consisting of all 87,934 concept inclusions

- NCI$^\star(\equiv)$, consisting of all 10,366 equivalences

AMEX and STAR: Random Signatures

Our first experiment concerns extracting modules from each NCI fragment using random signatures. For each fragment we considered random signatures of sizes: 100, 250, 500, 750 and 1000 concepts, and role percentages 0%, 25%, 50%, 75% and 100% and took 1000 random signatures for each concept/role combination — a total of 25,000 signatures considered for each ontology — and extracted both an AMEX and a STAR module for each of these signatures.

The table in Figure 6.3 summarises the result of these experiments, showing the average size of the modules produced by the AMEX and STAR procedures for each separate concept/role percentage combination, and an additional column showing the percentage change (either increase or decrease) between the average sizes of the AMEX modules in comparison to the corresponding STAR modules. It can be seen that:

- in NCI$^\star(\equiv)$, AMEX-modules are significantly smaller than STAR-modules (between 73% and 82% smaller);
- in NCI$^\star(\sqsubseteq)$, AMEX-modules are, on average, slightly larger than STAR modules (between 3% and 21% larger);
- in NCI*, AMEX-modules are still significantly smaller than STAR-modules, but less so than in NCI$^\star(\equiv)$ (between 51% and 72% smaller).

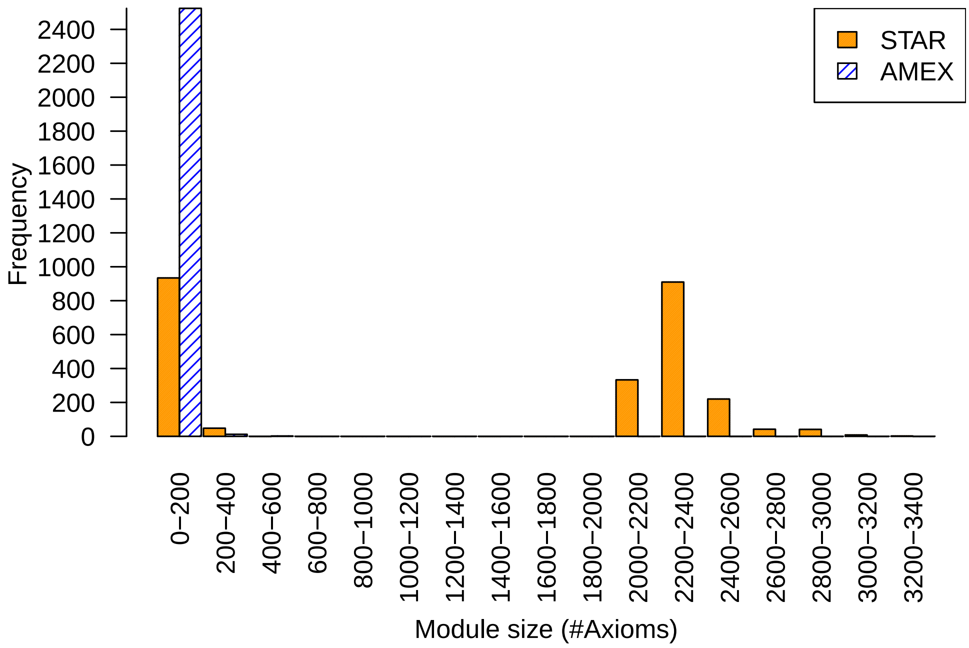
Moreover, for NCI* and NCI$^\star(\equiv)$ where AMEX-modules are considerably smaller, the difference in terms of relative module sizes is much more significant. Across all signatures we found for both NCI* and NCI$^\star(\equiv)$ the AMEX-module can be up to 5,595 and 4,624 axioms smaller than the corresponding STAR modules respectively, whereas for NCI$^\star(\sqsubseteq)$ the AMEX-modules are only up to 321 axioms larger.

The huge difference between modules for NCI$^\star(\sqsubseteq)$ and NCI$^\star(\equiv)$ can be explained if we recall that for acyclic $\mathcal{ELI}$ terminologies consisting of only concept inclusions, that STAR and AMEX modules are known to coincide [Kon+13]. This is not the case for acyclic $\mathcal{ALCQI}$ terminologies with RCIs (there can be axioms in STAR-modules that are not AMEX-modules and vice versa), but since the vast majority of axioms in NCI$^\star(\sqsubseteq)$ are $\mathcal{EL}$ concept inclusions one should not expect any significant difference between the two types of modules. So it is exactly those acyclic terminologies that contain many concept equalities, such as both NCI$^\star(\equiv)$ and NCI*, where a difference in module sizes can be observed.

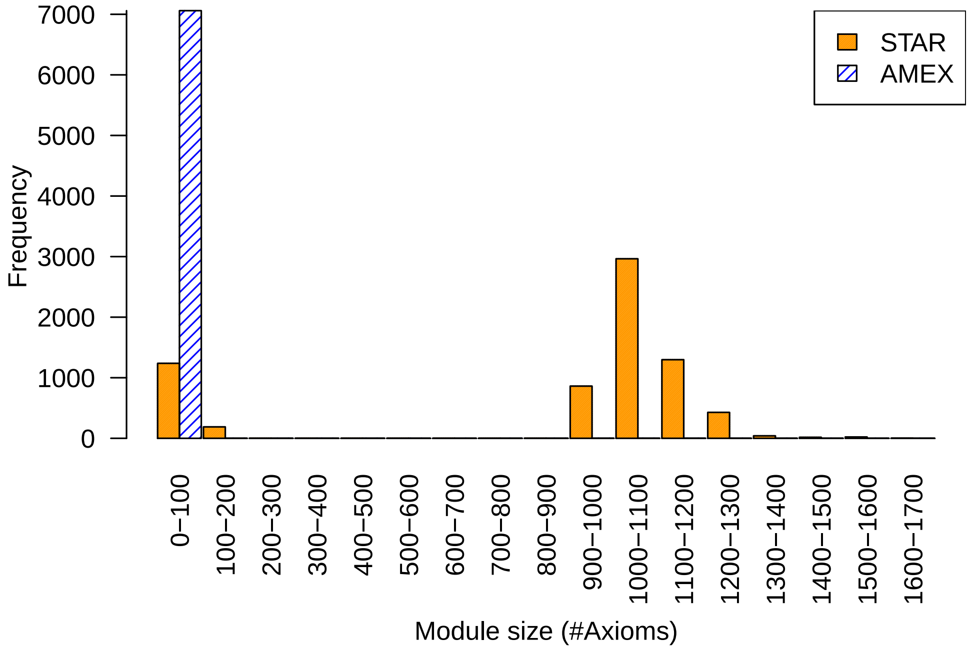
The reason why a difference is realised in the presence of concept equivalences comes down to how the two approximations handle these axioms, the nature of the STAR approximation means even simple concept equivalences can be needlessly included in the extracted modules: take the very simple acyclic $\mathcal{EL}$ terminology $\mathcal{T} = \{A \equiv B\}$ and the signature $\Sigma = \{B\}$ clearly $\mathcal{T} \equiv_\Sigma \emptyset$, for every interpretation $\mathcal{I}$ there always exists a $\mathcal{J}$ model of $\mathcal{T}$ with $\mathcal{I}|_\Sigma = \mathcal{J}|_\Sigma$ achieved by taking $A^\mathcal{J} = B^\mathcal{J} = B^\mathcal{I}$. Yet, the axiom $A \equiv B$ is neither $\top$-local nor $\bot$-local, for every interpretation of $B^\mathcal{I}$ one cannot find a model $\mathcal{J}$ of $\mathcal{T}$ where $B^\mathcal{J} = \emptyset$ or $B^\mathcal{J} = \Delta^\mathcal{J}$ whilst still maintaining that $\mathcal{I}|_\Sigma = \mathcal{J}|_\Sigma$, so the axiom $A \equiv B$ would end up in the STAR-module extracted from $\mathcal{O}$ for Σ. If we then consider how the AMEX procedure behaves in this case: Starting with $\mathcal{M} = \emptyset$ we find $\mathcal{T} \setminus \mathcal{M}$ contains no direct axiom $\Sigma \cup \mathsf{sig}(\mathcal{M})$-dependencies and so we can check with **R2** of AMEX that $\mathcal{T} \setminus \mathcal{M} \equiv_{\Sigma \cup \mathsf{sig}(\mathcal{M})} \emptyset$, so the axiom $A \equiv B$ isn't considered as belong to the AMEX-module. It is differences of this nature which can lead to large differences in the sizes of modules between the two extraction procedures. As the STAR procedure adds such equivalences to its module $\mathcal{M}$ it must consider inseparability over a larger signature — as depleting Σ-modules are decided over $\Sigma \cup \mathsf{sig}(\mathcal{M})$— which in turn pulls in more axioms, further increasing the size of the module.

AMEX and STAR: Axiom signatures

We next consider comparing the STAR and AMEX procedures by extracting modules for axiom signatures from each of the NCI fragments. This involved extracting both a STAR and AMEX-module for signatures based on 20,000 randomly selected axioms from both NCI* and NCI$^\star(\sqsubseteq)$ and each of 10,366 axioms from NCI$^\star(\equiv)$. Figure 6.2 summarises the results of this experiment. The figure shows the frequency of AMEX and STAR-modules of a given size within NCI* and NCI$^\star(\equiv)$ for the cases when the modules differ — which is in 13% and 68% of cases, respectively. For NCI$^\star(\equiv)$ in the cases in which we find a difference the STAR module is always larger than the corresponding AMEX module with an average difference of 865.6 axioms. For NCI* in a few (87 cases) the STAR modules are smaller than the corresponding AMEX ones by an average difference of 6.9 axioms whereas in the rest of the cases the STAR modules are much larger with an average difference of 1427 axioms. Over both ontologies we generally find AMEX-modules are consistently small whereas the size of STAR-modules vary considerably. We do not show the results for NCI$^\star(\sqsubseteq)$ since, as we explained above for the experiments with random signatures, there is essentially no difference between the AMEX and STAR-modules.



(a) NCI*



(b) NCI$^\star(\equiv)$

Figure 6.2: Frequency of genuine module sizes for NCI* and NCI$^\star(\equiv)$

Role%	0%			25%			50%			75%			100%		
$\lvert\Sigma\rvert$	Star	AMEX	Change %	Star	AMEX	Change %	Star	AMEX	Change %	Star	AMEX	Change %	Star	AMEX	Change %
NCI*															
100	3835.7	676.6	-82%	3848.6	943.7	-75%	3891.7	984.0	-75%	3929.4	1014.7	-74%	3929.8	1016.5	-74%
250	5310.2	1725.9	-67%	5365.6	1795.2	-67%	5463.1	1871.5	-66%	5506.3	1919.3	-65%	5505.4	1918.0	-65%
500	6985.9	2735.9	-61%	7109.6	2844.9	-60%	7165.5	2930.3	-59%	7252.8	3002.1	-59%	7245.9	2990.1	-59%
750	8223.3	3572.7	-57%	8355.2	3698.8	-56%	8464.4	3806.1	-55%	8538.5	3878.7	-55%	8526.1	3872.0	-55%
1000	9276.7	4333.6	-53%	9397.2	4458.4	-53%	9492.8	4573.9	-52%	9564.9	4627.1	-52%	9565.3	4642.7	-51%
NCI* ($\sqsubseteq$)															
100	55.5	65.0	+17%	232.8	281.9	+21%	286.1	318.8	+11%	312.6	333.7	+7%	339.8	351.7	+3%
250	328.3	390.8	+19%	559.6	657.4	+17%	651.1	718.6	+10%	712.9	759.1	+6%	765.3	796.5	+4%
500	852.9	1007.3	+18%	1046.4	1190.4	+14%	1193.8	1301.8	+9%	1278.5	1355.1	+6%	1378.3	1436.0	+4%
750	1326.0	1541.3	+16%	1517.7	1692.2	+12%	1675.4	1808.4	+8%	1802.6	1905.3	+6%	1921.1	1993.6	+4%
1000	1786.3	2039.7	+14%	1973.8	2174.0	+10%	2157.3	2314.2	+7%	2299.0	2416.3	+5%	2440.8	2527.3	+4%
NCI* ($\equiv$)															
100	2784.3	316.3	-89%	2793.0	319.7	-89%	2785.5	318.5	-89%	2770.3	318.6	-88%	2779.0	318.4	-89%
250	3982.2	622.7	-84%	3988.5	626.2	-84%	3984.8	624.0	-84%	3989.9	624.6	-84%	3982.6	625.9	-84%
500	4976.0	1001.2	-80%	4988.1	1003.8	-80%	4984.7	1002.1	-80%	4983.2	1004.1	-80%	4988.7	1004.0	-80%
750	5529.9	1310.0	-76%	5540.6	1315.3	-76%	5533.7	1309.2	-76%	5532.0	1310.8	-76%	5531.0	1311.7	-76%
1000	5899.9	1577.4	-73%	5897.4	1576.9	-73%	5891.8	1576.7	-73%	5894.1	1574.6	-73%	5900.4	1578.1	-73%

Figure 6.3: Comparing AMEX and STAR across NCI fragments

Hybrid modules and minimality

We also evaluated the sizes of modules produced by the hybrid STAR-AMEX extraction procedure, which — as both AMEX and STAR are extraction procedures for each of the NCI fragments — produces a module no larger than the one extracted by either AMEX or STAR independently. This may help improve on the particular cases where STAR is slightly smaller than AMEX. So in a similar experiment we considered random signatures of 100 to 1000 concepts, and role percentages of 0%, 50% and 100%, and for each of the NCI fragments we extracted a STAR, AMEX and hybrid STAR-AMEX-module over 200 signatures for each concept/role percentage combination. In addition, we also were interested to know how close each of our approximations came in size to the minimal depleting Σ-module, so for every signature we considered we also extracted the minimal 1-depleting module.

The table in Figure 6.4 shows a summary of the results of these experiments, showing the average sizes of each of the 3 upper approximations and the lower approximation over each of the NCI fragments and for each combination of different signatures sizes. In addition in each case we give the number of signatures (out of 200) in which there was a difference between the hybrid module and the minimal 1-depleting module. It can be seen that:

- in NCI* and NCI$^\star(\sqsubseteq)$ the hybrid module almost always coincides with the minimal 1-depleting module. Thus the hybrid module almost always coincides with the minimal depleting module.
- in NCI$^\star(\equiv)$, the hybrid module coincides with the minimal 1-depleting module for approximately 50% of all signatures. Moreover, on average the minimal 1-depleting module is less than 0.3% smaller than the hybrid module.
- in all three ontologies, hybrid modules are only slightly smaller than AMEX-modules.

What we can take from this is that both AMEX and the hybrid STAR-AMEX procedures produce a successful approximation for minimal depleting Σ-modules over fragments of the NCI ontology, coinciding or coming very in size to the ideal minimal module. When the hybrid module and 1-depleting module do not coincide, we cannot say with certainty whether it is the hybrid module which over-approximates the minimal module or the 1-depleting module which under-approximates it (for acyclic $\mathcal{ALC}$ it is of course undecidable to tell if a module coincide with the minimal depleting Σ-module). The only way to establish this is to check "by hand", and we did find a few examples where the 1-depleting module was not a depleting module (Example 5.1.1 from the previous chapter is based on such a module) but the task is currently too labour intensive to establish a general pattern.

6.3.2 Full NCI

Unlike AMEX the hybrid STAR-AMEX procedure facilitates extraction from the whole NCI ontology without splitting it into fragments. The experiments shown in Figure 6.5 are based on 200 signatures, again over sizes 100 to 1000 concepts and 0%, 50% and 100% percent of all role names and compare the average size of modules extracted using STAR-extraction, hybrid extraction, and also 1-depleting module extraction.

The results are very similar to the results for NCI*. Hybrid modules are on average significantly smaller than STAR modules and are often identical to the minimal 1-depleting module (and so the minimal depleting module). In fact, over this small sample of signatures, we found no hybrid module that does not coincide with the corresponding minimal 1-depleting module.

Role%	0%					50%					100%				
$\|\Sigma\|$	Star (S)	AMEX (A)	Hybrid (H)	1-dep (D)	Diff/200	Star (S)	AMEX (A)	Hybrid (H)	1-dep (D)	Diff/200	Star (S)	AMEX (A)	Hybrid (H)	1-dep (D)	Diff/200
NCI*															
100	3834.21	722.21	710.65	671.68	10	3887.17	972.68	960.44	960.39	3	3915.18	1013.23	1000.79	1000.70	4
250	5310.96	1721.28	1705.71	1705.61	4	5452.52	1882.65	1870.87	1870.83	4	5539.39	1924.77	1912.95	1912.89	5
500	6977.33	2725.74	2700.00	2699.96	2	7186.09	2933.90	2919.23	2919.15	3	7237.22	2987.75	2977.62	2977.58	2
750	8235.36	3573.97	3542.57	3542.49	2	8437.07	3801.24	3786.05	3786.01	2	8579.98	3902.12	3892.36	3892.26	4
1000	9273.62	4341.25	4305.41	4305.38	1	9525.81	4570.55	4553.91	4553.81	4	9542.00	4621.42	4612.19	4606.46	3
NCI* ($\sqsubseteq$)															
100	58.74	69.53	58.74	58.74	0	291.91	326.68	291.91	291.89	2	345.01	357.58	345.01	344.89	5
250	330.79	386.45	330.79	330.78	1	652.09	716.64	652.09	652.09	0	775.00	808.03	775.00	775.00	0
500	852.14	1007.20	852.14	852.14	0	1173.34	1274.27	1173.34	1173.34	0	1387.67	1444.68	1387.67	1387.67	0
750	1352.47	1571.46	1352.47	1352.47	0	1681.12	1816.79	1681.12	1681.12	0	1935.47	2009.62	1935.47	1935.47	0
1000	1788.02	2046.62	1788.02	1788.02	0	2152.83	2315.19	2152.83	2152.83	0	2434.06	2519.63	2434.06	2434.06	0
NCI* ($\equiv$)															
100	2760.96	310.25	310.25	309.21	122	2759.11	319.08	319.11	318.23	114	2782.54	318.79	318.79	317.73	130
250	3989.74	622.65	622.63	621.89	110	4000.93	623.38	623.25	622.50	104	3973.78	624.51	624.23	623.47	102
500	4994.77	1003.76	1003.75	1002.95	108	4983.10	1002.14	1002.04	1001.32	101	4986.77	999.87	999.87	999.08	101
750	5539.78	1310.33	1310.31	1309.38	124	5531.60	1313.51	1311.54	1310.67	90	5525.28	1307.71	1307.71	1306.85	106
1000	5886.91	1573.06	1573.14	1572.11	122	5901.34	1577.34	1572.14	1571.10	102	5903.37	1576.95	1571.18	1570.08	103

Figure 6.4: Comparing upper and lower approximations across NCI fragments

Role%	0%				50%				100%			
$\lvert\Sigma\rvert$	Star(S)	Hybrid (H)	1-depl (D)	Diff/200	Star(S)	Hybrid (H)	1-depl (D)	Diff/200	Star(S)	Hybrid (H)	1-depl (D)	Diff/200
100	5274.7	1905.8	1905.8	0	5409.9	2010.5	2010.5	0	5452.8	2079.8	2079.8	0
250	7306.6	3269.6	3269.6	0	7329.9	3329.1	3329.1	0	7360.4	3365.4	3365.4	0
500	9477.8	4833.9	4833.9	0	9575.1	4880.0	4880.0	0	9572.7	4920.8	4920.8	0
750	11044.9	6050.5	6050.5	0	11132.7	6105.7	6105.7	0	11121.4	6133.9	6133.9	0
1000	12393.1	7117.7	7117.7	0	12440.6	7165.5	7165.5	0	12455.4	7215.3	7215.3	0

Figure 6.5: Modules of NCI

In a similar vein we also extracted modules for axiom signatures. The results of this can be seen in Figure 6.6 which is based on the signatures of 20,000 axioms randomly selected from the whole of NCI for which 13.2% of axioms showed a difference in size between the extracted hybrid and STAR-modules. The average difference, for those cases when there was a difference, was 2264.5 axioms.

Performance

We found AMEX to be incredibly efficient, over all experiments performed over the NCI fragments, a single extraction took just under 3 seconds and the maximum time taken was 15 seconds. This can be attributed to the distribution of the workload over each of AMEX's rules; interestingly in 97% of all experiments only **R1** of AMEX located any axioms to add to the module, so in those cases the module was computed purely syntactically with a single call to the QBF solver to verify that **R2** was not applicable and that the extracted module was indeed depleting. Over the remaining 3% of experiments, where **R2** was applicable and separability causing axioms were identified, the maximal number of separability axioms located for a single extraction was 4, with 73 being the maximum number of calls to the QBF solvers.

The hybrid STAR-AMEX procedure we also found to be very efficient. For any one extraction (which begins with the extraction of a STAR-module) we found the additional time needed to extract a module compared to STAR extraction alone was at most only 2.2 seconds. Over the fragments of NCI, for

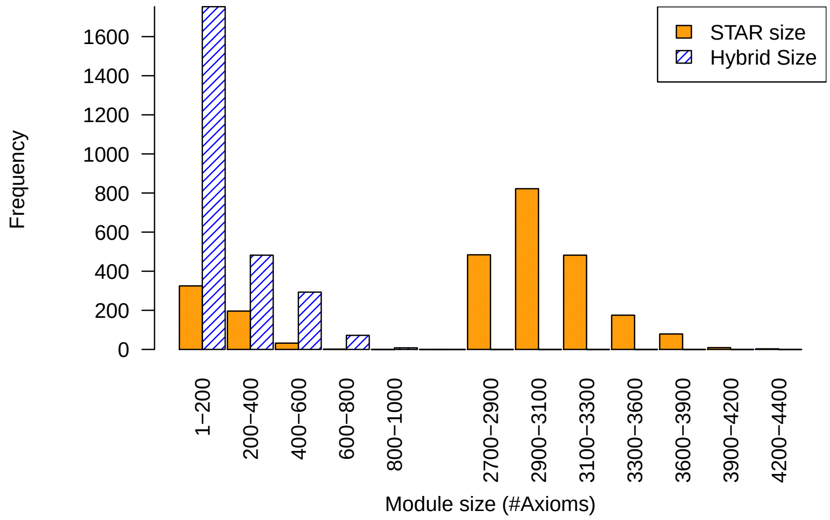


Figure 6.6: Frequency of genuine module sizes for NCI

extractions using the hybrid procedure, we saw exactly 2 alternations of STAR module extraction whereas the AMEX extractions varied between 1 and 2 times. For full NCI it was much the same pattern except for a single extraction that involved 3 alternations of the STAR procedure with 2 from AMEX. Over all experiments we found AMEX tended towards a single alternation as the signature sizes grew and the differences between the respective modules became smaller.

The computation of 1-depleting modules were also reasonably efficient over such a large ontology but still significantly slower than any of the upper approximations, even with the additional preprocessing step. The time taken to compute a 1-depleting module varied considerably over each of the NCI fragments: for NCI$^\star(\sqsubseteq)$ a single 1-depleting extraction took no more than 2 minutes, for NCI* and NCI$^\star(\equiv)$ however it took up to 30 minutes. The increased amount of time can be attributed to the number of calls to the QBF solver required to locate the 1-separability causing axioms within each of the ontologies: for NCI$^\star(\sqsubseteq)$ the maximum number of QBF calls was 5,052, for NCI* 193,993 and for NCI$^\star(\equiv)$ 433,546 calls were necessary.

We also attempted to extract the minimal 2-depleting module across the NCI fragments for those cases where the hybrid-module and the corresponding 1-

depleting module did not coincide in order to try and reduce the gap between the lower and upper approximations. This turned out to be infeasible in practice, where for some signatures a 2-depleting module was not computed after several hours of computation. That considered, even when our upper approximations did not coincide with the lower approximation they were still sufficiently close in size to be considered successful approximations even if we could not show that they were minimal.

6.4. Experiments over the experimental corpus

Of the 172 ontologies in the experimental corpus a large percentage are neither acyclic $\mathcal{ALCQI}$ nor terminologies which limits those to which we can apply the AMEX procedure. For this reason we focus on comparing STAR-modules to hybrid STAR-AMEX-modules across the ontologies of the corpus. For the small number ontologies to which AMEX can be applied, since both AMEX and STAR are subset preserving procedures, it follows from Lemma 4.1.5 that a module extracted using the hybrid STAR-AMEX procedure will be at least as small as the corresponding module extracted by AMEX procedure alone.

We decided to focus on axiom signatures only across the experimental corpus for a number of reasons. Due the both the sheer quantity and size of the ontologies in the corpus we would need to consider an huge variety of signatures of different sizes to give a meaningful result, coupled with the poor performance of extracting the lower approximation from NCI which could also be apparent over the ontologies in the corpus, such experiments would not be feasible in a reasonable amount of time. Moreover, axiom signatures give the most meaningful result as they correspond to genuine modules which are representative of *all* modules, and differences between approximations over axioms signatures can translate into large differences over random signatures as we observed in our experiments over NCI.

Our experiments involved taking each ontology of the experimental corpus in turn and extracting a STAR and hybrid STAR-AMEX-module for each axiom signature taken from that ontology. We also wanted to investigate the difference in size between an upper approximation and its corresponding minimal module, ideally increasing the lower approximation beyond 1-depleting modules to see if that would give a better estimation into to which of our upper approximations coincide with the ideal minimal. We conjecture that although it was not possible to compute a 2-depleting module for NCI in a reasonable amount of time, the ontologies of the experimental corpus are comparatively smaller, so although 2-depleting modules might take long time to compute, since we are only considering axiom signatures it should be feasible within a decent time bound.

So, in addition, to evaluate the success of the both upper approximations, for each signature considered we also extracted the corresponding minimal 2-depleting module.

6.4.1 Differences in upper approximations

We found that 66 of the 172 ontologies (~38%) of the corpus contained axioms whose signature revealed some difference in size between the STAR and corresponding hybrid modules. Figure 6.7 shows for each of the 66 ontologies which percentage of the total axiom signatures taken from the ontology was there a difference observed between the STAR and corresponding hybrid modules. For example take the Galen (GAL) ontology which contains 4,735 axioms, out of these we observed a difference in module sizes over 2,644 of them ~56%, so this ontology contributes to the bar for 50-60% in the figure.

Exploring these results in more detail we can see that at the lower end of the scale 28 out of 66 (~42%) of all ontologies only showed a difference between 1% and 10% of their total axiom signatures, the worse of these being the Terminological and Ontological Knowledge Resources Ontology (TOK) where a difference was observed on just 1 of its 370 axioms (~0.27%), at the other end of the scale we found 5 ontologies where between 90% and 100% of all axiom signatures revealed a difference in module sizes, the best of these being

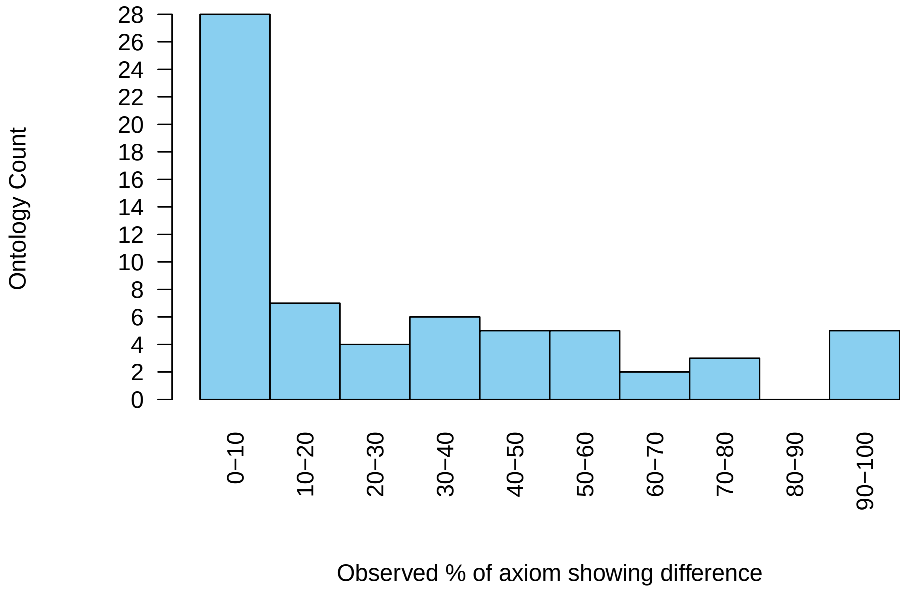


Figure 6.7: Axiom signatures showing differences between STAR and Hybrid STAR-AMEX modules

the Atom Complex (ATC) ontology where each of its 119 signatures (100%) revealed a difference in comparative module sizes.

Over the signatures where a difference was observed the hybrid modules were on average ∼35% smaller than the corresponding STAR modules. The table in Figure 6.8 shows the 38 ontologies for which the difference, when there was a difference, was more than 2 axioms. The first column is the abbreviation which is used for the ontology, the next two columns of the table show the number of axioms of each ontology and which percentage of those axioms corresponded to a signature where a difference was observed, the last four show both the average size of STAR and hybrid modules when a difference was observed and the absolute difference in size between the averages, and finally the percentage change in size the hybrid is compared to the corresponding STAR module. The results for the other 28 ontologies where a difference smaller than 2 axioms was observed are deferred to Appendix B.

It can be seen that for certain ontologies the hybrid-module is considerably

Ont	Axs	Diff	Star	Hybrid	Size Diff.	%Change
ATC	119	100.00%	119.00	1.99	117.01	-98.33%
SitBAC	464	11.42%	210.91	140.66	70.25	-33.31%
TAM	595	50.59%	129.48	75.51	53.97	-41.68%
GALO	9645	44.84%	659.54	613.86	45.68	-6.93%
ATM	89	59.55%	33.62	1.19	32.43	-96.46%
PIZ	694	95.97%	99.18	67.14	32.03	-32.30%
CHMC	103	47.57%	28.14	1.04	27.10	-96.30%
GEOS	653	93.87%	25.99	3.49	22.50	-86.58%
GAL	4735	55.84%	100.54	82.21	18.33	-18.23%
ProPreO	598	92.64%	20.97	5.95	15.02	-71.64%
PER	58	62.07%	18.64	4.83	13.81	-74.07%
EDAS	576	8.33%	37.92	27.83	10.08	-26.59%
GRO	933	50.80%	98.24	88.72	9.53	-9.70%
GRI	901	26.86%	41.61	32.46	9.15	-22.00%
AER	120	71.67%	11.98	3.22	8.76	-73.11%
PEO	70	61.43%	10.09	3.56	6.53	-64.75%
SUB2	458	73.36%	37.10	30.68	6.42	-17.31%
PAR	270	45.19%	23.31	17.48	5.83	-25.00%
SPO	678	27.73%	175.82	170.01	5.81	-3.31%
TBM	173	31.79%	11.64	5.84	5.80	-49.84%
PHOT	523	5.16%	44.48	39.63	4.85	-10.91%
OPB	1254	24.80%	9.29	4.77	4.52	-48.69%
BHO	2393	0.17%	34.00	29.75	4.25	-12.50%
BT	1152	96.79%	213.22	209.21	4.01	-1.88%
CMTC	195	1.54%	5.00	1.00	4.00	-80.00%
HRT	343	71.43%	308.94	304.96	3.98	-1.29%
SDO	2664	21.51%	301.25	297.36	3.88	-1.29%
YBC	162	17.90%	6.31	2.45	3.86	-61.20%
UNIB	89	8.99%	5.88	2.25	3.63	-61.70%
FHHO	926	6.80%	4.29	1.00	3.29	-76.67%
UNI	162	3.70%	7.00	4.17	2.83	-40.48%
CNR	168	11.90%	36.55	33.75	2.80	-7.66%
KBCF	665	13.98%	9.92	7.19	2.73	-27.52%
LiPrO	2375	3.20%	20.21	17.64	2.57	-12.70%
NUM	264	8.33%	4.91	2.45	2.45	-50.00%
DCCL	313	34.50%	26.93	24.48	2.44	-9.08%
SIO	2205	9.57%	153.84	151.50	2.35	-1.52%
MHC	287	51.22%	30.30	28.01	2.29	-7.57%

Figure 6.8: Differences between STAR and hybrid STAR-AMEX modules over axiom signatures

smaller than the STAR-module, the best result coming again from the Atom Complex (ATC) ontology, which not only showed a difference for each of its axiom signatures but the difference was both largest in terms of relative module sizes and percentage change. We looked into exactly why this was the case and it came down to again how the STAR procedure handles equivalences as previously discussed. The ATC ontology describes the periodic table of of elements and contains a single axiom of the form $\mathsf{Atom} \sqsubseteq \exists \mathsf{hasPart.Proton}$ and an axiom of the form

$$\mathsf{Element} \equiv \mathsf{Atom} \sqcap (= \langle \mathit{atomic\_number} \rangle\ \mathsf{hasPart.Proton})$$

for each of the 118 elements of periodic table, where the concept $\mathsf{Element}$ each and $\langle \mathit{atomic\_number} \rangle$ are distinct values corresponding to the name of an element and its associated atomic number e.g. $\alpha = \mathsf{Iron} \equiv \mathsf{Atom} \sqcap (= 26\ \mathsf{hasPart.Proton})$.

To see why there is such a large difference in module sizes, suppose we take $\Sigma = \mathsf{sig}(\alpha)$, then we have $\{\alpha\} \not\equiv_\Sigma \emptyset$, for any interpretation $\mathcal{I}$ where $\mathsf{Iron}^\mathcal{I} \neq (\mathsf{Atom} \sqcap (= 26\ \mathsf{hasPart.Proton}))^\mathcal{I}$ there is clearly no corresponding model of α that coincides on Σ. But if we consider another axiom of this form for the same signature, say $\beta = \mathsf{Zinc} \equiv \mathsf{Atom} \sqcap (= 30\ \mathsf{hasPart.Proton})$, since $\mathsf{Zinc} \notin \Sigma$, for any interpretation $\mathcal{I}$ we can find a model $\mathcal{J}$ of β by w.r.t $\mathsf{Zinc}^\mathcal{J} = (\mathsf{Atom} \sqcap (= 30\ \mathsf{hasPart.Proton}))^\mathcal{I}$ whilst still ensuring $\mathcal{I}|_\Sigma = \mathcal{J}|_\Sigma$. The problem in the module extraction setting, if $\mathsf{sig}(\alpha)$ is chosen, then β is neither $\top$ or $\bot$-local w.r.t Σ, nor are any of the other axioms whose signature was not considered for extraction. It's for this reason the whole ontology is extracted into a STAR-module over each of the axiom signatures.

Other particularly notable results from this table include the People (PEO) and Galen (GAL) ontologies which are well known for having disproportionally large locality-based modules [Gra+08; Del+11] for which we found on average the hybrid modules were 18.23% and 64.75% smaller than the corresponding

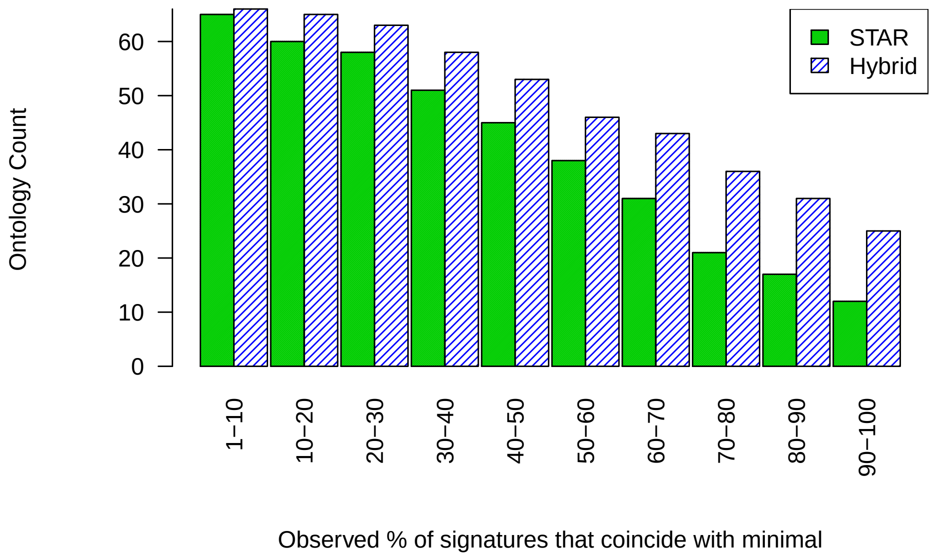
STAR modules respectively. We also saw smaller modules on average over the signatures of Tambis (TAM), miniTambis (TMB) and University (UNI) some of the other ontologies which have previously been studied in research surrounding modularity.

6.4.2 Minimality

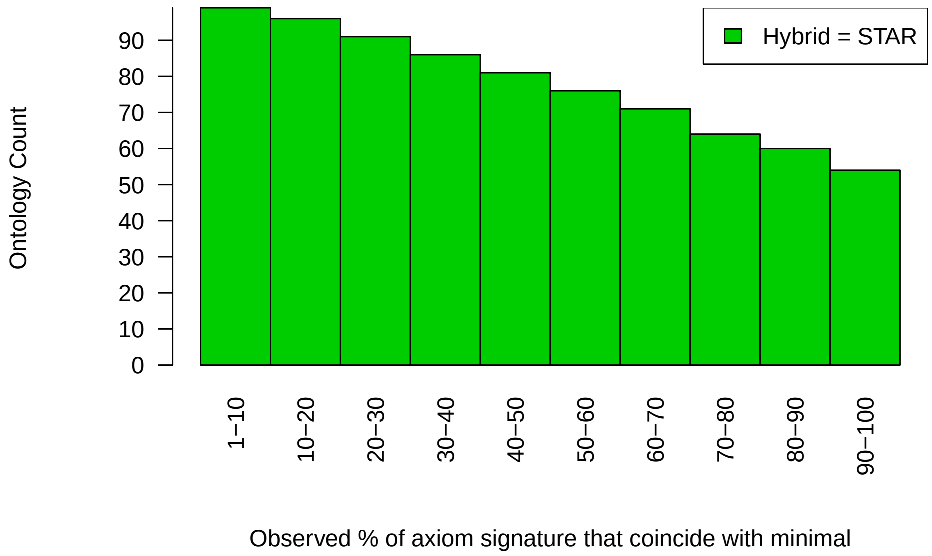
We have established which cases the hybrid modules are smaller than the corresponding STAR modules, but we also investigated into how often our upper approximations coincided with the corresponding 2-depleting modules to estimate if they are a successful approximation. Figure 6.9 shows a summary of the results from extracting 2-depleting modules.

The figures the show the frequency of ontologies where we found the upper approximations (STAR and hybrid-modules) coinciding with the lower approximation (2-depleting modules) on at least X% of their total axiom signatures. The top chart shows those 66 ontologies where a difference in size was observed between hybrid and STAR-modules, and where potentially hybrid modules can coincide more often with the lower approximation. The bottom chart shows the other 106 ontologies where the hybrid and STAR-modules always coincide.

It can be seen that for the cases where the hybrid and STAR-modules differed, there were significantly more axioms signatures where their corresponding hybrid-modules coincided with the lower approximation, and therefore the minimal depleting Σ-module. Over all axiom signatures of these ontologies the hybrid-module coincided with 2-depleting module on 6,800 more signatures than the STAR-module, and there were 36 ontologies for which over 70% of all axioms signatures taken from that ontology produced a hybrid module which coincide with the 2-depleting module, compared to just 21 in the case of STAR-modules. Moreover, there were 25 ontologies where between 90% and 100% of all axiom signatures coincided with the lower approximation, this included 7 for which the hybrid module coincided over all axiom signatures taken from that ontology, namely: Atom Common (ATM),



(a) Difference Observed



(b) No difference observed

Figure 6.9: Observed modules that coincide with minimal

Atom Complex (ATC), CMT conf (CMTC), Family Health (FHHO), GeoSkills (GEOS), Particle (PAR) and Worm Phenotype (WORM). Comparatively, out of these 66 ontologies, there were only 12 ontologies for which the STAR and 2-depleting modules coincided on over 90% of signatures, and we did not find any ontology where the STAR-modules and 2-depleting modules coincide over all axiom signatures.

Over the other ontologies, where there was no difference in the sizes of STAR and corresponding hybrid modules, it can be seen that over half (56) of ontologies saw each of the upper approximations coincide with the lower approximation on between 90% and 100% of their total axiom signatures, 39 of these coinciding on all axiom signatures. In these cases where the STAR modules are already minimal one will never see an improvement by extracting the corresponding hybrid module.

If we look at the results across all ontologies there are still a number of signatures where the upper and lower approximations were not found to coincide, again it is either the lower approximation under-approximating the minimal module and/or an upper approximation over-approximating it. In these cases where the upper and lower approximations do not coincide there is on average an 18.5 axiom difference between the hybrid-module and the corresponding 2-depleting module, but this can be as many as 237 axioms in which case we found the 2-depleting module to be over 99% smaller than the corresponding hybrid module. As we mention in our experiments over NCI, establishing why there is a difference between the upper and lower approximations can only be done by hand and is incredibly labour intensive. Further investigation may help establish exactly what is happening in these cases, and it may be the case that computing n-depleting modules for larger values of n will help close the gap between the upper and lower approximations, although extracting such modules may take a significant amount of time.

6.4.3 Performance

Over all ontologies and axiom signatures we found the STAR procedure to be efficient as expected, since all computation is done syntactically, and for any one extraction it took less than 1 second, and it most cases it was a matter of milliseconds. The additional time need to compute a hybrid module, in addition to the initial STAR extraction, was also on average very efficient over all ontologies the average time taken was well under a second. We did find a few cases where the maximum time needed for a single hybrid extraction was above average, the Inverterbrata (INV) ontology saw a single extraction taking up to 8 seconds and the Data mining (DMK) ontology saw one which took up to 53 seconds.

The encouraging performance of the hybrid procedure can again be attributed to the majority of the workload being done by the syntactic rule **R1** of the AMEX procedure. In fact, in only 49 of the 172 ontologies saw **R2** of AMEX being applicable, where a separability causing axiom needed locating after **R1** was exhaustively applied. The ontology for which **R2** was applicable the most and subsequently the most calls to the QBF solvers were used, was the myGrid (GRI) ontology for which a single extraction located up to 11 separability causing axioms using 48 calls to the QBF solvers in the process. For all signatures over the other 123 remaining ontologies the modules were computed purely syntactically with a single call to the QBF solver to ensure a depleting module was extracted.

Within the hybrid procedure those 106 ontologies where the STAR and hybrid-modules always coincided in size only saw 1 alternation of the STAR extraction procedure and 1 alternation of the AMEX extraction procedure as expected, since extracting an AMEX module from a STAR module saw no reduction in size, the fixpoint was reached and the algorithm terminated. For those 66 ontologies where there was a difference, 61 of them saw at most 2 STAR alternation and 1 AMEX alternation, the remaining 5 saw up to 3 STAR alternations and 2 AMEX alternations.

For the lower approximation, for all but 22 ontologies there was at least one signature where it was necessary to take the union of both the minimal exactly 1 and exactly 2-depleting modules to produce the lower approximation, that is where we found the exactly 1-depleting module and corresponding hybrid module did not coincide. On average the time needed to extract a 2-depleting Σ-module over all ontologies and signatures was just under 2 seconds, but for some ontologies in the worst case took significantly more time. The worst of these was Open Galen (GALO) for which a single 2-depleting module extraction took on average 21 seconds but in one case took more than 3 hours!

6.5. Conclusion

This chapter looked at answering several research questions about the comparative size, success and time to extract AMEX, hybrid STAR-AMEX modules in comparison to STAR modules.

In results we presented for NCI and its fragments, we saw that AMEX and the hybrid STAR-AMEX-procedure can offer significant reductions in the size of the depleting Σ-modules they produce in comparison to corresponding STAR modules. We also saw that these improved approximations came very close to, and often coincided with, the lower approximation we computed and therefore the minimal depleting Σ-module.

For experiments over the experimental corpus we found 66 ontologies for which the hybrid extraction procedure produced modules that were often smaller than corresponding STAR-modules over axiom signatures. In the cases where we did see a difference in size between the two module notions we found that the hybrid-modules coincided more frequently with the corresponding lower approximation. Conversely, there were some signatures for which there was no difference recorded between STAR and hybrid-module notions. This lack of difference was explainable in some cases in that the STAR approximations already coincided with the ideal minimal and were already as a successful approximation as possible, for others there was still a difference between our upper and lower approximations and would need further evaluation in order to fully understand the reason why these cases occur.

CHAPTER 7

Conclusions

7.1. Conclusions

The focus of this thesis was the development, improvement, and evaluation of algorithms which approximate minimal depleting Σ-modules.

In Chapter 3 we introduced the AMEX extraction algorithm which was developed by extending existing notions of model-theoretic Σ-inseparability to produce an approximation of the minimal depleting Σ-module for acyclic $\mathcal{ALCQI}$ terminologies which may additionally contain repeated concept inclusions. Due to the restrictive family of ontologies to which AMEX can be applied, in Chapter 4 we introduced a hybrid module extraction algorithm which generalises the already successful STAR approximation, combining two module extraction procedure together by iteratively extracting their modules from one another until a fixpoint is reached. This in turn spawned the hybrid STAR-AMEX procedure which was designed specifically to help minimise the size of the STAR approximations using AMEX but without losing the inclusive nature of the STAR procedure, so that approximations could still be extracted from general $\mathcal{SROIQ}$ ontologies. In Chapter 5 we introduced a way of evaluating how well one can approximate minimal modules, by treating each sound approximation procedure as an upper approximation we then introduced a lower approximation by the means of n-depleting Σ-modules which we proved are always contained in the minimal depleting Σ-module. The difference between an upper and lower approximation then giving an estimate to how close the upper approximation is to the ideal minimal and therefore how successful the approximation is. Finally we brought all of these results together over a large empirical investigation in chapter

Chapter 6 which was evaluated over a corpus of real-world ontologies. What was revealed is that both AMEX and hybrid STAR-AMEX modules were not only often significantly smaller than the corresponding STAR modules but also coincided with the corresponding minimal depleting Σ-modules much more frequently. In addition we found both our AMEX and hybrid extraction procedures to work very efficiently over the real-world ontologies we examined, certainly all modules we extracted were computed within a matter of seconds. Generally n-depleting modules up to 2-depleting were also computed reasonably efficiently with a few anomalous cases taking considerably longer. Such cases could become problematic if considering computing the lower approximation for higher values for n, the exponential growth of the QBF reduction may see computing the lower approximation becoming unfeasible. In summary, considering that hybrid modules are at least as small as the corresponding STAR modules, but can be significantly smaller (and often minimal) and also the fact that they can computed very efficiently we have presented strong empirical evidence to prefer hybrid STAR-AMEX modules to just STAR modules on their own.

In summary, we have developed several new algorithms for approximating minimal depleting Σ-modules in expressive description logics, and also a methodology for evaluating how successful these resulting approximations are. What was found our new approximations were very successful, that they could be computed very efficiently in practice, were often significantly smaller than the most popular rival approximation, and often coincided with the minimal depleting Σ-module.

7.2. Future Work

Comparison to datalog modules. In Section 2.4.3 we described a notion of modularity based on datalog reasoning. These modules are not depleting

Σ-modules by default but theoretically such modules can be produced using this method. When an implementation becomes available, and we are able to compute depleting Σ-module via datalog reasoning, it would be interesting to see how this approximation compares to approximations we have considered across this thesis. In addition to a direct comparison, it would also be interesting to know how far we can improve on existing approximations by using datalog modules in a hybrid approximation algorithm with either STAR or AMEX or a combination of both.

Improving approximations. By the introduction of the lower approximation we have a means of estimating how close our approximations are to the ideal minimal. Then by investigating those cases for which there is a large gap between the upper and lower approximations we could hope to gain insight into why such approximations are unsuccessful, and in turn this could help improve our approximations so that we are able to approximate minimal modules more accurately. In addition, it would also be useful to optimise the computation of the lower approximation itself, even for small values of n it often took a long time to compute an n-depleting module.

Extending experimental evaluation. Over the experimental evaluation in Chapter 6 we saw a number of ontologies from the experimental corpus for which the hybrid approximation was smaller than corresponding STAR approximation, but this difference was not always large, and the modules were only compared over axiom signatures. With more time we would seek to carry out a more extensive investigation on these ontologies, considering a larger and more varied selection of input signatures to fully understand the difference between these two approximation notions.

Appendix A

Experimental Ontologies

This appendix shows statistics for the ontologies from Chapter 6 which are used in the experimental evaluation.

The columns of the table below give the ontology's name (Name) and the abbreviation (Abbrev.) we use in the experimental evaluation, the description logic the ontology is formulated in (DL) and the number of axioms it contains (Axioms), whether or not it contains terminological cycles (Cyclic) where Y = Yes and N = No, and finally the $\mathsf{N_C}$ and $\mathsf{N_R}$ columns are a count of the concept and role names in the ontology's signature.

Name	Abbrev.	DL	Axioms	Cyclic	$\mathsf{N_C}$	$\mathsf{N_R}$
Adolena	ADO	$\mathcal{SRIQ}$	230	Y	141	16
Adverse Events	AER	$\mathcal{ALCHQ}$	120	Y	42	4
Alignment Initiative	IASTD	$\mathcal{ALCIN}$	348	Y	139	38
Allen Brain Atlas	ABA	$\mathcal{ALCI}$	3441	N	913	2
Amino Acid	AMI	$\mathcal{ALCF}$	464	Y	46	5
Animal Natural History	ANH	$\mathcal{ALUF}$	385	N	361	14
Atom Common	ATM	$\mathcal{ALCHI}$	89	N	14	5
Atom Complex	ATC	$\mathcal{ALEQ}$	119	N	120	1
Atom Primitive	ATP	$\mathcal{ALH}$	136	N	124	5
Basic Formal	BFO	$\mathcal{ALC}$	95	Y	39	0
Bilaterian	BIL	$\mathcal{ALEHI}+$	139	N	113	5
Biochemistry	BIC	$\mathcal{ALC}$	136	N	64	0
Biocode	BOC	$\mathcal{ALC}$	84	Y	41	6

Continued on next page..

...Continued from previous page

Name	Abbrev.	DL	Axioms	Cyclic	N_C	N_R
Bioinformatics Web Service	OBIws	$\mathcal{SRIQ}$	511	Y	228	24
Biological Imaging	FBbi	$\mathcal{S}$	550	N	517	1
Biological Processes	BPO	$\mathcal{SHIN}$	267	Y	69	55
Biomedical Resource	BRO	$\mathcal{ALHI}+$	80	N	0	52
biopax-level2	BOP	$\mathcal{ALCHN}$	224	Y	41	33
BioTop	BT	$\mathcal{SRI}$	1152	Y	389	82
Bleeding History	BHO	$\mathcal{ALCIF}$	2393	Y	544	33
Bone Dysplasia	BDO	$\mathcal{SIF}$	3970	Y	3663	12
Cancer	CNR	$\mathcal{ALCHF}$	168	N	88	13
Cancer Chemoprevention	CanCO	$\mathcal{ALEH}$	171	N	94	39
Cancer Research	CRM	$\mathcal{SRIQ}$	5261	Y	1755	232
Cell Behaviour	CBO	$\mathcal{SR}$	654	N	241	20
Chemical	CHM	$\mathcal{ALCH}$	90	N	48	9
Chemical Biology	CHB	$\mathcal{SHIF}$	337	N	104	33
Chemical Information	CHEMINF	$\mathcal{ALCRI}$	305	Y	220	40
Chemistry complex	CHMC	$\mathcal{ALCHQ}$	103	Y	84	14
Chemistry-primitive	CHMP	$\mathcal{ALHI}+$	171	N	158	8
Cluster Analysis	CAO	$\mathcal{SHIQ}$	438	Y	204	35
CMT conf	CMTC	$\mathcal{ALCIN}$	195	N	30	49
CMT tool	CMTT	$\mathcal{SIN}$	354	Y	68	62
Cocus	COC	$\mathcal{ALCIQ}$	161	Y	54	33
Comparative Data Anaylsis	CDAO	$\mathcal{SRIQ}$	391	Y	132	68
Conference	CONF	$\mathcal{ALCHIF}$	234	Y	59	46
Conference Management	EDAS	$\mathcal{ALCIN}$	576	N	104	30
Confious	COFI	$\mathcal{SHIN}$	262	Y	57	52
confOf	confOf	$\mathcal{SIF}$	123	Y	39	13
Cooking	COOK	$\mathcal{ALCF}$	50	Y	23	5
Costal Observation	OBOE	$\mathcal{ALCQ}$	63	Y	23	9

Continued on next page..

...Continued from previous page

Name	Abbrev.	DL	Axioms	Cyclic	N_C	N_R
Countries	CNT	$\mathcal{ALCIN}$	77	N	11	12
CRS CMT	CRS-CMT	$\mathcal{ALCIN}$	268	Y	44	64
CRS Conf1	CRS1	$\mathcal{SIF}$	192	Y	53	28
CRS Conf2	CRS2	$\mathcal{SIF}$	196	Y	53	28
CRS Dr	CRSDR	$\mathcal{ALCIF}$	59	N	14	15
CRS EKAW	CRSE	$\mathcal{SHIN}$	308	Y	88	48
CRS PCS	PCS	$\mathcal{ALCIF}$	177	Y	38	39
Cystic Fibrosis	KBCF	$\mathcal{ALCHIF}$	665	Y	408	34
Data	DAT	$\mathcal{ALHN}$	255	N	174	32
Data Mining	DMC	$\mathcal{SHIQ}$	1067	Y	468	30
Data Mining KDD	DMK	$\mathcal{SHI}$	1999	Y	263	30
Datatypes	DTO	$\mathcal{SHI}$	354	N	143	7
Dendric Cells	DCCL	$\mathcal{ALC}$	313	N	148	9
Descriptive	DOLCE	$\mathcal{SHIF}$	351	Y	37	70
Detection Mechanisms	DET	$\mathcal{ALC}$	124	N	35	2
Diagnostic	DIA	$\mathcal{ALCF}$	234	Y	96	4
Digital Assets	DAM	$\mathcal{ALUH}$	184	N	90	51
DIKB Evidence	DEVI	$\mathcal{ALCI}$	304	Y	93	37
DUL	DUL	$\mathcal{SHIN}$	568	Y	72	103
Eagle-I Research	ERO	$\mathcal{SHIF}$	4237	Y	3451	112
Earthrealm	EART	$\mathcal{ALCH}$	873	Y	557	85
Economy	ECON	$\mathcal{ALCH}$	563	N	332	38
Ekaw2	EKAW2	$\mathcal{SHIN}$	390	Y	112	46
Evidence and Conclusion	ECO	$\mathcal{ALE}$	363	N	283	1
Expression	EXP	$\mathcal{ALCHI}$	176	Y	38	34
Family Health	FHHO	$\mathcal{ALCHIF}$	930	N	238	431
Family Tree	FLT	$\mathcal{SRIF}$	157	N	12	52
Galen	GAL	$\mathcal{ALEHIF}+$	4735	Y	2748	413

Continued on next page..

...Continued from previous page

Name	Abbrev.	DL	Axioms	Cyclic	N_C	N_R
Gene	GO	$\mathcal{AL}$	158	N	161	0
Gene Regulation	GRO	$\mathcal{ALCHIQ}$	823	Y	420	18
GeoSkills	GEOS	$\mathcal{ALCHIN}$	654	N	589	19
Heart	HRT	$\mathcal{SHI}$	343	Y	75	29
Homology	HOM	$\mathcal{ALC}$	83	N	65	0
Hospital Equiptment	HOSP	$\mathcal{ALC}$	50	N	29	0
Human Activities	HUM	$\mathcal{AL}$	163	N	157	7
Image Quality	IDQA	$\mathcal{ALRIF}$	220	N	179	16
Immunogenetics	IMGT	$\mathcal{ALCIN}$	2260	Y	286	4
Information Artifact	IAO	$\mathcal{ALRIF}+$	374	N	174	50
Information Exchange	IEDM	$\mathcal{ALUN}$	655	N	195	222
Inverterbrata	INV	$\mathcal{ALCRIF}$	2096	N	557	32
ISO	ISO	$\mathcal{ALIN}$	124	N	41	22
JERM Systems	JERM	$\mathcal{SHI}$	443	Y	263	20
Knowledge Acquisition 1	KA1	$\mathcal{AL}$	166	Y	96	60
Knowledge Acquisition 2	KA2	$\mathcal{AL}$	166	Y	96	60
Lipid	LiPrO	$\mathcal{ALCHIN}$	2375	Y	716	46
Major Histocompatability Complex	MHC	$\mathcal{ALCIQ}$	287	Y	118	7
Medically Related	OMRSE	$\mathcal{ALCHIQ}$	86	N	74	6
Menelas Top	MTOP	$\mathcal{ALCH}$	1381	N	524	298
Micro	MIC	$\mathcal{ALCIN}$	92	Y	32	17
MicroRNA Targetting	OMIT	$\mathcal{ALCHIQ}$	808	Y	376	27
miniTAMBIS	TBM	$\mathcal{ALCN}$	173	Y	178	35
Smoking Behaviour Risk	SBRO	$\mathcal{ALEI}+$	185	N	121	12
Molecular Function	MOLF	$\mathcal{ALE}$	315	N	632	3
Movie	MOV	$\mathcal{ALCN}$	140	N	57	21
myGrid	GRI	$\mathcal{SHIN}$	902	Y	543	66

Continued on next page..

...Continued from previous page

Name	Abbrev.	DL	Axioms	Cyclic	N_C	N_R
myGrid Simple	GRIS	$\mathcal{ALCHIF}$	1959	Y	475	8
myReview	REV	$\mathcal{ALCIN}$	324	Y	39	49
Neomark	NEO	$\mathcal{ALCHQ}$	1212	Y	55	105
Neural ElectroMagnetic	NEMO	$\mathcal{SHIQ}$	2468	Y	1674	89
New Upper Level	NULO	$\mathcal{AL}$	100	N	11	50
NIF Cell	NIFC	$\mathcal{S}$	399	N	374	1
Nif Subcell	NIFS	$\mathcal{ALC}$	890	N	408	3
Normal	NORM	$\mathcal{SHI}$	137	Y	68	22
Numerics	NUM	$\mathcal{SI}$	264	N	130	38
OpenGalen	GALO	$\mathcal{ALEHIF}$	9645	Y	4699	922
Oral Cancer (NEO)	ORO	$\mathcal{SHIQ}$	399	Y	325	26
Paperdyne	PAP	$\mathcal{ALCHIN}$	338	Y	48	58
Parasite Lifecycle	OPL	$\mathcal{SHIF}$	860	N	360	12
Particle	PAR	$\mathcal{ALCQ}$	270	N	73	5
PCS	PCS	$\mathcal{ALCIF}$	106	N	24	24
PCS Conf	PCSC	$\mathcal{SIF}$	243	Y	62	37
PCS EKAW	PCSE	$\mathcal{SHIN}$	361	Y	97	57
People	PEO	$\mathcal{ALCHIN}$	70	Y	60	13
Periodic Table	PER	$\mathcal{ALU}$	58	N	165	0
Phama Primitive	PHARP	$\mathcal{ALHI}+$	50	N	31	15
Pharma Complex	PHARC	$\mathcal{ALC}$	256	Y	145	3
Pharmacogenomic Rel.	PHARE	$\mathcal{ALCHI}$	450	Y	228	73
phenomena	PHEN	$\mathcal{ALUH}$	395	Y	317	35
Phenotype	RPO	$\mathcal{ALF}$	2029	N	1544	157
Phenotypic	PATO	$\mathcal{SH}$	1989	N	1480	17
photography	PHOT	$\mathcal{SRIQ}$	527	Y	170	25
Phylogenetic Ontology	PhylOnt	$\mathcal{ALCH}$	224	Y	148	17

Continued on next page..

...Continued from previous page

Name	Abbrev.	DL	Axioms	Cyclic	N_C	N_R
Physical Medicine and Rehabilitation	PMR	$\mathcal{ALU}$	163	N	137	14
Physics for Biology	OPB	$\mathcal{ALCHIQ}$	1254	Y	679	33
Pipeline Infrastructure	CPTAC	$\mathcal{ALC}$	855	N	19	342
Pizza	PIZ	$\mathcal{SHIN}$	694	N	99	7
Plant	PO	$\mathcal{SHIF}$	157	N	50	43
Pol	POL	$\mathcal{ALCIF}$	75	Y	21	10
Process1	PRO1	$\mathcal{ALCH}$	2054	Y	1514	98
Process2	PRO2	$\mathcal{ALUH}$	160	N	152	4
Property	PRO	$\mathcal{AL}$	375	N	343	20
Property Complex	PROC	$\mathcal{ALE}$	156	N	155	2
Proteomics Data	ProPreO	$\mathcal{SHIN}$	598	Y	399	32
Protein	PROT	$\mathcal{ALCF}$	306	Y	45	50
Quantitative Imaging	QIBO	$\mathcal{ALUIF}$	788	N	619	54
Reaction	REA	$\mathcal{ALCHIQ}$	96	N	40	22
Relative Places	REL	$\mathcal{SHIF}$	79	Y	7	16
RNA	RNAO	$\mathcal{SRIQ}$	585	Y	203	106
Semantic Integration	SIO	$\mathcal{SRIQ}$	2206	Y	1336	201
Sequence	SEQ	$\mathcal{SHI}$	1943	N	1576	19
SIGKDD	SIGKDD	$\mathcal{ALCIF}$	167	Y	64	32
SIGKDD-EKAW	SIGK	$\mathcal{SHIN}$	345	Y	123	50
SIGKDD2	sigkdd2	$\mathcal{ALEI}$	94	N	50	17
Situation Based Access	SitBAC	$\mathcal{ALCN}$	464	Y	178	40
Skin Physiology	SPO	$\mathcal{ALERIF}+$	678	Y	339	34
Sleep Domain	SDO	$\mathcal{SHIQ}$	2666	Y	1374	75
Software1	SOF1	$\mathcal{ALHN}$	61	N	18	25
Software2	SOF2	$\mathcal{ALCHIQ}$	2095	Y	735	15
Spatial	BSPO	$\mathcal{ALEHI}+$	262	Y	128	32

Continued on next page..

...Continued from previous page

Name	Abbrev.	DL	Axioms	Cyclic	N_C	N_R
Spatial	SPA	$\mathcal{ALEH}+$	190	N	106	13
Statistics	STAT	$\mathcal{ALCHIN}$	171	N	123	7
Student Health	SHR	$\mathcal{ALH}$	414	N	344	36
Study Design	STUD	$\mathcal{ALC}$	132	Y	86	1
Subatomic Particle	SUBA	$\mathcal{ALC}$	75	N	52	1
Subcelular Anatomy1	SAO1	$\mathcal{SHIF}$	2524	Y	736	36
Subcelular Anatomy2	SAO2	$\mathcal{SHIF}$	2659	Y	795	40
Substance1	SUB1	$\mathcal{ALUH}$	491	N	351	11
Substance2	SUB2	$\mathcal{ALCF}$	458	N	81	3
Syndromic Surveillance	SSO	$\mathcal{ALIF}$	197	N	171	11
Tambis	TAM	$\mathcal{SHIN}$	595	Y	395	100
Terminological and Ontological Knowledge Resources	TOK	$\mathcal{SRIQ}$	370	Y	193	85
Time Entry	TIME	$\mathcal{SHIN}$	86	Y	18	28
Time Modification	TIMM	$\mathcal{ALUHIF}$	80	N	48	14
Transportation	TRAN	$\mathcal{ALCH}$	923	Y	429	77
University	UNI	$\mathcal{ALEI}+$	184	N	161	9
University Bench	UNIB	$\mathcal{ALEHI}+$	89	N	43	25
Variables and Values	OoEVV	$\mathcal{ALU}$	76	N	34	28
Vertebrate Skeletal Anatomy	VSAO	$\mathcal{ALERI}+$	457	Y	273	12
VIVO	VIVO	$\mathcal{ALEHIN}+$	638	N	165	190
Worm Phenotype	WORM	$\mathcal{ALE}$	1173	N	1841	30
Yeas Biology Primitive	YBP	$\mathcal{ALCI}$	147	N	140	10
Yeast Biology Complex	YBC	$\mathcal{ALCH}$	162	N	106	17

Appendix B

Experimental Results : Comparing Upper Approximations

This appendix shows the full experimental result which were presented in Section 6.4.1 in which the size of STAR and corresponding hybrid STAR-AMEX-modules were compared over every axiom signature for each ontology in the experimental corpus.

This table is the extended version of Figure 6.8 which shows the relative sizes of the two module extraction procedures for signatures when difference in size was observed. The columns have the same meaning as those in the original figure.

Ont	Axs	Diff	Star	Hybrid	Size Diff.	%Change
AER	120	71.67%	11.98	3.22	8.76	-73.11%
ATM	89	59.55%	33.62	1.19	32.43	-96.46%
ATC	119	100.00%	119.00	1.99	117.01	-98.33%
BT	1152	96.79%	213.22	209.21	4.01	-1.88%
BHO	2393	0.17%	34.00	29.75	4.25	-12.50%
BDO	3970	0.03%	2.00	1.00	1.00	-50.00%
CNR	168	11.90%	36.55	33.75	2.80	-7.66%
CRM	5255	1.58%	83.01	82.01	1.00	-1.20%
CHEMINF	305	3.93%	6.92	5.50	1.42	-20.48%
CHMC	103	47.57%	28.14	1.04	27.10	-96.30%

Continued on next page..

...Continued from previous page

Ont	Axs	Diff	Star	Hybrid	Size Diff.	%Change
CMTC	195	1.54%	5.00	1.00	4.00	-80.00%
CDAO	390	36.15%	56.24	55.24	1.00	-1.78%
CONF	234	1.71%	21.00	20.00	1.00	-4.76%
EDAS	576	8.33%	37.92	27.83	10.08	-26.59%
KBCF	665	13.98%	9.92	7.19	2.73	-27.52%
DCCL	313	34.50%	26.93	24.48	2.44	-9.08%
DEVI	304	5.92%	18.17	16.94	1.22	-6.73%
EART	873	18.79%	3.12	1.93	1.20	-38.28%
FHHO	926	6.80%	4.29	1.00	3.29	-76.67%
FLT	157	4.46%	18.29	17.29	1.00	-5.47%
GAL	4735	55.84%	100.54	82.21	18.33	-18.23%
GOSLIM	158	48.10%	3.00	2.00	1.00	-33.33%
GRO	933	50.80%	98.24	88.72	9.53	-9.70%
GEOS	653	93.87%	25.99	3.49	22.50	-86.58%
HRT	343	71.43%	308.94	304.96	3.98	-1.29%
LiPrO	2375	3.20%	20.21	17.64	2.57	-12.70%
MHC	287	51.22%	30.30	28.01	2.29	-7.57%
TBM	173	31.79%	11.64	5.84	5.80	-49.84%
GRI	901	26.86%	41.61	32.46	9.15	-22.00%
NEMO	2468	2.19%	285.00	284.00	1.00	-0.35%
NUM	264	8.33%	4.91	2.45	2.45	-50.00%
GALO	9645	44.84%	659.54	613.86	45.68	-6.93%
PAR	270	45.19%	23.31	17.48	5.83	-25.00%
PEO	70	61.43%	10.09	3.56	6.53	-64.75%
PER	58	62.07%	18.64	4.83	13.81	-74.07%
PHARE	448	6.70%	2.00	1.00	1.00	-50.00%
PHEN	395	4.05%	2.13	1.13	1.00	-47.06%
PHOT	523	5.16%	44.48	39.63	4.85	-10.91%

Continued on next page..

...Continued from previous page

Ont	Axs	Diff	Star	Hybrid	Size Diff.	%Change
OPB	1254	24.80%	9.29	4.77	4.52	-48.69%
PIZ	694	95.97%	99.18	67.14	32.03	-32.30%
PO	157	1.27%	2.00	1.00	1.00	-50.00%
PRO1	2048	9.42%	2.98	1.64	1.34	-45.04%
PRO2	160	16.25%	2.81	1.38	1.42	-50.68%
PRO	375	4.27%	2.00	1.00	1.00	-50.00%
ProPreO	598	92.64%	20.97	5.95	15.02	-71.64%
REA	96	32.29%	24.81	23.45	1.35	-5.46%
RNAO	585	5.81%	84.94	82.94	2.00	-2.35%
SIO	2205	9.57%	153.84	151.50	2.35	-1.52%
SEQ	1943	2.62%	21.22	19.61	1.61	-7.58%
SIGKKD2	94	9.57%	7.33	6.33	1.00	-13.64%
SitBAC	464	11.42%	210.91	140.66	70.25	-33.31%
SPO	678	27.73%	175.82	170.01	5.81	-3.31%
SDO	2664	21.51%	301.25	297.36	3.88	-1.29%
SOF2	2095	18.47%	2.70	1.31	1.39	-51.39%
STAT	171	34.50%	23.32	21.95	1.37	-5.89%
SUBA	75	6.67%	3.40	1.60	1.80	-52.94%
SUB1	491	46.84%	2.75	1.58	1.17	-42.56%
SUB2	458	73.36%	37.10	30.68	6.42	-17.31%
TAM	595	50.59%	129.48	75.51	53.97	-41.68%
TOK	370	0.27%	3.00	1.00	2.00	-66.67%
UNI	162	3.70%	7.00	4.17	2.83	-40.48%
UNIB	89	8.99%	5.88	2.25	3.63	-61.70%
VIVO	638	0.63%	3.50	2.00	1.50	-42.86%
WORM	1173	33.50%	2.52	1.09	1.42	-56.57%
YBC	162	17.90%	6.31	2.45	3.86	-61.20%

Bibliography

[AHV95] Serge Abiteboul, Richard Hull and Victor Vianu, eds. *Foundations of Databases: The Logical Level*. 1st. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1995. ISBN: 0201537710.

[Baa+03] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi and Peter F. Patel-Schneider, eds. *The Description Logic Handbook: Theory, Implementation, and Applications*. New York, NY, USA: Cambridge University Press, 2003. ISBN: 0521781760.

[BBL05] Franz Baader, Sebastian Brandt and Carsten Lutz. "Pushing the EL Envelope". In: *Proceedings of the 19th International Joint Conference on Artificial Intelligence*. IJCAI'05. Edinburgh, Scotland: Morgan Kaufmann Publishers Inc., 2005, pp. 364–369.

[BCH06a] Jie Bao, Doina Caragea and Vasant G. Honavar. "Modular Ontologies – a Formal Investigation of Semantics and Expressivity". In: *Proceedings of the First Asian Conference on The Semantic Web*. ASWC'06. Beijing, China: Springer-Verlag, 2006, pp. 616–631. ISBN: 3-540-38329-8, 978-3-540-38329-1.

[BCH06b] Jie Bao, Doina Caragea and Vasant G Honavar. "On the semantics of linking and importing in modular ontologies". In: *In International Semantic Web Conference (ISWC*. Springer, 2006, pp. 72–86.

[Ben04] M. Benedetti. *sKizzo: a QBF Decision Procedure based on Propositional Skolemization and Symbolic Reasoning*. Tech. rep. 04-11-03. ITC-irst, 2004.

[Bie+09] A. Biere, A. Biere, M. Heule, H. van Maaren and T. Walsh. *Handbook of Satisfiability: Volume 185 Frontiers in Artificial Intelligence and Applications*. Amsterdam, The Netherlands, The Netherlands: IOS Press, 2009. ISBN: 1586039296, 9781586039295.

[BL99a] Hans K. Buning and T. Letterman. *Propositional Logic: Deduction and Algorithms*. New York, NY, USA: Cambridge University Press, 1999. ISBN: 0521630177.

[BL99b] Hans Kleine Büning and Theodor Lettmann. *Propositional Logic: Deduction and Algorithms (Cambridge Tracts in Theoretical Computer Science)*. Cambridge University Press, 1999. ISBN: 0521630177.

[BM07] John-Adrian Bondy and U. S. R. Murty. *Graph theory*. Graduate texts in mathematics. OHX. New York, London: Springer, 2007. ISBN: 9781846289699.

[BS03] Alex Borgida and Luciano Serafini. *Distributed Description Logics: Assimilating Information from Peer Sources*. 2003.

[BS08] Franz Baader and Boontawee Suntisrivaraporn. "Debugging SNOMED CT using axiom pinpointing in the description logic $\mathcal{EL}++$". In: *Proceedings of the International Conference on Representing and Sharing Knowledge Using SNOMED (KR-MED 2008), Phoenix, Arizona*. 2008.

[Cal+07] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini and Riccardo Rosati. "Tractable Reasoning and Efficient Query Answering in Description Logics: The *DL-Lite* Family". In: *J. Autom. Reasoning* 39.3 (2007), pp. 385–429.

[Cla10] Optimising Ontology Classification. "Birte Glimm and Ian Horrocks and Boris Motik and Giorgos Stoilos". In: *Proc. of the 9th Int. Semantic Web Conf. (ISWC 2010)*. Ed. by Peter F. Patel-Schneider, Yue Pan, Pascal Hitzler, Peter Mika, Lei Zhang, Jeff Z. Pan, Ian

Horrocks and Birte Glimm. Vol. 6496. LNCS. Shanghai, China: Springer, Nov. 2010, pp. 225–240.

[CPS06] Bernardo Cuenca Grau, Bijan Parsia and Evren Sirin. “Combining OWL Ontologies Using $\mathcal{E}$-Connections”. In: *Web Semantics* 4.1 (Jan. 2006), pp. 40–59. ISSN: 1570-8268.

[Dan+01] Evgeny Dantsin, Thomas Eiter, Georg Gottlob and Andrei Voronkov. “Complexity and Expressive Power of Logic Programming”. In: *ACM Comput. Surv.* 33.3 (2001), pp. 374–425. ISSN: 0360-0300.

[Deg+08] Kirill Degtyarenko, Paula de Matos, Marcus Ennis, Janna Hastings, Martin Zbinden, Alan McNaught, Rafael Alcántara, Michael Darsow, Mickaël Guedj and Michael Ashburner. “ChEBI: a database and ontology for chemical entities of biological interest”. In: *Nucleic Acids Research* 36.suppl 1 (2008), pp. D344–D350.

[Del+11] Chiara Del Vescovo, Bijan Parsia, Ulrike Sattler and Thomas Schneider. “The Modular Structure of an Ontology: Atomic Decomposition and Module Count.” In: *WoMO*. 2011, pp. 25–39.

[Del+12] Chiara Del Vescovo, Pavel Klinov, Bijan Parsia, Uli Sattler, Thomas Schneider and Dmitry Tsarkov. “Syntactic vs. Semantic Locality: How Good Is a Cheap Approximation?” In: *CoRR* abs/1207.1641 (2012).

[Dij+76] Edsger Wybe Dijkstra, Edsger Wybe Dijkstra, Edsger Wybe Dijkstra and Edsger Wybe Dijkstra. *A discipline of programming*. Vol. 1. prentice-hall Englewood Cliffs, 1976.

[EB05] Niklas Eén and Armin Biere. “Effective Preprocessing in SAT Through Variable and Clause Elimination”. In: *Proceedings of the 8th International Conference on Theory and Applications of Satisfiability Testing*. SAT’05. St Andrews, UK: Springer-Verlag, 2005, pp. 61–75. ISBN: 3-540-26276-8, 978-3-540-26276-3.

[Gan+02] Aldo Gangemi, Nicola Guarino, Claudio Masolo, Alessandro Oltramari and Luc Schneider. "Sweetening Ontologies with DOLCE". In: *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web*. EKAW '02. London, UK, UK: Springer-Verlag, 2002, pp. 166–181. ISBN: 3540442685.

[GF95] M. Grüninger and M. Fox. "Methodology for the Design and Evaluation of Ontologies". In: *IJCAI'95, Workshop on Basic Ontological Issues in Knowledge Sharing, April 13, 1995*. 1995.

[GKW13] William Gatens, Boris Konev and Frank Wolter. "Module Extraction for Acyclic Ontologies". In: *WoMO*. Vol. 1081. CEUR Workshop Proceedings. CEUR-WS.org, 2013.

[GKW14] William Gatens, Boris Konev and Frank Wolter. "Lower and Upper Approximations for Depleting Modules of Description Logic Ontologies." In: *ECAI*. 2014, pp. 345–350.

[GLW06] S. Ghilardi, C. Lutz and F. Wolter. "Did I Damage my Ontology? A Case for Conservative Extensions in Description Logics". In: *Proceedings of the Tenth International Conference on Principles of Knowledge Representation and Reasoning (KR'06)*. Ed. by Patrick Doherty, John Mylopoulos and Christopher Welty. AAAI Press, 2006, pp. 187–197.

[Gol+11] J. Golbeck, G. Fragoso, F. Hartel, J. Hendler, J. Oberthaler and B. Parsia. "The National Cancer Institute's Thesaurus and Ontology". In: *Web Semantics: Science, Services and Agents on the World Wide Web* 1.1 (2011), pp. 75–80.

[GPS12a] Rafael S. Gonçalves, Bijan Parsia and Ulrike Sattler. "Concept-Based Semantic Difference in Expressive Description Logics". In: *Proceedings of the 2012 International Workshop on Description Logics, DL-2012, Rome, Italy, June 7-10, 2012*. 2012.

[GPS12b] Rafael S. Gonçalves, Bijan Parsia and Ulrike Sattler. "Performance Heterogeneity and Approximate Reasoning in Description Logic Ontologies." In: ed. by 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. Lecture Notes in Computer Science. Springer, 2012, pp. 82–98.

[Gra+07] Bernardo Cuenca Grau, Ian Horrocks, Yevgeny Kazakov and Ulrike Sattler. "Just the right amount: extracting modules from ontologies". In: *Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May 8-12, 2007*. 2007, pp. 717–726.

[Gra+08] Bernardo Cuenca Grau, Ian Horrocks, Yevgeny Kazakov and Ulrike Sattler. "Modular Reuse of Ontologies: Theory and Practice". In: *J. Artif. Intell. Res. (JAIR)* 31 (2008), pp. 273–318.

[Gra+10] Bernardo Cuenca Grau, Christian Halaschek-Wiener, Yevgeny Kazakov and Boontawee Suntisrivaraporn. "Incremental Classification of Description Logics Ontologies". In: *JAR* 44.4 (2010), pp. 337–369.

[Gro04] OWL Working Group. *OWL Web Ontology Language Guide*. https://www.w3.org/TR/owl-guide/. Accessed: 2016-05-06. 2004.

[GSG04] Pierre Grenon, Barry Smith and Louis Goldberg. "Biodynamic Ontology: Applying BFO in the Biomedical Domain". In: IOS Press, 2004, pp. 20–38.

[GY05] Jonathan L. Gross and Jay Yellen. *Graph Theory and Its Applications, Second Edition (Discrete Mathematics and Its Applications)*. Chapman & Hall/CRC, 2005. ISBN: 158488505X.

[Haa+12] Volker Haarslev, Kay Hidde, Ralf Möller and Michael Wessel. "The RacerPro knowledge representation and reasoning system". In: *Semantic Web Journal* 3.3 (2012), pp. 267–277.

[HB11] M. Horridge and S. Bechhofer. “The OWL API: A Java API for OWL ontologies”. In: *Semantic Web* 2.1 (2011), pp. 11–21.

[Hit+09] Pascal Hitzler, Markus Krötzsch, Bijan Parsia, Peter F. Patel-Schneider and Sebastian Rudolph, eds. *OWL 2 Web Ontology Language: Primer*. Available at `http://www.w3.org/TR/owl2-primer/`. W3C Recommendation, 27 October 2009.

[HKH08] Riikka Henriksson, Tomi Kauppinen and Eero Hyvönen. “Core Geographical Concepts: Case Finnish Geo-Ontology”. In: Apr. 2008.

[HKS06] Ian Horrocks, Oliver Kutz and Ulrike Sattler. “The Even More Irresistible $\mathcal{SROIQ}$”. In: *Proc. of the 10th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR 2006)*. AAAI Press, 2006, pp. 57–67. ISBN: 9781577352716.

[Hoe+07] Rinke Hoekstra, Joost Breuker, Marcello Di Bello and Er Boer. “The LKIF Core ontology of basic legal concepts”. In: *In Pompeu Casanovas, Maria Angela Biasiotti, Enrico Francesconi, and Maria Teresa Sagri, editors, Proceedings of the Workshop on Legal Ontologies and Artificial Intelligence Techniques (LOAIT 2007*. 2007.

[HT02] Ian Horrocks and Sergio Tessaris. “Querying the Semantic Web: a Formal Approach”. In: *Proc. of the 13th Int. Semantic Web Conf. (ISWC 2002), number 2342 in Lecture Notes in Computer Science*. Springer-Verlag, 2002, pp. 177–191.

[Jim+08] Ernesto Jimenez-Ruiz, Bernardo Cuenca Grau, Thomas Schneider, Ulrike Sattler and Rafael Berlanga. “Safe and Economic re-use of ontologies: a logic-based methodology and tool support”. In: *OWLEd 2008, Proceedings of the 4th International Workshop: OWL Experiences and Directions, April 1-2, 2008*. Another version of this paper will also appear in Procedings of the 21st Description Logics Workshop (DL-2008). 2008.

[Kal+06a] Aditya Kalyanpur, Bijan Parsia, Evren Sirin and Bernardo Cuenca-Grau. "Repairing Unsatisfiable Concepts in OWL Ontologies". In: *Proceedings of the 3rd European Conference on The Semantic Web: Research and Applications*. ESWC'06. Budva, Montenegro: Springer-Verlag, 2006, pp. 170–184. ISBN: 3-540-34544-2, 978-3-540-34544-2.

[Kal+06b] Aditya Kalyanpur, Bijan Parsia, Evren Sirin, Bernardo Cuenca Grau and James Hendler. "Swoop: A Web Ontology Editing Browser". In: *Web Semant.* 4.2 (June 2006), pp. 144–153. ISSN: 1570-8268.

[Kaz08] Yevgeny Kazakov. "$\mathcal{RIQ}$ and $\mathcal{SROIQ}$ Are Harder than $\mathcal{SHOIQ}$". In: *KR*. AAAI Press, 2008, pp. 274–284.

[Kon+08a] Boris Konev, Carsten Lutz, Dirk Walther and Frank Wolter. "Semantic Modularity and Module Extraction in Description Logics". In: *ECAI 2008 - 18th European Conference on Artificial Intelligence, Patras, Greece, July 21-25, 2008, Proceedings*. 2008, pp. 55–59.

[Kon+08b] Boris Konev, Carsten Lutz, Dirk Walther and Frank Wolter. "Semantic Modularity and Module Extraction in Description Logics". In: *ECAI*. 2008, pp. 55–59.

[Kon+09a] Boris Konev, Carsten Lutz, Dirk Walther and Frank Wolter. "Formal Properties of Modularisation". In: *Modular Ontologies: Concepts, Theories and Techniques for Knowledge Modularization*. 2009, pp. 25–66.

[Kon+09b] R. Kontchakov, L. Pulina, U. Sattler, T. Schneider, P. Selmer, F. Wolter and M. Zakharyaschev. "Minimal Module Extraction from DL-lite Ontologies Using QBF Solvers". In: *Proceedings of the 21st International Jont Conference on Artifical Intelligence*. IJCAI'09. Pasadena, California, USA: Morgan Kaufmann Publishers Inc., 2009, pp. 836–841.

[Kon+13] Boris Konev, Carsten Lutz, Dirk Walther and Frank Wolter. “Model-theoretic inseparability and modularity of description logic ontologies”. In: *Artif. Intell.* 203 (2013), pp. 66–103.

[KS13] Patrick Koopmann and Renate Schmidt. “Forgetting Concept and Role Symbols in ALCH-Ontologies”. In: *Logic in Programming, Artificial Intelligence and Reasoning (LPAR)*. Springer, 2013.

[Kut04] Oliver Kutz. “$\mathcal{E}$-Connections and Logics of Distance”. PhD thesis. The University of Liverpool, 2004.

[KWZ10] Roman Kontchakov, Frank Wolter and Michael Zakharyaschev. “Logic-based ontology comparison and module extraction, with an application to DL-Lite”. In: *Artif. Intell.* 174.15 (2010), pp. 1093–1141.

[LB10a] Jens Lehmann and Lorenz Bühmann. “ORE - A Tool for Repairing and Enriching Knowledge Bases”. In: *Proceedings of the 9th International Semantic Web Conference (ISWC2010)*. Lecture Notes in Computer Science. Springer, 2010, pp. 177–193.

[LB10b] Florian Lonsing and Armin Biere. “DepQBF: A Dependency-Aware QBF Solver.” In: *JSAT* 7.2-3 (2010), pp. 71–76.

[LK13] Michel Ludwig and Boris Konev. “Towards Practical Uniform Interpolation and Forgetting for $\mathcal{ALC}$ TBoxes”. In: *Proceedings of the 26th International Workshop on Description Logics (DL-2013)*. (Ulm, Germany). Vol. 1014. CEUR-WS. 2013, pp. 377–389.

[LK14] Michel Ludwig and Boris Konev. “Practical Uniform Interpolation and Forgetting for ALC TBoxes with Applications to Logical Difference”. In: *Proceedings of the 14th International Conference on Principles of Knowledge Representation and Reasoning (KR’14)*. Ed. by Chitta Baral, Giuseppe De Giacomo and Thomas Eiter. AAAI Press, 2014.

[LPW11] Carsten Lutz, Robert Piro and Frank Wolter. “Description Logic TBoxes: Model-theoretic Characterizations and Rewritability”. In: *CoRR* (2011).

[LST13] Birger Lantow, Kurt Sandkuhl and Vladimir Tarasov. “Selecting Content Ontology Design Patterns for Ontology Quality Improvement”. In: *Proceedings of the 6th International Workshop on Information Logistics, Knowledge Supply and Ontologies in Information Systems, Warzaw, Poland, September 23rd, 2013*. 2013, pp. 68–79.

[LW10] Carsten Lutz and Frank Wolter. “Deciding inseparability and conservative extensions in the description logic EL”. In: *J. Symb. Comput.* 45.2 (2010), pp. 194–228.

[LWW07] Carsten Lutz, Dirk Walther and Frank Wolter. “Conservative Extensions in Expressive Description Logics”. In: *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007*. 2007, pp. 453–458.

[NBM13] Riku Nortje, Katarina Britz and Thomas Meyer. “Reachability Modules for the Description Logic $\mathcal{SRIQ}$.” In: *LPAR*. Ed. by Kenneth L. McMillan, Aart Middeldorp and Andrei Voronkov. Vol. 8312. Lecture Notes in Computer Science. Springer, 2013, pp. 636–652. ISBN: 9783642452208.

[NK10] Immanuel Normann and Oliver Kutz. “Ontology Reuse and Exploration via Interactive Graph Manipulation”. In: *Proceedings of the 1st Workshop On Semantic Repositories For The Web, SERES-2010*. 2010.

[NM03] Natalya F. Noy and Mark A. Musen. “The PROMPT Suite: Interactive Tools for Ontology Merging and Mapping”. In: *Int. J. Hum.-Comput. Stud.* 59.6 (Dec. 2003), pp. 983–1024. ISSN: 1071-5819.

[RGH12] Ana Armas Romero, Bernardo Cuenca Grau and Ian Horrocks. "MORe: Modular Combination of OWL Reasoners for Ontology Classification". In: *Proceedings of the 11th International Semantic Web Conference (ISWC 2012)*. LNCS. Springer, 2012.

[Ric+14] Mariela Rico, María Laura Caliusco, Omar Chiotti and María Rosa Galli. "OntoQualitas: A framework for ontology quality assessment in information interchanges between heterogeneous systems." In: *Computers in Industry* 65.9 (2014), pp. 1291–1300.

[Rom+14] Ana Armas Romero, Mark Kaminski, Bernardo Cuenca Grau and Ian Horrocks. "Ontology Module Extraction via Datalog Reasoning". In: *CoRR* (2014).

[Sha81] Micha Sharir. "A strong-connectivity algorithm and its applications in data flow analysis". In: *Computers & Mathematics with Applications* 7.1 (1981), pp. 67–72.

[Sir+07] Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur and Yarden Katz. "Pellet: A Practical OWL-DL Reasoner". In: *Web Semant.* 5.2 (June 2007), pp. 51–53. ISSN: 1570-8268.

[Spa00] K.A. Spackman. "Managing clinical terminology hierarchies using algorithmic calculation of subsumption: Experience with SNOMED-RT". In: *J. of the American Medical Informatics Association (Fall Symposium Special Issue)* (2000).

[SS09] Heiner Stuckenschmidt and Anne Schlicht. "Structure-Based Partitioning of Large Ontologies." In: *Modular Ontologies*. Ed. by Heiner Stuckenschmidt, Christine Parent and Stefano Spaccapietra. Vol. 5445. Lecture Notes in Computer Science. Springer, 15th June 2009, pp. 187–210. ISBN: 9783642019067.

[SSZ09] Ulrike Sattler, Thomas Schneider and Michael Zakharyaschev. "Which Kind of Module Should I Extract?" In: *Proceedings of the 22nd International Workshop on Description Logics (DL 2009), Oxford, UK, July 27-30, 2009*. 2009.

[Sun+08] Boontawee Suntisrivaraporn, Guilin Qi, Qiu Ji and Peter Haase. "A Modularization-Based Approach to Finding All Justifications for OWL DL Entailments". In: *The Semantic Web, 3rd Asian Semantic Web Conference, ASWC 2008, Bangkok, Thailand, December 8-11, 2008. Proceedings*. 2008, pp. 1–15.

[Tar72] Robert Tarjan. "Depth-first search and linear graph algorithms". In: *SIAM journal on computing* 1.2 (1972), pp. 146–160.

[TH06] Dmitry Tsarkov and Ian Horrocks. "FaCT++ Description Logic Reasoner: System Description". In: *Proceedings of the Third International Joint Conference on Automated Reasoning*. IJCAR'06. Seattle, WA: Springer-Verlag, 2006, pp. 292–297. ISBN: 3-540-37187-7, 978-3-540-37187-8.

[The12] The Gene Ontology Consortium. "The Gene Ontology: enhancements for 2011". In: *Nucleic Acids Research* 40.D1 (2012), pp. D559–D564.

[TNM08] Tania Tudorache, Natalya Fridman Noy and Mark A. Musen. "Collaborative Protege: Enabling Community-based Authoring of Ontologies". In: *International Semantic Web Conference (Posters & Demos)*. 2008.

[TP12a] Dmitry Tsarkov and Ignazio Palmisano. *Chainsaw: A Metareasoner for Large Ontologies*. 2012.

[TP12b] Dmitry Tsarkov and Ignazio Palmisano. "Chainsaw: a Metareasoner for Large Ontologies." In: ed. by Ian Horrocks, Mikalai Yatskevich and Ernesto Jiménez-Ruiz. Vol. 858. CEUR Workshop Proceedings. 2012.

[Tsa12] Dmitry Tsarkov. "Improved Algorithms for Module Extraction and Atomic Decomposition". In: *Proceedings of the 2012 International Workshop on Description Logics, DL-2012, Rome, Italy, June 7-10, 2012*. 2012.

[Tse68] G.S Tseitin. “On the complexity of derivation in propositional calculus”. In: *Studies in Constructive Mathematics and Mathematical Logic* (1968), pp. 115–125.

[Ves+10] Chiara del Vescovo, Bijan Parsia, Uli Sattler and Thomas Schneider. “The Modular Structure of an Ontology: An Empirical Study”. In: *Proceedings of the 2010 Conference on Modular Ontologies: Proceedings of the Fourth International Workshop (WoMO 2010)*. Amsterdam, The Netherlands, The Netherlands: IOS Press, 2010, pp. 11–24. ISBN: 9781607505433.

[Ves+12] Chiara Del Vescovo, Pavel Klinov, Bijan Parsia, Ulrike Sattler, Thomas Schneider and Dmitry Tsarkov. “Syntactic vs. Semantic Locality: How Good Is a Cheap Approximation?” In: *Proceedings of the 6th International Workshop on Modular Ontologies, Graz, Austria, July 24, 2012*. 2012.

[Ves+13] Chiara Del Vescovo, Pavel Klinov, Bijan Parsia, Ulrike Sattler, Thomas Schneider and Dmitry Tsarkov. “Empirical Study of Logic-Based Modules: Cheap Is Cheerful”. In: *Informal Proceedings of the 26th International Workshop on Description Logics, Ulm, Germany, July 23 - 26, 2013*. 2013, pp. 144–155.

[Ves13] Chiara Del Vescovo. “The Modular Structure of an Ontology: Atomic Decomposition and its Applications”. PhD thesis. The University of Manchester, 2013.

[Whe+11] P. Whetzel, N. Fridam Noy, N. Shah, P. Alexander, C. Nyulas, T. Tudorache and M. Musen. “BioPortal”. In: *Nucleic Acids Research* Web-Server-Issue (2011), pp. 541–545.

[WPH06] Taowei David Wang, Bijan Parsia and James Hendler. “A Survey of the Web Ontology Landscape”. In: *Proceedings of the 5th International Conference on The Semantic Web*. ISWC’06. Athens, GA, 2006, pp. 682–694. ISBN: 3-540-49029-9, 978-3-540-49029-6.