THE MODULAR STRUCTURE OF AN ONTOLOGY: ATOMIC DECOMPOSITION AND ITS APPLICATIONS

A thesis submitted to the University of Manchester
for the degree of Doctor of Philosophy
in the Faculty of Engineering and Physical Sciences

2013

By
Chiara Del Vescovo
School of Computer Science

Contents

Word Count: 44,300

List of Tables

List of Figures

Abstract

Ontologies are descriptions of the knowledge about a domain of interest encoded in computer processable languages, e.g., Description Logics, which are decidable fragments of First Order Logic. The main aim of ontologies is to define unambiguous vocabularies to facilitate knowledge sharing and integration.

A critical issue with ontologies consists of their increasing complexity. To address this problem several notions of modularity have been recently proposed. Modularity notions can help in two ways: 1) For identifying in a principled way the appropriate sub-part of an ontology that we want to work with; 2) for defining a modular structure, induced by the notion of module, which allows users to explore the entire ontology in a sensible manner (perhaps finding appropriate sub-parts to work on). However, the most popular notion—locality based modules—while excelling at modular extraction have thus far resisted attempts to induce a modular structure. Indeed, due to their nature, ontologies tend to have an unfeasible number of such modules, i.e., up to exponential in the ontology's size.

We tackle this problem by identifying basic building blocks of modules as sets of axioms which "cling together", that is, such that if any element appears in a module, then the remaining of the set also occurs. This notion of an "atom" proves key to defining a useful family of locality based modular structures, the (Labelled) Atomic Decompositions ((L)ADs).

In this thesis, we define (L)AD and explore its properties. We show that ADs are efficiently computable and, with appropriate labellings, provide a reasonably terse representation of the entire set of locality based modules. From ADs, we are able to distinguish so-called "genuine" modules, i.e., modules that cannot be decomposed further as the union of two or more modules.

Finally, we explore several of the applications to which (L)ADs have been applied including module extraction, ontology comprehension, and modular reasoning.

Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright

Acknowledgements

I really need to thank all the people involved in the process of writing my thesis for their wonderful support. I feel so lucky to have you in my life!

I want to thank my examiners David Rydeheard and Sebastian Rudolph for the insightful comments they gave me both during and after my thesis defense.

A special thank goes to my supervisors Uli and Bijan who have helped a lot throughout the 3 years of my PhD, for the patience of unravelling my thoughts, and for pushing me to write down everything since a theorem does not exist until its proof is written somewhere.

Thanks to my colleagues Pavel, Thomas and Dmitry: working with you has shade plenty of light to my research!

Thanks to Eleni, Fabio, Francesco, Ignazio, and Nico, who have helped me greatly with the implementation, read a large part of my thesis, made great suggestions, and corrected my English...

Thanks to all my Mancunian friends who have regularly listened to my moaning: Julie, Marialuisa, Martin, Matthew, Mohammad, Patrick, Rafa & Catarina, Roman, Riccardo, Rishi, Sam, and Sergio & Nathäele. A special special thank goes to Verena for finding and sharing with me the cutest pictures of koalas... without a PhD!

Many thanks to my friends all around the world, from Rome to wherever you are in Italy and beyond, from Belfast to Berlin, from London to Prague, from Warsaw to Sevilla, and then up to Japan, Nigeria, and the US. I could have not completed my thesis without you all sending me energy and positivity even when I have disappeared to write up. I cannot mention you all, but I cannot avoid to thank Paolo, a friend, a mentor, my father-in-logic.

Thanks to my family back in Rome: we have been apart, but love does not fear the distance (even if we do).

Finally, least but not last at all, thank to Carlo, who has suffered (not in silence!) for my stress, and has supported me in more than one way.

Chapter 1

Introduction

An ontology is a computer processable description of the knowledge about the relationships between the terms occurring about a domain of interest. Ontologies are typically encoded in a Knowledge Representation language. A major class of ontology languages is based on the family of *Description logics* (DLs) [BCNP03], which generally are decidable fragments of First Order Logic. Hence, an ontology may be viewed as a set of formulae in FOL, called *axioms*, and it is therefore a logical theory. A notable example of an ontology language based on DLs is the Web Ontology Language OWL [HPSvH03], a World Wide Web Consortium Standard which has been used both in academic and industrial environments since its first version became a W3C Recommendation in 2004.

The logical foundation of an ontology $\mathcal{O}$ provides the vocabulary described in $\mathcal{O}$ with a well-defined, unambiguous meaning. Moreover, it enables the use of *reasoners* that are able to draw inferences logically derivable from the ontology. In other words, a reasoner is able to make explicit the knowledge that the ontology implicitly encodes. Prominent examples of reasoners are FaCT++ [TH06] and its Java port JFact, HermiT [MSH07], Pellet [SPC+07], Racer [HM01], CB [Kaz08], `CEL` [BLS06], and ELK [KKS11]. We will also briefly describe in Section 6.3 the recently developed reasoners Chainsaw [TP12] and MORe [ACH12] which are of particular interest for this thesis.

The potential of providing domains with an unambiguous vocabulary has attracted the interest of many domain experts from diverse areas, such as medicine,

bioinformatics, and geography. In some examples a strong effort is put in modelling and maintaining ontologies, a notable example of this process being the National Cancer Institute thesaurus[1] [GFH+11]: NCI-t has been updated monthly since 2003, and many versions of the NCI-t ontology are freely downloadable. The progression of the NCI-t corpus has been analyzed in [GPS11]: by comparing each version with the next one, the evidence has emerged of a continuous restructuring of some of the already encoded knowledge, both in its syntax and in its semantics. In particular, one can wonder whether all these changes were planned.

Modelling an ontology is clearly a hard task to perform. This is mainly due to the mismatch between the intended and the actual representation: whilst the knowledge of the domain of interest is generally well-understood, or at least agreed on, it is not trivial to foresee and understand the logical consequences of adding, removing, or modifying an axiom, especially when the terminology is highly interconnected. In particular, there is no inherent localization of the semantic effects of an axiom. The order of the axioms is semantically irrelevant, ontologies suffer from an inherent lack of structure, and are often treated as monolithic objects.

Beside lacking a structure, ontologies grow in size and expressivity. These are all factors that contribute to the *complexity* of an ontology, which can be divided into two major categories: *computational* complexity and *cognitive* complexity.

The computational complexity of an ontology refers at the amount of time needed to execute a procedure, e.g., classification. In this area extensive research has been carried out, aimed at finding the complexity class of reasoning over ontologies encoded in a certain DL. DL languages vary in the complexity class they belong to, from low expressive languages whose standard reasoning tasks can be performed in polynomial time [Baa03, BBL05, CDL+05], to highly expressive DLs whose complexity class is N2ExpTime [Kaz09] in the size of the input.

Even though highly expressive languages seem intractable, the extreme optimizations of the reasoners means that even large and complex ontologies can be used in practice. Indeed, the reasoning time for an ontology $\mathcal{O}$ is not, in general, proportional to the size of $\mathcal{O}$ and to the complexity class that the language in which $\mathcal{O}$ is encoded belongs to. A comparison of different reasoners' performances over a selection of biomedical ontologies is reported in [GPS12] where the number

[1] `https://wiki.nci.nih.gov/display/VKC/NCI+Thesaurus+Terminology`

of axioms clearly is not the main factor influencing the reasoning time.

The cognitive complexity of an ontology aims at measuring how hard it is for a human to understand an ontology. The term "understand" is vague, and to be evaluated needs to be operationalized by defining a specific task to perform and measuring the empirical observations of the outcome. Several tasks involve the understanding of an ontology: from the development and maintainance, to the use and re-use of ontologies. In these scenarios, the lack of structure is especially problematic since an ontology does not group semantically related axioms together, so the users potentially have to keep every axiom in mind as they go through them.

For this reason, several approaches have been attempted to provide ontology engineers with a way to read the axioms of an ontology $\mathcal{O}$ in order to understand whether the actual modelling matches the intended modelling, to support the development, the maintenance, and the debug of ontologies. Among the plethora of applications providing this kind of information, we mention:

- Tools that reveal *axiom pinpointing* or *justifications*, defined to be minimal sets of axioms from $\mathcal{O}$ sufficient for an entailment η to hold [BPS07, Hor11]; in this case, the user is helped in reading an ontology because they do not have to go through all the axioms of $\mathcal{O}$, and can focus only on those that play an active role in explaining why η is derivable from the ontology.

- Tools that aim at ordering axioms according to their impact on the rest of the ontology, especially in the cases where some entailments are clearly wrong, as it is generally the case for unsatisfiable classes [Kal06, NRG12, MMV11]. These services aim at supporting the ontology engineers in repairing single entailments, i.e., removing the wrong ones, even though by repairing one axiom one could obtain that many wrong inferences are removed. In particular, the engineers have to actively search for modelling errors by checking that each entailment is correct. Since there can be infinitely many inferences this approach is unfeasible.

- Tools, such as Protégé, sensitive to the syntactic relations between an axiom and a term that allow the exploration of an ontology via linking all terms with the axioms in which they occur. However, the more expressive a language is, the more complex the logical relations between axioms and terms can be, and this view is more prone to be misleading on what is

relevant for a given term in an ontology: in particular, logically relevant axioms to a term could be excluded from such a view, and some extraneous ones may creep in.

Another viewpoint takes into account the fact that the users generally do not interact with an ontology $\mathcal{O}$ as a whole, but they focus on a limited part of $\mathcal{O}$. The Systematized Nomenclature of Medicine—Clinical Terms[2] [SCC97] (SNOMED CT), for example, contains over 500,000 axioms and covers a broad range of areas, such as diseases, symptoms, operations, treatments, devices and drugs, and describes their interconnections and commonalities. However, the interest of a single user is likely to be focused only on one medical specialization, for example `Cardiology` rather than `Radiology`. So, while on the one hand the users are interested in taking advantage of the effort put in modelling SNOMED CT, on the other hand they would prefer to avoid importing the whole ontology since a large part of it is irrelevant to their purposes.

In this perspective, the idea has recently arisen to explore the notion of *modularity* in ontologies. Ideally, the user selects a "topic" they are interested in, and then, based on the application the module is to be used for, to identify in $\mathcal{O}$ the "relevant part" of $\mathcal{O}$ for the topic specified. This "topical" approach can be of interest also for ontology engineers: by looking at the parts identified and how they are related, one can also gain an insight on how $\mathcal{O}$ is modelled.

1.1 Modules

In recent times, several approaches at modularity have been explored. We can divide these approaches into two main categories:

1. the approaches that go through the axioms of an ontology and employ heuristic methods in order to determine whether an axiom belongs to the module for the "topic" selected by the user that generally is specified as a signature $\Sigma \subseteq \widetilde{\mathcal{O}}$.

2. the approaches that aim at identifying modules that satisfy logical properties, e.g., to preserve the knowledge defined by the ontology for a suitable signature $\Sigma \subseteq \widetilde{\mathcal{O}}$. For the purposes of this thesis, knowledge can be defined in two flavours: a model-theoretic flavour (preserving the models over Σ)

[2] `http://www.ihtsdo.org/snomed-ct/`

or a deduction-oriented flavour (preserving the entailments over Σ). This property is called *coverage*.

Among the structural approaches, we mention the PROMPT-FACTOR tool described in [NM03], and the Web ontology segmentation technique analysed in [SR06]. The first approach extracts a module $\mathcal{M} \subseteq \mathcal{O}$ for a given signature Σ as follows: first, all the axioms mentioning some terms from Σ are added to $\mathcal{M}$; then, the original signature Σ is expanded with the terms mentioned in the axioms now included in $\mathcal{M}$; finally, the procedure is repeated until a fixpoint is reached. The second approach instead works as follows: first, for each term $\mathtt{t} \in \Sigma$ all those axioms are identified that describe `t` as being equivalent to other terms; these axioms are then added to the module $\mathcal{M}$, and the original signature Σ is expanded with the terms mentioned in the axioms now included in $\mathcal{M}$; finally, the procedure is repeated until a fixpoint is reached. As a consequence, this approach exploits "upwards" the hierarchical structure of the concepts described in an ontology, i.e., the inferred forest of concepts summarizing which ones are a specialization of others (e.g., `Man` and `Woman` are specializations of `Person`).

These approaches are clearly syntax-based, and suffer from not fully capturing and exploiting the semantics of the ontologies. Examples of modules extracted with these methods that either do not preserve all the entailments over Σ, or include axioms that have no role in preserving such entailments can be found in [CHKS07].

For the purpose of this thesis, the modularisation methods of interest are those that produce modules satisfying logical properties, one above all being coverage: given the effort put in representing the domain knowledge into an ontology we do not want to lose any information that the ontology could provide. Following [CK07], we distinguish two main categories of approaches:

1. the *a priori* approaches that involve the design of formalisms (*modular ontology languages*) to control the interactions between the modules at developing time.

2. the *a posteriori* approaches, aiming at revealing in a given ontology $\mathcal{O}$ the logical relations between the components (terms, axioms, domain) of $\mathcal{O}$.

A priori approaches have been defined in [BLSW02] to cope with the high computational complexity of languages that allow many different constructors. The basic idea is to keep separate those tractable fragments of the logics that,

if merged, lead to intractable fragments by avoiding in a principled way that different fragments share terms; indeed, delegating the reasoning over the single components is less complex than designing new reasoning methods from scratch, if they can exist at all. Among these approaches we mention Package-based Description Logics [BCH06], Distributed Description Logics [BS03], and $\mathcal{E}$-connections [KLWZ04, CPS06]. However, this choice imposes further limitations to what can be expressed in an ontology since it is crucial to avoid semantic interrelations to arise between the modules, or unwanted entailments can creep in the ontology.

A further question that one can ask is whether the modelling implemented at development time matches the domain conceptualization. Adherence to predefined modelling is not a trivial task, especially in the case where the ontology is underspecified, and some entailments are accidentally left out: examples have been reported in [CPSK06] of ontologies whose logical structure differs substantially from the modelling of the domain.

The a posteriori approaches can be divided into two categories:

1. those notions that aim at identifying a *modular structure* in an ontology $\mathcal{O}$ obtained by fragmenting the ontology in coherent parts (the modules) and by exposing the logical interactions between these modules, in order to gain an insight on the actual modelling of $\mathcal{O}$; in this category fall Parikh's signature decomposition [Par99], the signature Δ-decomposition [KLPW10], and the converse use of $\mathcal{E}$-connections as described in [CPSK06].

2. those notions that aim at extracting from $\mathcal{O}$ a small, less complex fragment to be used in applications instead of the whole ontology; in this category fall the different notions of modules based on conservativity [GLW06, KLWW08], those based on inseparability for a query language [KPS$^+$09, KWZ10], and those based on locality [CHKS08, KLWW09].

The target of this thesis are the notions of modules that are a posteriori identified. It has to be noted, though, that there is a strong relationship between the a priori and the a posteriori approaches. Cuenca Grau and Kutz investigated in [CK07] the relationships between the various modular ontology languages, locality and conservative extensions. In particular, the modular ontology languages can be expressed in terms of locality and conservative extensions, but not *vice versa*.

1.2 Contributions of this Thesis

From the discussion in the previous section, it is clear that logic-based modularity in ontologies has been recently investigated under several viewpoints.

This thesis makes new advances in this area by providing a unified overview of the theory of modularity. In particular, we discuss the advantages of choosing notions of modules that satisfy some logical properties. We then define the modular structure of an ontology $\mathcal{O}$ as a pair $(\mathcal{F}(\mathcal{O}), \rightarrow)$ where $\mathcal{F}(\mathcal{O})$ is a set of fragments of $\mathcal{O}$ that represents all the modules, and $\rightarrow$ is a logical relation between the modules, defined to hold between two modules if one needs to import the other to preserve some logical property.

Having an abstract notion of modular structure allows us to compare different kinds of logical modules. We have (1) those notions of modules that come from identifying a modular structure in an ontology; and (2) those notions that derive from the extraction of suitable subsets that behave for some purposes as the ontology does. For the modules of kind (2) we first define a notion of basic modules (*genuine*) that build up all the other modules, and then we show that the simple set-inclusion relation induces over the set of genuine modules a mathematical structure. This structure is called the Atomic Decomposition of an ontology $\mathcal{O}$, and it is a succinct representation of all the modules of $\mathcal{O}$ that can be feasibly obtained. These results are quite surprising considering that an ontology $\mathcal{O}$ can contain exponentially many modules of kind (2) in the size of $\mathcal{O}$. Finally, we investigate a wide range of applications where the ADs can be exploited.

The contributions of this thesis are described in what follows organized by chapters.

Chapter 3 We provide a unified theory of modularity by analysing the existing approaches, identifying their commonalities and differences, and defining a framework to evaluate the modular structures of ontologies. The paper [DPS11] partially addresses this subject, and it has been published at the *International Conference on Conceptual Structure (ICCS-11)*.

Chapter 4 We define the *Atomic Decomposition* of an ontology, i.e., a well-defined, succinct modular structure for the class of Σ-modules that generates in general exponentially many modules in the size of an ontology. Set n to be the number of axioms of $\mathcal{O}$, we devise an algorithm that requires n modules

extractions to compute an AD, hence it is polynomial in the size of the ontology provided that the notion of Σ-module used is polynomial, as those based on syntactic locality. Moreover, we also provide strong empirical evidence that the computation of AD is also feasible in practice. The paper [DPSS11a] addresses this subject, and it has been published at the *International Joint Conference on Artificial Intelligence (IJCAI-11)*.
We define a logic-based reading order for axioms via the notion of *Chain of Conservative Extensions* of an ontology, i.e., sequences of modules enlargements in $\mathcal{O}$ such that by adding axioms at each step the meaning of the terms is preserved. The paper [DPS12] introduces this notion, and it has been presented at the *International Workshop in Description Logics (DL-2012)*.

Chapter 5 We define the *Labelled Atomic Decomposition* of an ontology, i.e., the AD enriched with suitable labels that reveal the relations between modules and signatures. LADs are introduced in [Del11], presented at the *International Workshop in Description Logics (DL-2011)*.

Chapter 6 The two notions of ADs and LADs have already shown to be of high interest for the research in Description Logics. We measure their impact by describing some of the applications already investigated:
Module Count, i.e., an investigation of the use of ADs in the estimation of the number of modules of an ontology. This subject is addressed in [DPSS11b], and the paper is published at the *International Workshop on Modular Ontologies (WoMO-11)*.
Offline Module Extraction, i.e., how to allow the extraction of a module directly from the AD of $\mathcal{O}$, without the need to load the ontology into an editor, or to transfer the data between two agents. The paper [DGK+11] addresses this subject, and it has been published at the *International Semantic Web Conference (ISWC-11)*.
Modular Reasoning, i.e., the exploitation of modules to improve the performance of reasoners. The new meta-reasoner Chainsaw, implemented by Tsarkov and Palmisano and described in [TP12] put to use the LADs in the case where the ontology is intractable and needs to be "chopped up" to be reasoned over.
Patterns Evaluation, i.e., an analysis of design patterns identified with the RIO framework by Mikroyannidi et al. In particular, the sensitivity of the AD of an ontology both to the syntactic and to the semantic aspects of ontologies are

exploited, and the AD is used as an external criterion to evaluate the quality of the patterns identified.

Support Understanding, e.g., how to provide the user with a logic-based overview of the ontology. In this context, the tool DeMoSt has been implemented and presented at the Demo session of the *International Semantic Web Conference (ISWC-11)*.

Chapter 2

Background

We assume the reader to be familiar with the basic definitions for First Order Logic (FOL), Second Order Logic (SOL), and with the associated model-theoretic semantics. In this chapter, we first introduce the foundations of Description Logic (DL) languages, their syntax and semantics, and briefly describe the standardized language OWL (Web Ontology Language) underpinned by DLs. Then, we introduce the notions of Conservative Extensions (CEs) and of inseparability relations that capture the idea of preserving knowledge key in the definition of logic-based notions of modules.

In Section 2.3 we will introduce the notion of Σ-*modules* which is a class of modules designed to be suitably small subsets of an ontology $\mathcal{O}$ that satisfy a number of logical properties, among which coverage is fundamental, i.e., the property of a set of axioms to preserve all the entailments of $\mathcal{O}$ over a selected signature Σ.

Finally, we will briefly recap some basic notions of algebra used in this thesis. The presentation of this subject is rather schematic and it serves only to fix the notation. For a thorough presentation, we refer the reader to [Sch03].

Aim of this chapter is to fix the notations to be used in the rest of the thesis, and to refer the interested readers to the relevant literature. For a comprehensive presentation of the main topics concerning DLs we refer the reader to "The Description Logic Handbook: Theory, Implementation, and Applications" [BCNP03].

2.1 Foundations of Description Logics

A Description Logic (DL) is a decidable fragment of First Order Logic used for describing the knowledge about a domain of interest, by defining its relevant concepts and their interrelations. Syntactically, the vocabulary of a DL is obtained by using:

- *atomic concepts*, also called concept names, corresponding to unary predicates in FOL, denoted by symbols as $\mathtt{A}, \mathtt{B}$;
- *atomic roles*, also called role names, that in FOL correspond to binary predicates, denoted by symbols as $\mathtt{r}, \mathtt{s}$;
- *individuals*, that correspond to constants in FOL, denoted by letters in italics, like *a, b*;
- *constructors*, particular logical symbols that allow for the inductive construction of *complex concepts* or *complex roles*.

We will refer to individuals, atomic concepts, and atomic roles by calling them *terms* or *entities*. A set of terms will be called a *signature*.

The expressive power of a DL can be identified by the constructors allowed. The associated semantics is given by an *interpretation* $\mathcal{I}$, that is a pair $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where $\Delta^{\mathcal{I}}$ is a non-empty set, and $\cdot^{\mathcal{I}}$ is a function that maps each individual a to an element $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$, each atomic concept $\mathtt{A}$ to a subset $\mathtt{A}^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, and each atomic role $\mathtt{r}$ to a set of pairs $\mathtt{r}^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$.

Throughout this thesis we want to avoid that the signature where the interpretation function is defined over includes arbitrary terms that do not occur in any of the terms that are to be interpreted. Hence, we will adopt the convention that the interpretation function $\cdot^{\mathcal{I}}$ is defined over a minimal signature. This choice allows us to define the following notions:

- given an interpretation function $\mathcal{I}$ defined over a signature $\Sigma_{\mathcal{I}}$, and a signature $\Sigma \subseteq \Sigma_{\mathcal{I}}$, we define the Σ-*projection* $\mathcal{I}|_{\Sigma}$ of $\mathcal{I}$ as the interpretation function defined over Σ obtained by projecting $\mathcal{I}$ over Σ, i.e., $\mathcal{I}|_{\Sigma}$ is such that $\mathtt{t}^{\mathcal{I}|_{\Sigma}} = \mathtt{t}^{\mathcal{I}}$ for each term $\mathtt{t} \in \Sigma$;
- let $(\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ be an interpretation, and let $\Sigma_{\mathcal{I}}$ be the signature that the interpretation function $\mathcal{I}$ is defined over; we say that an interpretation $(\Delta^{\mathcal{J}}, \cdot^{\mathcal{J}})$ is an *expansion* of $\mathcal{I}$ if $\Delta^{\mathcal{J}}|_{\Sigma_{\mathcal{I}}} = \Delta^{\mathcal{I}}$ and if $\mathcal{I}$ is a $\Sigma_{\mathcal{I}}$-projection of $\mathcal{J}$;

- two interpretation functions $\mathcal{I}$ and $\mathcal{J}$ are said to *coincide on a signature* Σ, in symbols $\mathcal{I}|_\Sigma = \mathcal{J}|_\Sigma$, if $\Delta^\mathcal{I} = \Delta^\mathcal{J}$ and $\mathtt{t}^\mathcal{I} = \mathtt{t}^\mathcal{J}$ for all $X \in \Sigma$;

- let $(\Delta^{\mathcal{I}_1}, \mathcal{I}_1)$ and $(\Delta^{\mathcal{I}_1}, \mathcal{I}_2{}^{\mathcal{I}_2})$ be two interpretations such that $\Delta^{\mathcal{I}_1} = \Delta^{\mathcal{I}_2}$ and such that the interpretation functions $\mathcal{I}_1$ and $\mathcal{I}_2$ are respectively defined over two signatures Σ_1 and Σ_2 such that $\Sigma_1 \cap \Sigma_2 = \emptyset$; we define the *sum interpretation* $\mathcal{J} = \mathcal{I}_1 + \mathcal{I}_2$ as the interpretation defined over $\Sigma_\mathcal{J} = \Sigma_1 \cup \Sigma_2$ such that $\mathcal{J}|_{\Sigma_1} = \mathcal{I}_1$ and $\mathcal{J}|_{\Sigma_2} = \mathcal{I}_2$.

The two basic DLs that we consider throughout this thesis are $\mathcal{EL}$ and $\mathcal{AL}$. The constructors allowed in these languages, their syntax, and their semantics, are described in Table 2.1.

Language	Name	Syntax	Semantics
$\mathcal{EL}$	Top	$\top$	$\Delta^\mathcal{I}$
	Intersection	$\mathtt{C} \sqcap \mathtt{D}$	$\mathtt{C}^\mathcal{I} \cap \mathtt{D}^\mathcal{I}$
	Existential quantification	$\exists \mathtt{r}.\mathtt{C}$	$\{a \in \Delta^\mathcal{I} \mid \exists b.\, (a,b) \in \mathtt{r}^\mathcal{I} \wedge b \in \mathtt{C}^\mathcal{I}\}$
$\mathcal{AL}$	Top	$\top$	$\Delta^\mathcal{I}$
	Bottom	$\bot$	$\emptyset$
	Intersection	$\mathtt{C} \sqcap \mathtt{D}$	$\mathtt{C}^\mathcal{I} \cap \mathtt{D}^\mathcal{I}$
	Atomic negation	$\neg \mathtt{A}$	$\Delta^\mathcal{I} \setminus \mathtt{A}^\mathcal{I}$
	Limited existential quantification	$\exists \mathtt{r}$	$\{a \in \Delta^\mathcal{I} \mid \exists b.\, (a,b) \in \mathtt{r}^\mathcal{I}\}$
	Value restriction	$\forall \mathtt{r}.\mathtt{C}$	$\{a \in \Delta^\mathcal{I} \mid \forall b.(a,b) \in \mathtt{r}^\mathcal{I} \rightarrow b \in \mathtt{C}^\mathcal{I}\}$

Table 2.1: Constructors in $\mathcal{EL}$ and in $\mathcal{AL}$

The basic DLs just described can be extended with more expressive constructors. The extended DL obtained is then denoted by the name of its basic DL followed by the list of all other constructors allowed. For example, $\mathcal{ALC}$ corresponds to $\mathcal{AL}$ extended by allowing complex concept negation. In Table 2.2 we list the constructors of interest in this thesis. For more details please refer to [BCNP03].

A logical axiom, denoted by a Greek letter, is a variable-free well-formed formula that uses special logical operators. In Figure 2.3 we list name, syntax,

Name	Syntax	Semantics	Symbol
Union	$\mathtt{C} \sqcup \mathtt{D}$	$\mathtt{C}^{\mathcal{I}} \cup \mathtt{D}^{\mathcal{I}}$	$\mathcal{U}$
Negation	$\neg\mathtt{C}$	$\Delta^{\mathcal{I}} \setminus \mathtt{C}^{\mathcal{I}}$	$\mathcal{C}$
Existential quantification	$\exists\mathtt{r}.\mathtt{C}$	$\{a \in \Delta^{\mathcal{I}} \mid \exists b.\, (a,b) \in \mathtt{r}^{\mathcal{I}} \wedge b \in \mathtt{C}^{\mathcal{I}}\}$	$\mathcal{E}$
Unqualified number restriction	$\geq n\mathtt{r}$ $\leq n\mathtt{r}$ $= n\mathtt{r}$	$\{a \in \Delta^{\mathcal{I}} \mid \#\{b \in \Delta^{\mathcal{I}} \mid (a,b) \in \mathtt{r}^{\mathcal{I}}\} \geq n\}$ $\{a \in \Delta^{\mathcal{I}} \mid \#\{b \in \Delta^{\mathcal{I}} \mid (a,b) \in \mathtt{r}^{\mathcal{I}}\} \leq n\}$ $\{a \in \Delta^{\mathcal{I}} \mid \#\{b \in \Delta^{\mathcal{I}} \mid (a,b) \in \mathtt{r}^{\mathcal{I}}\} = n\}$	$\mathcal{N}$
Qualified number restriction	$\geq n\mathtt{r}.\mathtt{C}$ $\leq n\mathtt{r}.\mathtt{C}$ $= n\mathtt{r}.\mathtt{C}$	$\{a \in \Delta^{\mathcal{I}} \mid \#\{b \in \mathtt{C}^{\mathcal{I}} \mid (a,b) \in \mathtt{r}^{\mathcal{I}}\} \geq n\}$ $\{a \in \Delta^{\mathcal{I}} \mid \#\{b \in \mathtt{C}^{\mathcal{I}} \mid (a,b) \in \mathtt{r}^{\mathcal{I}}\} \leq n\}$ $\{a \in \Delta^{\mathcal{I}} \mid \#\{b \in \mathtt{C}^{\mathcal{I}} \mid (a,b) \in \mathtt{r}^{\mathcal{I}}\} = n\}$	$\mathcal{Q}$
Nominal	$\{a\}$	$\{a\}^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ with $\#\{a\}^{\mathcal{I}} = 1$	$\mathcal{O}$
Data types	the interpretation of number symbols is standard		$(\mathcal{D})$
Inverse role	$\mathtt{r}^{-}$	$\{(b,a) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid (a,b) \in \mathtt{r}^{\mathcal{I}}\}$	$\mathcal{I}$
Composition	$\mathtt{r} \circ \mathtt{s}$	$\mathtt{r}^{\mathcal{I}} \circ \mathtt{s}^{\mathcal{I}}$	$\circ$
Transitive closure	$\mathtt{r}^{+}$	$\bigcup_{n \geq 1} (\mathtt{r}^{\mathcal{I}})^n$	$+$

Table 2.2: Expressive constructors

and semantics of the basic axioms. Please note that in some applications non-logical axioms, as annotations, are used. For the purpose of this thesis, though, only logical axioms are of interest. Hence, in the remainder of this thesis the simple term "axiom" means a logical axiom.

Name	Syntax	Semantics
Concept inclusion	$\mathtt{C} \sqsubseteq \mathtt{D}$	$\mathtt{C}^{\mathcal{I}} \subseteq \mathtt{D}^{\mathcal{I}}$
Concept equality	$\mathtt{C} \equiv \mathtt{D}$	$\mathtt{C}^{\mathcal{I}} \equiv \mathtt{D}^{\mathcal{I}}$
Concept assertion	$\mathtt{C}(a)$ or $(a : \mathtt{C})$	$a^{\mathcal{I}} \in \mathtt{C}^{\mathcal{I}}$
Role assertion	$\mathtt{r}(a,b)$	$(a^{\mathcal{I}}, b^{\mathcal{I}}) \in \mathtt{r}^{\mathcal{I}}$
Transitivity	$\mathrm{Trans}(\mathtt{r})$	$(\mathtt{r}^{+})^{\mathcal{I}} = \mathtt{r}^{\mathcal{I}}$

Table 2.3: Basic logical axioms in DLs

More complex axioms are also allowed in extended DLs. In Table 2.4 we list the kinds of axioms that we will use in this thesis.

Name	Syntax	Semantics	Symbol
Role hierarchy	$\mathtt{r} \sqsubseteq \mathtt{s}$	$\mathtt{r}^{\mathcal{I}} \subseteq \mathtt{s}^{\mathcal{I}}$	
and role equality	$\mathtt{r} \equiv \mathtt{s}$	$\mathtt{r}^{\mathcal{I}} \equiv \mathtt{s}^{\mathcal{I}}$	$\mathcal{H}$
Functionality	$\mathrm{Func}(\mathtt{r})$	$\forall a \in \Delta^{\mathcal{I}}$, $\#\{b \in \Delta^{\mathcal{I}} \mid (a,\, b) \in \mathtt{r}^{\mathcal{I}}\} \leq 1$	$\mathcal{F}$
Reflexivity	$\mathrm{Refl}(\mathtt{r})$	$\forall a \in \Delta^{\mathcal{I}}, (a, a) \in \mathtt{r}^{\mathcal{I}}$	
and Irreflexivity	$\mathrm{Irrefl}(\mathtt{r})$	$\forall a \in \Delta^{\mathcal{I}}, (a, a) \notin \mathtt{r}^{\mathcal{I}}$	$\mathcal{R}$

Table 2.4: More expressive logical axioms in DLs

Some DLs do not follow the standard naming scheme. The exceptions of interest for this thesis are:

$\mathcal{EL}++$	equivalent to $\mathcal{ELOR}$
$\mathcal{S}$	equivalent to $\mathcal{ALC}+$
DL-lite	a family of sublanguages of $\mathcal{SHIF}(\mathcal{D})$ for which the usual DL reasoning tasks are polynomial in the size of the ontology

Given an axiom α, we will denote the set of terms occurring in α with the symbol $\widetilde{\alpha}$. Given a concept name $\mathtt{A}$, a role name $\mathtt{r}$, a pair of individuals a, b, a pair of complex concepts $\mathtt{C}, \mathtt{D}$, a pair of complex roles $\mathtt{r}, \mathtt{s}$, and a datatype R, we inductively define the *length* $\ell(\alpha)$ of an axiom α as follows:

$\ell(\mathtt{C} \sqsubseteq \mathtt{D}) = \ell(\mathtt{C} \equiv \mathtt{D}) := \ell(\mathtt{C}) + \ell(\mathtt{D})$,

$\ell(\mathtt{r} \sqsubseteq \mathtt{s}) = \ell(\mathtt{r} \equiv \mathtt{s}) := \ell(\mathtt{r}) + \ell(\mathtt{s})$,

$\ell(\mathtt{C}(a)) := \ell(\mathtt{C}) + 1$,

$\ell(\mathtt{r}(a, b)) := 3$,

$\ell(\mathrm{Trans}(\mathtt{r})) = \ell(\mathrm{Func}(\mathtt{r})) = \ell(\mathrm{Refl}(\mathtt{r})) = \ell(\mathrm{Irrefl}(\mathtt{r})) := 1$,

where

$\ell(\top) = \ell(\bot) = \ell(R) := 0$,

$\ell(\mathtt{A}) = \ell(\{a\}) = \ell(\mathtt{r}) = \ell(\mathtt{r}^{-}) = \ell(\mathtt{r}^{+}) := 1$,

$\ell(\neg\mathtt{C}) := \ell(\mathtt{C})$,

$\ell(\mathtt{C} \sqcap \mathtt{D}) = \ell(\mathtt{C} \sqcup \mathtt{D}) := \ell(\mathtt{C}) + \ell(\mathtt{D})$,

$\ell(\mathtt{r} \circ \mathtt{s}) = \ell(\mathtt{r}) + \ell(\mathtt{s})$,

$\ell(\exists\mathtt{r}.\mathtt{C}) = \ell(\forall\mathtt{r}.\mathtt{C}) = \ell(\leq n\mathtt{r}.\mathtt{C}) = \ell(\geq n\mathtt{r}.\mathtt{C}) := 1 + \ell(\mathtt{C})$.

This measure captures the complexity of parsing an axiom, and it will be used in this chapter after Definition 2.3.13, and in Section 4.4.

For our purposes, an ontology is a finite set $\mathcal{O}$ of logical axioms. Given an ontology $\mathcal{O}$, its signature, denoted $\widetilde{\mathcal{O}}$, is the set of terms occurring in one of $\mathcal{O}$'s axioms. The *size* $s(\mathcal{O})$ of an ontology $\mathcal{O}$ is the number $\sum_{\alpha\in\mathcal{O}} \ell(\alpha)$ obtained by adding up the length of each axiom $\alpha \in \mathcal{O}$. The *expressivity* of an ontology $\mathcal{O}$ is defined to be the expressivity of the minimal DL needed to express all the axioms in $\mathcal{O}$. Please note that logically equivalent ontologies might have different expressivities.

We say that an interpretation $\mathcal{I}$ *satisfies an axiom* α if the formula $\alpha^{\mathcal{I}}$ obtained by mapping each term in $\widetilde{\alpha}$ via the interpretation function $\cdot^{\mathcal{I}}$ is true. The interpretation $\mathcal{I}$ satisfies an ontology $\mathcal{O}$ if $\mathcal{I}$ satisfies each element of $\mathcal{O}$.

If $\mathcal{I}$ satisfies an axiom α (resp. a set of axioms $\mathcal{O}$), then we say that $\mathcal{I}$ is a *model* of α (resp. $\mathcal{O}$), and we write $\mathcal{I} \models \alpha$ (resp. $\mathcal{I} \models \mathcal{O}$). If every possible interpretation $\mathcal{I}$ is a model for α, we say that α is a *tautology*. It is useful to compare models of different sets of axioms $\mathcal{O}_1$ and $\mathcal{O}_2$: if all models of $\mathcal{O}_1$ are also models for $\mathcal{O}_2$, we write that $\mathcal{O}_1 \models \mathcal{O}_2$; for example, this happens if $\mathcal{O}_2 \subseteq \mathcal{O}_1$. A concept name $\mathtt{A}$ is said to be *satisfiable* in α (resp. in $\mathcal{O}$) if there exists a model $\mathcal{I}$ for α (resp. in $\mathcal{O}$) such that $\mathtt{A}^{\mathcal{I}} \neq \emptyset$. We say that an ontology is *consistent* if there exists an interpretation $\mathcal{I}$ over $\widetilde{\mathcal{O}}$ such that $\mathcal{I} \models \mathcal{O}$.

OWL The Web Ontology Language (OWL) is a standardized language for defining ontologies, and a W3C recommendation. The underpinning logic is based on DLs, however OWL offers more (logical and non logical) features, for example the use of datatypes for managing data assertions, and annotations. In this thesis we will consider only the logical aspects of ontologies, so the non-logical features of OWL will be generally disregarded.

The latest version of OWL is called OWL 2, which offers a number of *profiles*, i.e., sublanguages with different expressivity power and computational complexity properties. The profiles of interest for this thesis are OWL EL, OWL Lite (called OWL QL in OWL 2), and OWL DL. OWL EL is based on the logic $\mathcal{EL}++$ [BBL05]. OWL QL instead is based on the logic DL-lite [CDL$^+$07]. For OWL EL and OWL QL, the standard reasoning tasks available are polynomial. OWL DL corresponds to $\mathcal{SROIQ}$ and is the most expressive decidable language in the OWL family.

2.2 Conservative Extensions and inseparability relations

As discussed in Chapter 1, the need has emerged for managing ontologies in a modular way, to reduce both the cognitive and the computational complexity that easily arise even for supposed simple ontologies. In this thesis, all the notions of modules considered are built upon the notion of *Conservative Extensions* (CEs) defined in what follows. We will discuss and motivate this choice in Chapter 3.

There are a number of variants of the notion of CEs, which capture the preservation of knowledge to different degrees. We focus on the following basic ones.

Definition 2.2.1 (Ghilardi *et al.*, [GLW06], Konev et al., [KLWW09])**.** Let $\mathcal{O}$ be a $\mathcal{SROIQ}$-ontology, $\mathcal{M} \subseteq \mathcal{O}$, and Σ a signature. We say that:
1. $\mathcal{O}$ is a *deductive* Σ*-conservative extension* (Σ*-dCE*) of $\mathcal{M}$ if, for all $\mathcal{SROIQ}$-axioms α with $\widetilde{\alpha} \subseteq \Sigma$, it holds that $\mathcal{M} \models \alpha$ if and only if $\mathcal{O} \models \alpha$;
2. $\mathcal{O}$ is a *model* Σ*-conservative extension* (Σ*-mCE*) of $\mathcal{M}$ if $\{\mathcal{I}|_\Sigma \mid \mathcal{I} \models \mathcal{M}\} = \{\mathcal{J}|_\Sigma \mid \mathcal{J} \models \mathcal{O}\}$;
3. $\mathcal{M}$ is a *dCE-based (mCE-based) module for* Σ of $\mathcal{O}$ if $\mathcal{O}$ is a Σ-dCE (Σ-mCE) of $\mathcal{M}$;
4. If $\mathcal{M}$ is a dCE-based module for Σ, we also say that $\mathcal{M}$ *covers* or *provides coverage to* $\mathcal{O}$ for Σ.

Since $\mathcal{M} \subseteq \mathcal{O}$, the monotonicity of the logic $\mathcal{SROIQ}$ implies that the "only if" direction of Definition 2.2.1.1 holds trivially. Please also note that, for each language $\mathcal{L} \subseteq \mathcal{SROIQ}$, a module preserving the models over Σ clearly preserves also all the $\mathcal{L}$-entailments over Σ. In contrast, the converse statement is not always true: there can be an mCE-based module $\mathcal{M}$ of $\mathcal{O}$ for a signature Σ for which the set of the $\mathcal{L}$-entailments over Σ is not preserved since $\mathcal{L}$ could be not expressive enough. In other words, whilst the model-theoretic notion of module is also able to identify all the deduction-theoretic modules, not all mCE-based modules can be identified by the notion of dCE. To sum up, all the dCE-based modules are also mCE-based modules, whilst the *vice versa* is, in general, false.

In order to abstract from the requirement that $\mathcal{M}$ is a subset of $\mathcal{O}$, the notion of inseparability relation has been introduced.

Definition 2.2.2 (Konev et al., [KLWW09])**.** Let $\mathcal{O}_1$ and $\mathcal{O}_2$ be two ontologies, and Σ a signature. We say that:

1. $\mathcal{O}_1$ and $\mathcal{O}_2$ are *model inseparable w.r.t.* Σ if $\{\mathcal{I}|_\Sigma \,|\, \mathcal{I} \models \mathcal{O}_1\} = \{\mathcal{I}|_\Sigma \,|\, \mathcal{I} \models \mathcal{O}_2\}$. In this case, we write that $\mathcal{O}_1 \equiv_\Sigma^{\text{mCE}} \mathcal{O}_2$.
2. $\mathcal{O}_1$ and $\mathcal{O}_2$ are *deduction inseparable w.r.t.* Σ if, for all entailments η over Σ, $\mathcal{O}_1 \models \eta \Leftrightarrow \mathcal{O}_2 \models \eta$. In this case, we write that $\mathcal{O}_1 \equiv_\Sigma^{\text{dCE}} \mathcal{O}_2$.

The equivalence relations $\equiv^R$ are defined upon the notion $R \in \{\text{mCE, dCE}\}$ which is called an *inseparability relation.*

Please note that other notions of inseparability relations can be defined, even though for the purposes of this thesis we will consider only inseparability relations $R \in \{\text{mCE, dCE}\}$.

The following property of *monotonicity* is of crucial importance for the inseparability relations just defined, and will have a deep impact on the properties of modules of interest in this thesis.

Definition 2.2.3 (Kontchakov et al., [KPS$^+$09])**.** An inseparability relation R is called *monotone* if, for all ontologies $\mathcal{O}_1, \mathcal{O}_2$ and $\mathcal{O}_3$ and each signature Σ, it satisfies the following conditions:

(M_{sig}) if $\mathcal{O}_1 \equiv_\Sigma^R \mathcal{O}_2$, then $\mathcal{O}_1 \equiv_{\Sigma'}^R \mathcal{O}_2$ for each $\Sigma' \subseteq \Sigma$;

(M_{axs}) if $\mathcal{O}_1 \subseteq \mathcal{O}_2 \subseteq \mathcal{O}_3$ and $\mathcal{O}_1 \equiv_\Sigma^R \mathcal{O}_3$, then $\mathcal{O}_1 \equiv_\Sigma^R \mathcal{O}_2$.

The property of monotonicity for an inseparability relation $R \in \{\text{mCE, dCE}\}$ is guaranteed by the monotonicity of the DLs. Please note though that not all inseparability relations are monotonic. Examples can be found in [CHKS08].

Inseparability relations induce modules defined as follows.

Definition 2.2.4 (Kontchakov et al., [KPS$^+$09])**.** Let R be an inseparability relation, $\mathcal{M}$ and $\mathcal{O}$ two ontologies, and Σ a signature. We call $\mathcal{M}$:

1. an *R_Σ-module of* $\mathcal{O}$ if $\mathcal{M} \equiv_\Sigma^R \mathcal{O}$;
2. a *self-contained R_Σ-module of* $\mathcal{O}$ if $\mathcal{M} \equiv^R_{\Sigma \cup \widetilde{\mathcal{M}}} \mathcal{O}$;
3. a *depleting R_Σ-module of* $\mathcal{O}$ if $\emptyset \equiv^R_{\Sigma \cup \widetilde{\mathcal{M}}} \mathcal{O} \setminus \mathcal{M}$.

$\mathcal{M}$ is called a *minimal* (self-contained, depleting) R_Σ-module of $\mathcal{O}$ if $\mathcal{M}$, but no proper subset of $\mathcal{M}$, is a (self-contained, depleting) R_Σ-module of $\mathcal{O}$.

To ease the notation, we will omit the symbol Σ from the notion R of an inseparability relation whenever the specific signature Σ is irrelevant to our purposes.

Unfortunately, deciding whether a fragment $\mathcal{M}$ of an ontology $\mathcal{O}$ is a module in this sense is in general computationally hard: in its model-theoretic formulation, deciding conservativity is PTIME for acyclic $\mathcal{EL}$ ontologies [KLWW08], coNEXP-hard for some fragments of DL-lite [KWZ10] and undecidable already for general $\mathcal{EL}$ ontologies and acyclic $\mathcal{ALC}$ ontologies [LWW07]; other approaches limit the selection of the signature Σ to contain only concept names (such Σ is then called a *concept signature*), and obtain that the complexity of deciding mCE-based modules goes down to PTIME for general $\mathcal{ELI}$ ontologies, and to Π_2^p for $\mathcal{ALCI}$ [KLWW08]. For the deduction-oriented formulation better results have been found: deciding whether $\mathcal{M}$ is a dCE-based module of $\mathcal{O}$ for Σ is EXPTIME-complete in $\mathcal{EL}$ [LW10], 2EXPTIME-complete in $\mathcal{ALC}$ [GLW06] and $\mathcal{ALCQI}$ [LWW07], and undecidable for expressive DLs as $\mathcal{ALCQIO}$ [LWW07].

To cope with this various approximations have been devised: the idea is to define an oracle "x-check" able to decide whether easier conditions hold to guarantee that a set of axioms $\mathcal{M}$ is an R-module. As a consequence, the x-check will not be able to find *all* the R-modules. However, if we require the x-check to decide whether a set of axioms $\mathcal{M}$ is a self-contained module for a signature Σ, then we will also have that $\mathcal{M}$ is a module for any $\Sigma' \subseteq \widetilde{\mathcal{M}}$, even if not minimal. The modules found via the oracle x-check will be called x-modules.

Before going to the next section to describe several notions of x-modules and the conditions to extract them, we complete the description of the properties of inseparability relations of interest for this thesis.

Definition 2.2.5 (Kontchakov et al., [KPS+09])**.** We say that an inseparability relation R is *robust under replacement* if, for all ontologies $\mathcal{O}, \mathcal{O}_1$, and $\mathcal{O}_2$, and for each signature Σ, we have that $\mathcal{O}_1 \cup \mathcal{O} \equiv_\Sigma^R \mathcal{O}_2 \cup \mathcal{O}$ whenever $\mathcal{O}_1 \equiv_\Sigma^R \mathcal{O}_2$ and $\widetilde{\mathcal{O}} \cap (\widetilde{\mathcal{O}}_1 \cup \widetilde{\mathcal{O}}_2) \subseteq \Sigma$.

This property guarantees that for any signature Σ, the R_Σ-module $\mathcal{M}$ can be used instead of $\mathcal{O}$, and hence it is fundamental in application scenarios as the reuse of ontologies. Moreover, R-modules based on inseparability relations that are robust under replacement benefit from other properties described in what follows.

Proposition 2.2.6 (Kontchakov et al., [KPS+09])**.** *Let R be a monotone inseparability relation that is robust under replacement. Then, every depleting R_Σ-module is a self-contained R_Σ module.*

Theorem 2.2.7 (Kontchakov et al., [KPS+09]). *Let R be a monotone inseparability relation that is robust under replacement, $\mathcal{O}$ an ontology, and Σ a signature. Then, there is a unique minimal depleting R_Σ-module $\mathcal{M} \subseteq \mathcal{O}$.*

By Theorem 2.2.7 we know that, if R is monotone and robust under replacement, then there is a unique minimal depleting R_Σ-module $\mathcal{M} \subseteq \mathcal{O}$, which by Proposition 2.2.6 is also self-contained. In the remainder of this thesis, we will deal with modules that satisfy all of these properties. Hence we define a class of modules that includes them all.

Definition 2.2.8. We define the *class of Σ-modules* as the set of all those notions x of modules that satisfy the properties of self-containment and depletion as in Definition 2.2.4, and such that, for each ontology $\mathcal{O}$ and each signature Σ, the x-module is uniquely determined.

For each notion x of a Σ-module, we denote the x-module for Σ in an ontology $\mathcal{O}$ with the symbol $x\text{-mod}(\Sigma, \mathcal{O})$. If ontology and module notion are clear from the context, we will use the short form $\mathcal{M}_\Sigma$.

2.3 Extracting Σ-modules

In the previous section we have defined Σ-modules as those notions x of R-modules that are self-contained, depleting, and uniquely determined. We now define two more properties of modules of interest when it comes to extract them. The uniqueness of Σ-modules guarantees that such modules satisfy these properties.

Definition 2.3.1 (Sattler *et al.*, [SSZ09]). Let x be a notion of module. Then we say that:

- for any two signatures Σ_1 and Σ_2, and any ontology $\mathcal{O}$, x satisfies the property of *monotonicity by signature enlargement* if the following condition holds:

$$(m_{\text{sig}})\ \Sigma_1 \subseteq \Sigma_2 \implies x\text{-mod}(\Sigma_1, \mathcal{O}) \subseteq x\text{-mod}(\Sigma_2, \mathcal{O});$$

- for any two ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$, and any signature Σ, x satisfies the property of *monotonicity by ontology enlargement* if the following condition holds:

$$(m_{\text{axs}})\ \mathcal{O}_1 \subseteq \mathcal{O}_2 \implies x\text{-mod}(\Sigma, \mathcal{O}_1) \subseteq x\text{-mod}(\Sigma, \mathcal{O}_2).$$

From the definition and the properties of Σ-modules an algorithm can be designed to extract the module $x\text{-mod}(\Sigma, \mathcal{O})$ for any signature Σ and any ontology

$\mathcal{O}$. Let x be a notion of Σ-module, and let us denote by x-check the oracle that decides whether an axiom needs to be included into the module for Σ. Note that an axiom α of interest can include terms that do not occur in Σ, and these terms can further interact on defining models or entailments over Σ, so the initial signature needs to be extended to $\Sigma \cup \widetilde{\alpha}$ in order to ensure depletion and self-containment. Hence, an x-module can be obtained by applying Algorithm 1.

Algorithm 1 General algorithm for computing a depleting x-module

1: **Input:** An ontology $\mathcal{O}$; a notion x of Σ-module; a signature Σ.
2: **Output:** The x-module $\mathcal{M}$ for Σ.
3: $\mathcal{M} \leftarrow \emptyset$
4: **repeat**
5: $\quad \Sigma_{\text{prev}} \leftarrow \Sigma \cup \widetilde{\mathcal{M}}$
6: $\quad$ **for** each $\alpha \in \mathcal{O}$ **do**
7: $\quad\quad$ **if** the x-check returns that α needs to be included in the module for $\Sigma \cup \widetilde{\mathcal{M}}$ **then**
8: $\quad\quad\quad \mathcal{M} \leftarrow \mathcal{M} \cup \{\alpha\}$
9: $\quad\quad$ **end if**
10: $\quad$ **end for**
11: **until** $\Sigma \cup \widetilde{\mathcal{M}} = \Sigma_{\text{prev}}$
12: **return** $\mathcal{M}$

This procedure terminates because a module is a subset of $\mathcal{O}$, so at each repeat-until iteration, either the module in enlarged and the set of axioms not in $\mathcal{M}$ is strictly smaller than at the previous step, or the module does not change, and the condition $\Sigma \cup \widetilde{\mathcal{M}} = \Sigma_{\text{prev}}$ stops the loop. In particular, the algorithm needs to perform at most polynomially many x-checks in the number of axioms of $\mathcal{O}$. Finally, please note that two notions x_1 and x_2 can be nested to obtain the module x_2-mod$(\Sigma, x_1\text{-mod}(\Sigma, \mathcal{O}))$ that is not larger than x_1-mod$(\Sigma, \mathcal{O})$.

Coverage, depletion, self-containment, and monotonicity are guaranteed to hold by the extraction Algorithm 1, provided that a suitable oracle is called in line 7. The notions of Σ-modules that we are going to analyse are modules based on conservativity and locality.

2.3.1 Conservativity-based Modules

Conservativity-based modules (CBMs) are minimal dCE- or mCE-modules in the sense of Definition 2.2.4. We already mentioned that these modules are

hard, or even impossible to compute in highly expressive DLs. However, for a few lightweight description logics in the $\mathcal{ELI}$ and DL-Lite families, CBMs can be computed efficiently. For example, Konev *et al.* in [KLWW08] describe a polynomial-time procedure for computing minimal depleting mCE-modules of acyclic $\mathcal{ELI}$-terminologies and an implementation in the system MEX. The oracle to be called in line 7 of Algorithm 1 decides, for each axiom $\alpha = \mathtt{A} \sqsubseteq \mathtt{C}$, whether some concept names that are (possibly indirectly) used in the definition of the concept name $\mathtt{A}$ in the complement of $\mathcal{M}$ are in the extended signature $\Sigma \cup \widetilde{\mathcal{M}}$. For $\alpha = \mathtt{A} \equiv \mathtt{C}$, a similar but more complex test is used. In [KLWW08] the same approach is describes for acyclic $\mathcal{ALCI}$ terminologies, but with a computationally more complex test for the case $\mathtt{A} \equiv \mathtt{C}$.

2.3.2 Modules based on other inseparability notions

Kontchakov *et al.* in [KPS+09, KWZ10] investigate the computation of different kinds of minimal modules for ontologies in different DL-lite dialects, including minimal dCE-modules (which are not necessarily depleting), and minimal depleting modules based on query-based inseparability (called MDQMs). Without going into the details of the underlying notion of query-based inseparability, we can say that MDQMs fit into the context of this thesis because the inseparability relation guarantees such module to be Σ-modules. The oracle to be called in line 7 decides whether $\mathcal{O} \setminus \mathcal{M} \cup \{\alpha\} \equiv^{q}_{\Sigma \cup \widetilde{\mathcal{M}}} \emptyset$.

MDQMs are not polynomial-time computable, but the approach described in [KPS+09, KWZ10] reduces the inseparability tests to satisfiability of quantified Boolean formulas (QBF) and employs state-of-the-art QBF solvers.

2.3.3 Modules based on Locality

Locality-based modules (LBMs) are an approximation of CBMs in the following sense: every LBM is a depleting dCE-module, but not necessarily a minimal one. That is, LBMs may contain axioms irrelevant for preserving entailments. In particular, for ontologies that fall into the lightweight fragments mentioned above, LBMs always contain the corresponding modules computed via the respective approaches. The major advantage of modules based on locality is that they are considerably easier to extract, and have been defined, implemented and used for extracting modules from ontologies encoded in expressive languages up to

$\mathcal{SROIQ}$ [CHKS08]. They come in two flavours – based on semantic and syntactic locality.

Semantic locality

The intention behind locality is to decide, for each axiom α, independently of the other axioms whether α is included in the module.

Definition 2.3.2. Let α be an axiom, Σ be a signature, and $(\Delta^{\mathcal{J}_1}, \cdot^{\mathcal{J}_1})$ be an interpretation over $\widetilde{\alpha} \setminus \Sigma$. Then, α is said to be $\mathcal{J}_1$-*local* w.r.t. the signature Σ if, for any interpretation $(\Delta^{\mathcal{J}_1}, \cdot^{\mathcal{J}_2})$ over Σ, the interpretation $(\Delta^{\mathcal{J}_1}, \cdot^{\mathcal{I}})$ over $\Sigma \cup \widetilde{\alpha}$ such that the interpretation function $\mathcal{I}$ expands both $\mathcal{J}_1$ and $\mathcal{J}_2$ is a model for α.

Intuitively, if an axiom α is $\mathcal{J}_1$-local w.r.t. a given signature Σ, then it is irrelevant to Σ since any expansion $\mathcal{I}$ of $\mathcal{J}_1$ leaves any interpretation of Σ possible.

If the interpretation $\mathcal{J}_1$ as in Definition 2.3.2 is defined by mapping any concept name $\mathtt{A} \in \widetilde{\alpha} \setminus \Sigma$ to S and each role name $\mathtt{r} \in \widetilde{\alpha} \setminus \Sigma$ to $S \times S$ for a subset $S \subseteq \Delta$, the notion of $\mathcal{J}_1$-locality will be simply called S-locality. In this case, instead of considering all interpretations over $\widetilde{\alpha} \setminus \Sigma$, we consider only "trivial interpretations".

Example 2.3.3. Let the domain Δ be $\{a\}$, let α be the axiom $\mathtt{A} \sqsubseteq \mathtt{B}$, and let Σ be $\{\mathtt{A}\}$. Then, α is not $\emptyset$-local w.r.t. Σ since the interpretation $(\Delta, \cdot^{\mathcal{I}})$ where $\mathtt{A}^{\mathcal{I}} = \Delta$ and $\mathtt{B}^{\mathcal{I}} = \emptyset$ is not a model for α. However, α is Δ-local w.r.t. Σ since any interpretation $(\Delta, \cdot^{\mathcal{I}})$ where $\mathtt{A}^{\mathcal{I}} = S$ for $S \subseteq \Delta$ and $\mathtt{B}^{\mathcal{I}} = \Delta$ is a model for α.

The different behaviour of locality depending on the choice of $\mathcal{J}_1$ can intuitively be seen as a way to determine *how* the interpretation of symbols in Σ is constrained by the axiom α. In the first case, α not being $\emptyset$-local means that $\mathtt{A}$ is indeed constrained by α *from above*, t.i. α bounds the maximal size of possible interpretations of $\mathtt{A}$ once a model for α is chosen. In the second case, α being Δ-local shows that $\mathtt{A}$ is not constrained by α *from below*, t.i. α does not bound the minimal size of possible interpretations of $\mathtt{A}$ once a model for α is chosen. Now, on the one hand is useful to be able to separate the different consequences that the notions of semantic locality show on an axiom. On the other hand, Example 2.3.3 shows that defining a "local axiom" as an "irrelevant axiom" *tout-court* is not appropriate since, depending on the notion of locality, we have different values of locality.

Beside having an interesting intuition behind them, the two notions of $\emptyset$- and of Δ-locality are clearly independent on the choice of the interpretation domain Δ. From now on we will consider only these notions of semantic locality.

Definition 2.3.4 (Cuenca Grau *et al.*. [CHKS08]). A $\mathcal{SROIQ}$-axiom α is called *$\emptyset$-local (Δ-local) w.r.t. the signature* Σ if, for each interpretation function $\mathcal{I}$, there exists an interpretation function $\mathcal{J}$ such that $\mathcal{I}|_\Sigma = \mathcal{J}|_\Sigma$, $\mathcal{J} \models \alpha$, and for each concept name $\mathtt{X} \in \widetilde{\alpha} \setminus \Sigma$, $\mathtt{X}^\mathcal{J} = \emptyset$ (for each $\mathtt{C} \in \widetilde{\alpha} \setminus \Sigma$, $\mathtt{C}^\mathcal{J} = \Delta$ and for each role name $\mathtt{r} \in \widetilde{\alpha} \setminus \Sigma$, $\mathtt{r}^\mathcal{J} = \Delta \times \Delta$).

Modules based on semantic locality are extracted using Algorithm 1 where the oracle to be called in line 7 decides $\emptyset$-locality (Δ-locality, respectively) of α w.r.t. $\Sigma \cup \widetilde{\mathcal{M}}$. This test can be performed using available DL-reasoners [CHKS08], which makes this problem considerably easier than testing conservativity. However, reasoning in expressive DLs is still complex, i.e., N2ExpTime-complete for $\mathcal{SROIQ}$ [Kaz08]. The output of Algorithm 1 is the $\emptyset$-module (Δ-module, respectively) of the input ontology $\mathcal{O}$ for Σ.

Notation 2.3.5. Given an ontology $\mathcal{O}$ and a signature Σ, the set of all axioms in $\mathcal{O}$ that are x-local w.r.t. Σ is denoted by $x\text{-}local(\Sigma, \mathcal{O})$.

Remark 2.3.6. A tautology is, by definition, a valid axiom τ that is valid in all interpretations $\mathcal{I}$ over $\widetilde{\tau}$. In particular, a tautology will always be x-local for any $x \in \{\emptyset, \Delta\}$, and no module based on semantic locality and computed via Algorithm 1 will contain a tautology.

From the previous remark, we have that in general the set of all modules based on semantic locality is does not cover $\mathcal{O}$, i.e., $\bigcup_\mathcal{M} \mathcal{M} \neq \mathcal{O}$. This is not an issue since they do not add any knowledge to an ontology $\mathcal{O}$. In other words, the ontology $\mathcal{O}' = \{\alpha \in \mathcal{O} \mid \alpha \text{ is not a semantic tautology}\}$ is logically equivalent to $\mathcal{O}$.

The following result proves that testing for locality defines a sufficient condition for $\mathcal{M}$ to be a dCE-based module for Σ of $\mathcal{O}$.

Proposition 2.3.7 (Cuenca Grau *et al.*, [CHKS08]). *Let $\mathcal{M}$ be a subset of an ontology $\mathcal{O}$ such that all axioms in $\mathcal{O} \setminus \mathcal{M}$ are $\emptyset$-local (or Δ-local) w.r.t. $\Sigma \cup \widetilde{\mathcal{M}}$. Then, $\mathcal{M}$ is a dCE-based module of $\mathcal{O}$ w.r.t. Σ.*

The inverse implication of Proposition 2.3.7 does not hold, as the following example shows.

Example 2.3.8. Let $\mathcal{O} = \{\mathtt{A} \equiv \mathtt{B}\}$ and $\Sigma = \{\mathtt{A}\}$. Then the single axiom is neither $\emptyset$- nor Δ-local w.r.t. Σ, but $\mathcal{O}$ is a dCE of the empty ontology w.r.t. Σ.

The following example shows a special case of a key property of $\emptyset$-modules: if a concept name $\mathtt{A}$ occurs in an $\emptyset$-module $\mathcal{M}$ of an ontology $\mathcal{O}$, then all the superconcepts $\mathtt{B}$ of $\mathtt{A}$ occur in $\mathcal{M}$.

Example 2.3.9. Let us now define the family of ontologies Chain_n for $n > 2$ defined as follows:

$$\alpha_1 : \mathtt{A}_0 \sqsubseteq \mathtt{A}_1$$
$$\dots$$
$$\alpha_{n-1} : \mathtt{A}_{n-2} \sqsubseteq \mathtt{A}_{n-1}$$
$$\alpha_n : \mathtt{A}_{n-1} \sqsubseteq \mathtt{A}_n.$$

Let $\overline{n}$ be fixed, $\mathsf{Chain}_{\overline{n}}$ the corresponding ontology, and $\Sigma = \{\mathtt{A}_0\}$ be the signature we want to extract the $\emptyset$-module $\mathcal{M}_\Sigma$ for. It is easy to see that α_1 is the only axiom to be non-local w.r.t. Σ, so that $\mathcal{M}_\Sigma \ni \alpha_1$. For the extraction of $\mathcal{M}_\Sigma$, then, the signature Σ is extended to include $\mathtt{A}_1$, and the $\emptyset$-check is run again over $\mathsf{Chain}_{\overline{n}}$. The same line of reasoning triggers a chain reaction that leads $\mathcal{M}_\Sigma$ to coincide with the whole ontology. In particular, all the concepts $\mathtt{A}_i$ with $i > 0$ that are superconcepts of $\mathtt{A}_0$ end up in $\mathcal{M}_\Sigma$.

Dually to $\emptyset$-modules, Δ-modules satisfy a similar property: if a concept name $\mathtt{A}$ occurs in an $\emptyset$-module $\mathcal{M}$ of an ontology $\mathcal{O}$, then all the subconcepts $\mathtt{B}$ of $\mathtt{A}$ occur in $\mathcal{M}$. These results are summarized in the following proposition.

Proposition 2.3.10 (Cuenca Grau *et al.*, [CHKS08], Corollary 47)**.** *Let $\mathcal{O}$ be a $\mathcal{SROIQ}$ ontology, and let $\mathtt{A}$ and $\mathtt{B}$ be two concept names in $\widetilde{\mathcal{O}}$. Then, the following conditions are equivalent:*

(1) $\mathcal{O} \models \mathtt{A} \sqsubseteq \mathtt{B}$*;*

(2) $\emptyset$-mod$(\{\mathtt{A}\}, \mathcal{O}) \models \mathtt{A} \sqsubseteq \mathtt{B}$*;*

(3) Δ-mod$(\{\mathtt{B}\}, \mathcal{O}) \models \mathtt{A} \sqsubseteq \mathtt{B}$.

From Proposition 2.3.7, we have that locality-based modules preserve entailments over their seed signatures. However, these modules are sometimes quite large; for example, given the ontology $\mathcal{O} = \{\mathtt{C}_i \sqsubseteq \mathtt{D} \mid 1 \leq i \leq n\}$, the Δ-module $\mathcal{M}_1 = \Delta\text{-mod}(\mathcal{O}, \{\mathtt{C}_1, \mathtt{D}\})$ contains the whole ontology. Now, we can find a smaller

ontology that preserves all the entailments over $\Sigma = \{\mathtt{C}_1, \mathtt{D}\}$ by extracting an $\emptyset$-module from $\mathcal{M}_1$, because by Proposition 2.3.7 we know that the result is still an mCE for Σ. In practice, we can then extract the $\emptyset$-module[1] for Σ from the ontology $\mathcal{M}_1$. The resulting set is again an mCE-based module, denoted by $\emptyset\Delta\text{-mod}(\Sigma, \mathcal{O})$. Please note that we can also start from an $\emptyset$-module, then extract a Δ-module from the previously extracted module, and denote the resulting module by $\Delta\emptyset\text{-mod}(\Sigma, \mathcal{O})$. Finally, we can keep nesting the extraction until a fixpoint is reached. The number of steps needed to reach this fixpoint can be at most as big as the number of axioms in $\mathcal{O}$ [SSZ09].

In general, $\emptyset$- and Δ-modules for a signature Σ in an ontology $\mathcal{O}$ are different, hence by nesting and alternating the two notions of modules we expect that the module obtained by starting with the extraction of an $\emptyset$-module is also different from the one obtained by starting with the extraction of a Δ-module. However, this difference is removed when a fixpoint is reached, i.e., the fixpoint module $\emptyset\Delta^*\text{-mod}(\Sigma, \mathcal{O})$ coincides with the fixpoint module $\Delta\emptyset^*\text{-mod}(\Sigma, \mathcal{O})$. We include here the unpublished proof for this result by Kazakov.

Proposition 2.3.11. *Given an ontology $\mathcal{O}$ and a signature Σ, we have that the two modules $\Delta\emptyset^*\text{-mod}(\Sigma, \mathcal{O})$ and $\emptyset\Delta^*\text{-mod}(\Sigma, \mathcal{O})$ coincide.*

Proof. By the notation $\Delta\emptyset\text{-mod}(\Sigma, \mathcal{O})$ we mean that the first extraction of a module is based on $\emptyset$-locality. Because of the fixpoint, we have:

$$\Delta\emptyset^*\text{-mod}(\Sigma, \mathcal{O}) = \emptyset\Delta^*\emptyset\text{-mod}(\Sigma, \mathcal{O}) = \emptyset\Delta^*\text{-mod}(\Sigma, \emptyset\text{-mod}(\Sigma, \mathcal{O})).$$

By Definition 2.3.1 of monotonicity for Σ-modules, it also holds that

$$\begin{array}{ccc} \emptyset\Delta^*\text{-mod}(\Sigma, \emptyset\text{-mod}(\Sigma, \mathcal{O})) & \subseteq & \emptyset\Delta^*\text{-mod}(\Sigma, \mathcal{O}) \\ & || & \\ \Delta\emptyset^*\text{-mod}(\Sigma, \mathcal{O}) & & \end{array}$$

This proves that $\Delta\emptyset^*\text{-mod}(\Sigma, \mathcal{O}) \subseteq \emptyset\Delta^*\text{-mod}(\Sigma, \mathcal{O})$. The converse inclusion can be proven in the same way. □

[1] The extraction of a Δ-module w.r.t. Σ from $\mathcal{M} = \Delta\text{-mod}(\Sigma, \mathcal{O})$ results in the same module $\mathcal{M}$ since all axioms in $\mathcal{M}$ are clearly non Δ-local w.r.t. Σ

Syntactic Locality

Since checking an axiom for locality against a signature is as hard as reasoning, we have that extracting a locality-based module has the same complexity. Now, since reasoning in expressive DLs is infeasible in the worst case, approximations have been defined. Here we focus on the notion of *syntactic locality*. The idea behind this notion is that the locality of an axiom α for a signature Σ can be often revealed already by α's syntax.

The syntactic identification of local axioms can be conceptually divided into two steps. The first step consists of identifying those complex expressions that can be trivially interpreted, whatever interpretation function $\mathcal{I}$ over Σ is taken into account.

Definition 2.3.12. Let $\mathcal{O}$ be an ontology, $\Sigma \subseteq \widetilde{\mathcal{O}}$ a signature, $\mathtt{A}$ a concept name, $\mathtt{C}, \mathtt{D}$ two (possibly complex) concepts, $\mathtt{r}$ a role name, and n a positive natural number. For each concept name $\mathtt{A} \notin \Sigma$ and each role name $\mathtt{r} \notin \Sigma$ we define two aliases denoted by using either the superscript $\bot$ or the superscript $\top$ (e.g., $\mathtt{A}^{\bot}$ and $\mathtt{A}^{\top}$). For the two notions $\bot, \top$ of locality, we inductively define the sets of $\bot$-*terms* $\mathrm{Bot}(\Sigma)$ and of $\top$-*terms* $\mathrm{Top}(\Sigma)$ is given in Table 2.5.

(a) $\bot$-locality

$\mathrm{Bot}(\Sigma) ::= \bot \mid \mathtt{A}^{\bot} \mid \mathtt{r}^{\bot} \mid \neg \mathtt{C}^{\top} \mid \mathtt{C} \sqcap \mathtt{C}^{\bot} \mid \mathtt{C}^{\bot} \sqcap \mathtt{C} \mid \exists \mathtt{r}.\mathtt{C}^{\bot} \mid \leq n\, \mathtt{r}.\mathtt{C}^{\bot} \mid \exists \mathtt{r}^{\bot}.\mathtt{C} \mid \leq n\, \mathtt{r}^{\bot}.\mathtt{C}$
$\mathrm{Top}(\Sigma) ::= \top \mid \neg \mathtt{C}^{\bot} \mid \mathtt{C}_1^{\top} \sqcap \mathtt{C}_2^{\top} \mid \leq 0\, \mathtt{r}.\mathtt{C}$

(b) $\top$-locality

$\mathrm{Top}(\Sigma) ::= \top \mid \mathtt{A}^{\top} \mid \mathtt{r}^{\top} \mid \neg \mathtt{C}^{\bot} \mid \mathtt{C}^{\top} \sqcap \mathtt{D}^{\top} \mid \exists \mathtt{r}^{\top}.\mathtt{C}^{\top} \mid \leq n\, \mathtt{r}^{\top}.\mathtt{C}^{\top} \mid \leq 0\, \mathtt{r}.\mathtt{C}$
$\mathrm{Bot}(\Sigma) ::= \bot \mid \neg \mathtt{C}^{\top} \mid \mathtt{C} \sqcap \mathtt{C}^{\bot} \mid \mathtt{C}^{\bot} \sqcap \mathtt{C} \mid \exists \mathtt{r}.\mathtt{C}^{\bot} \mid \leq n\, \mathtt{r}.\mathtt{C}^{\bot}$

Table 2.5: $\mathrm{Bot}(\Sigma)$ and $\mathrm{Top}(\Sigma)$ for the notions of locality $\bot$ and $\top$

The second step needed for the identification of the syntactically Σ-local axioms of an ontology $\mathcal{O}$ consists of parsing each axiom $\alpha \in \mathcal{O}$, with the aim of establishing whether an axiom's syntax enforces only a vacuous relationship between the terms in Σ and the complex expressions that are described in Definition 2.3.12.

Definition 2.3.13. Let $\mathcal{O}$ be an ontology, $\Sigma \subseteq \widetilde{\mathcal{O}}$ a signature, $\mathtt{A}$ a concept name, $\mathtt{C}, \mathtt{D}$ two (possibly complex) concepts, $\mathtt{r}$ a role name, n a positive natural number, and a an individual. An axiom $\alpha \in \mathcal{O}$ is called

(a) *syntactically $\bot$-local w.r.t. Σ* if it has one of the following forms:

$$\mathtt{C}^{\bot} \sqsubseteq \mathtt{C},\ \mathtt{C} \sqsubseteq \mathtt{C}^{\top},\ \mathtt{C}^{\bot} \equiv \mathtt{C}^{\bot},\ \mathtt{C}^{\top} \equiv \mathtt{C}^{\top},\ \mathtt{r}^{\bot} \sqsubseteq \mathtt{r},\ \mathrm{Trans}(\mathtt{r}^{\bot}), \mathrm{Func}(\mathtt{r}^{\bot})$$

where any expression with the superscript $\bot$ belongs to the set $\mathrm{Bot}(\Sigma)$, and any expression with the superscript $\top$ belongs to the set $\mathrm{Top}(\Sigma)$ as defined in Table 2.5(a);

(b) *syntactically $\top$-local w.r.t. Σ* if it has one of the following forms:

$$\mathtt{C}^{\bot} \sqsubseteq \mathtt{C},\ \mathtt{C} \sqsubseteq \mathtt{C}^{\top},\ \mathtt{C}^{\bot} \equiv \mathtt{C}^{\bot},\ \mathtt{C}^{\top} \equiv \mathtt{C}^{\top},\ \mathtt{r} \sqsubseteq \mathtt{r}^{\top},\ \mathrm{Trans}(\mathtt{r}^{\top}), \mathrm{Func}(\mathtt{r}^{\top}), (a : \mathtt{C}^{\top})$$

where any expression with the superscript $\bot$ belongs to the set $\mathrm{Bot}(\Sigma)$, and any expression with the superscript $\top$ belongs to the set $\mathrm{Top}(\Sigma)$ as defined in Table 2.5(b).

Given an ontology $\mathcal{O}$, a signature $\Sigma \subseteq \widetilde{\mathcal{O}}$, and a notion of locality $x \in \{\bot, \top\}$, the extraction of an x-module can be performed by applying Algorithm 1 where the x-check is defined upon Definition 2.3.13. As for semantic locality, the two notions can be nested until a fixpoint is reached, and the corresponding module notion will be denoted by the symbol $\top\bot^*$.

Clearly, checking an axiom α against locality is polynomial in $\ell(\alpha)$, and hence extracting a module based on syntactic locality is polynomial in the size of the ontology $\ell(\mathcal{O})$. Most importantly, syntactic modules approximate semantic modules, as stated in what follows.

Proposition 2.3.14 (Cuenca Grau *et al.*, [CHKS08])**.** *Let $\mathcal{O}$ be an ontology. If $\mathcal{M} \subseteq \mathcal{O}$ is a $\bot$-module w.r.t. a signature Σ, then $\mathcal{M}$ is also an $\emptyset$-module w.r.t. Σ. Similarly, if $\mathcal{M}$ is a $\top$-module w.r.t. Σ, then $\mathcal{M}$ is a Δ-module w.r.t. Σ.*

As for semantic locality, modules based on syntactic locality also provide coverage; that is, they capture *all* the relevant entailments, but not necessarily *only* those: Example 2.3.8 is applicable to this case too [CHKS08, JCS+08]. Syntactic locality was at first defined for the DL $\mathcal{SHIQ}$ in [CHKS08], and then extended to $\mathcal{SHOIQ}(\mathcal{D})$ in [JCS+08]. A locality-based module extractor is implemented in the OWL API.[2] The implementation provided is highly optimized, and it exploits

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

theoretical results and technical tricks to reduce the complexity down to $O(n \cdot s)$, where n is the number of axioms of $\mathcal{O}$ and $s = \max_\alpha \ell(\alpha)$. In particular, each axiom α is checked against locality at most $|\widetilde{\alpha}|$ times, i.e., only if the enlarged signature Σ_i at the i-th iteration includes at least a term $\mathtt{t} \in \widetilde{\alpha}$ that did not occur in Σ_{i-1}. The algorithm used has been devised by Tsarkov and it is described in [Tsa12].

Since $\bot$-modules are approximations of $\emptyset$-modules, and $\top$-modules are approximations of Δ-modules, we have that the same result to that described in Proposition 2.3.10 holds.

Corollary 2.3.15. *Let $\mathcal{O}$ be a $\mathcal{SROIQ}$ ontology, and let $\mathtt{A}$ and $\mathtt{B}$ be two concept names in $\widetilde{\mathcal{O}}$. Then, the following conditions are equivalent:*
(1) $\mathcal{O} \models \mathtt{A} \sqsubseteq \mathtt{B}$*;*
(2) $\bot$-mod$(\{\mathtt{A}\}, \mathcal{O}) \models \mathtt{A} \sqsubseteq \mathtt{B}$*;*
(3) $\top$-mod$(\{\mathtt{B}\}, \mathcal{O}) \models \mathtt{A} \sqsubseteq \mathtt{B}$.

Syntactic locality does not exploit any reasoning service, which is required in order to identify tautologies. However, particularly simple kinds of tautologies can be syntactically identified as such.

Definition 2.3.16. Let $x \in \{\bot, \top\}$ be a notion of syntactic locality. We say that an axiom α is a *syntactic tautology* if, for every signature $\Sigma \subseteq \widetilde{\alpha}$, α is x-local w.r.t. Σ.

Remark 2.3.17. In contrast with the syntactic tautologies, there are tautologies that cannot be syntactically identified, as it is the case for $\tau : \mathtt{A} \sqcap \mathtt{B} \sqsubseteq \mathtt{A} \sqcup \mathtt{B}$: by checking τ against $\bot$-locality w.r.t. the signature $\{\mathtt{A}, \mathtt{B}\}$ we see that both sides are non trivial complex concepts, hence τ is a non $\bot$-local axiom.

Similarly to tautologies for modules based on semantic locality, syntactic tautologies will not belong to any module based on syntactic locality and computed via Algorithm 1 in locality-based modules.

As for all notions of Σ-modules, we can nest $\bot$- and $\top$-modules to reduce the size of the module for a signature Σ. An analogous result to Proposition 2.3.11 holds.

Proposition 2.3.18. *Given an ontology $\mathcal{O}$ and a signature Σ, we have that the two modules $\top\bot^*\text{mod}(\Sigma, \mathcal{O})$ and $\bot\top^*\text{mod}(\Sigma, \mathcal{O})$ coincide.*

Proof. The proof is analogous to the proof for Proposition 2.3.11. □

2.4 Basic Notions of Algebra

We want to describe the relationships between an ontology $\mathcal{O}$ and a family $\mathcal{F}(\mathcal{O})$ of subsets thereof by means of a well-understood structure. To this end, we introduce in what follows some notions of algebra.

Definition 2.4.1. A *weak partial ordering* $\leq$ defined on a set X is a binary relation over any two elements $x_1, x_2 \in X$ satisfying 3 properties:

$$\begin{array}{lll} (1) & x_1 \leq x_1 & \text{(reflexivity)} \\ (2) & x_1 \leq x_2 \text{ and } x_1 \neq x_2 \implies x_2 \not\leq x_1 & \text{(weak antisymmetry)} \\ (3) & x_1 \leq x_2 \text{ and } x_2 \leq x_3 \implies x_1 \leq x_3 & \text{(transitivity)} \end{array}$$

In this case, the pair $(X, \leq)$ is called *partially ordered set* or *poset*.

Given a poset $(X, \leq)$ we can also dually define the binary relation $\geq$ over the set X to hold between the two elements of the ordered pair $(x_1, x_2) \in X \times X$ if, and only if, $x_2 \leq x_1$ holds. The resulting relation $\geq$ is also a *weak partial order*, and the structure $(X, \geq)$ a *poset*.

Definition 2.4.2. A *strict partial order* $<$ defined on a set X is a binary relation over any two elements $x_1, x_2 \in X$ satisfying 2 properties:

$$\begin{array}{lll} (1) & x_1 < x_2 \implies x_2 \not< x_1 & \text{(strict antisymmetry)} \\ (2) & x_1 < x_2 \text{ and } x_2 < x_3 \implies x_1 < x_3 & \text{(transitivity)} \end{array}$$

Also in this case the pair $(X, <)$ is called *partially ordered set* or *poset*.

Remark 2.4.3. For each weak partial order $\leq$ over a set X there is a correspondent strict partial order $<$ defined over X as the binary relation between any two elements $x_1, x_2 \in X$ such that $x_1 \neq x_2$ and $x_1 \leq x_2$. *Vice versa*, given a strict partial order $<$ over X there is a correspondent weak partial order $\leq$ defined over X as the binary relation between any two elements $x_1, x_2 \in X$ such that $x_1 < x_2$ or such that $x_1 = x_2$.

Remark 2.4.4. Given a strict partial order $<$ over a set X a Directed Acyclic Graph (V, E) is defined, where the set of vertexes V equals the set X, and the set of edges E equals the set of pairs (x_1, x_2) with $x_1 < x_2$. Such DAG is called the *Hasse Diagram* of the poset $(X, <)$. From Remark 2.4.3 we know that, given

any weak partial order $\leq$ we can derive the corresponding strict partial order $<$, and hence a corresponding Hasse diagram.

Definition 2.4.5. Two elements x, y of a poset $(X, <)$ are called *comparable* if $x < y$ or $y < x$. Otherwise they are said to be *incomparable*. A subset of X in which any two elements are comparable is a *chain*. A subset of X in which any two elements are incomparable is an *antichain*. If X is finite, we can define its *width* to be the maximal size of an antichain in $(X, <)$ and its *length* to be the maximal size of a chain in $(X, <)$ minus 1.

Definition 2.4.6. Given a poset $(X, <)$, an element $x \in X$ is called *minimal* if there exists no element y of X with $y < x$. Similarly, an element $x \in X$ is called *maximal* if there is no element $y \in X$ such that $x < y$.

Not all posets have minimal elements. However, if the number of elements is finite, then the minimal elements exist.

We will make use of the notions of *ideal* and of *principal ideal*, defined as follows.

Definition 2.4.7. For an element $x \in X$, the set $\downarrow x := \{y \in X \mid y \leq x\}$ is called the *principal ideal* of x. Given a subset $S = \{x_1, \ldots, x_\kappa\}$ of X, we define the *ideal* of S as the union of the principal ideals of all elements in S, t.i., $\bigcup_{x_i \in S} \downarrow x_i$.

Chapter 3

Modularity

In Section 2.2 we have described Conservative Extensions, and we have commented on the reasons that have lead to the definition of such a notion, i.e., to reduce the complexity of an ontology. Ideally, this notion is at the heart of the definition of *logical modules*, intended as fragments of an ontology $\mathcal{O}$ that preserve the entailments of $\mathcal{O}$ over a given signature Σ, and hence that provide *coverage* as defined in Definition 2.2.1. A module then can be used instead of $\mathcal{O}$ for many purposes, for example when it comes to reusing the knowledge modelled in comprehensive, possibly huge, well-designed ontologies.

However, deciding if a fragment $\mathcal{M}$ of an ontology $\mathcal{O}$ is a module in this sense is computationally hard: it is ExpTime-complete in $\mathcal{EL}$ [LW10], 2ExpTime-complete in $\mathcal{ALC}$ [GLW06] and $\mathcal{ALCQI}$ [LWW07], and undecidable for expressive DLs such as $\mathcal{ALCQIO}$ [LWW07]. Hence, a strong effort has recently been put in identifying feasible approximations of the notion of module. Extracting modules from an ontology is now a well studied task, and some of the notions of modules defined satisfy also more interesting logical properties, discussed in Section 3.1, of depletion, self-containment, and uniqueness, which CE-based modules do not satisfy in general. In this chapter we are going to analyse the following notions of modules:

- Parikh's approach to split a logical theory [Par99]

- the Δ-decomposition of an ontology [KLPW10]

- decomposition of an ontology using $\mathcal{E}$-connections [CPSK06]

- Σ-modules as defined in Section 2.3, that include some notions of modules

based on inseparability relations [KPS+09, KWZ10], conservativity-based modules that can be extracted by using the MEX system [KLWW08], and the modules based on semantic and syntactic locality [CHKS08, SSZ09, JCS+08].

Historically, the first three notions of modules have been defined by identifying, within an ontology $\mathcal{O}$, a decomposition of axioms, terms, or of the domain into representative fragments that may or may not have some logical interactions. By "representative" we mean that can be eventually combined to get the whole set of modules. The set of these fragments together with their interactions is called a *modular structure*. The usage of finding a modular structure includes:

- to reveal information about the *topics* described in $\mathcal{O}$ and on how these topics are described

- to check whether the intended modelling is adequately encoded in $\mathcal{O}$ by identifying connected vs. isolated parts of $\mathcal{O}$.

Quite differently, Σ-modules, designed for different purposes, are not provided with a modular structure: the substantial overlapping of Σ-modules causes an exponential blowup in their cardinality, and finding a set of representative modules is a task previously attempted with little results. However, a trivial modular structure can be defined as follows: given the set of all Σ-modules, their logical interrelations are defined by the set-theoretic inclusion relation.

In Section 3.2 we suggest an evaluation framework to summarize the properties of a modular structure. The evaluation framework is based upon four parameters: *coherence*, *granularity*, *dependence* or its dual concept of *independence*, and *computability*. The modular structures defined or induced by the aforementioned logical modules are then evaluated in Section 3.3 following this framework, and conclusions on the structures' applicability are drawn.

3.1 Desirable Properties of Modules

Modularity in Computer Science has been studied since the late 1960s. In 1972 in the area of Software Engineering, Parnas recommended the use of *functional modules*, i.e. incorporating a function, rather than *sequential modules*, i.e. determined by the temporal sequence of actions to perform. The modules are combined via

calls to the modules' *interface*, that is, the abstraction of the specification parameters (i.e. input and output). The system obtained is then independent of the specific implementation of the functional modules. This property is now called *encapsulation*, even though the term does not occur in Parnas' paper.

Parnas claims that the choice of the parameters to be exposed to the system needs to be defined upon a principled criterion, called *information hiding*: at design time, the engineer has to identify the implementation parameters (e.g. local variables). To preserve the system's correctness these parameters should be inaccessible by other modules, otherwise their values could uncontrollably change.

In what follows we are going to analyse whether these properties can be translated into analogous properties of interest in ontologies or in logical theories.

In ontologies several approaches to modularity have been designed and implemented, but we limit our investigation to those modules that preserve logical properties.

Definition 3.1.1 (Garson, 1989)**.** Given a logical theory $\mathcal{T}$, a fragment $\mathcal{T}'$ (not necessarily a subset) of $\mathcal{T}$ is a *logical module* if, for some background logic, the following two conditions hold:
1. $\mathcal{T}'$ is *locally correct*, i.e. any sentence provable in $\mathcal{T}'$ should be provable in $\mathcal{T}$.
2. $\mathcal{T}'$ is *locally complete*, i.e. every sentence in the signature of $\mathcal{T}'$ that is provable in $\mathcal{T}$ should be provable also in $\mathcal{T}'$.

Local correctness means that $\mathcal{T}'$ does not have unexpected entailments, i.e., entailments that do not already follow from $\mathcal{T}$. If $\mathcal{T}'$ is a subset of $\mathcal{T}$, then local correctness follows from the monotonicity of the underlying logic. Local completeness, on the other hand, guarantees that a logical module $\mathcal{T}'$ preserves all entailments of $\mathcal{O}$ over the signature $\widetilde{\mathcal{T}'}$, and hence $\mathcal{T}'$ can be used in place of the original theory $\mathcal{T}$ for everything that concerns $\widetilde{\mathcal{T}'}$. If $\mathcal{T}'$ is a subset of $\mathcal{T}$, then local completeness implies that $\mathcal{T}'$ is a self-contained $\equiv^{\mathrm{dCE}}_{\widetilde{\mathcal{T}'}}$-module of $\mathcal{T}$ in the sense of Definition 2.2.4. Because $\equiv^{\mathrm{dCE}}_{\widetilde{\mathcal{T}'}}$ is robust under vocabulary restrictions [KLWW09], it follows that $\mathcal{T}'$ covers $\mathcal{T}$ for the signature of $\mathcal{T}'$.

Definition 3.1.1 has a deductive flavour. Definition 2.2.4 is indeed a reformulation and generalization that uses different inseparability relations, including the model-theoretic counterpart $\equiv^{\mathrm{mCE}}$ of $\equiv^{\mathrm{dCE}}$.

Let us discuss the notions of coverage and self-containment in more detail. Coverage guarantees that the use of a module instead of the original theory is safe because all relevant entailments are preserved. Self-containment instead

aims at making all terms in a module equal in order to provide a module with a well-defined interface, which simply is the module's signature. The following is a simple example showing the roles that coverage and self-containment play in determining a logical module.

Example 3.1.2. Let us consider the following ontology:

$$\begin{aligned}\mathcal{O} = \{\alpha :& \mathtt{A} \sqsubseteq \mathtt{B} \sqcap (\mathtt{C} \sqcup \mathtt{D}),\\ \beta :& \mathtt{C} \sqsubseteq \mathtt{D},\\ \gamma :& \mathtt{B} \sqsubseteq \mathtt{C}\}.\end{aligned}$$

Then, the axiom α preserves all the entailments of $\mathcal{O}$ over $\Sigma_1 = \{\mathtt{A}, \mathtt{B}\}$. However, not all entailments of $\mathcal{O}$ over $\Sigma_2 = \{\mathtt{A}, \mathtt{C}, \mathtt{D}\}$ are preserved by $\{\alpha\}$ since $\eta : \mathtt{A} \sqsubseteq \mathtt{C}$ is lost. The set of axioms $\mathcal{M} = \{\alpha, \beta\}$ instead preserves all axioms over both Σ_1 and Σ_2, and it would make sense to consider both Σ_1 and Σ_2 as an interface for $\mathcal{M}$. However, $\mathcal{M}$ does not preserve all the entailments over the *union* $\Sigma_1 \cup \Sigma_2$ of the two interfaces, that also coincides with $\widetilde{\mathcal{M}}$, and thus $\mathcal{M}$ is clearly not self-contained. In particular, if we now were to use $\mathcal{M}$ in a reuse scenario, we would need a reminder of the signature that $\mathcal{M}$ is used for.

Providing coverage is a necessary condition for a module $\mathcal{M}$ of a logical theory to be considered functional: since $\mathcal{T}'$ preserves the entailments of $\mathcal{T}$ over the signature $\widetilde{\mathcal{T}'}$ it can be, for example, used instead of $\mathcal{T}$ for querying $\mathcal{T}$ over a restricted signature. The interface of a logical module is then the set of terms Σ over which the module $\mathcal{M}$ preserves the entailments.

Modules providing coverage are called *weak modules* in [KLWW08] because coverage alone cannot guarantee encapsulation: it can happen that the same logical consequences of a Σ-module $\mathcal{M}$ are entailed also by $\mathcal{O} \setminus \mathcal{M}$, as described in the following example.

Example 3.1.3. Let Dolphin be the following ontology:

$$\begin{aligned}\{\mathtt{Dolphin} &\sqsubseteq \mathtt{Fish},\\ \exists\mathtt{hasFin}.\top &\sqsubseteq \mathtt{Fish},\\ \mathtt{Dolphin} &\sqsubseteq \exists\mathtt{hasCaudalFin}.\top,\\ \mathtt{hasCaudalFin} &\sqsubseteq \mathtt{hasFin},\\ \mathtt{Mammal} &\sqsubseteq \neg\mathtt{Fish},\\ \mathtt{Dolphin} &\sqsubseteq \mathtt{Mammal}\}.\end{aligned}$$

Let us suppose that we want to revise the ontology to remove the entailment η : $\texttt{Dolphin} \sqsubseteq \bot$. We start by extracting the following subset $\mathcal{M}$, which is a logical module for the signature $\Sigma = \{\texttt{Dolphin}\}$, and therefore preserves the entailment $\eta : \texttt{Dolphin} \sqsubseteq \bot$.

$$\begin{aligned}\{&\texttt{Dolphin} \sqsubseteq \texttt{Fish},\\ &\texttt{Mammal} \sqsubseteq \neg\texttt{Fish},\\ &\texttt{Dolphin} \sqsubseteq \texttt{Mammal}\}.\end{aligned}$$

We can repair the module $\mathcal{M}$ by removing the axiom $\alpha : \texttt{Dolphin} \sqsubseteq \texttt{Fish}$, and we obtain the set of axioms $\mathcal{M}' = \mathcal{M} \setminus \{\alpha\} = \{\texttt{Mammal} \sqsubseteq \neg\texttt{Fish}, \texttt{Dolphin} \sqsubseteq \texttt{Mammal}\}$ that does not entail η anymore. However, α is not the only axiom that causes the unintended entailment η since by applying the same repair to $\mathsf{Dolphin}$ we obtain the ontology $\mathsf{Dolphin}' = \mathsf{Dolphin} \setminus \{\alpha\}$ that still entails η. In other words, a logical module for a signature Σ in general does not contain, i.e., *encapsulate*, all the reasons for the entailments over Σ to hold.

The issue discussed in Example 3.1.3 does not arise in logical modules that satisfy the additional property of depletion, see Definition 2.2.4. A module $\mathcal{M} \subseteq \mathcal{O}$ is depleting if $\mathcal{O} \setminus \mathcal{M}$ has no non-trivial entailments η over Σ, i.e., if the set of axioms in $\mathcal{O} \setminus \mathcal{M}$ is indistinguishable w.r.t. Σ from the empty set. Hence, if we want to look at the reasons in $\mathcal{O}$ for η to hold, we can restrict our attention to modify only the module $\mathcal{M}$ for $\widetilde{\eta}$ in $\mathcal{O}$. Depleting modules are called *strong modules* in [KLWW08].

The information hiding principle described by Parnas aims at making the system robust under design modifications. However, a substantial difference between modules in Software Engineering and modules in Ontology Engineering consists of the difficulty of defining a criterion such as information hiding in logical theories, as in the case of *global axioms* described in the following example.

Example 3.1.4. Let us consider the ontology:

$$\begin{aligned}\mathsf{Child} = \{&\texttt{Child} \sqsubseteq (=1\,\texttt{hasMother}.\texttt{Mother}),\\ &\texttt{Child} \sqsubseteq (=1\,\texttt{hasFather}.\texttt{Father}),\\ &\texttt{Mother} \sqsubseteq \neg\texttt{Father},\\ &\mathrm{Irrefl}(\texttt{hasMother}),\\ &\mathrm{Irrefl}(\texttt{hasFather})\}\end{aligned}$$

Child is clearly a module for its signature $\Sigma = \{\texttt{Child}, \texttt{Mother}, \texttt{Father}\}$. Suppose we now want to describe the situation as in the Bible story of Adam and Eve in the garden of Eden into an ontology Child′ obtained by adding to Child the following axiom:

$$\alpha : \top \sqsubseteq \{Adam\} \sqcup \{Eve\}.$$

The signature $\widetilde{\alpha}$ is clearly disjoint from the ontology's signature Σ. However, the ontology $\mathsf{Child} \cup \{\alpha\}$ generates the new entailment $\texttt{Child} \sqsubseteq \bot$. Hence, it is not possible to protect the meaning of the terms in a module $\mathcal{M}$ by preventing the changes to use terms from the module's interface $\widetilde{\mathcal{M}}$.

This example shows that importing a module $\mathcal{M}$ of an ontology $\mathcal{O}$ into another ontology $\mathcal{O}'$ could have a logical impact—both on $\mathcal{M}$ and on $\mathcal{O}'$—whose scope is difficult to evaluate, and this is the reason why the information hiding principle is hard to follow for logical theories.

Another property of modules of ontologies that we want to investigate is their computability. Decidability is one of the key properties of DL ontologies that makes ontologies useful in practice, and a lot of effort has been put in identifying tractable fragments of DL languages/optimizing reasoners implementations so that ontologies encoded in intractable dialects can still be efficiently used for reasoning tasks. With a similar spirit, a module of an ontology should be useful in practice, hence low complexity is another key aspect to be considered when modules are used.

3.2 Evaluating a Modular Structure

As mentioned in Section 3.1, the original aim of modularity in Computer Science was to decompose complex systems into more manageable, loosely related fragments. However, the system functionality is still the key of the decomposition and the composition of modules. Hence, a modularised system can be represented as the set of its components, and the interrelations between the components occur through the components' interfaces.

For ontologies, we define the notion of *modular structure* to be a pair $(\mathfrak{F}, \rightarrow)$, where $\mathfrak{F}$ is a representative set of modules of an ontology $\mathcal{O}$, and $\rightarrow$ is a binary relation between the elements of $\mathfrak{F}$ that captures suitable logical interactions between the modules of $\mathcal{O}$; in particular $\mathcal{M}_1 \rightarrow \mathcal{M}_2$ means that $\mathcal{M}_1$ needs to

import $\mathcal{M}_2$ to preserve some logical properties. One possible approach to reveal the impact of adding, removing, or modifying an axiom in an ontology consists of comparing the two modular structures of the initial ontology $\mathcal{O}$ and of the modified ontology $\mathcal{O}'$. Clearly, there can be more than one notion of modular structure since there is more than one notion of module. Depending on the notion used, different kinds of impact can be revealed. By evaluating a modular structure, then, we can understand what it means for two modules to be distinct, or to be related, and how coarse the modularisation is.

In the context of Software Engineering, Constantine in 1974 suggested two measures to be considered when defining functional modules to obtain a structured system: *cohesion* and *coupling* [SMC74]. Cohesion is a measure for evaluating how relevant the components of a module are to the function the module is designed for: the higher the cohesion, the better the module. Coupling is a measure for the substitutability of a logical module, and it tries to capture the degree of encapsulation of a module by measuring the interactions between modules: the looser the coupling, the better the modularisation. Despite being different concepts, cohesion and coupling often anti-relate: a modularised system that shows a high cohesion often also shows a low coupling, and *vice versa*.

Similar aspects can be studied for evaluating the modular structures of a logical theory. In particular, we will take into account:

1. The internal logical cohesion of a module, called *coherence*.
2. The coupling between the elements of different modules expressed in terms of their *dependence/independence*.

We also want to evaluate the general behaviour of a modular structure. To this aim, we define another aspect that takes into account the modular structure as a whole:

3. The coarseness of the set of modules, expressed in terms of its *granularity*.

3.3 Evaluation of the existing Modular Structures

In this section we describe and evaluate the modules identified in [Par99, KLPW10, CPSK06]. In these cases, the modules of an ontology $\mathcal{O}$ (or of a logical theory $\mathcal{T}$) are obtained by *decomposing* $\mathcal{O}$, and the basic fragments obtained together with their interrelations define different modular structures.

3.3.1 Parikh's Approach

The logical formalisation of this first notion of modular structure was introduced by Parikh in 1999 in the context of Belief Revision [Par99]. The question addressed originally is essentially the following: if we want to revise a theory with a new piece of knowledge that contradicts some of what is entailed, do we have to check it against the whole theory? Or do we have some kind of safety that allows us not to touch those parts that are irrelevant for this new finding? He formalises a way to "split" a logical theory $\mathcal{T}$ into signature-disjoint parts.

Definition 3.3.1 (Parikh, 1999)**.** Let $\mathcal{T}$ be a logical theory over the signature $\widetilde{\mathcal{T}}$ and let $\{\widetilde{\mathcal{T}_1}, \widetilde{\mathcal{T}_2}\}$ be a partition of $\widetilde{\mathcal{T}}$. We say that $\widetilde{\mathcal{T}_1}, \widetilde{\mathcal{T}_2}$ *split* the theory $\mathcal{T}$ if there are formulae α over $\widetilde{\mathcal{T}_1}$ and β over $\widetilde{\mathcal{T}_2}$ such that the logical closure of $\{\alpha, \beta\}$ coincides with the logical closure of $\mathcal{T}$. In this case, we say that $\{\widetilde{\mathcal{T}_1}, \widetilde{\mathcal{T}_2}\}$ is a $\mathcal{T}$-splitting. In general, we say that (mutually disjoint) signatures $\widetilde{\mathcal{T}}, \ldots, \widetilde{\mathcal{T}_n}$ split $\mathcal{T}$ if there exist formulae $\alpha_i \in \widetilde{\mathcal{T}_i}$ for $i = 1, \ldots, n$ such that $\mathcal{T}$ is the logical closure of $\{\alpha_1, \ldots, \alpha_n\}$.

In Parikh's approach, then, a logical module is a set $\mathcal{T}_1$ such that the signature partition $\{\widetilde{\mathcal{T}_1}, \widetilde{\mathcal{T}} \backslash \widetilde{\mathcal{T}_1}\}$ splits the theory $\mathcal{T}$. It has to be pointed out that the theories $\mathcal{T}_i$ are not in general subsets of formulae from $\mathcal{T}$ since rewriting could be required to express a logical theory $\mathcal{T}_j'$ over Σ_j equivalent to $\mathcal{T}_j$. Please note that Parikh is focusing on SOL, so any finite set of formulae can be equivalently expressed by a single formula. Hence, it may be the case that the set $\mathcal{T}$ is a singleton, but Parikh's splitting is non trivial.

Example 3.3.2. Let us consider the singleton ontology:

$$\mathsf{One} = \{\top \sqsubseteq (\neg \mathtt{A}_1 \sqcup \mathtt{B}_1) \sqcap (\neg \mathtt{A}_2 \sqcup \mathtt{B}_2) \sqcap \ldots \sqcap (\neg \mathtt{A}_n \sqcup \mathtt{B}_n)\}$$

One can be rewritten into the logically equivalent ontology

$$\begin{aligned} \mathsf{All} = \{&\mathtt{A}_1 \sqsubseteq \mathtt{B}_1, \\ &\mathtt{A}_2 \sqsubseteq \mathtt{B}_2 \\ &\ldots \\ &\mathtt{A}_n \sqsubseteq \mathtt{B}_n\} \end{aligned}$$

Hence, the signature partition $\Sigma_1 = \{\mathtt{A}_1, \mathtt{B}_1\}, \ldots, \Sigma_n = \{\mathtt{A}_n, \mathtt{B}_n\}$ splits the ontology One.

A nice property of this approach is that, for each theory $\mathcal{T}$ the $\mathcal{T}$-splitting is uniquely determined.

Theorem 3.3.3 (Parikh, 1999)**.** *Given a theory $\mathcal{T}$ over the signature $\widetilde{\mathcal{T}}$, there is a unique finest $\mathcal{T}$-splitting of $\widetilde{\mathcal{T}}$, i.e. one which refines every other $\mathcal{T}$-splitting.*

The uniqueness of theory splitting allows us to identify a well-defined modular structure of any logical theory $\mathcal{T}$: in this case, the representative modules are those that determine the finest $\mathcal{T}$-splitting. A module in $\mathcal{T}$ is simply the union of some of these basic modules.

Example 3.3.4. Let us consider the following ontology:[1]

$$\begin{aligned}\mathsf{Tree} = \{&\mathtt{Tree} \sqsubseteq \exists \mathtt{isMadeFrom}.\mathtt{Seed},\\ &\mathtt{Seed} \sqsubseteq \exists \mathtt{isMadeFrom}.\mathtt{Fruit},\\ &\mathtt{Fruit} \sqsubseteq \exists \mathtt{isMadeFrom}.\mathtt{Flower},\\ &\mathtt{Flower} \sqsubseteq \exists \mathtt{isMadeFrom}.\mathtt{Branch},\\ &\mathtt{Branch} \sqsubseteq \exists \mathtt{isMadeFrom}.\mathtt{Tree},\\ &\mathtt{Mountain} \sqsubseteq \exists \mathtt{isMadeFrom}.\mathtt{Soil}\}.\end{aligned}$$

Parikh's splitting consists of one module containing all the terms in Tree. However, there is a clear logical difference between the set of terms $\Sigma_1 = \{\mathtt{Tree}, \mathtt{Seed}, \mathtt{Fruit}, \mathtt{Flower}, \mathtt{Branch}\}$ and the set $\Sigma_2 = \{\mathtt{Mountain}, \mathtt{Soil}\}$: by interpreting $\mathtt{Tree}$ as the empty set, all the other terms in Σ_1 are unsatisfiable, whilst there is no constraint in the interpretation of the terms in Σ_2. Similarly, by interpreting $\mathtt{Soil}$ as the empty set we get that $\mathtt{Mountain}$ is unsatisfiable, whilst there is no constraint in the interpretation of the terms in Σ_1. However, we cannot separate these two sets because they share the term $\mathtt{isMadeFrom}$, so that Tree cannot be rewritten as the union of two signature-disjoint theories.

Example 3.3.4 shows that the induced notion of coherence is rather loose. Indeed, two terms $\mathtt{A}$ and $\mathtt{B}$ belong to the same module if there is an arbitrary long sequence of intermediate terms $\{\mathtt{A}_i \mid i \leq n, n \in \mathbb{N}\}$ such that $\mathtt{A}$ interacts (in

[1] This ontology is an adaptation and translation into DL of some fragments from the Italian song *Ci vuole un fiore* (It takes a flower), by G. Rodari, L. Enriquez, and S. Endrigo.

the model-theoretic sense described above) with $\mathtt{A}_0$, $\mathtt{A}_i$ interacts with $\mathtt{A}_{i+1}$ for all $i \in \{1, \dots n\}$, and $\mathtt{A}_n$ interacts with $\mathtt{B}$. However, $\mathtt{A}$ and $\mathtt{B}$ can still not interact with each other. At the same time, distinct modules are completely unrelated. This situation is described in the following example.

Example 3.3.5. Let us consider the following family of ontologies:

$$\begin{aligned}\mathsf{Zigzag}_n = \{&\mathtt{A}_1 \sqsubseteq \mathtt{B}_1,\\ &\mathtt{A}_2 \sqsubseteq \mathtt{B}_1,\\ &\mathtt{A}_2 \sqsubseteq \mathtt{B}_2,\\ &\dots,\\ &\mathtt{A}_n \sqsubseteq \mathtt{B}_n,\\ &\mathtt{A}_{n+1} \sqsubseteq \mathtt{B}_n\}\end{aligned}$$

The relation between the concepts in Zigzag_n is represented in Figure 3.1.

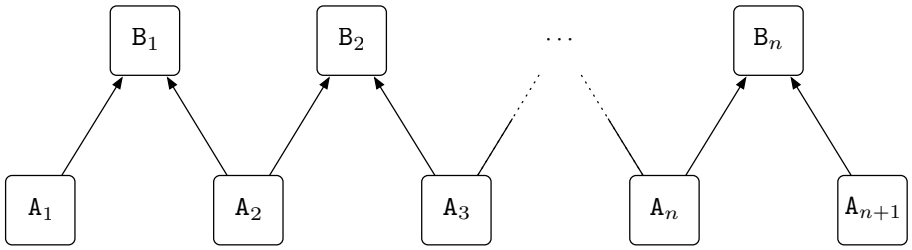


Figure 3.1: Concept relations in Zigzag_n

For $n \geq 3$, the concepts $\mathtt{A}_1$ and $\mathtt{A}_{n+1}$ are completely unrelated since Zigzag_n has only trivial entailments over the signature $\{\mathtt{A}_1, \mathtt{A}_{n+1}\}$. However, $\mathtt{A}_1$ and $\mathtt{A}_{n+1}$ still belong to the same Parikh's module.

The notion of dependence in Parikh is trivial since the modules, when two or more of them exist, do not interact at all.

Please note that two equivalent logical theories generate the same partitioning. Hence, the granularity of this modular structure reveals subterminologies that are completely unrelated, regardless of the syntactic encoding of the logical theory.

To sum up, the Parikhian modular structure can be evaluated as follows:
- Coherence: loose;
- Dependence relations: trivial;
- Granularity: coarse.

Parikh is not concerned in studying the practical feasibility of computing the signature decomposition of a logical theory. However, some insight on this issue are provided by the computational analysis of signature Δ-decompositions, i.e. a generalization of signature decomposition proposed in [KLPW10] and described in the next subsection.

3.3.2 Signature Δ-decomposition

The modules defined by Parikh have proved to be too coarse-grained for both Tree and Zigzag_n: all the axioms ended up into the same and only module, regardless of whether we could easily point out intuitively independent fragments.

Parikh's notion of coherence can be made stronger by imposing that any (part of an) ontology is coherent if its signature cannot be decomposed into disjoint subsets *by discarding some special, common terms*. These common terms can be distinctive of the whole area, but can be used in different contexts, and enforce substantially different logical relations. In Example 3.3.4, `isMadeFrom` is such a special term.

In [Pon08] this idea has been formalized and proposed by introducing *signature Δ-decompositions*. This approach has been further investigated in [KLPW10], where the authors explain that some terms behave like logical symbols under certain points of view. So, the idea is to identify a set Δ of these terms and then to decompose the signature of a logical theory $\mathcal{T}$ into disjoint subsignatures Σ_i, such that there exist theories $\mathcal{T}_i$ over the signatures $\Sigma_i \cup \Delta$ with $\bigcup_i \mathcal{T}_i \equiv \mathcal{T}$.

Definition 3.3.6 (Konev *et al.*, 2010)**.** Let $\mathcal{T}$ a finite theory of formulae in SOL, $\Delta \subseteq \widetilde{\mathcal{T}}$ and $\mathcal{L}$ a fragment of SOL. A partition $\Sigma_1, \ldots, \Sigma_n$ of $\widetilde{\mathcal{T}} \setminus \Delta$ is called a *signature Δ-decomposition of $\mathcal{T}$ in $\mathcal{L}$* if there are $\mathcal{T}_1, \ldots, \mathcal{T}_n$ theories of formulae in $\mathcal{L}$ such that
- $\widetilde{\mathcal{T}_i} \subseteq \Sigma_i \cup \Delta$ for $i = 1, \ldots, n$
- $\mathcal{T}_1 \cup \ldots \cup \mathcal{T}_n \equiv \mathcal{T}$.
$\mathcal{T}_1, \ldots, \mathcal{T}_n$ is called a *realization* of the signature Δ-decomposition $\Sigma_1, \ldots, \Sigma_n$ in $\mathcal{L}$.

For $\mathcal{L} =$ SOL, the realization of a signature Δ-decomposition always exists, and, as in Parikh's approach, is unique.

Theorem 3.3.7 (Konev *et al.*, 2010)**.** *Let $\mathcal{T}$ a finite theory of SO formulae, $\Delta \subseteq \widetilde{\mathcal{T}}$, and let $\Sigma_1, \ldots, \Sigma_n$ and $\Pi_1, \ldots, \Pi_m$ be Δ-decompositions of $\mathcal{T}$ in SOL.*

Then, the partition $\Sigma_i \cap \Pi_j$ *for all* i, j *with* $\Sigma_i \cap \Pi_j \neq \emptyset$ *of* $\widetilde{\mathcal{T}} \setminus \Delta$ *is a* Δ*-decomposition of* $\mathcal{T}$ *in SOL. Thus, there exists a unique finest* Δ*-decomposition of* $\mathcal{T}$ *in SOL called* Unique Decomposition Realization *(UDR).*

Set $\mathcal{T}_\Delta$ to be a theory over Δ such that, for any axiom η over $\widetilde{\mathcal{T}}_\Delta$, $\mathcal{T}_\Delta \models \eta \Leftrightarrow \mathcal{T} \models \eta$. Then, we define a Δ-*module* to be a union of $\mathcal{T}_\Delta$ with some theories $\mathcal{T}_i$ belonging to the UDR.

In contrast with what happens in the case of SOL, a realization of a signature Δ-decomposition in a language $\mathcal{L}$ different from SOL does not necessarily exist. In other words, in less expressive logics, like some DLs, the UDR does not exists in general. However, for some languages we have the guarantee that a UDR always exists, and when it does, it coincides with the UDR determined in SOL. A sufficient condition for this to hold is the language $\mathcal{L}$ to satisfy the *Parallel Interpolation Property* (*PIP*) defined as follows:

Definition 3.3.8 (Konev *et al.*, 2010)**.** Let $\mathcal{L}$ be a fragment of SO, $(\mathcal{T}_1, \mathcal{T}_2)$ be two sets of SO formulae, α an SO formula with $\mathcal{T}_1 \cup \mathcal{T}_2 \models \alpha$, and Δ a signature. A pair $\mathcal{T}_1', \mathcal{T}_2'$ with $\mathcal{T}_i'$ encoded in a fragment of $\mathcal{L}$ for $i = 1, 2$ is called a Δ-*parallel intermediate* of $(\mathcal{T}_1, \mathcal{T}_2)$ and α in $\mathcal{L}$ if the following conditions hold:

- $\mathcal{T}_i \models \mathcal{T}_i'$ for $i = 1, 2$;
- $\widetilde{\mathcal{T}}_i' \setminus \Delta \subseteq \widetilde{\mathcal{T}}_i \cap \widetilde{\alpha}$ for $i = 1, 2$;
- $\mathcal{T}_1' \cup \mathcal{T}_2' \models \alpha$.

$\mathcal{L}$ has the *parallel interpolation property* (*PIP*) if, for all $\mathcal{T}_1, \mathcal{T}_2$ encoded in $\mathcal{L}$, all α in $\mathcal{L}$, and all signatures Δ such that

1. $\widetilde{\mathcal{T}}_1 \cap \widetilde{\mathcal{T}}_2 \subseteq \Delta$,
2. $\mathcal{T}_1 \cup \mathcal{T}_2 \models \alpha$,
3. $\mathcal{T}_1$ and $\mathcal{T}_2$ are Δ-inseparable w.r.t. $\mathcal{L}$

there exists a Δ-parallel intermediate of $(\mathcal{T}_1, \mathcal{T}_2)$ and α in $\mathcal{L}$.

Theorem 3.3.9 (Konev *et al.*, 2010)**.** *Let* $\mathcal{L}$ *be a fragment of SOL with the PIP. Then:*

1. $\mathcal{L}$*-decompositions coincide with SOL-decompositions;*
2. $\mathcal{L}$ *has UDR.*

The previous result makes sense of analysing, for ontologies encoded in a language with the PIP, the modular structure defined by the unique finest Δ-decomposition in SOL. In this case, the representative set of Δ-modules is the

set of theories $\{\mathcal{T}_\Delta, \mathcal{T}_1 \cup \mathcal{T}_\Delta, \ldots, \mathcal{T}_n \cup \mathcal{T}_\Delta, \}$. The dependence between modules is slightly more interesting than the Parikhian one, and it consists of one independent module ($\mathcal{T}_\Delta$) and n modules ($\mathcal{T}_i \cup \mathcal{T}_\Delta$) dependent on the common module $\mathcal{T}_\Delta$.

Example 3.3.10. Let us consider the Zigzag_n family of ontologies as in Example 3.3.5 for $n \geq 3$. Then, we see that by choosing Δ to be a singleton set $\{\mathtt{A}_i\}$, where $2 \leq i \leq n\}$ the Δ-decomposition obtained consists of three parts: Δ, $\{\mathtt{A}_1, \ldots, \mathtt{A}_{i-1}, \mathtt{B}_1, \ldots, \mathtt{B}_{i-1}\}$, and $\{\mathtt{A}_{i+1}, \ldots, \mathtt{A}_{n+1}, \mathtt{B}_i, \ldots, \mathtt{B}_n\}$. Similarly, if $\Delta = \{\mathtt{B}_i\}$ with $1 \leq i \leq n$, the corresponding Δ-decomposition is Δ, $\{\mathtt{A}_1, \ldots, \mathtt{A}_i, \mathtt{B}_1, \ldots, \mathtt{B}_{i-1}\}$, and $\{\mathtt{A}_{i+1}, \ldots, \mathtt{A}_{n+1}, \mathtt{B}_{i+1}, \ldots, \mathtt{B}_n\}$.

If we analyse all these cases, we notice that, for all $i \in \{1, \ldots, n\}$ the terms $\mathtt{A}_i$ and $\mathtt{B}_i$ are always either in the same part, or in related parts. The same happens for the terms $\mathtt{A}_{i+1}$ and $\mathtt{B}_i$. In contrast with this situation, all the other pairs of terms can be separated by a suitable choice of Δ, so it is clear that the logical relation between these pairs is looser than the one between the pairs listed before. Hence, the Δ-decomposition reveals stronger logical relations between the terms of an ontology $\mathcal{O}$.

Example 3.3.11. Let us consider again the ontology Tree as in Example 3.3.4. Then, by setting $\Delta = \{\mathtt{isMadeFrom}\}$, we obtain a decomposition of the signature of Tree into the two fragments $\Sigma_1 = \{\mathtt{Tree}, \mathtt{Seed}, \mathtt{Fruit}, \mathtt{Flower}, \mathtt{Branch}\}$ and $\Sigma_2 = \{\mathtt{Mountain}, \mathtt{Soil}\}$. This decomposition captures exactly the desired logical relation as described in Example 3.3.4.

Please note that the choice of the set Δ hugely influences the decomposition obtained: given a different set $\Delta' = \{\mathtt{isMadeFrom}, \mathtt{Tree}, \mathtt{Fruit}\}$, we obtain a Δ'-split consisting of 3 parts: $\Sigma'_1 = \{\mathtt{Seed}\}$, $\Sigma'_2 = \{\mathtt{Flower}, \mathtt{Branch}\}$, and $\Sigma'_3 = \{\mathtt{Mountain}, \mathtt{Soil}\}$.

From Example 3.3.11 it is clear that the major challenge for Δ-decomposing an ontology is the selection of a suitable set Δ. As the authors say, they "do not expect signature decompositions to be a push-button technique, but rather envision an iterative and interactive process of understanding and improving the structure of an ontology, where the designer repeatedly chooses sets Δ and analyzes the impact on the resulting decomposition". Hence, this method does not produce any stable, intrinsic modular structure—at least, we still do not have conditions to ensure such a property.

In [KLPW10] the computability of signature Δ-decompositions has been analysed for those languages that are shown to have the PIP. In particular, finding the finest Δ-decomposition is polynomial in DL-Lite, and in $\mathcal{EL}$ with further constraints (either $\Delta = \emptyset$ or $\mathcal{O}$ being role-acyclic); ExpTime-complete in $\mathcal{ALC}, \mathcal{ALCI}, \mathcal{ALCQ}$ or $\mathcal{ALCQI}$, and in $\mathcal{ALCH}$ or $\mathcal{ALCHI}$ under the condition that Δ contains all the role and individual names from $\widetilde{\mathcal{O}}$. For those languages that do not have the PIP the existence of a Δ-decomposition is not guaranteed.

To sum up, the modular structure based on the signature Δ-decomposition can be evaluated as follows:
- Coherence: ranging from loose to strict, depending on Δ;
- Dependence relations: almost trivial, and of depth at most 1;
- Granularity: ranging from fine to coarse, depending on Δ.
- Computability: when the UDR exists, it depends on the language, and it can range from polynomial to ExpTime.

3.3.3 $\mathcal{E}$-connections

An orthogonal approach to decomposing an ontology into modules consists of trying to capture different *kinds* of things in an ontology, and how the ontology describes some things in terms of other ones. For example we could have an ontology dealing with the concept of `Person` described both in terms of what she eats and of her occupational status. Ideally, a modular structure should be able to reveal these logical interactions, and hence to retrieve a non trivial structuring of the ontology's modules.

In [CPSK06] the authors apply $\mathcal{E}$-connections to *decompose* ontologies, instead of composing them as originally defined in [KLWZ04]. The notion of module they are searching for is such that no subsumption relations exist between a concept (as in DLs, i.e. unary predicates) inside a given module and concepts outside this module. This intuition leads to the following notion of module.

Definition 3.3.12 (Cuenca Grau *et al.*, 2006)**.** A TBox $\mathcal{M}_{\mathtt{A}} \subseteq \mathcal{O}$ is a *module* for a concept $\mathtt{A} \in \widetilde{\mathcal{O}}$ if:
1. $\mathcal{M}_{\mathtt{A}}$ is a logical module in $\mathcal{O}$ as in Definition 3.1.1
2. for every concept $\mathtt{B} \in \widetilde{\mathcal{O}}$, the following holds:
   (a) $\mathcal{M}_{\mathtt{A}} \models \{\mathtt{A} \sqsubseteq \mathtt{B}\} \iff \mathcal{O} \models \{\mathtt{A} \sqsubseteq \mathtt{B}\}$
   (b) $\mathcal{M}_{\mathtt{A}} \models \{\mathtt{B} \sqsubseteq \mathtt{A}\} \iff \mathcal{O} \models \{\mathtt{B} \sqsubseteq \mathtt{A}\}$

3. there are no concepts $\mathtt{C}, \mathtt{D} \in \widetilde{\mathcal{O}}$ such that $\mathtt{C} \in \widetilde{\mathcal{M}_{\mathtt{A}}}$, $\mathtt{D} \notin \widetilde{\mathcal{M}_{\mathtt{A}}}$ and either $\mathcal{O} \models \mathtt{C} \sqsubseteq \mathtt{D}$ or $\mathcal{O} \models \mathtt{D} \sqsubseteq \mathtt{C}$.

To obtain such modules from an ontology, the authors describe a 3-steps algorithm: a safety-check, a partitioning algorithm, and the identification and extraction of modules.

The safety-check aims at identifying if the ontology $\mathcal{O}$ contain any axiom that constrains the domain size, as the axiom α in the extended ontology $\mathsf{Child} \cup \{\alpha\}$ in Example 3.1.4. If some of these axioms occur in $\mathcal{O}$, then the ontology is said to be not safe, and this algorithm cannot be applied.

After applying the partition algorithm, the concepts are grouped in parts that do not have to share elements in their extensions (see Theorem 3 in [CPSK06]). In symbols, there exists a model $\mathcal{I}$ for $\mathcal{O}$ such that $\mathtt{C}^{\mathcal{I}} \cap \mathtt{D}^{\mathcal{I}} = \emptyset$ for all concepts $\mathtt{C}$ and $\mathtt{D}$ that belong to different parts. In particular, we obtain a partition of the *domain*, whose parts can be of three types: (Red) those which import vocabulary from others, (Blue) those whose vocabulary is imported, and (Green) isolated parts. Intuitively, this property means that either the parts correspond to actual non-overlapping subject matters, or the ontology is underspecified and some of the parts correspond to "unused information".

In contrast to signature Δ-decompositions, and like in Parikh's approach, the procedure described here is completely automatic, so for any ontology $\mathcal{O}$ there is a unique decomposition into $\mathcal{E}$-connections. The ontology $\mathcal{O}$ gets partitioned into fragments that correspond to independent parts of the domain, connected by non trivial dependence relations. In particular, distinct parts can share part of the vocabulary, but no instances. By automatically labelling each part with the highest common concept name in the concept hierarchy, we obtain the so-called Partitioning Graph. One example can be found in Fig. 3.2, which represents the $\mathcal{E}$-connection of the toy ontology Koala[2] that contains 42 axioms.

To obtain a logical module in $\mathcal{E}$-connections, after selecting the parts of interest one must import all the parts whose vocabulary is imported. In other words, the smallest module containing the selection can be obtained by computing the transitive closure over the outgoing edges from the red parts. In particular, for Koala we have 8 non-empty modules: one for each blue fragment `Gender`, `Habitat`, and `Degree`, then any combination of these three, and finally the module for `Animal` that imports the vocabulary from the other parts. The

[2] `http://protege.stanford.edu/plugins/owl/owl-library/koala.owl`

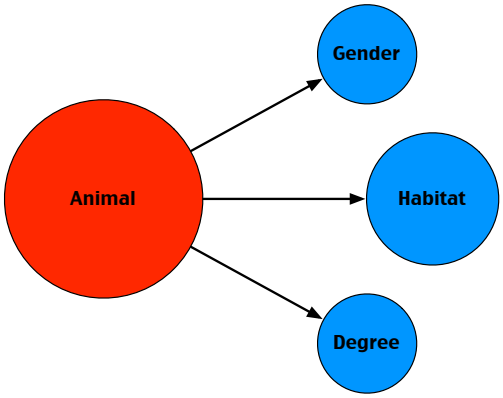


Figure 3.2: $\mathcal{E}$-connections-based Partitioning Graph of the ontology Koala

representative set of modules consists clearly of the 4 fragments labelled with `Animal`, `Gender`, `Habitat`, and `Degree`.

As mentioned before, this technique fails sometimes in partitioning an ontology, and it returns a unique block, even if the ontology seems in principle well structured.

Example 3.3.13. Let us consider again the family of ontologies Zigzag_n as in Example 3.3.5. Then, for each $i \in \{1, \ldots, n\}$ we have that $\mathtt{A}_i$ and $\mathtt{B}_i$ belong to the same module since $\mathsf{Zigzag}_n \models \mathtt{A}_i \sqsubseteq \mathtt{B}_i$. The same argument can be expressed for $\mathtt{A}_{i+1}$ and $\mathtt{B}_i$. Finally, all terms end up in the same module whose label is $\top$.

Similar unexpected aggregation can occur also in cases where the relation between concepts is different from "$\sqsubseteq$".

Example 3.3.14. Let us consider the ontology Tree as in Example 3.3.4. Then, we see that the first axiom $\mathtt{Tree} \sqsubseteq \exists \mathtt{isMadeFrom}.\mathtt{Seed}$ connects the two concepts `Tree` and `Seed`. Similarly, the second axiom connects `Seed` and `Fruit`, and so on, until the fifth axiom connects `Branch` back to `Tree`. Since these terms are connected to one another, they end up in the same partition. In contrast, the terms `Mountain` and `Soil` are connected to each other, but not related to the first partition. The Partitioning Graph of Tree using $\mathcal{E}$-connections is represented in Figure 3.3.

A more notable example of an ontology where the Partitioning Graph has a coarse-grained structure is the ontology Periodic[3] that describes the *distinct* elements of the periodic table, and their properties. Despite the domain being

[3] `www.cs.man.ac.uk/~stevensr/ontology/periodic_full_06012009.owl`

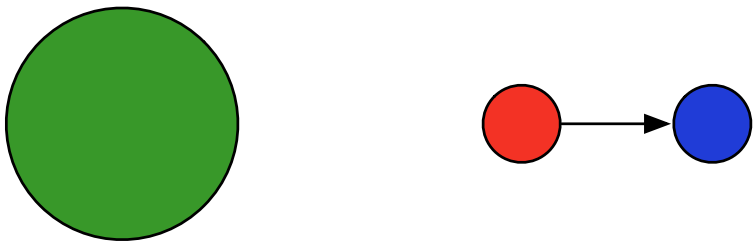


Figure 3.3: $\mathcal{E}$-connections-based Partitioning Graph of the ontology Tree

naturally partitioned, and the ontology expressing this separation, all concepts end up in the same partition, and the Partitioning Graph consists of one green circle.

To sum up, the modular structure based on $\mathcal{E}$-connections can be evaluated as follows:
- Coherence: from very tight to very loose, depending on how the elements of the domain are interrelated: sometimes, as in Koala, the decomposition is finer than in Parikh, sometimes instead it is as coarse as the Parikh's one, while other approaches are able to identify a finer structure, as it is the case for the Δ-decomposition of Zigzag_n;
- Dependence relations: when $\mathcal{E}$-connections succeeds, the dependence relations between the modules can reflect intuitive dependencies between elements of the domain, and the Partitioning Graph provides the user with interesting information for comparing the intended modelling of a domain with the actual modelling of the ontology;
- Granularity: when the ontology does not interrelate the elements of the domain too much, the granularity of the modular structure based on $\mathcal{E}$-connections is able to reveal the differences in the specification of the domain.
- Computability: worst-case quadratic in the size of the input ontology.

3.3.4 Σ-modules

As already mentioned, all notions of Σ-modules have not been designed with a notion of representative set of modules in mind, and hence with a notion of modular structure. However, a family of Σ-modules is simply a set of sets, so we can consider the whole family of modules defined by fixing a notion x of Σ-module, equipped with the natural structure induced by the "$\subseteq$" relation between sets. In particular, we can analyse the modular structure induced by a notion x

of a Σ-module by disregarding the specific notion x since the modular structures induced have a similar behaviour, and can be all treated at once.

Notation 3.3.15. Given an ontology $\mathcal{O}$ and a notion x of Σ-module, we denote by $\mathcal{F}^x(\mathcal{O})$ the *set of all x-modules* in $\mathcal{O}$. When the notion x of module is clear, or irrelevant, we use the notation $\mathcal{F}(\mathcal{O})$.

Let us now evaluate the modular structure $(\mathcal{F}(\mathcal{O}), \subseteq)$ by following the framework defined in Section 3.2 and applied to the modular structures described so far. The following example shows that a module $\mathcal{M} \in \mathcal{F}(\mathcal{O})$ can be extremely incoherent.

Example 3.3.16. Let us consider the ontology All as defined in Example 3.3.2. Recall that All consists of n axioms whose signatures are pairwise disjoint. Now, for each subset S of All, there is a signature Σ (namely $\widetilde{S}$) such that $\mathrm{mod}(\Sigma, \mathsf{All}) = S$. In particular, every subset S of All is a module. However, we can easily see that any two disjoint modules $\mathcal{M}_1, \mathcal{M}_2 \subseteq \mathsf{All}$ do not interact at all since, given any two models $\mathcal{I}_1 \models \mathcal{M}_1$ and $\mathcal{I}_2 \models \mathcal{M}_2$ defined over the same domain Δ, we have that the sum interpretation function $\mathcal{I}_1 + \mathcal{I}_2$ is a model for $\mathcal{M}_1 \cup \mathcal{M}_2$.

The set-inclusion relation $\subseteq$ induces a natural structure over the set of all modules $\mathcal{F}(\mathcal{O})$ of an ontology $\mathcal{O}$, and this structure can be represented by the complete join-semilattice defined by considering as 1-element the whole ontology $\mathcal{O}$, and as the join of any two modules $\mathcal{M}_1$ and $\mathcal{M}_2$ the smallest module containing both $\mathcal{M}_1$ and $\mathcal{M}_2$, well-defined as discussed in Section 2.3. Please note that this structure also has a 0-element, defined to be the module $\mathcal{M}_0 = \mathrm{mod}(\emptyset, \mathcal{O})$. However, this modular structure is not able to reveal interesting logical interactions between modules, as shown in the following example.

Example 3.3.17. Let us consider the ontology $\mathsf{Diamond} = \{\alpha_1 : \mathtt{A}_1 \sqsubseteq \mathtt{A}_2, \alpha_2 : \mathtt{A}_2 \sqsubseteq \mathtt{A}_3, \beta : \mathtt{B}_1 \sqsubseteq \mathtt{B}_2\}$ and let $x = \top\bot^*$. Then, each subset of $\mathsf{Diamond}$ is a module, hence the induced modular structure can be represented as in Figure 3.4. Clearly, the modules $\mathcal{M}_1 = \{\alpha_1, \alpha_2\}$ and $\mathcal{M}_2 = \{\alpha_1, \beta\}$ are different since the set of the entailments deriving from $\mathcal{M}_1$ is strictly larger than the union of the entailments deriving from $\mathcal{M}_1$'s components $\{\alpha_1\}$ and $\{\alpha_2\}$, whilst there is no entailment deriving from $\mathcal{M}_2$ that was not entailed either by $\{\alpha_1\}$ or by $\{\beta\}$. However, this difference is not reflected in the representation of the modular structure $(\mathcal{F}(\mathcal{O}), \subseteq)$. Hence, the granularity of the induced modular structure does not capture in general the logical differences between modules.

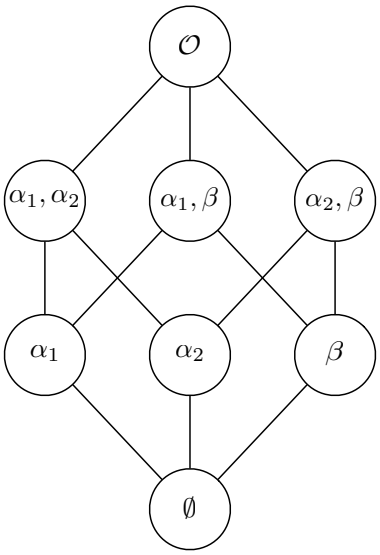


Figure 3.4: Induced modular structure of the Diamond ontology

Finally, the computability of the set $\mathcal{F}(\mathcal{O})$ of all modules of an ontology $\mathcal{O}$ is in general an unfeasible task since the number of modules of an $\mathcal{O}$ is potentially exponential in the size of the $\mathcal{O}$, as we have seen in Example 3.3.16: All has 2^n modules, where n is the number of axioms in All. All is a logically disconnected ontology, and Parsia & Schneider in [PS10] have wondered whether the exponential blowup of the number of modules is a feature of this kind of ontologies. Hence, they have tried to compute $\mathcal{F}(\mathcal{O})$ for several existing ontologies that they evaluate as well designed, i.e. covering a specific domain to a certain level of detail, and sufficiently diverse, i.e. different in sizes, expressivities, ratios of axiom and term numbers, and cover different domains. Unfortunately, despite the optimizations, they have succeeded in fully modularising only 2 small ontologies, i.e. Koala, containing 42 axioms and generating $3,660$ $\top\bot^*$-modules, and Mereology, containing 44 axioms and generating $1,952$ $\top\bot^*$-modules. So, they hypothesize that the number of modules of an ontology is in general exponential in its size. They evaluate this hypothesis against the experimental results obtained by extracting all the submodules from a family of size-increasing modules, and the trendlines they obtain suggest that the numbers of modules of the ontologies considered show indeed an exponential behaviour.

To sum up, the modular structure $(\mathcal{F}(\mathcal{O})^x, \subseteq)$ induced by a given notion x of Σ-module can be evaluated as follows:
- Coherence: very loose.

- Dependence relations: reflect faithfully the interrelations between modules.
- Granularity: unhelpful, unable to reveal logical differences between modules.
- Computability: unfeasible.

Chapter 4

The Atomic Decomposition of an Ontology

In Chapter 3 we have surveyed different notions of logical modules. Some of these notions induce an interesting modular structure, in particular with a well-definable set of representative modules. For Σ-modules, instead, we have discussed how the respective sets of all modules contain "uninteresting" modules. In this chapter we are going to introduce the notion of *genuine module*, i.e., a Σ-module that cannot be decomposed as the union of two or more incomparable modules w.r.t. set inclusion. As a consequence of this definition, we have that every module can be obtained as the union of suitable genuine modules. In this sense, genuine modules are representative of all *general* (as opposed to genuine) Σ-modules. The set of all genuine modules of an ontology $\mathcal{O}$ will be denoted by $\mathfrak{G}(\mathcal{O})$.

From now on, we will deal only with Σ-modules. Hence, for simplicity we will omit Σ, and say modules to mean Σ-modules, unless we want to highlight that we refer specifically to this notion, rather than to a logical module of another kind.

The definition of genuine module suggests that for finding the set $\mathfrak{G}(\mathcal{O})$ we could go through the whole set $\mathcal{F}(\mathcal{O})$ of the modules of $\mathcal{O}$, compare the modules pairwise, and discard those modules that can be decomposed. However, due the the possible exponential number of general modules (which seems to be the common case), discussed in Subsection 3.3.4, the naïve approach to finding genuine modules is infeasible. Hence, other methods to identify genuine modules need to be devised.

The identification of genuine modules out of all modules starts with an abstract investigation on the features of the set $\mathcal{F}(\mathcal{O})$ of all modules of $\mathcal{O}$. Given $\mathcal{F}(\mathcal{O})$, we define an *atom* to be a maximal set of axioms such that, for each module $\mathcal{M} \in \mathcal{F}(\mathcal{O})$, either they all occur in $\mathcal{M}$, or none of them does. Since the relation of co-occurring in a module is an equivalence relation, atoms are disjoint, and their union is the whole ontology $\mathcal{O}$. Hence, atoms form a partition of $\mathcal{O}$. In particular, there are at most as many atoms as axioms. We denote by $\mathcal{A}(\mathcal{O})$ the set of all the atoms of an ontology.

The set of all the modules also induces a dependency relation between atoms: we say that an atom $\mathfrak{a}$ is dependent on an atom $\mathfrak{b}$ if all modules containing $\mathfrak{a}$ contain also $\mathfrak{b}$. The set of atoms together with the dependency relation is called the *Atomic Decomposition* (AD) of an ontology. We prove that this relation, denoted $\succ$, is a partial ordering, so the AD is a poset (recall Definition 2.4.2), and can be represented by a Hasse diagram.

In Section 4.3 we investigate the AD of an ontology as the modular structure induced by a notion of Σ-modules: from the properties of Σ-modules it follows that there is a 1-1 correspondence between atoms and genuine modules, and each genuine module can be obtained from an atom $\mathfrak{a}$ by identifying the set of axioms in the transitive closure of $\succ$ from $\mathfrak{a}$. Moreover, we study the semantic properties of ADs by defining the notion of a *chain of conservative extensions*, which provides a meaning to the dependence relation $\succ$ defined over the set of atoms in terms of logical difference between two related atoms.

The computation of the AD of an ontology $\mathcal{O}$ does not need to go through the extraction of all the modules of $\mathcal{O}$, approach that would be unfeasible: we prove that the genuine modules are indeed *α-modules* of $\mathcal{O}$, that is, the modules $\mathcal{M}$ for which there exists an axiom $\alpha \in \mathcal{O}$ such that $\mathcal{M} = \mathrm{mod}(\widetilde{\alpha}, \mathcal{O})$. Hence, we only need to perform as many module extractions as axioms in $\mathcal{O}$ to obtain $\mathcal{A}(\mathcal{O})$.

The low number of modules extractions required to obtain the set $\mathfrak{G}(\mathcal{O})$ of the genuine modules of an ontology $\mathcal{O}$ suggests that by using modules' notions whose extraction is feasible, the computation of the AD of an ontology is feasible as well. Recall that modules based on syntactic locality can be extracted in polynomial time in the size s of $\mathcal{O}$. Then, extracting an x-AD, with $x \in \{\bot, \top, \top\bot^*\}$, is also polynomial in s. Furthermore, the optimized implementation devised by Tsarkov and described in [Tsa12] allows us to extract the ADs from a large corpus of ontologies, described in Appendix A: in almost all cases the ADs can be obtained

in a few seconds on a standard machine. In Section 4.4 we will analyse the results of this experiment, discuss on similarities/differences between the different notions of ADs, and comment on the decomposability of the ontologies in our corpus.

The preference of module notions based on syntactic locality is justified by their efficiency. However, as discussed in Section 2.3, these notions "approximate" the other notions of Σ-modules. Hence, by looking at $\bot$, $\top$, and $\top\bot^*$-ADs, we have a first insight on the ADs of the other notions. In particular, an empirical comparison between semantic vs. syntactic modules has been carried out in [DKP+12], and the experimental analysis shows that the syntactic notions in general approximate quite well the semantic notions. A comparison of locality-based modules with the other notions of Σ-modules is included in our future work.

Finally, we apply the framework defined in Section 3.2 to evaluate the AD of an ontology as a modular structure, and discuss on general coherence, dependence structure, and granularity of the AD of an ontology.

4.1 Genuine Modules

A first attempt at distinguishing a set of "interesting" Σ-modules is carried out in [PS10], with the aim of identifying the modular structure that these modules induce. The authors notice that some modules can be split into two or more signature-disjoint modules, so they define a *genuine* module*, as opposed to a *fake* module*, to be a module that cannot be decomposed into the union of signature-disjoint modules.[1] In other words, they try to get rid of the uninteresting modules as those in the ontology All described in Example 3.3.16. In this case, then, their approach generates a partition of the ontology All, and this corresponds to the intuition that genuine* modules are the building blocks of an ontology.

In general, however, this notion of genuinity fails at identifying a partition of the ontology. Let us consider the $\top\bot^*$ notion of module, and an ontology Zigzag_n for $n \geq 2$. Then, we can represent the modules of Zigzag_n as we do for

[1] We will see later that these notions do not capture the intuition of being the basic elements for the modules of an ontology, and have been replaced by the notion of genuine and fake modules defined in [DPSS11a]. To avoid confusion, then, we distinguish the old notions by using the $*$ symbol as a superscript.

the ontology in Figure 3.1, so that each module is a union of segments from the representation of Zigzag_n. Then, the genuine* modules are all and only those that are represented by a connected component. Hence, each genuine* module has a "starting point" and an "ending point" that are symbols from the signature of Zigzag_n, and contains all the terms "in between", as represented in Figure 4.1. In particular, the whole ontology is a module, as well as each single axiom is. So, on the one end these modules are not signature disjoint, on the other hand $\mathcal{O}$ is the union of disjoint modules. Hence, some modules are uninteresting since they can be looked at one chunk at a time.

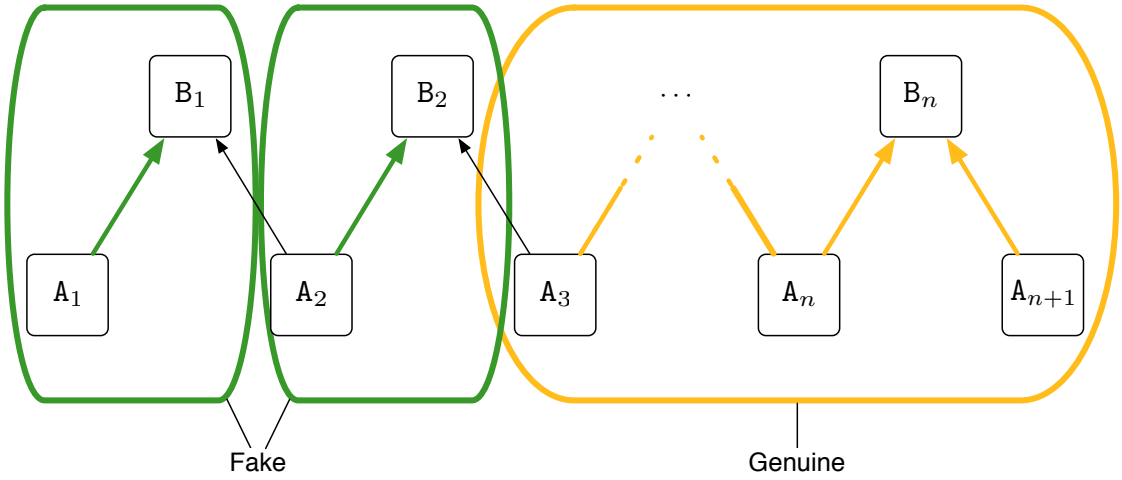


Figure 4.1: Genuine vs. Fake modules in Zigzag_n

The ontology Zigzag_n contains $2n$ axioms and $2n(n+1)$ genuine* modules, and we know that there are genuine* modules that we would not want to include within the set of representative modules. We will use Zigzag_n as the baseline for a simple analysis on the number of genuine* modules: set $\mathfrak{G}(\mathcal{O})$ to be the set of genuine* modules of an ontology $\mathcal{O}$, we can *prima facie* suppose that if $\#\mathfrak{G}(\mathcal{O}) > \#\mathfrak{G}(\mathsf{Zigzag}_n)$, then the set $\mathfrak{G}(\mathcal{O})$ needs to be further refined as $\mathfrak{G}(\mathsf{Zigzag}_n)$ does. We make this rough assumption even though we do not want to compare the structure of $\mathcal{O}$ with the structure of Zigzag_n. Unfortunately, we can apply this analysis only to Koala, containing 42 axioms, and Mereology, containing 44 axioms, since these are the only ontologies for which we have a complete modularisation, as discussed in Subsection 3.3.4. We compare them to Zigzag_{21} containing 42 axioms, and obtain that $\#\mathfrak{G}(\mathsf{Zigzag}_{21}) = 924$, $\#\mathfrak{G}(\mathsf{Koala}) = 2,143$, and $\#\mathfrak{G}(\mathsf{Mereology}) = 252$. Whilst Mereology seems to have a "low enough" number of genuine* modules, $\#\mathfrak{G}(\mathsf{Koala}) \gg \#\mathfrak{G}(\mathsf{Zigzag}_n)$. Hence, we expect a high

number of genuine* modules in Koala to be uninteresting. This analysis suggests that a different notion of genuine module needs to be devised.

As a simple example, let us consider the Person ontology defined as follows:

$$\{\alpha_1 : \texttt{Vegetarian} \sqsubseteq \texttt{Person},$$
$$\alpha_2 : \texttt{Student} \sqsubseteq \texttt{Person},$$
$$\alpha_3 : \texttt{Person} \sqsubseteq \texttt{Animal}\}$$

Since all axioms share the concept `Person` it is clear that there is only one genuine* module, and it coincides with the whole ontology. However, the two concepts `Vegetarian` and `Student` can clearly be independently interpreted—whilst it is evident that the interpretations for both are dependent on the concept `Person`. Intuitively we want to separate α_1 from α_2. In Software Engineering, it is common to have distinct modules with common dependencies. In the same spirit, we discard modules that can be decomposed into $\subseteq$-incomparable modules to obtain the following new definition for the notion of genuine module.

Definition 4.1.1. Given an ontology $\mathcal{O}$ and the set $\mathcal{F}(\mathcal{O})$ of all its modules, we define a *fake module* to be a module $\mathcal{M} \subseteq \mathcal{O}$ such that there are (at least) two modules $\mathcal{M}_1$ and $\mathcal{M}_2$ of $\mathcal{O}$, with neither $\mathcal{M}_1 \subseteq \mathcal{M}_2$ nor $\mathcal{M}_2 \subseteq \mathcal{M}_1$, and $\mathcal{M} = \mathcal{M}_1 \cup \mathcal{M}_2$. A module is called *genuine module* if it is not fake.

In principle, we could find all genuine modules by going through all the modules of $\mathcal{O}$, but we know that computing $\mathcal{F}(\mathcal{O})$ is in general an infeasible task. The following discussion will lead to a characterization of the set of genuine modules in terms of a feasibly computable set of modules.

4.2 Atoms and their Dependence Relation

By analysing the results of the experiments implemented and described in [PS10], we have noticed, and reported in [DPSS10], that the co-occurrence of axioms in a module is not arbitrary. In particular, some axioms occur only if some others do. We formalize these observations as follows.

Definition 4.2.1. Let x be a notion of Σ-module, and let $\mathcal{F}^x(\mathcal{O})$ be the set of all x-modules of an ontology $\mathcal{O}$. The *x-relation* $\approx_x$ is the binary relation over $\mathcal{O}$

defined to hold between two axioms $\alpha, \beta \in \mathcal{O}$ if, for all $\mathcal{M}^x \in \mathcal{F}^x(\mathcal{O})$, $\alpha \in \mathcal{M}^x$ if, and only if, $\beta \in \mathcal{M}^x$.

As usual, if x is clear from the context, or irrelevant, we simply drop it, and denote the corresponding relation by $\approx$.

Proposition 4.2.2. *The relation $\approx$ as in Definition 4.2.1 is an equivalence relation.*

Proof. The property of of two elements to co-occur in a set of sets is trivially reflexive, transitive, and symmetric. □

Definition 4.2.3. Let $\mathcal{O}$ be an ontology, $\mathcal{F}(\mathcal{O})$ the set of all modules of $\mathcal{O}$, and $\approx$ the relation between axioms as in Definition 4.2.1. We define an *atom* $\mathfrak{a}$ of $\mathcal{O}$ to be an equivalence class $[\alpha]_\approx$ for an axiom $\alpha \in \mathcal{O}$. The *set of atoms* of $\mathcal{O}$ is denoted by $\mathcal{A}(\mathcal{O})$, or $\mathcal{A}^x(\mathcal{O})$ if we want to emphasize the module notion x.

Remark 4.2.4. Some axioms will not appear in any module. These axioms are tautologies for all the semantic notions of Σ-modules, and syntactic tautologies (recall Definition 2.3.17) for modules based on syntactic locality. In all cases, having fixed a notion of Σ-modules, the equivalence class for any such axiom τ contains all tautologies, and the corresponding atom is denoted by $\mathfrak{t}$.

Corollary 4.2.5. *Let $\mathcal{O}$ be an ontology. Then:*
(1) *$\mathcal{A}(\mathcal{O})$ is a partition of $\mathcal{O}$.*
(2) *Each atom $\mathfrak{a} \in \mathcal{A}(\mathcal{O})$ is a finite set of axioms.*
(3) *$\#\mathcal{A}(\mathcal{O}) \leq \#\mathcal{O}$.*

Proof. It is an immediate consequence of Proposition 4.2.2 and of the finiteness of $\mathcal{O}$. □

Next, we define a binary relation $\succeq$ over the set $\mathcal{A}(\mathcal{O})$ of atoms of an ontology $\mathcal{O}$ to capture the occurrence of some axioms in a module only if some other axioms occur, as mentioned at the beginning of this subsection. As the notation suggests, this relation is a partial ordering.

Definition 4.2.6. Let x be a notion of Σ-module, and let $\mathfrak{a}$ and $\mathfrak{b}$ be two atoms induced by $\approx_x$ over an ontology $\mathcal{O}$. Then, $\mathfrak{a}$ is said to be *x-dependent* on $\mathfrak{b}$ (written $\mathfrak{a} \succeq_x \mathfrak{b}$ or, dually, $\mathfrak{b} \preceq_x \mathfrak{a}$) if, for every x-module $\mathcal{M} \in \mathcal{F}^x(\mathcal{O})$ such that $\mathfrak{a} \subseteq \mathcal{M}$, we have $\mathfrak{b} \subseteq \mathcal{M}$.

As usual, we omit the notion x of module if clear from the context, or irrelevant. In this case the relation $\succeq_x$ is simply called "dependence" and denoted by $\succeq$.

Proposition 4.2.7. *The binary relation $\succeq$ as in Definition 4.2.6 is a partial ordering over the set $\mathcal{A}(\mathcal{O})$ of atoms of an ontology $\mathcal{O}$.*

Proof. $\succeq$ satisfies the following 3 properties:

Reflexivity: trivial.

Antisymmetry: let us consider two atoms $\mathfrak{a}$ and $\mathfrak{b}$ such that: (1) $\mathfrak{a} \succeq \mathfrak{b}$; (2) $\mathfrak{b} \succeq \mathfrak{a}$. We want to prove that in this case $\mathfrak{a}$ and $\mathfrak{b}$ coincide. Equivalently, we can prove without loss of generality that there is no module $\mathcal{M}$ such that $\mathcal{M}$ contains $\mathfrak{a}$ but not $\mathfrak{b}$. By contraposition, let $\mathcal{M}$ be such a module. Then, by the definition of $\succeq$ we have that $\mathfrak{a} \not\succeq \mathfrak{b}$, and this contradicts the hypothesis (2).

Transitivity: let $\mathfrak{a} \succeq \mathfrak{b}$ and $\mathfrak{b} \succeq \mathfrak{c}$; that is, every module containing $\mathfrak{a}$ contains also $\mathfrak{b}$; but since such a module contains $\mathfrak{b}$, then it also contains $\mathfrak{c}$. Hence, $\mathfrak{a}$ is dependent on $\mathfrak{c}$.

Hence, $\succeq$ is a partial ordering. □

From now on we will consider the strict poset $(\mathcal{A}(\mathcal{O}), \succ)$, defined by removing all pairs $(\mathfrak{a}, \mathfrak{a})$ for any $\mathfrak{a} \in \mathcal{A}(\mathcal{O})$ from the graph of $\succeq$. Hence, the dependence relation between atoms can be represented by means of a Hasse diagram.

Definition 4.2.8. Given an ontology $\mathcal{O}$, we call *Atomic Decomposition* (AD) the pair $(\mathcal{A}(\mathcal{O}), \succ)$.

In the cases where specifying the notion x of Σ-module is important, we will explicitly mention x, and speak of the *x-Atomic Decomposition* (x-AD) of an ontology $\mathcal{O}$ as the partially ordered set $(\mathcal{A}^x(\mathcal{O}), \succ_x)$.

We will see in Subsection 4.4.1 how to efficiently compute the AD of an ontology. Before proceeding further we provide the reader with some examples of ontologies and their ADs.

Example 4.2.9. Let us consider the Dog ontology defined as follows:

$$\{\alpha_1 : \texttt{Poodle} \sqsubseteq \texttt{Dog} \sqcap \exists \texttt{hasCoat.Curly},$$
$$\alpha_2 : \texttt{Dog} \sqsubseteq \texttt{Mammal} \sqcap \exists \texttt{hasPart.Tail},$$
$$\alpha_3 : \texttt{Mammal} \sqsubseteq \texttt{Animal} \sqcap \exists \texttt{givesBirth.Life}\}.$$

For each notion $x \in \{\bot, \top, \top\bot^*\}$ we give the set of all the x-modules:

$$\mathcal{F}(\mathsf{Dog})^{\bot} = \{\emptyset, \{\alpha_3\}, \{\alpha_2, \alpha_3\}, \mathsf{Dog}\}.$$
$$\mathcal{F}(\mathsf{Dog})^{\top} = \{\emptyset,\ \{\alpha_1\},\ \{\alpha_1, \alpha_2\},\ \mathsf{Dog}\}.$$
$$\mathcal{F}(\mathsf{Dog})^{\top\bot^*} = \{\emptyset, \{\alpha_1\}, \{\alpha_2\}, \{\alpha_3\}, \{\alpha_1, \alpha_2\}, \{\alpha_2, \alpha_3\}, \mathsf{Dog}\}.$$

Then, the corresponding x-ADs can be represented by the the Hasse diagrams as in Figure 4.2.

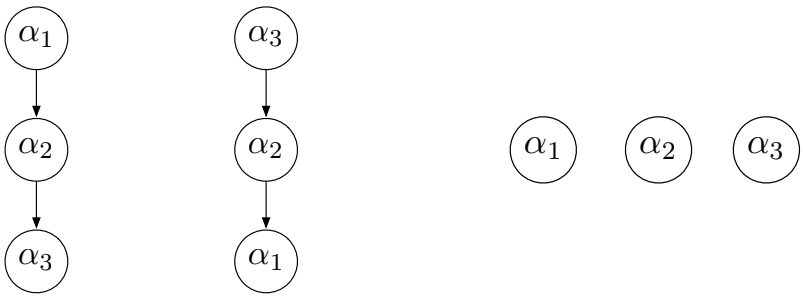


Figure 4.2: $\bot$-, $\top$-, and $\top\bot^*$-ADs of the Dog ontology

We can see that both in the $\bot$-AD and in the $\top$-AD all the atoms are comparable. This is in contrast with the $\top\bot^*$-AD, which is totally disconnected since each atom is incomparable with any other atom. However, at a closer inspection we can see that the set $\{\alpha_1, \alpha_3\}$ is not a $\top\bot^*$-module, so that the smallest module containing both α_1 and α_3 is the whole ontology. Intuitively, we can see that the $\bot$-AD shows the dependence of axioms dealing with more specific concepts (i.e., `Poodle`) from the axioms dealing with more general concepts (i.e., `Mammal`). Dually, the $\top$-AD shows the dependence of axioms dealing with more general concepts from the axioms dealing with more specific concepts. The $\top\bot^*$-AD, instead, shows the independence of some axioms even from those that deal with more specific or general concepts. To sum up, we can say that the $\bot$- and the $\top$-AD reveal a subsumption-preserving kind of dependence, whilst the $\top\bot^*$-AD shows a finer notion of independence than those revealed by the other kinds of ADs. We will further investigate this phenomenon in Subsection 4.3.2.

In Figure 4.3 we show the $\bot$-AD of the ontology Koala, and we see that there are 19 atoms, and 1 connected component. To better appreciate how fine-grained the $\bot$-AD is, please compare this figure with the $\mathcal{E}$-connections partitioning graph for Koala as represented in Figure 3.2, which consists of 4 parts and 1 connected component.

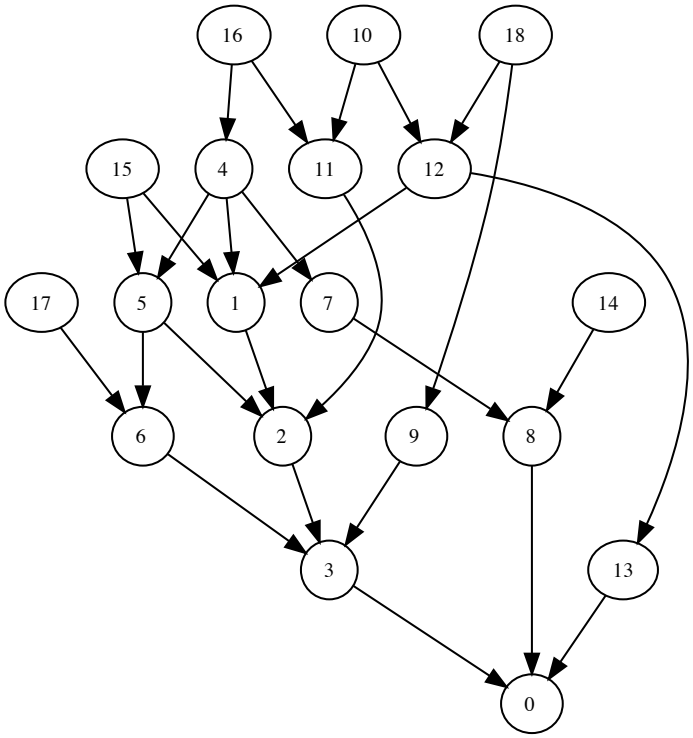


Figure 4.3: $\bot$-AD of the ontology Koala

4.3 The AD as a Modular Structure

In the previous section we have defined the Atomic Decomposition of an ontology $\mathcal{O}$ as the pair $(\mathcal{A}(\mathcal{O}), \succ)$, where $\mathcal{A}(\mathcal{O})$ is the set of the atoms induced by the modules of $\mathcal{O}$, and $\succ$ is a partial order between the atoms. In this section we will investigate the AD of $\mathcal{O}$ as the modular structure induced over $\mathcal{O}$ by the Σ-modules.

The first aspect, analysed in Subsection 4.3.1, consists of two consequences over ADs that the desirable properties of self-containment, depletion, and monotonicity as described in Definition 2.3.1, have:

1. for each atom $\mathfrak{a} \in \mathcal{A}(\mathcal{O})$, the principal ideal $\downarrow \mathfrak{a} = \mathfrak{a} \cup \bigcup_{\mathfrak{b} | \mathfrak{a} \succ \mathfrak{b}} \mathfrak{b}$ in $(\mathcal{A}(\mathcal{O}), \succ)$ is a genuine module of $\mathcal{O}$;

2. there is a 1-1 correspondence between the genuine modules $\mathfrak{G}(\mathcal{O})$ of $\mathcal{O}$ and the set of atoms $\mathcal{A}(\mathcal{O})$.

As a consequence, the AD of an ontology is a succint representation of the set of genuine modules $\mathfrak{G}(\mathcal{O})$.

The semantic properties of ADs are the second aspect to be analysed in Subsection 4.3.2. These properties are captured by the notion of *chain of Conservative Extensions*, that provides a meaning to the dependence relation $\succ$ defined over the set of atoms in terms of logical difference between two related atoms.

4.3.1 Atoms vs. Genuine Modules

Throughout this subsection, the properties of self-containment, depletion, and monotonicity (see Definition 2.2.4 and Definition 2.2.3) play a crucial role: they make the AD of an ontology $\mathcal{O}$ to be a succint representation of the genuine Σ-modules of $\mathcal{O}$.

Lemma 4.3.1. *Let x be a notion of module that is monotonic, self-contained, and depleting. Then, for every atom $\mathfrak{a}$ in $\mathcal{A}(\mathcal{O}) \setminus \{\mathfrak{t}\}$, and every axiom $\alpha \in \mathfrak{a}$ we have that x-mod$(\widetilde{\alpha}, \mathcal{O})$ is the smallest module containing $\mathfrak{a}$.*

Proof. The axiom $\alpha \in \mathfrak{a}$ is clearly a non tautological axiom (recall Remark 4.2.4). Let us consider the module $\mathcal{M}_\alpha := x\text{-mod}(\widetilde{\alpha}, \mathcal{O})$. Then:
(1) $\mathcal{M}_\alpha$ is not empty since it contains α.
(2) $\mathcal{M}_\alpha \supseteq \mathfrak{a}$, by the definition of atoms.
As discussed in Section 2.3, $\mathcal{M}_\alpha$ is the unique, and thus smallest, module for the seed signature $\widetilde{\alpha}$. Let now $\mathcal{M}'$ be a module containing α, and hence containing $\mathfrak{a}$ as well by the definition of atoms. Then, by self-containment we have that $\mathcal{M}' = x\text{-mod}(\widetilde{\mathcal{M}'}, \mathcal{O}) = x\text{-mod}(\widetilde{\mathcal{M}'} \cup \widetilde{\alpha}, \mathcal{O})$. Moreover, by monotonicity we have that any module obtained by enlarging the seed signature $\widetilde{\alpha}$ is either equal to, or a superset of $\mathcal{M}_\alpha$, hence $\mathcal{M}' \supseteq \mathcal{M}_\alpha$. Finally, by the arbitrary choice of $\alpha \in \mathfrak{a}$ we have that $\mathcal{M}_\alpha$ is the smallest module containing $\mathfrak{a}$ for each choice of such an axiom α. □

Corollary 4.3.2. *Given an atom $\mathfrak{a}$, we have that x-mod$(\widetilde{\mathfrak{a}}, \mathcal{O}) = \mathcal{M}_\alpha$ for any axiom $\alpha \in \mathfrak{a}$. Moreover, $\mathfrak{a}$ is dependent on all the atoms contained in $\mathcal{M}_\alpha \setminus \mathfrak{a}$.*

In the proof of Lemma 4.3.1, the module $\mathcal{M}_\alpha$ plays a major role. We will see that this kind of axioms is fundamental for finding the representative set of Σ-modules.

Definition 4.3.3. A module $\mathcal{M} \subseteq \mathcal{O}$ is called *α-module* if, there exists an axiom α such that $\mathcal{M} = \text{mod}(\widetilde{\alpha}, \mathcal{O})$. In this case, $\mathcal{M}$ will be denoted by $\mathcal{M}_\alpha$.

Let α be an axiom and $\mathfrak{a}$ be the module such that $\alpha \in \mathfrak{a}$. By Corollary 4.3.2 we know that $\mathcal{M}_\alpha = x\text{-mod}(\widetilde{\mathfrak{a}}, \mathcal{O})$. Hence, we can equivalently use one these two notions.

Notation 4.3.4. For each atom $\mathfrak{a} \in \mathcal{A}(\mathcal{O})$, the module $\mathcal{M}_\mathfrak{a} := x\text{-mod}(\widetilde{\mathfrak{a}}, \mathcal{O})$ is an α-module, and vice versa. Depending on the context, then, we could use the notation $\mathcal{M}_\mathfrak{a}$ to highlight that the module extraction is obtained by considering the atom's signature.

Corollary 4.3.5. *$\mathfrak{a}$ is an atom, and α is an axiom in $\mathfrak{a}$ $\iff$ $\mathcal{M}_\alpha = \mathcal{M}_\mathfrak{a}$.*

From the set of all modules $\mathcal{F}(\mathcal{O})$ of an ontology $\mathcal{O}$, we can then determine the AD $(\mathcal{A}(\mathcal{O}), \succ)$ of the ontology. A natural question now arising is, whether the AD is a representation of the modules of $\mathcal{O}$.

Theorem 4.3.6. *Let $\mathcal{O}$ be an ontology, and let $(\mathcal{A}(\mathcal{O}), \succ)$ be an AD. Then, for each atom $\mathfrak{a} \in \mathcal{A}(\mathcal{O})$, the principal ideal $\downarrow \mathfrak{a}$ is a module.*

Proof. Let $\mathfrak{a}$ be an atom, and let us fix an axiom $\gamma \in \mathfrak{a}$. We want to compare the principal ideal $\downarrow \mathfrak{a} = \{\mathfrak{b} \mid \mathfrak{a} \succeq \mathfrak{b}\}$ with the α-module $\mathcal{M}_\gamma$. By the definition of atoms, $\mathcal{M}_\gamma \supseteq \downarrow \mathfrak{a}$. We still need to prove that the equality holds. By contraposition, let $\mathcal{M}_\gamma$ be a proper superset of $\downarrow \mathfrak{a}$. Then it contains at least one atom $\mathfrak{b}$ which $\mathfrak{a}$ is not dependent on. Let β be an axiom in $\mathfrak{b}$, and let us consider $\mathcal{M}_\beta$. By Lemma 4.3.1, $\mathcal{M}_\beta$ is the smallest module containing $\mathfrak{b}$. Then, $\mathcal{M}_\beta$ is contained in $\mathcal{M}_\gamma$, and since the latter is the smallest module containing $\mathfrak{a}$, this means that $\mathfrak{a}$ is dependent on $\mathfrak{b}$. This last fact contradicts the assumption set by contraposition. □

Corollary 4.3.7. *A module $\mathcal{M}$ is an α-module if, and only if, there is an atom $\mathfrak{a} \subseteq \mathcal{M}$ such that $\mathcal{M} = \downarrow \mathfrak{a}$.*

Corollary 4.3.8. *There is a 1-1 correspondence between α-modules and atoms in $\mathcal{A}(\mathcal{O}) \setminus \mathfrak{t}$.*

So far we have proved that the α-modules of an ontology $\mathcal{O}$ can easily be identified by looking at the AD of $\mathcal{O}$. We show now that *all* modules are represented in the AD, even if their identification is not so straightforward as for α-modules.

Lemma 4.3.9. *A module is a disjoint finite union of atoms.*

Proof. From the construction of atoms as in Definition 4.2.3, we have that for any atom $\mathfrak{a} \in \mathcal{A}(\mathcal{O})$, there does not exist a module $\mathcal{M}$ such that $\mathcal{M} \cap \mathfrak{a} \subsetneqq \mathfrak{a}$. Moreover, atoms are disjoint. Finally, since any ontology $\mathcal{O}$ contains only finitely many axioms, a module $\mathcal{M} \subseteq \mathcal{O}$ can contain only finitely many atoms. □

Proposition 4.3.10. *Every module $\mathcal{M}$ is determined by selecting in the atomic Hasse diagram one suitable antichain $\mathfrak{a}_1, \ldots \mathfrak{a}_k$, $k \in \mathbb{N}$, and by taking the union of principal ideals of these atoms:*

$$\mathcal{M} = \bigcup_{i=1}^{k} \downarrow \mathfrak{a}_i.$$

Proof. From Lemma 4.3.9, we have that every module $\mathcal{M}$ is a disjoint finite union of atoms. Now, if $\mathcal{M}$ contains an atom $\mathfrak{a}$, then it contains also all atoms which $\mathfrak{a}$ is dependent on, that is, the set $\{\mathfrak{b} \mid \mathfrak{a} \succeq \mathfrak{b}\}$; this set corresponds exactly to the principal ideal $\downarrow \mathfrak{a}$. □

In other words, Proposition 4.3.10 says that, each module $\mathcal{M}$ has a unique decomposition into incomparable principal ideals. Hence, the set A of α-modules forms a *base* for the set $\mathcal{F}(\mathcal{O})$ of all modules of $\mathcal{O}$. In Definition 4.1.1 we have already set the notion of a base of modules by introducing the set $\mathfrak{G}(\mathcal{O})$ of genuine modules. In what follows, we prove that these notions coincide.

Theorem 4.3.11. *The set $\mathfrak{G}(\mathcal{O})$ of genuine modules and the set A of α-modules coincide.*

Proof. We use the characterization of α-modules as principal ideals in $(\mathcal{A}(\mathcal{O}), \succ)$ stated in Corollary 4.3.7. Hence, we have to prove that, if a module is genuine, then it is the principal ideal $\downarrow \mathfrak{a}$ of an atom $\mathfrak{a}$, and *vice versa*. We prove both directions by contraposition.

α-module $\Rightarrow$ genuine module: By contraposition, let $\mathcal{M}$ be a fake module. Then there are two incomparable modules $\mathcal{M}_1$ and $\mathcal{M}_2$ such that $\mathcal{M} = \mathcal{M}_1 \cup \mathcal{M}_2$. From Lemma 4.3.9, we have that there exist suitable atoms such that $\mathcal{M}_1 = \mathfrak{a}_1 \cup \ldots \cup \mathfrak{a}_\kappa$ and $\mathcal{M}_2 = \mathfrak{b}_1 \cup \ldots \cup \mathfrak{b}_\ell$; since the modules are incomparable, then there is at least one atom $\mathfrak{a}_k \notin \{\mathfrak{b}_1, \ldots, \mathfrak{b}_\kappa\}$; similarly, there is at least one atom $\mathfrak{b}_l \notin \{\mathfrak{a}_1, \ldots, \mathfrak{a}_\kappa\}$. Moreover, there is no atom $\mathfrak{c} \in \mathcal{M} = \{\mathfrak{a}_1, \ldots, \mathfrak{a}_\kappa, \mathfrak{b}_1, \ldots, \mathfrak{b}_\ell\}$ dependent both on $\mathfrak{a}_k$ and on $\mathfrak{b}_l$, otherwise these atoms would be both in $\mathcal{M}_1$ and in $\mathcal{M}_2$; in particular, $\mathcal{M}$ is not an α-module.

genuine module $\Rightarrow$ α-module: By contraposition, let $\mathcal{M}$ be a module such that there exist atoms $\mathfrak{a}_1, \ldots, \mathfrak{a}_\kappa$ such that $\mathcal{M} = \downarrow \mathfrak{a}_1 \cup \ldots \cup \downarrow \mathfrak{a}_\kappa$, with $\kappa \geq 2$. Without loss of generality, we can assume the atoms $\mathfrak{a}_1, \ldots, \mathfrak{a}_\kappa$ to be pairwise independent. Then, by Theorem 4.3.6 we have that the principal ideal of every atom is a module. Hence $\mathcal{M} = \downarrow \mathfrak{a}_1 \cup \ldots \cup \downarrow \mathfrak{a}_\kappa$ is a union of incomparable modules, and more specifically, fake. □

Corollary 4.3.12. *There is a 1-1 correspondence between atoms and genuine modules.*

We will see in Section 4.4 that a consequence of Theorem 4.3.11 is the feasibility of computing the set $\mathfrak{G}(\mathcal{O})$ of all genuine modules of an ontology for the notions of modules that can be efficiently extracted: it suffices to extract a module for each axiom $\alpha \in \mathcal{O}$, hence it requires a linear number of module extractions in the size of the ontology.

We saw in Proposition 4.3.10 that each module can be uniquely represented in the AD of the ontology as the ideal of a suitable set of atoms. However, not all ideals are modules, as shown by the following example.

Example 4.3.13. Let us consider the ontology Girl defined as follows:

$$\begin{aligned}\{&\alpha_1 : \texttt{Woman} \sqsubseteq \texttt{Person},\\ &\alpha_2 : \texttt{Child} \sqsubseteq \texttt{Person},\\ &\alpha_3 : \texttt{Woman} \sqcap \texttt{Child} \sqsubseteq \texttt{Girl}\}\end{aligned}$$

Then, the $\bot$-AD of Girl is represented in Figure 4.4. Now, the atom $\mathfrak{a}_1 = \{\alpha_1\}$

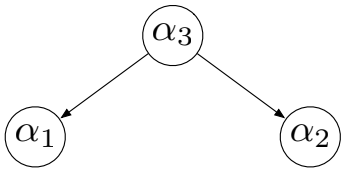


Figure 4.4: $\bot$-AD of the ontology Girl

is a module for the signature $\Sigma_1 = \{\texttt{Woman}, \texttt{Person}\}$, and the atom $\mathfrak{a}_2 = \{\alpha_2\}$ is a module for the signature $\Sigma_2 = \{\texttt{Child}, \texttt{Person}\}$. However, their union is not a module since, whenever we include both the terms $\texttt{Woman}, \texttt{Child}$ in a signature Σ, then also the axiom α_3 is non local, and it is included in the module for Σ.

More in general, not all unions of atoms are modules. This means that recognizing modules in the AD of an ontology is possible only by enumeration, and we know that this task is infeasible. The representation of all the modules of an ontology can be indeed done by enriching the AD with suitable *labels*, as we discussed in [DGK+11]. We will describe the theoretical solution in Section 5.2, and the experimental results carried on its implementation in Section 6.2.

4.3.2 Chains of Conservative Extensions

The interrelations between the genuine modules are better understood in terms of the notions of Conservative Extensions underpinning each notion x of Σ-module. Let $R \in \{\text{mCE, dCE}\}$ be an inseparability relation, and let us consider two ontologies $\mathcal{O}_1 \subsetneqq \mathcal{O}_2$ such that $\mathcal{O}_1 \equiv^R_{\widetilde{\mathcal{O}}_1} \mathcal{O}_2$. When $R = \text{dCE}$ then, by definition, all entailments of $\mathcal{O}_2$ over $\widetilde{\mathcal{O}}_1$ already follow from $\mathcal{O}_1$, so we can restrict our attention to $\mathcal{O}_1$ *without missing the entailments of* $\mathcal{O}_2$ *over* $\widetilde{\mathcal{O}}_1$. Similarly, when $R = \text{mCE}$, we have by its definition that $\{\mathcal{I} \,|\, \mathcal{I} \models \mathcal{O}_1\} = \{\mathcal{J}|_{\widetilde{\mathcal{O}_1}} \,|\, \mathcal{J} \models \mathcal{O}_2\}$. Intuitively, we can think of $\mathcal{O}_2$ as extending $\mathcal{O}_1$ *without spoiling the models* already identified by $\mathcal{O}_1$ over its signature.

Let us consider two comparable atoms $\mathfrak{a}_1 \prec \mathfrak{a}_2 \in \mathcal{A}(\mathcal{O})$ such that there is no atom $\mathfrak{a}_3 \in \mathcal{A}(\mathcal{O})$ for which $\mathfrak{a}_1 \prec \mathfrak{a}_3 \prec \mathfrak{a}_2$ holds, and let us consider the corresponding genuine modules $\mathcal{G}_1 =\!\downarrow \mathfrak{a}_1$ and $\mathcal{G}_2 =\!\downarrow \mathfrak{a}_2$ in $\mathfrak{G}(\mathcal{O})$. From the condition $\mathfrak{a}_1 \prec \mathfrak{a}_2$ we have that $\mathcal{G}_1$ can be safely enlarged with the set of axioms $\mathcal{G}_2 \setminus \mathcal{G}_1$, while the meaning of $\mathcal{G}_1$ over its signature $\widetilde{\mathcal{G}_1}$ is preserved. Moreover, since there is no genuine module $\mathcal{G}_3 \in \mathfrak{G}(\mathcal{O})$ with $\mathcal{G}_1 \subsetneq \mathcal{G}_3 \subsetneq \mathcal{G}_2$ we have that the enlargement with the set $\mathcal{G}_2 \setminus \mathcal{G}_1$ is x-minimal, i.e., that there is no proper subset $\mathfrak{S}$ of $\mathcal{G}_2 \setminus \mathcal{G}_1$ such that $\mathcal{G}_1$ is an x-module for $\mathcal{G}_1 \cup \mathfrak{S}$ over $\widetilde{\mathcal{G}_1}$.

This discussion suggests us the following definition that formalizes safe enlargements of subontologies.

Definition 4.3.14. Let $R \in \{\text{mCE, dCE}\}$ be an inseparability relation, and let $\mathcal{O}$ be an ontology. An *R-chain of CEs* is a finite sequence of subontologies $\mathcal{O}_1 \subsetneq \mathcal{O}_2 \subsetneq \ldots \subsetneq \mathcal{O}_m = \mathcal{O}$ such that, for each $i \in \{1, \ldots, m-1\}$, we have that $\mathcal{O}_i \equiv^R_{\widetilde{\mathcal{O}}_i} \mathcal{O}_{i+1}$.

If R is clear from the context, or irrelevant because the discussion abstracts from the specific notion of R, we simply omit it.

We can *prima facie* suppose that, the longer an identified chain of CEs in $\mathcal{O}$ is, the easier is looking at each subontology $\mathcal{O}_i$ since the number of axioms added at each step will be on average as small as possible. We have already commented on the hardness of deciding if $\mathcal{O}$ is a CE. However, given an AD it is straightforward to find chains of CEs of genuine modules.

Proposition 4.3.15. *Let $(\mathcal{A}(\mathcal{O}), \succ)$ be the AD of an ontology $\mathcal{O}$, and let the atoms $\mathfrak{a}, \mathfrak{b}$ be such that $\mathfrak{a} \succ \mathfrak{b}$. Then, $\downarrow \mathfrak{b} \subsetneq \downarrow \mathfrak{a} \subsetneq \mathcal{O}$ is a chain of CEs.*

Proof. Trivial: $\downarrow \mathfrak{b}$ is a module of $\mathcal{O}$, and it is contained in the module $\downarrow \mathfrak{a}$, hence it also holds that $\downarrow \mathfrak{b} \equiv_{\widetilde{\downarrow \mathfrak{b}}} \downarrow \mathfrak{a}$. □

Corollary 4.3.16. *Given a chain $\mathfrak{a}_1 \succ \ldots \succ \mathfrak{a}_\kappa$ of κ atoms, we have that $\downarrow \mathfrak{a}_\kappa \subsetneq \ldots \subsetneq \downarrow \mathfrak{a}_1 \subsetneq \mathcal{O}$ is a chain of CEs.*

Remark 4.3.17. Please note that the empty set is not necessarily a module of $\mathcal{O}$, hence $\emptyset \subsetneq \mathcal{O}$ is not in general a chain of CEs. Let us recall Example 3.1.4. The global axiom $\top \sqsubseteq \{Adam\} \sqcup \{Eve\}$ is included in all modules since it constrains the domain size, and as a consequence all terms in $\widetilde{\mathcal{O}}$. However, all global axioms belong to the module $\mathcal{M}_0 = \mathrm{mod}(\emptyset, \mathcal{O})$, and $\mathcal{M}_0 \subsetneq \mathcal{O}$ is a chain of CEs whenever $\mathcal{M}_0 \neq \mathcal{O}$.

The aim in defining the chains of CEs is clear: to find a logical preordering between axioms, so that it makes sense to say that one axiom depends on others. In the next examples we try to define such a preordering in the Child' ontology as defined in Example 3.1.4.

Example 4.3.18. In Figure 4.5 we show the $\bot$-AD for Child', where:

$$\begin{aligned}
\mathfrak{a}_0 &= \{\alpha_0 : \top \sqsubseteq \{Adam\} \sqcup \{Eve\}\} \\
\mathfrak{a}_1 &= \{\alpha_1 : \mathrm{Irrefl}(\texttt{hasMother})\} \\
\mathfrak{a}_2 &= \{\alpha_2 : \texttt{Mother} \sqsubseteq \neg\texttt{Father}\} \\
\mathfrak{a}_3 &= \{\alpha_3 : \mathrm{Irrefl}(\texttt{hasFather})\} \\
\mathfrak{a}_4 &= \{\alpha_4 : \texttt{Child} \sqsubseteq (=1\texttt{hasMother.Mother}), \\
&\qquad \alpha_5 : \texttt{Child} \sqsubseteq (=1\texttt{hasFather.Father})\}
\end{aligned}$$

We have highlighted one chain of $\bot$-atoms in $(\mathcal{A}(\mathcal{O}), \succ)$: $\mathfrak{a}_0 \prec \mathfrak{a}_1 \prec \mathfrak{a}_4$. The corresponding chain of CEs is $\{\alpha_0\} \subsetneq \{\alpha_0, \alpha_1\} \subsetneq \mathsf{Child}'$.

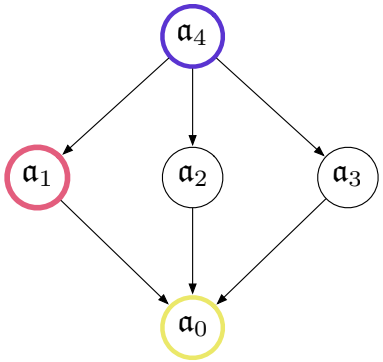


Figure 4.5: A chain of CEs in the Child′ ontology

First of all, please note that all axioms depend on the global axiom $\top \sqsubseteq \{Adam\} \sqcup \{Eve\}$ since this axiom constrains even the domain Δ to be at most of size 2. Another interesting observation concerns the independence of the axioms Irrefl(`hasMother`), Irrefl(`hasFather`), and the axiom $\texttt{Mother} \sqsubseteq \neg\texttt{Father}$: indeed there is no logical relation between these axioms expressed in the ontology, i.e., we can separately interpret the terms in these axioms with no logical consequence on the others. Quite differently, and unsurprisingly, the axioms describing `Child` depend on all the other axioms, so the corresponding module $\downarrow \mathfrak{a}_4$ includes also atoms that are not selected: in particular, this behaviour reveals that the unsatisfiability of the term `Child` derives from a combination of logical interrelations between all the axioms of the ontology.

From the observations just pointed out, we can derive a meaningful reading order for sets of axioms in Child′ as follows: $\{\alpha_0\}, \{\alpha_1\}, \{\alpha_2, \alpha_3, \alpha_4\}$. The notion of chain of CE will be analyzed in Section 6.3 for application in the field of modular reasoning.

4.4 Computation of ADs

Previously in this chapter we have defined the Atomic Decomposition of an ontology $\mathcal{O}$ as the modular structure induced by a given notion of Σ-module. So far, though, we have not discussed the feasibility of computing an AD.

However, the results described in the previous section suggest that computing an AD is indeed feasible: firstly, please note that we can compute the AD of an ontology $\mathcal{O}$ by pairwise comparing genuine modules rather than all the modules, and obtain the same set of atoms and the same dependencies since any module is

the union of genuine modules. Moreover, a hint of the complexity for computing an AD can be derived from Theorem 4.3.11: the identification of genuine modules requires only the extraction of one module for each axiom of $\mathcal{O}$, which is clearly linear in the size of $\mathcal{O}$.

In this section we will determine the complexity of computing an AD given a notion of Σ-module, and we will enter more in the specific of computing the AD for the efficiently extractable modules based on syntactic locality. Finally, we will use the AD as a mean to better understand the notions of syntactic locality by analysing the results obtained by computing the AD for a large corpus of ontologies in the biomedical domain.

4.4.1 The AD Algorithm and its Complexity

We have discussed in the previous section how the computation of the AD of an ontology can be done by extracting only the genuine modules. In Algorithm 2 we give the pseudocode of a simple procedure to compute the AD. Now we want to establish the complexity of computing the AD of an ontology $\mathcal{O}$.

Proposition 4.4.1. *Let $\mathcal{O}$ be an ontology, and n be the number of axioms in $\mathcal{O}$. Algorithm 2 correctly computes the AD of $\mathcal{O}$ and it requires n module extractions.*

Proof. The correctness is guaranteed by Theorem 4.3.11 and by the following description of the algorithm.

Let n be the number of axioms of $\mathcal{O}$. Clearly, $n \leq s = \sum_{\alpha} \ell(\alpha)$. On line 5 in Algorithm 2 we see that for each axiom $\alpha \in \mathcal{O}$ the module $\mathsf{Module}(\alpha) = \mathcal{M}_\alpha$ is computed. If $\mathsf{Module}(\alpha) = \emptyset$ then the axiom is put in the atom $\mathfrak{t}$ that contains all those axioms that the notion of module recognizes as tautologies of $\mathcal{O}$. Otherwise, Algorithm 2 checks whether there is an axiom β used in a previous iteration for which the module $\mathsf{Module}(\beta) = \mathsf{Module}(\alpha)$. If not, then α is included in the set GenAxs of the generating axioms, so that at the end of the **for** cycle in line 22 there will be one generating axiom for each atom. If instead the axiom α is not generating a new module, then there is a generating axiom β such that $\mathsf{Module}(\beta) = \mathsf{Module}(\alpha)$, and the atom that contains β is updated to include α.

Now, we only need to prove the tractability of the algorithm. The procedure described above requires the extraction of n modules, and at most $i - 1$ comparisons for the axiom α into account at the i-th iteration. The extraction of a module requires at least a linear number of operations in the size of $\mathcal{O}$, hence

Algorithm 2 Algorithm for computing the AD of an ontology

1: **Input:** An ontology $\mathcal{O}$; a notion x of Σ-module.
2: **Output:** The set $\mathfrak{G}(\mathcal{O})$ of genuine modules; the poset of atoms $(\mathcal{A}(\mathcal{O}), \succ)$; a set of generating axioms GenAxs.
3: GenAxs $\leftarrow \emptyset$
4: TAtom $\leftarrow \emptyset$
5: **for** each $\alpha \in \mathcal{O}$ **do**
6: Module$(\alpha) \leftarrow x\text{-mod}(\widetilde{\alpha}, \mathcal{O})$
7: **if** Module$(\alpha) = \emptyset$ **then**
8: TAtom $\leftarrow$ TAtom $\cup \{\alpha\}$
9: **else**
10: *new* $\leftarrow$ *true*
11: **for** each $\beta \in$ GenAxs **do**
12: **if** Module$(\alpha) =$ Module(β) **then**
13: Atom$(\beta) \leftarrow$ Atom$(\beta) \cup \{\alpha\}$
14: *new* $\leftarrow$ *false*
15: **end if**
16: **end for**
17: **end if**
18: **if** *new* = *true* **then**
19: Atom$(\alpha) \leftarrow \{\alpha\}$
20: GenAxs $\leftarrow$ GenAxs $\cup \{\alpha\}$
21: **end if**
22: **end for**
23: **for** each $\alpha \in$ GenAxs **do**
24: **for** each $\beta \in$ GenAxs **do**
25: **if** $\beta \in$ Module(α) **then**
26: Atom$(\beta) \succ$ Atom(α)
27: **end if**
28: **end for**
29: **end for**
30: $\mathcal{A}(\mathcal{O}) \leftarrow \{\text{Atom}(\alpha) \mid \alpha \in \text{GenAxs}\}$
31: $\mathfrak{G} \leftarrow \{\text{Module}(\alpha) \mid \alpha \in \text{GenAxs}\}$
32: **return** $[\mathfrak{G}(\mathcal{O}), (\mathcal{A}(\mathcal{O}), \succ),$ GenAxs$]$

the whole module extraction is at least quadratic, but can be unfeasible for some notions of modules when applied to ontologies encoded in a highly expressive DL. The comparison procedure, needed to obtain the partial ordering $\succ$ and described after the **for** cycle in lines 3-22, is determined by selecting two generating axioms $\alpha, \beta \in$ GenAxs and checking whether one is included in the genuine module for the other. If so, then the corresponding atoms are comparable. This operation

requires $n(n-1)$ comparisons, hence quadratic in the ontology's size. Hence, the extraction of the n modules is at least as expensive timewise as the comparison, and dominates the complexity of computing the AD of $\mathcal{O}$. □

The complexity of computing the AD of an ontology heavily depends on the notion of module used. Next, we determine the complexity of computing ADs when we use the implementation of the extraction of modules based on syntactic locality as described by Tsarkov in [Tsa12].

Proposition 4.4.2. *The computation of the AD of an ontology $\mathcal{O}$ as performed by Algorithm 2 for one of the notions $\{\bot, \top, \top\bot^*\}$ of syntactic locality is quadratic in the number of axioms of $\mathcal{O}$.*

Proof. We have seen in Subsection 2.3.3 that in the process of extracting a module, each axiom is checked for locality at most $|\widetilde{\alpha}|$ times, and that in the worst case when the locality checks bring in exactly one axiom, then the extraction of a module requires at most $n \cdot c$ locality checks [Tsa12], where $c = \max_\alpha |\widetilde{\alpha}|$. The locality check in the case of syntactic modules requires to parse an axiom α, and to check it against the given rules, in order to verify whether α is or not local. The number of rules to check against depends on the language in which $\mathcal{O}$ is encoded, but clearly it contributes to the complexity only as a multiplicative constant e, hence the computational complexity does not depend on the language expressivity. To sum up, a module extraction requires at most $k \cdot n$ operations, where $k = c^2 \cdot e$ is a constant. Hence, it is linear in the size of $\mathcal{O}$. Finally, the extraction of the sets $\mathfrak{G}(\mathcal{O})$ of genuine modules, GenAxs of the generating axioms, and $\mathcal{A}(\mathcal{O})$ of all atoms, is quadratic in the ontology's size.

Hence, computing the AD of an ontology $\mathcal{O}$ is quadratic in the size of $\mathcal{O}$. □

Algorithm 2 can be further optimized as described by Tsarkov in [Tsa12]: let us suppose that, for an axiom $\alpha \in \mathcal{O}$ the module $\mathcal{M}_\alpha$ contains an axiom $\beta \neq \alpha$. Then, we know that $\mathcal{M}_\beta \subseteq \mathcal{M}_\alpha$, hence we do not need to check all the axioms in $\mathcal{O}$ in order to find $\mathcal{M}_\beta$, but only the ones in $\mathcal{M}_\alpha$. A further optimization, finally, consists of exploiting the previous module extractions during the computation of the AD in the following way: whenever a previously analysed axiom α is negative against a locality-check during a successive iteration, then the corresponding genuine module $\mathcal{M}_\alpha$ is automatically included in the current extraction. This optimization, however, does not change the asymptotic complexity.

4.4.2 Experiment: Design and Results

In this subsection, we are going to describe the experiment carried out to analyze the AD of an ontology under three main points of view.

The first analysis aims at evaluating the time taken for computing the AD of an ontology $\mathcal{O}$: from the previous subsection, we know that such a computation for the notions of modules based on syntactic locality is quadratic in the size of $\mathcal{O}$. Please note that the number of axioms of an ontology can be as large as 500K axioms (as it is the case for SNOMED CT), so if we had that a locality check takes $1ms$, then the overall time to decompose SNOMED CT would be higher than 10 years. We will empirically show that for a large corpus of biomedical ontologies the average time taken to perform a locality check is two or three orders of magnitude lower that $1ms$ (depending on the notion of locality). Moreover, we show that the optimizations implemented by Tsarkov empirically reduce the growth power from 2 to ~ 1.6. In particular, from this estimate we can predict that the AD of SNOMED CT should be performed in less than 4 hours.

The second analysis of the AD aims at investigating the notions of syntactic locality and the corresponding modules, and in particular we will provide an answer to some open questions that can be found in the literature. As already mentioned before, in [DPSS10] the authors try to fully modularize a small corpus of ontologies to understand whether the theoretical exponential blowup in the number of modules occurs in practice. Analyzing the results of the experiment, the authors observe that in some ontologies that fail at being fully modularized a high number of modules are disproportionately large, for example this happens for the People (PEO) ontology. A second observation consists of noticing that a high number of modules have a common "core", and differ only by a small number of axioms that do, or do not, occur. The authors thus wonder whether these observations are features of the ontology, or rather they depend on the choice of the signatures analyzed to extract the statistically significant sample of modules. Using the AD we will show that these observation are indeed features of some ontologies, and of course of the notion of module used.

As a third outcome of our experiment, we will analyse the ADs computed under the paradigms defined in the framework described in Chapter 3. This will allow for a comparison of the modular structures defined by the ADs of an ontology with one another, as well as with the modular structures already analyzed in Chapter 3. In particular, we will see that the $\bot$-AD is able to capture and

represent a notion of logical dependence that preserves the relation of subsumption within an ontology $\mathcal{O}$ (e.g., if two named concepts A and B are such that $\mathcal{O} \models \mathtt{A} \sqsubseteq \mathtt{B}$, then either A and B co-occur in the signature of one atom, or the atoms in whose signatures A occurs depend on some atoms where B occurs). Dually, the $\top$-AD preserves the relation of supersumption within $\mathcal{O}$. Finally, the $\top\bot^*$-AD captures a meaningful notion of independence between the axioms of $\mathcal{O}$.

Setting: For our experiments, we have built a corpus containing: (1) ontologies from the TONES repository[2] which have already been studied in previous work on modularity [DPSS10]: Koala, Mereology, University, People, miniTambis, OWL-S, Tambis, Galen; (2) all ontologies from the NCBO BioPortal ontology repository.[3] From this corpus, we have removed those ontologies with at least one of the following problems: the ontology is impossible to download; the .owl file is corrupted when downloaded; the file is not well-formed; the ontology is inconsistent; the ontology contains some constructors that the implementation provided by Tsarkov does not yet support, e.g., some kinds of datatypes. This selection results in a corpus of 253 ontologies designed and built by users (domain experts) for application purposes. These ontologies greatly differ in sizes, expressivities, ratios of axiom and term numbers, classification times, number of non-trivial entailments, and number of justifications per entailment [HPS11]. More details on the whole corpus can be found in Appendix A.

The machine used to perform the experiment is a MacBook Pro with 3.06 GHz Intel Core 2 Duo processor, 8Gb of memory and running Mac OS X version 10.6.8.

Time performance: We have decomposed all the ontologies in our corpus according to the three main notions of syntactic locality: $\bot$, $\top$, and $\top\bot^*$. An external script was designed to call each ontology $\mathcal{O}$ and determine the time needed to compute each AD of $\mathcal{O}$, then further data concerning the ADs have been collected. The experiment has been performed 5 times, and the median time has been considered in the result. Please note, however, that the time needed in general does not vary greatly between the different runs, as described in Table 4.1: the performance time for $\sim 80\%$ of the ontologies differ less than 5% from the average.

[2] http://owl.cs.manchester.ac.uk/repository/
[3] http://bioportal.bioontology.org

Notion of locality	# ontologies with variance time $< 5\%$	# ontologies with variance time $> 20\%$
$\bot$	197	2
$\top$	205	1
$\top\bot^*$	203	0

Table 4.1: Summary of the variance in the performance time

The extraction of the three kinds of ADs over our corpus took, in the median case, less than 67mins of overall time so divided: $23mins$ for the $\bot$-ADs, $11mins$ for the $\top$-ADs, and $33mins$ for the $\top\bot^*$-ADs. To concisely represent the time performance for all the ontologies, we show three graphs drawn as follows: for each ontology $\mathcal{O}$ we consider (1) the size $s(\mathcal{O})$ of $\mathcal{O}$ as defined in Section 2.1, (2) the time $t_{\mathcal{O}}$ to compute the AD in μs, and (3) the expressivity bin that $\mathcal{O}$ belongs to. We plot a graph of the points $(s_{\mathcal{O}}, t_{\mathcal{O}})$, where the colour of the point represents the expressivity bin of $\mathcal{O}$. Given the values involved, both axes are in logarithmic scale. The 3 graphs, one for each kind of AD, are represented in Figure 4.6 for $\bot$-ADs, in Figure 4.7 for $\top$-ADs, and in Figure 4.8 for $\top\bot^*$-ADs.

From the analysis of the computational complexity of Algorithm 2 we could infer that the language expressivity of the ontology has little influence on the AD computation time, and this conclusion is shown in the graphs since there is no substantial difference in general in the trendlines defined by ontologies from different bins. The equations described in the graphs are the best fitting power equations to the points represented: in particular we see that in most cases the exponents involved are ~ 1.6, so the quadratic worse-case is generally not met.

We see that in two of the graphs, the one for the $\bot$- and the one for the $\top\bot^*$-AD time, there is one point whose ordinate largely dominates the ordinates of the other points: in both cases the ontology is nci-thesaurus (NCI), whose ADs take respectively $\sim 21mins$ and $\sim 24mins$.[4] Surprisingly, the $\top$-AD computation for NCI takes only $2.2s$ to be performed. This observation is even more surprising if we consider that by discarding NCI and looking at the time needed to compute the remaining ADs we obtain the following times: $2mins$ for the $\bot$-ADs, $9mins$ for the $\top\bot^*$-ADs, and $11mins$ for the $\top$-ADs. In other words, generally the computation of the $\top$-AD is the one that takes more time among the three main notions, whilst for NCI the converse happens.

[4]Please note that NCI contains 130,945 axioms.

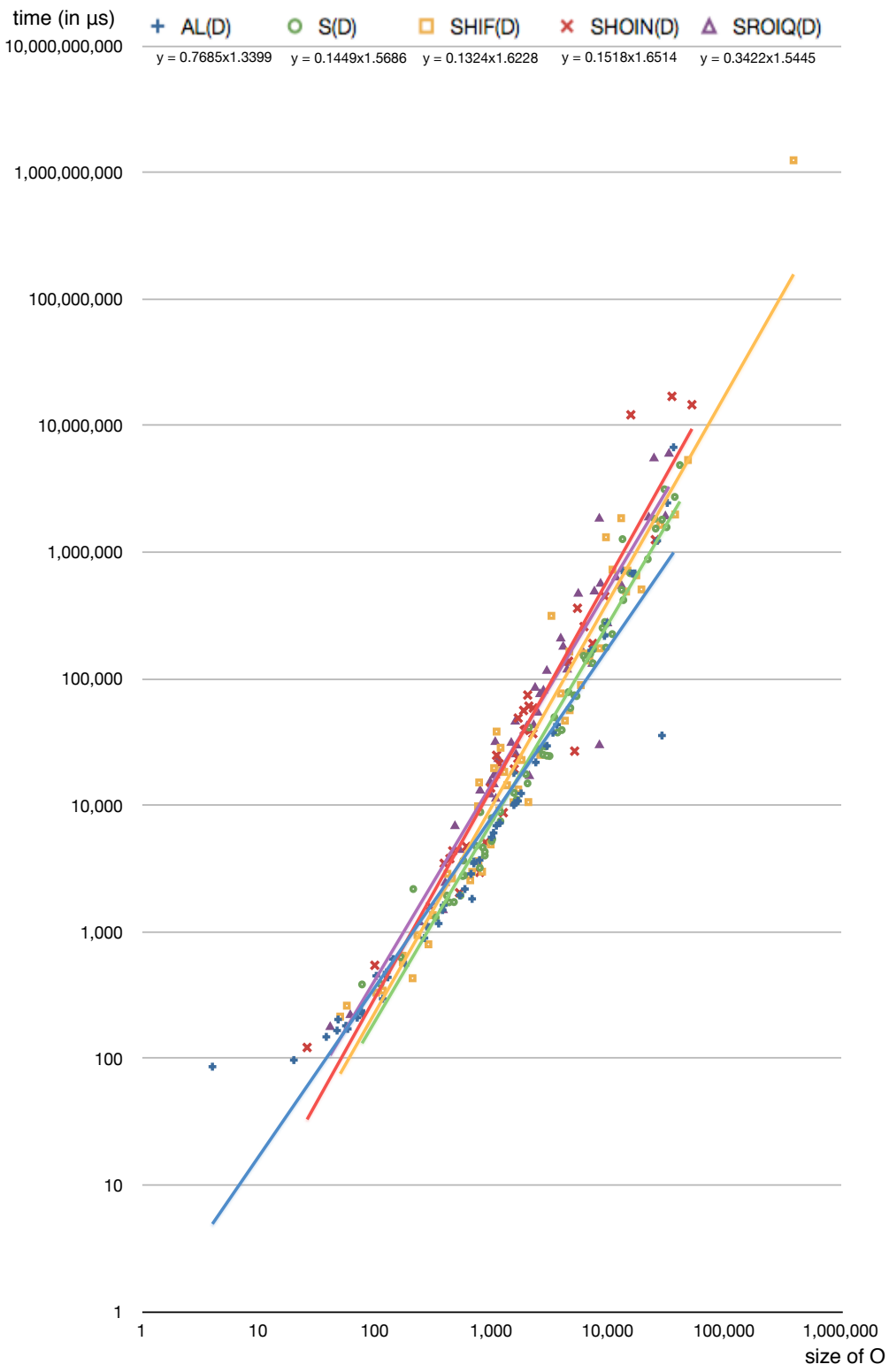


Figure 4.6: Time to $\bot$-AD vs. size of $\mathcal{O}$

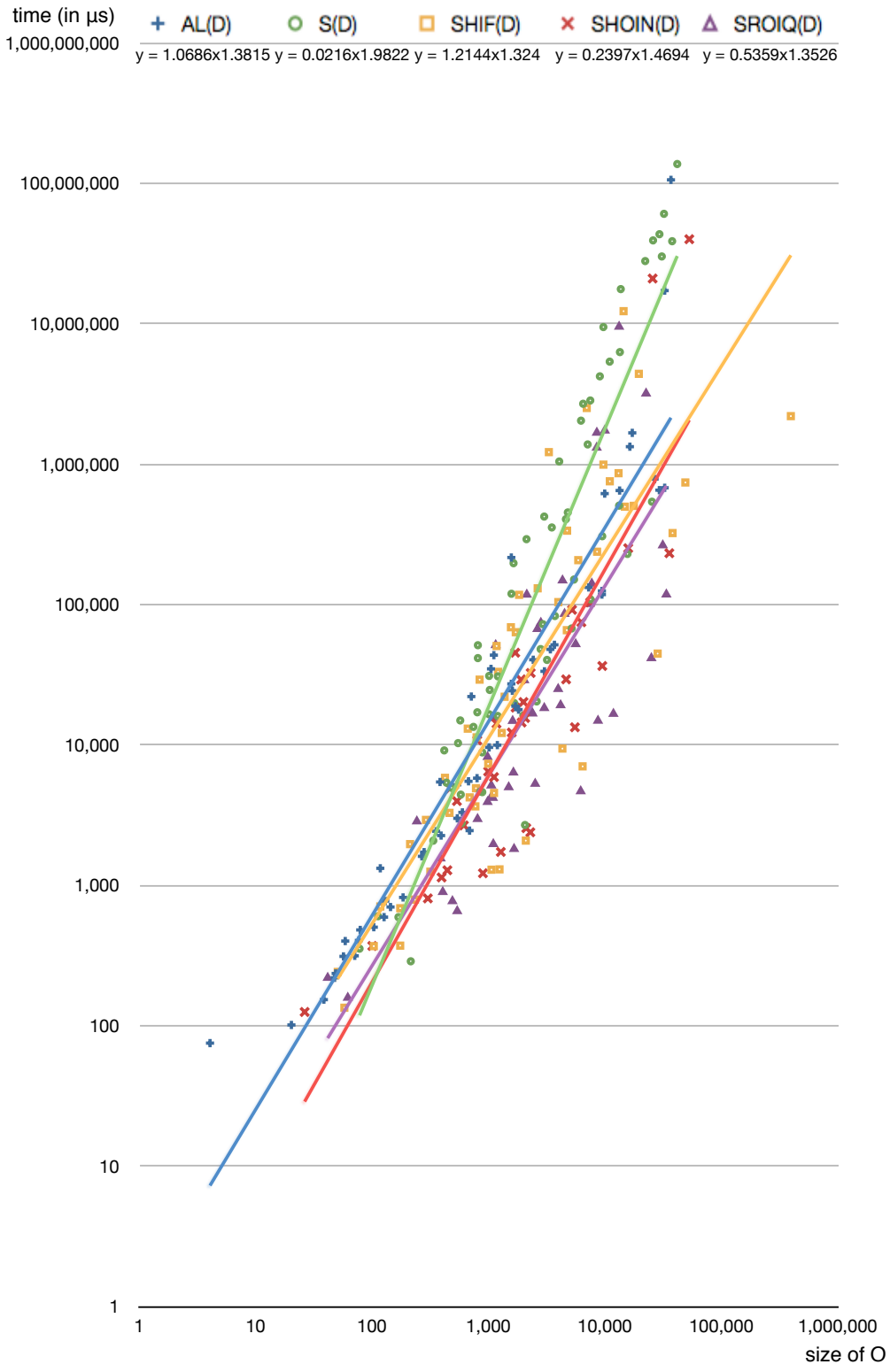


Figure 4.7: Time to $\top$-AD vs. size of $\mathcal{O}$

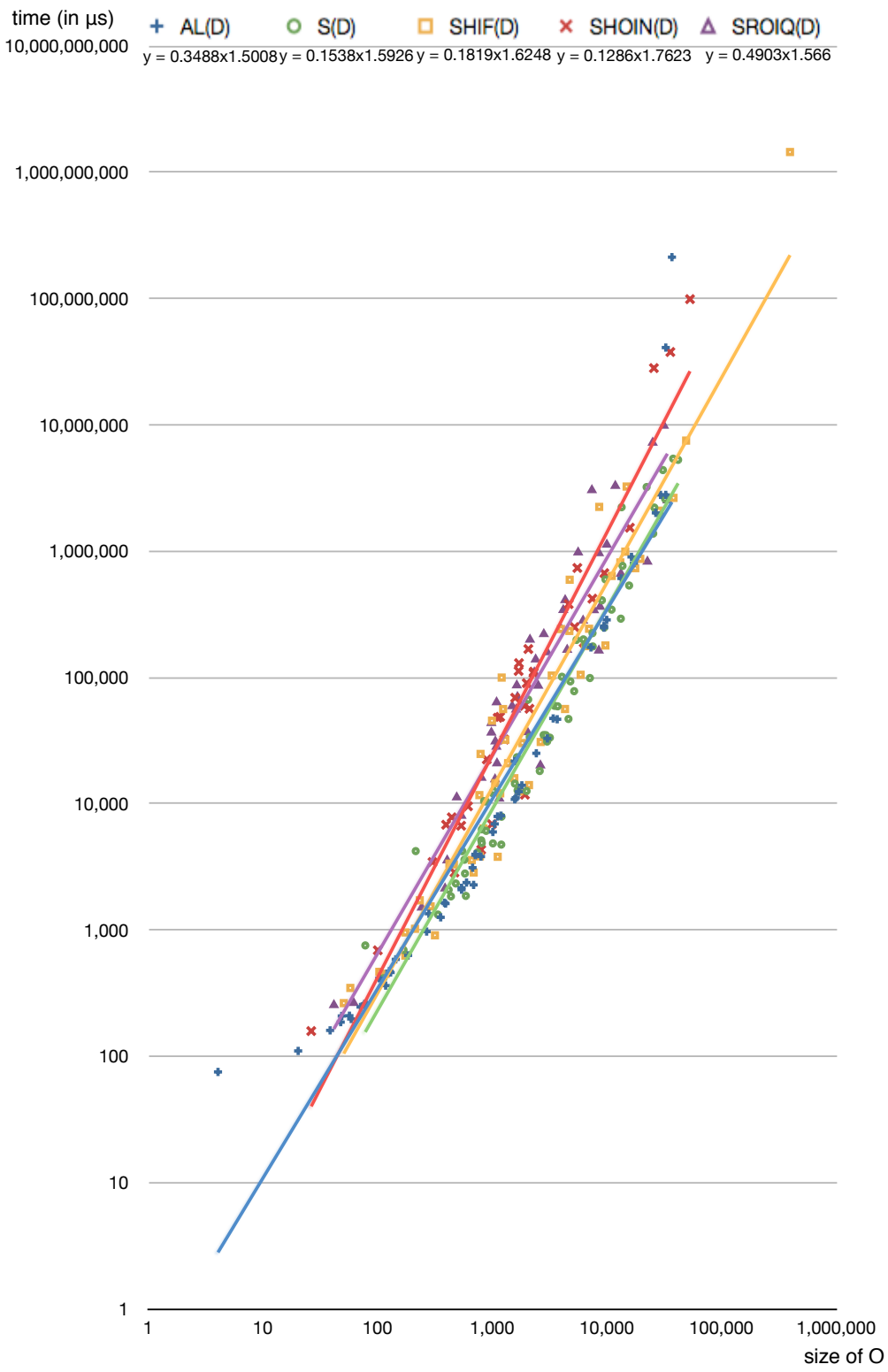


Figure 4.8: Time to $\top\bot^*$-AD vs. size of $\mathcal{O}$

Finally, we want summarize in Table 4.2 the number of ontologies, for each notion of locality, that are computed within a certain time interval. We see that the majority of the ontologies (i.e., $\sim 68\%$) are decomposed in less than $0.1s$, i.e., we can say that the decomposition is instantaneous. For more than a quarter of the ontologies, the decomposition is hardily noticeable by a human user, i.e., it takes less than $4s$. The remaining few cases consists of large and very large ontologies (i.e., the smallest is mouse-adult-gross-anatomy (MAA) with 3,776 axioms whose computation of the $\top$-AD takes $\sim 5.4s$).

Notion of locality	# ontologies with $t_{\mathcal{O}} < 0.1s$	# ontologies with $0.1s \leq t_{\mathcal{O}} < 4s$	# ontologies with $4s \leq t_{\mathcal{O}} < 10s$	# ontologies with $t_{\mathcal{O}} \geq 10s$
$\top\bot^*$	161	80	6	6
$\bot$	177	67	5	4
$\top$	175	59	6	13

Table 4.2: Summary of the performance time

Analysis of the modules through the ADs: In this paragraph we will answer to two open questions concerning modules by exploiting the ADs of the ontologies. In specific, we will investigate on the two question arisen in [DPSS10] for the peculiar behaviour of a statistically significant amount of modules of some ontologies. The first question comes from observing that in some ontologies (e.g., the People ontology) a high number of modules are disproportionately large. The second question consists of noticing that in some ontologies a high number of modules have a common "core", and differ only by a small amount of axioms that do, or do not, occur. When this happens, the modules of $\mathcal{O}$ are called to have a *Mexican hat* shape, represented in Figure 4.9: the big central core is a set of axioms that occur in most of the modules of $\mathcal{O}$. The small circles around the core represent the axioms that can combinatorially occur in the modules of $\mathcal{O}$.

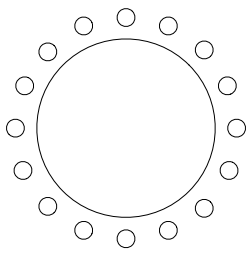


Figure 4.9: A Mexican hat

Both these cases can be explained if the corresponding ontology $\mathcal{O}$ has a *huge* atom $\mathfrak{a}$ in the lower part of an AD: for what concerns the average size of the modules of $\mathcal{O}$, we have that, if this happens, then it is quite likely that a non-empty module contains a large number of axioms. Moreover, if such atom $\mathfrak{a}$ is the lowest w.r.t. $\succ$ among all the atoms, then every non-empty module of $\mathcal{O}$ needs to contain $\mathfrak{a}$, thus it is "large". For what concerns the Mexican hat shape, we can explain the observation when the huge atom $\mathfrak{a}$ is in the lower part of an ontology, and many small atoms depend on $\mathfrak{a}$.

In Figure 4.10 we represent the $\top\bot^*$-AD of the ontology People, and we see that the empirical observations described in [DPSS10] are explained by the $\top\bot^*$-AD, consisting of 26 atoms, where the $\top\bot^*$-atom $\mathfrak{a}_0$ contains indeed 73 axioms out of the 108 that People is made of.

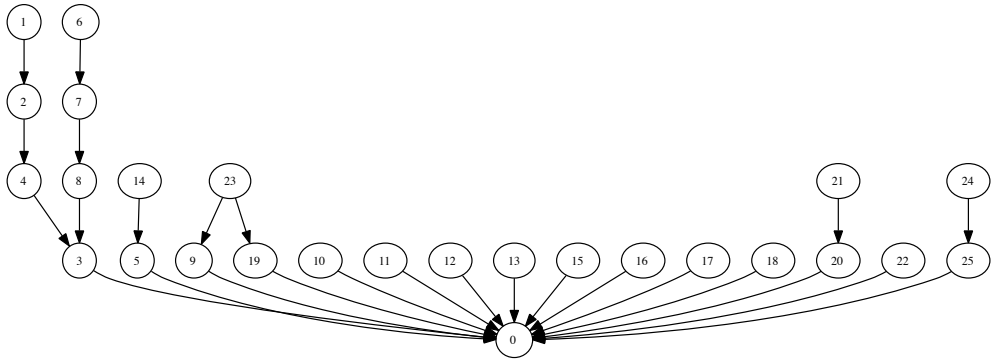


Figure 4.10: $\top\bot^*$-AD of the ontology People

The following proposition establishes a strong connection between the atoms of the three main notions of ADs based on syntactic locality: the $\top\bot^*$-atoms of an ontology $\mathcal{O}$ are a refinement w.r.t. set inclusion of both the $\bot$- and the $\top$-atoms of $\mathcal{O}$, so that ontologies have more, smaller $\top\bot^*$-atoms than $\bot$- or $\top$-atoms.

Proposition 4.4.3. *The* $\top\bot^*$-AD *is finer than both the* $\bot$-AD *and* $\top$-AD*, i.e., for any* $\top\bot^*$*-atom* $\mathfrak{a}$*, there exists a* $\bot$*-atom* $\mathfrak{b}$ *and a* $\top$*-atom* $\mathfrak{c}$ *with* $\mathfrak{a} \subseteq \mathfrak{b}$ *and* $\mathfrak{a} \subseteq \mathfrak{c}$*.*

Proof. We prove this result for $\bot$-modules. The proof for $\top$-modules is analogous.

Let $\mathfrak{a}^*$ be one star atom, and let $\alpha \in \mathfrak{a}^*$. Then, the $\bot$-module $\mathcal{M}_\alpha^\bot$ for α satisfies $\mathcal{M}_\alpha^\bot \supseteq \mathcal{M}_\alpha^* \supseteq \mathfrak{a}^*$. By the arbitrary choice of α in $\mathfrak{a}^*$, we have that $\forall \alpha \in \mathfrak{a}^*$, $\mathcal{M}_\alpha^\bot \supseteq \mathfrak{a}^*$.

We want now to prove that $\mathcal{M}_\alpha^\perp = \mathcal{M}_\beta^\perp$ for all $\beta \in \mathfrak{a}^*$.

By Lemma 4.3.1, $\mathcal{M}_\alpha^\perp$ is the smallest $\perp$-module containing α; since it contains β (because it contains $\mathfrak{a}^*$) we have that $\mathcal{M}_\alpha^\perp \supseteq \mathcal{M}_\beta^\perp$. For the same reason we have that $\mathcal{M}_\alpha^\perp \subseteq \mathcal{M}_\beta^\perp$. Then, for each axiom γ we have that $\mathcal{M}_\gamma^*$ is contained in $\mathcal{M}_\gamma^\perp$, so that $\top\perp^*$-atoms do not split between $\perp$-modules.

To sum up, the $\top\perp^*$-AD is finer-grained than the $\perp$-AD. □

The next results establishes a further connection of $\top\perp^*$-ADs with the $\perp$- and $\top$-ADs: if two atoms $\mathfrak{a}^*, \mathfrak{b}^*$ are related in the $\top\perp^*$-AD of an ontology $\mathcal{O}$, then either there exists a $\perp$-atom $\mathfrak{a}^\perp$ that contains both $\mathfrak{a}^*$ and $\mathfrak{b}^*$, or the two $\perp$-atoms $\mathfrak{a}^\perp$ cointaining $\mathfrak{a}^*$ and $\mathfrak{b}^\perp$ containing $\mathfrak{b}^*$ are comparable w.r.t. $\succ$. The analogous result holds for the $\top$-AD of $\mathcal{O}$.

Proposition 4.4.4. *Let $\mathcal{O}$ be an ontology, $\mathfrak{a}^*, \mathfrak{b}^*$ two $\top\perp^*$-atoms of $\mathcal{O}$ such that $\mathfrak{a}^* \succ \mathfrak{b}^*$, $x \in \{\perp, \top\}$ a notion of locality, and $\mathfrak{a}^x, \mathfrak{b}^x$ the two x-atoms of $\mathcal{O}$ such that $\mathfrak{a}^* \subseteq \mathfrak{a}^x$ and $\mathfrak{b}^* \subseteq \mathfrak{b}^x$. Then, either $\mathfrak{a}^x = \mathfrak{b}^x$, or $\mathfrak{a}^x \succ \mathfrak{b}^x$ in the x-AD of $\mathcal{O}$.*

Proof. Let $\alpha \in \mathfrak{a}^*$ and $\beta \in \mathfrak{b}^*$ be two axioms. Let us consider the module $\mathcal{M}_\alpha^* = \top\perp^*\text{-mod}(\widetilde{\alpha}, \mathcal{O})$. Then, by the construction of *tbstar*-modules we have that $\mathcal{M}_\alpha^* \subseteq \mathcal{M}_\alpha^x = x\text{-mod}(\widetilde{\alpha}, \mathcal{O})$. In particular, $\beta \in \mathcal{M}_\alpha^x$, and $\mathcal{M}_\beta^x = x\text{-mod}(\widetilde{\alpha}, \mathcal{O}) \subseteq \mathcal{M}_\alpha^x$. Finally, if $\mathcal{M}_\beta^x = \mathcal{M}_\alpha^x$ we have that $\mathfrak{a}^x = \mathfrak{b}^x$; otherwise, $\mathfrak{a}^x \succ \mathfrak{b}^x$. □

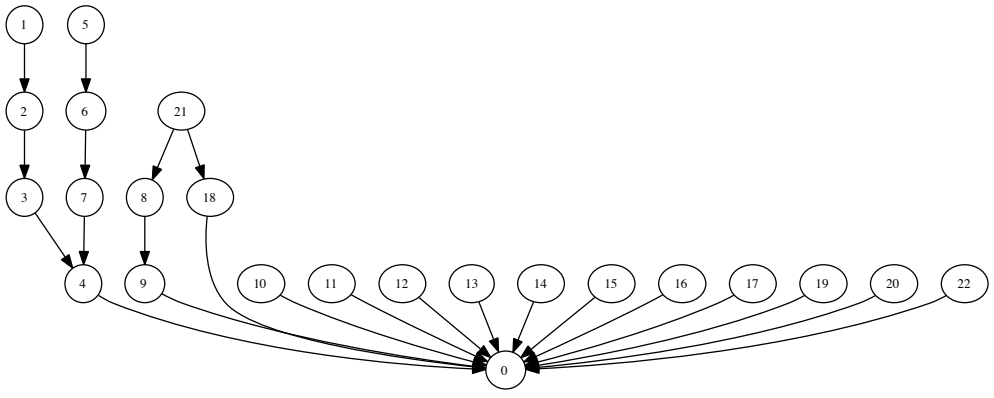


Figure 4.11: $\perp$-AD of the ontology People

In Figure 4.11 we can see the $\perp$-AD of the ontology People: compared to the $\top\perp^*$-AD represented in Figure 4.10, we have that the atoms $\mathfrak{a}_{14}^*$ and $\mathfrak{a}_9^*$ are merged into atom $\mathfrak{a}_8^\perp$, the atoms $\mathfrak{a}_{20}^*$ and $\mathfrak{a}_{21}^*$ are merged into atom $\mathfrak{a}_{19}^\perp$, and that the atom

$\mathfrak{a}^*_{25}$ is absorbed into the atom $\mathfrak{a}^{\perp}_0$. To sum up, the $\perp$-AD of the People ontology is made of just 23 atoms, and the lowest atom $\mathfrak{a}_0$ contains 75 axioms.

For the notion $\top$ of locality, the $\top\perp^*$-atoms of the ontology People get merged more frequently than in the $\perp$-AD: in Figure 4.12 we see that the $\top$-AD has only 4 atoms; moreover, the lowest atoms $\mathfrak{a}^{\top}_0$ contains 97 axioms.

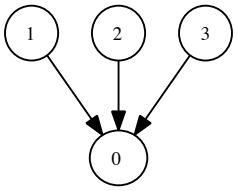


Figure 4.12: $\top$-AD of the ontology People

We have investigated the presence of huge atoms for every ontology $\mathcal{O}$ in our corpus: for each decomposition of $\mathcal{O}$, we have computed maximum, minimum, and average sizes of the atoms of $\mathcal{O}$. In Table 4.3 we show how the ontologies are distributed in terms of the absolute maximum size $m^a_{\mathfrak{a}}$ of their biggest atom. Similarly, in Table 4.4 we show the distribution of the ontologies in our corpus by the relative size $m^r_{\mathfrak{a}}$ of their biggest atom compared to the ontology's number of axioms.

Notion of locality	# ontologies with $m^a_{\mathfrak{a}} \leq 20$	# ontologies with $20 < m^a_{\mathfrak{a}} \leq 500$	# ontologies with $m^a_{\mathfrak{a}} > 500$
$\top\perp^*$	163	70	20
$\perp$	157	76	20
$\top$	32	121	100

Table 4.3: Distribution of the ontologies in our corpus by the absolute size of their biggest atoms

Notion of locality	# ontologies with $m^r_{\mathfrak{a}} \leq 20\%$	# ontologies with $20\% < m^r_{\mathfrak{a}} \leq 50\%$	# ontologies with $m^r_{\mathfrak{a}} > 50\%$
$\top\perp^*$	215	20	18
$\perp$	208	21	24
$\top$	73	36	144

Table 4.4: Distribution of the ontologies in our corpus by the relative size of their biggest atoms

We see that $\sim 63\%$ of the ontologies investigated decompose into small $\bot$- and $\top\bot^*$-atoms, and less than $\sim 8\%$ of the ontologies have huge $\bot$- and $\top\bot^*$-atoms. Differently, less than $\sim 13\%$ of the ontologies decompose into small $\top$-atoms, and almost $\sim 40\%$ of the ontologies have huge $\top$-atoms. The distribution of the ontologies does not differ much when we look at relative sizes of atoms.

Proposition 4.4.3 suggests to investigate on the presence of huge atoms by looking at the $\top\bot^*$-AD of an ontology. We report in Table 4.5 the relevant data for the ontologies with huge $\top\bot^*$-atoms in absolute terms, i.e., with more than 500 axioms, and in Table 4.6 the relevant data for the ontologies with huge $\top\bot^*$-atoms in relative terms, i.e., containing more than 50% of the axioms of the ontology.

Ontology $\mathcal{O}$ (Abbreviation)	# axioms in $\mathcal{O}$	#max atom $\mathfrak{a}$	Percentage $\#\mathfrak{a}/\#\mathcal{O}$
SYN	14,458	14,458	100%
ICF	19,223	12,198	63.46%
NAN	16,353	6,425	39.29%
SNP	11,199	3,233	28.87%
UNI	3,133	2,868	91.54%
CLL	3,996	2,444	61.16%
CCJ	13,612	2,236	16.43%
BTC	2,734	2,201	80.50%
MTP	2,364	1,790	75.72%
CTX	1,969	1,706	86.64%
ODG	1,867	1,601	85.75%
NCI	130,945	1,087	0.83%
CDS	4,322	843	19.50%
SSO	1,684	789	46.85%
IMG	2,114	769	36.38%
NEO	2,843	717	25.22%
ENM	931	650	69.82%
QIB	1,699	607	35.73%
ODM	2,353	607	25.80%
BCG	690	537	77.83%

Table 4.5: Ontologies with huge atoms in absolute value

We carried out a preliminary investigation of ontologies with huge atoms, trying to understand the reasons for the existence of huge atoms, and identified two reasons for the presence of huge atoms.

The first reason identified to cause the presence of huge atoms is the occurence in the ontology of global axioms, as it is the case for concept assertions, i.e.,

Ontology $\mathcal{O}$ (Abbreviation)	$\#\mathcal{O}$	#max atom $\mathfrak{a}$	Percentage $\#\mathfrak{a}/\#\mathcal{O}$
SYN	14,458	14,458	100%
BFO	95	89	93.68%
AMA	477	445	93.29%
UNI	3,133	2,868	91.54%
CTX	1,969	1,706	86.64%
ODG	1,867	1,601	85.75%
BTC	2,734	2,201	80.50%
BCG	690	537	77.83%
MTP	2,364	1,790	75.72%
ENM	931	650	69.82%
CBO	13	9	69.23%
PEO	108	73	67.59%
ICF	19,223	12,198	63.46%
CLL	3,996	2,444	61.16%
SDO	204	121	59.31%
GMS	216	122	56.48%
MSE	218	122	56.48%
GDP	773	399	51.62%

Table 4.6: Ontologies with relatively huge atoms

axioms of the kind $\mathtt{C}(a)$, and role assertions, i.e., axioms of the kind $\mathtt{r}(a, b)$. Global axioms can also pull non-global axioms to be included in a module in the following way: because of the property of self-containment, the seed signature Σ_0 of input for the extraction of a module is enlarged with the terms occurring in the global axioms. The ontology People falls in this case: it contains 25 concept assertions, among which the axiom $\alpha_1 = \mathtt{Person}(\mathit{Walt})$, and 12 role assertions, among which $\alpha_2 = \mathtt{hasPet}(\mathit{Walt}, \mathit{Dewey})$. Now, the axiom $\alpha_3 = \mathtt{DogOwner} \equiv \mathtt{Person} \sqcap (\exists \mathtt{hasPet}.\mathtt{Dog})$ is non local w.r.t. any signature that contains the terms Person and hasPet, hence α_3 belongs to any module of People because of α_1 and α_2. However, α_3 is not a global axiom *per se*. An even more peculiar example is the ontology SYN, which contains only concept assertions.

A second reason for huge atoms to occur in the AD of an ontology can be attributed to the abundance of Disjoint Covering Axioms (DCAs), and we assume that their abundance is due to a specific usage pattern of ontology editors. More precisely, one version of DCAs is a pair of axioms of the form $\{\mathtt{A} \equiv (\mathtt{B}_0 \sqcup \ldots \sqcup \mathtt{B}_n), \text{PairwiseDisjoint}(\mathtt{B}_0, \ldots, \mathtt{B}_n)\}$. Since our notion of modularity is based on

axioms and subsets of an ontology and is self-contained, any module that mentions $\mathtt{B}_i$ contains both axioms, and thus pulls in all axioms about $\mathtt{B}_j$ as well. When DCAs occur on many classes on all levels in the class hierarchy of an ontology, then this results, unsurprisingly, in a huge atom. Moreover, note that not only disjointness causes axioms to tie together, as the explicit covering axiom shows the same behaviour. For disjointness, however, this "pulling-in" effect does not occur if we rewrite the n-ary disjointness axiom into equivalent pairwise disjointness axioms or even make the disjointness implicit, as in the following example: $\{\mathtt{B}_0 \sqsubseteq \mathtt{A} \sqcap (= \ 0\,\mathtt{R}.\top), \ldots, \mathtt{B}_{n-1} \sqsubseteq \mathtt{A} \sqcap (= \ n-1\,\mathtt{R}.\top), \mathtt{B}_n \sqsubseteq \mathtt{A} \sqcap (\geq \ n\,\mathtt{R}.\top)\}$. Other patterns could also lead to huge atoms, and an investigation of possible patterns is part of our future work.

Evaluation of ADs as Modular Structures: We want now to evaluate the ADs of an ontology by following the framework defined in Section 3.2. Hence, we will analyse the AD of an ontology in terms of the coherence of its modules, the dependence/independence that the AD is sensitive to, and the granularity of an AD and which kinds of logical interrelations it can reveal. The computability has already been investigated at the beginning of this section.

To carry out this analysis, we first compute a basic set of metrics for each notion $\bot, \top$, and $\top\bot^*$ of locality: for each ontology, we compute the average and maximal size of atoms and Genuine Modules (GM) measured in numbers of axioms, and then we take the average of the resulting numbers over all the 253 ontologies in our corpus. The results are presented in Table 4.7.

Notion of locality	Average average axioms/atom	Average maximum axioms/atom	Average average axioms/GM	Average maximum axioms/GM
$\top\bot^*$	30.99	262.32	187.96	417.09
$\bot$	60.75	352.47	332.14	452.89
$\top$	481.70	1,955.41	1,716.26	2,870.51

Table 4.7: Overview of the experimental results for all the ontologies in our corpus

To show the influence on these numbers of those ontologies with huge atoms, we present in Table 4.8 the same metrics computed for the corpus of ontologies obtained by removing the 28 ontologies with (absolute or relative) huge atoms from our original corpus.

Notion of locality	Average average axioms/atom	Average maximum axioms/atom	Average average axioms/GM	Average maximum axioms/GM
$\top\bot^*$	1.42	32.10	34.20	100.50
$\bot$	1.75	37.52	46.05	115.96
$\top$	259.10	1,250.99	1,028.59	2,186.64

Table 4.8: Overview of the experimental results for ontologies without huge atoms

It can be seen that, if an ontology does not have huge atoms, then the $\top\bot^*$-AD is generally quite fine-grained: the average size of an atom is less than 2 axioms; indeed, 83 ontologies out of 253 have only singleton $\top\bot^*$-atoms, and only 98 ontologies have at least one $\top\bot^*$-atom whose size is greater than 10 axioms. Next, we observe that ontologies are also, in general, decomposable into small $\bot$-atoms: 39 ontologies out of 253 have only singleton $\bot$-atoms, and only 109 ontologies have at least one $\bot$-atom whose size is greater than 10 axioms.

In contrast, $\top$-AD is substantially coarser than both $\top\bot^*$ and $\bot$-ADs as the average atom is two orders of magnitude larger even if we consider only the ontologies without huge $\top\bot^*$-atoms. However, this datum does not imply that small *top*-atoms do not occur in practice: indeed, 206 ontologies out of 253 have at least one singleton $\top$-atom.

From the definition of atoms, we understand that the axioms in an atom $\mathfrak{a}$ must be always considered at once to grasp their meaning. This remark shows how tight the logical interrelations between these axioms are, hence the coherence of an atom is strong. Moreover, from the empirical evaluation described in Table **??** and in Table 4.8 we now know that atoms are, in general, and particularly for the $\top\bot^*$- and the $\bot$-ADs, quite small fragments, often even singletons, so the aggregation of more than one axiom reveals a tight model-theoretic relationship between the axioms.

Similarly, we can describe the logical coherence of a module $\mathcal{M} \subseteq \mathcal{O}$ as follows: since a module $\mathcal{M} \subseteq \mathcal{O}$ is genuine if there exists an axiom α such that $\mathcal{M} = \mathcal{M}_\alpha$, we can argue that $\mathcal{M}$ is designed to be a suitably small (depleting and self-contained) subset of an ontology to give $\mathcal{O}$'s meaning to α. In particular, set $\mathfrak{a}$ to be the atom such that $\mathcal{M} =\downarrow \mathfrak{a}$ we have that the chains of CEs defined in Subsection 4.3.2 reveals an order for importing sets of axioms in order to guarantee that $\mathcal{O}$'s meaning over $\mathfrak{a}$ is preserved. Dually, the independence of two atoms reveals also a fine-grained logical independence.

Finally, the granularity of an AD can be thought of in terms of the connected components of the diagram. In Table 4.9 we show more empirical results that provide an insight on how different kinds of ADs *structurally* differ in this sense. In particular, we have computed the depth, i.e., the length, in atoms, of the longest chain for each ontology, and the number of connected components of all the ADs, and then computed the averages of these measures across all the ontologies in the corpus for each notion $\{\bot, \top, \top\bot^*\}$ of locality.

Notion of locality	Average max. depth per ontology	Average nr. of connected components
$\top\bot^*$	7.73	1,015
$\bot$	10.53	33.06
$\top$	5.73	1.54

Table 4.9: Depth and connectedness of the ADs of the ontologies in our corpus

We notice that the $\top\bot^*$-AD looks rather disconnected since on average an ontology decomposes into more than a thousand unrelated aggregations of atoms, whilst the $\bot$-AD splits the ontology into 33 components, and the $\top$-AD into only 1 or 2 components. Finally, we see that the $\bot$-AD is the one kind that in general generates the longest chains of atoms among the three main notions of ADs.

Intuitively we can say, then, that the different notions of locality reveal different logical interactions between the axioms. Hence, as long as we are aware of the kind of interrelations, we can prefer one kind of AD of an ontology over another notion because the diagram of the former is "more interesting" that the latter. As a rule of thumb, though, the experimental results suggest that in general the $\bot$-AD is to prefer as a kind of AD for two reasons: (1) on average the $\bot$-AD is the notion with the highest length of chains and the lowest number of connected components, and (2) the atoms size is still rather small.

The evaluation of this modular structure can be summarized as follows:
- Coherence: rather tight since two axioms belong to the same atom only if the module notion x is not able to distinguish them.
- Dependence/Independence: the "$\succ$" relation is able to reveal subtle notions of logical dependence/independence, strongly connected to the notion of conservative extensions, by allowing the construction of chains of CEs.
- Granularity: especially for the two notions of modules $\bot$ and $\top\bot^*$, we have empirical results that show that on average the granularity of the AD is quite

fine-grained since the number of atoms is often more than $n/2$, where n is the number of axioms in $\mathcal{O}$.
- Computability: efficient in theory and feasible in practice.

Chapter 5

Labelled Atomic Decompositions

In Chapter 4 we have introduced the Atomic Decomposition $(\mathcal{A}(\mathcal{O}), \succ)$ of an ontology $\mathcal{O}$ as a structure that represents the set $\mathfrak{G}(\mathcal{O})$ of the genuine modules of $\mathcal{O}$, which is a base for all the modules of $\mathcal{O}$, as well as the logic-based relations between these modules. From Proposition 4.3.10 we know that each module $\mathcal{M}$ is uniquely determined in $(\mathcal{A}(\mathcal{O}), \succ)$ by a set of atoms $\{\mathfrak{a}_1, \ldots, \mathfrak{a}_\kappa\}$ such that $\mathcal{M} = \bigcup_{i=1}^{\kappa} \mathfrak{a}_i$. However, from the AD alone we cannot find the unique set of atoms which form the module for a given signature Σ since the bare structure does not represent the relation between Σ and the atoms in $\mathcal{A}(\mathcal{O})$.

The idea arising from this observation is that we need to enrich the AD of an ontology with a suitable set of *labels* to reveal the relations between atoms and signatures. This enriched structure is called *Labelled Atomic Decomposition* (*LAD*). Please note that there can be more than one relation between signatures and atoms of interest for different applications.

Throughout this chapter, we will analyse two main kinds of labels. Labels of the first kind are based on *Minimal Seed Signatures* (MSSs), i.e., minimal sets Σ of terms such that the module for Σ is $\downarrow \mathfrak{a}$. Labels of the second kind are simply based on $\widetilde{\mathfrak{a}}$, and reveals which terms are mentioned in an atom.

In Section 5.2 we prove that the LAD defined by using the sets of MSSs as labels is a suitable representation for extracting the module for a signature Σ directly from the LAD of $\mathcal{O}$. Hence, these labels capture the logical interrelations between the modules of $\mathcal{O}$ and the signatures in $\widetilde{\mathcal{O}}$. Intuitively, this means that by labelling the AD with this kind of label we gain insight onto which terms are indeed constrained by $\mathcal{O}$.[1]

[1]Here we disregard the fact that modules can in general contain unnecessary axioms

The labels based on signatures can help in providing the terms of an ontology with a logic-based ordering through the use of chains of CEs (recall Definition 4.3.14). Intuitively, then, it makes sense to say that an entity is defined in terms of another, e.g. `Person` is defined in terms of `Animal`.

In Section 5.4 we also discuss the idea that, by comparing these two approaches, one can understand which terms are indeed constrained, and which ones are just mentioned in the ontology. Such a comparison has interesting application scenarios to support ontology engineers in the development and maintainance of an ontology.

Finally, we introduce a preliminary investigation on the notions of *relevance* and *direct relevance* of an axiom to a term $\mathtt{t}$, aimed at formalising the idea that LADs can be used to identify the relevant of an ontology for a term. Intuitively, an axiom α is directly relevant to a term if, whenever a model $\mathcal{I}$ of α is fixed, then we can turn $\mathcal{I}$ into a non model of α just by changing the extension $\mathtt{t}^{\mathcal{I}}$. More in general, in the context of an ontology $\mathcal{O}$, an axiom is relevant to a term if $\mathtt{t}$ can be interpreted in $\mathcal{O} \setminus \{\alpha\}$ in a way which is independent on how the terms in $\widetilde{\alpha} \setminus \{\mathtt{t}\}$ are interpreted in $\mathcal{O}$.

It is not yet clear how the notions of relevance and of direct relevance are related to LADs. However, we have *prima facie* evidence that LADs can reveal the part of an ontology that is relevant for a given term. Completing the investigation of this issue is part of our future work.

5.1 Labels

Given the AD of an ontology $\mathcal{O}$, we want to provide each atom $\mathfrak{a}$ with a label sufficient to capture a given logical relation. As mentioned before, one possible kind of labelling aims at enabling the extraction of a module $\mathcal{M} = \mathrm{mod}(\Sigma, \mathcal{O})$ for the signature Σ from the AD, as described in the following example.

Example 5.1.1. Let us recall the Girl ontology from Example 4.3.13 and the $\bot$-AD for Girl as in Figure 4.4. The ontology's signature consists of 4 terms: $\mathtt{Woman}, \mathtt{Child}, \mathtt{Girl}$, and $\mathtt{Person}$; hence there are $2^4 = 16$ possible signatures to choose from for extracting a module.

We can summarize the possible signature choices as follows:
- If $\Sigma \subseteq \{\mathtt{Person}, \mathtt{Girl}\}$, then the module $\mathrm{mod}(\Sigma, \mathsf{Girl})$ is empty.
- If Σ contains the term $\mathtt{Woman}$ but not the term $\mathtt{Child}$, then $\mathrm{mod}(\Sigma, \mathsf{Girl}) = \{\alpha_1\}$.

- If Σ contains the term $\texttt{Child}$ but not the term $\texttt{Woman}$, then $\text{mod}(\Sigma, \mathsf{Girl}) = \{\alpha_2\}$.
- If $\Sigma \supseteq \{\texttt{Woman}, \texttt{Child}\}$, then $\text{mod}(\Sigma, \mathsf{Girl}) = \mathsf{Girl}$.

In order to reveal which atoms are included in the module $\mathcal{M}_\Sigma$ for a given signature Σ, we could label each atom with *all* the signatures that make the atom be included in $\mathcal{M}_\Sigma$, and obtain the following mapping:

$$\begin{aligned}
\mathfrak{a}_1 \mapsto \{&\{\texttt{Woman}\}, \{\texttt{Woman}, \texttt{Person}\}, \{\texttt{Woman}, \texttt{Girl}\}, \{\texttt{Woman}, \texttt{Person}, \texttt{Girl}\},\\
&\{\texttt{Child}, \texttt{Woman}\}, \{\texttt{Child}, \texttt{Woman}, \texttt{Person}\}, \{\texttt{Child}, \texttt{Woman}, \texttt{Girl}\},\\
&\{\texttt{Child}, \texttt{Woman}, \texttt{Person}, \texttt{Girl}\}\}\\
\mathfrak{a}_2 \mapsto \{&\{\texttt{Child}\}, \{\texttt{Child}, \texttt{Person}\}, \{\texttt{Child}, \texttt{Girl}\}, \{\texttt{Child}, \texttt{Person}, \texttt{Girl}\},\\
&\{\texttt{Child}, \texttt{Woman}\}, \{\texttt{Child}, \texttt{Woman}, \texttt{Person}\}, \{\texttt{Child}, \texttt{Woman}, \texttt{Girl}\},\\
&\{\texttt{Child}, \texttt{Woman}, \texttt{Person}, \texttt{Girl}\}\}\\
\mathfrak{a}_3 \mapsto \{&\{\texttt{Child}, \texttt{Woman}\}, \{\texttt{Child}, \texttt{Woman}, \texttt{Person}\}, \{\texttt{Child}, \texttt{Woman}, \texttt{Girl}\},\\
&\{\texttt{Child}, \texttt{Woman}, \texttt{Person}, \texttt{Girl}\}\}.
\end{aligned}$$

Please note that the same signature can belong to the label of more than one atom. In general, given an ontology $\mathcal{O}$, the total number of possible labels is 2^m where $m = \#\widetilde{\mathcal{O}}$. Hence listing all the labels λ that lead an atom to be included in the module $\mathcal{M}_\Sigma$ for $\Sigma = \lambda$ is unfeasible.

Before discussing specific choices for labels, we define a general notion of these.

Definition 5.1.2. Given an ontology $\mathcal{O}$, a *label* is a set $\lambda = \{\sigma_1, \ldots, \sigma_\ell\}$ of signatures $\sigma_i \in \wp(\widetilde{\mathcal{O}})$. The set of all labels will be denoted by the symbol Λ. A *labelling function* is a mapping $\mathsf{Lab} : \mathcal{A}(\mathcal{O}) \to \Lambda$ that maps each atom to its label. A *Labelled Atomic Decomposition* (*LAD*) for an ontology $\mathcal{O}$ is a triple $(\mathcal{A}(\mathcal{O}), \succ, \mathsf{Lab})$, where $(\mathcal{A}(\mathcal{O}), \succ)$ is an AD of $\mathcal{O}$, and $\mathsf{Lab} : \mathcal{A}(\mathcal{O}) \to \Lambda$ is a labelling function.

In the remainder of this chapter we discuss different options for the labelling function Lab and their applications.

5.2 LADs with Minimal Seed Signatures

In Example 5.1.1 we have shown a kind of labelling that can be easily used to determine which atoms are included in the module for a given signature Σ; hence

in principle a procedure for extracting the module $\mathrm{mod}(\Sigma, \mathcal{O})$ for a signature Σ consists of going through each atom $\mathfrak{a}$ and checking whether $\mathtt{Lab}(\mathfrak{a}) \ni \Sigma$. The labelling proposed, however, is clearly infeasible since it could involve 2^m signatures, m being the number of terms in $\widetilde{\mathcal{O}}$. However, it is obvious that the labels in Example 5.1.1 contain two kinds of redundancies:

1. We know that, if an atom $\mathfrak{a}$ is contained in a module $\mathcal{M}$, then all the atoms that $\mathfrak{a}$ depends on are also contained in $\mathcal{M}$. In the example we did not exploit the dependency structure of the LAD, so every atom contains in its label the signatures $\{\mathtt{Child}, \mathtt{Woman}\}$, $\{\mathtt{Child}, \mathtt{Woman}, \mathtt{Person}\}$, $\{\mathtt{Child}, \mathtt{Woman}, \mathtt{Girl}\}$, and $\{\mathtt{Child}, \mathtt{Woman}, \mathtt{Person}, \mathtt{Girl}\}$. However, for the atoms $\mathfrak{a}_1$ and $\mathfrak{a}_2$ this information is redundant since, whenever $\mathfrak{a}_3$ is contained in a module $\mathcal{M}$, then the dependency relation $\succ$ already reveals that also $\mathfrak{a}_1$ and $\mathfrak{a}_2$ are contained in $\mathcal{M}$.

2. At a closer inspection, we notice that, if an atom's label λ contains the signature $\sigma \subseteq \{\mathtt{Child}, \mathtt{Woman}\}$, then λ contains also the 3 labels $\sigma \cup \{\mathtt{Person}\}$, $\sigma \cup \{\mathtt{Girl}\}$, and $\sigma \cup \{\mathtt{Person}, \mathtt{Girl}\}$. In other words, the inclusion of either `Person` or `Girl` in a seed signature Σ does not change the result of the module extraction. In general, we can exploit the monotonicity over the signatures Σ as in Definition 2.3.1 to discard those terms from Σ, and include an atom $\mathfrak{a}$ in a module $\mathcal{M}_\Sigma = \mathrm{mod}(\Sigma, \mathcal{O})$ whenever its label $\lambda = \mathtt{Lab}(\mathfrak{a})$ contains a signature $\sigma \subseteq \Sigma$.

The observations just made allow us to shorten the labels in Example 5.1.1, so that the labelling function is defined as follows:

$$\mathfrak{a}_1 \mapsto \{\mathtt{Woman}\}$$
$$\mathfrak{a}_2 \mapsto \{\mathtt{Child}\}$$
$$\mathfrak{a}_3 \mapsto \{\mathtt{Child}, \mathtt{Woman}\}$$

Definition 5.2.1. Let $\mathcal{M}$ be a module in an ontology $\mathcal{O}$. We define a *minimal seed signature* for $\mathcal{M}$ to be a signature Σ such that $\mathcal{M} = \mathrm{mod}(\Sigma, \mathcal{O})$ and that is minimal w.r.t. set inclusion, i.e., there is no $\Sigma' \subsetneq \Sigma$ such that $\mathcal{M} = \mathrm{mod}(\Sigma', \mathcal{O})$. We denote the set of all minimal seed signatures of a module by $\mathtt{mssig}(\mathcal{M}, \mathcal{O})$. We call an atom $\mathfrak{a}$ *relevant to a signature* Σ if there exists $\Sigma' \in \mathtt{mssig}(\downarrow \mathfrak{a}, \mathcal{O})$ such that $\Sigma' \subseteq \Sigma$.

Please note that, in Definition 5.2.1, an atom is relevant to a signature, and not to a seed signature. Indeed an atom can belong to the module $\mathcal{M}_\Sigma = \text{mod}(\Sigma, \mathcal{O})$ even if it is not relevant to Σ, as the next example shows.

Example 5.2.2. Let us consider the ontology Apart defined as follows:

$$\{\alpha_1 : \mathtt{A} \sqsubseteq \mathtt{B} \sqcap \mathtt{C},$$
$$\alpha_2 : \mathtt{C} \sqcap \mathtt{Z} \sqsubseteq \bot,$$
$$\alpha_3 : \mathtt{X} \sqsubseteq \mathtt{Y} \sqcap \mathtt{Z}\}.$$

The $\bot$-LAD of Apart consists of three independent atoms containing one axiom each, and it is represented in Figure 5.1.

Figure 5.1: $\bot$-LAD of the ontology Apart

Let us now consider the signature $\Sigma = \{\mathtt{A}, \mathtt{X}\}$. Clearly, the two external atoms are relevant to Σ. However, $\text{mod}(\Sigma, \mathsf{Apart}) = \mathsf{Apart}$. Indeed, α_2 is needed to infer that $\mathtt{A} \sqsubseteq \neg\mathtt{X}$ which is an entailment of Apart over the seed signature Σ.

By looking closer at the process of extracting a module from a signature, we can see that during the extraction the seed signature Σ is enlarged with the terms that do not belong to Σ and that occur in those axioms that are non-local w.r.t. Σ to ensure that the module is self-contained. The alternation between checking the axioms against locality and enlarging the signature occurs until a fixpoint is reached, i.e., until Σ is not enlarged anymore. This procedure is monotonic only for $\bot$- and $\top$-modules. We can hence define another procedure, where the check of axioms against non-locality is substituted by the check for relevance of an atom against a signature. Then, by applying this procedure for $\bot$- or $\top$-locality, we obtain the module for the signature Σ, as expressed and proved in the next result.

Theorem 5.2.3. *Let $x \in \{\bot, \top\}$ and Σ_0 the input signature. Let us consider $\mathcal{M}_0^x = \{\alpha \in\downarrow \mathfrak{a} \,|\, \mathfrak{a}$ is relevant for $\Sigma_0\}$ and, for $i \geq 1$, $\mathcal{M}_i^x = \{\alpha \in\downarrow \mathfrak{a} \,|\, \mathfrak{a}$ is relevant for $\widetilde{\mathcal{M}_{i-1}^x} \cup \Sigma_0\}$. Then, the chain of inclusions $\mathcal{M}_0^x \subsetneq \mathcal{M}_1^x \subsetneq \ldots$ eventually stops, and for $\mathcal{M}_*^x$ the fixpoint, we have that $\mathcal{M}_*^x = x\text{-mod}(\Sigma_0, \mathcal{O})$.*

Proof. We prove the inclusions $\mathcal{M}_*^x \subseteq x\text{-mod}(\Sigma_0, \mathcal{O})$ and $\mathcal{M}_*^x \supseteq x\text{-mod}(\Sigma_0, \mathcal{O})$.

($\subseteq$) By induction over i we show that $\mathcal{M}_i^x \subseteq x\text{-mod}(\Sigma_0, \mathcal{O})$ for all $i \geq 0$.

For $i = 0$ we want to prove that $\mathcal{M}_0^x \subseteq x\text{-mod}(\Sigma_0, \mathcal{O})$:
$\mathcal{M}_0^x = \{\alpha \in \downarrow \mathfrak{a} \,|\, \mathfrak{a} \text{ is relevant for } \Sigma_0\}$; this means that there exists $\Sigma_\mathfrak{a} \in x\text{-}\mathtt{mssig}(\downarrow \mathfrak{a}, \mathcal{O})$ such that $\Sigma_\mathfrak{a} \subseteq \Sigma_0$; hence, by monotonicity we have that $\downarrow \mathfrak{a} \subseteq x\text{-mod}(\Sigma_0, \mathcal{O})$. Since this is true for all atoms that are relevant for Σ_0, we have that $\mathcal{M}_0^x \subseteq x\text{-mod}(\Sigma_0, \mathcal{O})$.

Let us now suppose $\mathcal{M}_i^x \subseteq x\text{-mod}(\Sigma_0, \mathcal{O})$. Then, by self-containment we have that $x\text{-mod}(\Sigma_0, \mathcal{O}) = x\text{-mod}(\widetilde{\mathcal{M}}_i^x \cup \Sigma_0, \mathcal{O})$. Moreover, $\mathcal{M}_{i+1}^x \subseteq x\text{-mod}(\widetilde{\mathcal{M}}_i^x \cup \Sigma_0, \mathcal{O})$ for reasons as above: by definition, for each atom $\mathfrak{a} \subseteq \mathcal{M}_{i+1}^x$ there exists a seed signature $\Sigma_\mathfrak{a}$ such that $\Sigma_\mathfrak{a} \subseteq \widetilde{\mathcal{M}}_i^x \cup \Sigma_0$. Then, by monotonicity we have that the module $\downarrow \mathfrak{a}$ is contained in $x\text{-mod}(\widetilde{\mathcal{M}}_i^x \cup \Sigma_0, \mathcal{O})$. Since this is true for all such atoms, then $\mathcal{M}_{i+1}^x \subseteq x\text{-mod}(\widetilde{\mathcal{M}}_i^x \cup \Sigma_0, \mathcal{O}) = x\text{-mod}(\Sigma_0, \mathcal{O})$.

Since the ontology is finite, this process eventually stops, and we find a fixpoint $\mathcal{M}_*^x \subseteq x\text{-mod}(\Sigma_0, \mathcal{O})$.

($\supseteq$) Let us describe the standard process of extracting a module as follows:
Let $\mathcal{N}_0^x$ be the set of all non-local axioms w.r.t. Σ_0 in $\mathcal{O}$. Then, the module $x\text{-mod}(\Sigma_0, \mathcal{O})$ is obtained by recursively extracting $\mathcal{N}_j^x = \{\alpha \in \mathcal{O} \,|\, \alpha \text{ is non-local w.r.t. } \widetilde{\mathcal{N}}_{j-1}^x \cup \Sigma_0\}$, until the fixpoint $\mathcal{N}_*^x$ is reached. Note that $\mathcal{N}_j^x \subseteq \mathcal{N}_{j+1}^x$, and that the fixpoint is the module $x\text{-mod}(\Sigma_0, \mathcal{O})$. It therefore suffices to show that $\mathcal{N}_j^x \subseteq \mathcal{M}_*^x$.

For this purpose, we prove by induction on j that, for each $j \geq 0$, there exists an $i \leq j$ such that $\mathcal{N}_j^x \subseteq \mathcal{M}_i^x$. We will use the following claim.

Claim($*$): if an axiom α is non-local w.r.t. a signature Σ, then its atom $\mathfrak{a}_\alpha$ is relevant for Σ.
This claim clearly follows from the definition of atoms and relevance of an atom for a signature, and from the fact that a genuine module is an α-module.

For $j = 0$, we have that $\mathcal{N}_0^x$ is the set of all non-local axioms w.r.t. Σ_0. Because of ($*$), for each axiom $\alpha \in \mathcal{N}_0^x$ we have that its atom is relevant to Σ_0, hence it is contained in $\mathcal{M}_0^x$.

Now let $j \geq 1$; we want to prove that, if there exists an $i \leq j$ such that $\mathcal{N}_j^x \subseteq \mathcal{M}_i^x$, then $\mathcal{N}_{j+1}^x \subseteq \mathcal{M}_{i+1}^x$ or $\mathcal{N}_{j+1}^x \subseteq \mathcal{M}_i^x$.

$\mathcal{N}^x_{j+1}$ is the set of all axioms that are non-local w.r.t. $\widetilde{\mathcal{N}}^x_j \cup \Sigma_0$. Since $\mathcal{N}^x_j \subseteq \mathcal{M}^x_i$ by inductive hypothesis, it follows that $\mathcal{N}^x_{j+1}$ is contained in the set of all axioms that are non-local w.r.t. $\widetilde{\mathcal{M}}^x_i \cup \Sigma_0$. Then, this set is contained in the set of all atoms relevant for $\widetilde{\mathcal{M}}^x_i \cup \Sigma_0$, i.e., $\mathcal{M}^x_{i+1}$. Because of the inclusion just obtained, we have that $\mathcal{N}^x_{j+1}$ can be already included into $\mathcal{M}^x_i$. □

As we already mentioned, the procedure described in Theorem 5.2.3 computes x-mod$(\Sigma, \mathcal{O})$ only for $x \in \{\top, \bot\}$. A counterexample for $x = \top\bot^*$ is provided in what follows.

Example 5.2.4. Let us consider an ontology Chain_n for a fixed $n > 2$ as defined in Example 2.3.9. Then, the $\top\bot^*$-ADs of this ontologies consist of n pairwise independent atoms. However, for each choice of two terms $\mathtt{A}_\kappa, \mathtt{A}_\ell$ with $\kappa < \ell$, the module for the seed signature $\Sigma = \{\mathtt{A}_\kappa, \mathtt{A}_\ell\}$ is the set $\top\bot^*$-mod$(\Sigma, \mathcal{O}) = \{\mathtt{A}_\kappa \sqsubseteq \mathtt{A}_{\kappa+1}, \ldots, \mathtt{A}_{\ell-1} \sqsubseteq \mathtt{A}_\ell\}$. The reason for this behaviour can be found in the extraction procedure: at each iteration i of the outer loop for the extraction of a $\bot$- or of a $\top$-module, either the set $\mathcal{M}^x_i$ grows, or the algorithm stops; in contrast, during the computation of a $\top\bot^*$-module, the intermediate sets of axioms can decrease in size when the procedure goes from one notion (either $\bot$ or $\top$) to the other.

Theorem 5.2.3 suggests a procedure for extracting modules from an ontology $\mathcal{O}$ without the need for loading $\mathcal{O}$ into memory, i.e., by *offline* performing a module extraction. Moreover, we see that the procedure described in the proof does not individually check all the axiom in $\mathcal{O}$ for locality against a signature Σ; quite differently, if the procedure finds that an atom $\mathfrak{a}$ is relevant for Σ, then all the axioms in $\downarrow \mathfrak{a}$ fall into the module for Σ. In particular, the offline extraction of modules is a promising technique from a performance point of view provided that the labels are not too large: recall that a genuine module $\mathcal{M}$ can still have exponentially many labels in the size of $\widetilde{\mathcal{M}}$, as described in the following example.

Example 5.2.5. Let us consider the the family of ontologies:

$$\begin{aligned}\mathcal{O}_n = \{&\alpha_i : \mathtt{A}_i \equiv \mathtt{A}_{i-1} \sqcup \mathtt{A}'_{i-1},\\ &\beta_i : \mathtt{B}_i \equiv \mathtt{B}_{i-1} \sqcup \mathtt{B}'_{i-1},\\ &\gamma_i : \mathtt{C}_i \equiv \mathtt{C}_{i-1} \sqcup \mathtt{C}'_{i-1},\\ &\delta_i : \mathtt{D}_i \equiv \mathtt{D}_{i-1} \sqcup \mathtt{D}'_{i-1},\\ &\eta_i : (\mathtt{A}_i \sqcup \mathtt{B}_i) \sqcap (\mathtt{C}_i \sqcup \mathtt{D}_i) \sqsubseteq \mathtt{X} \mid i = 1, \ldots, n\}.\end{aligned}$$

Then, $\#\mathcal{O}_n = 5n$, $\#\widetilde{\mathcal{O}}_n = 8n + 5$, the $\bot$-AD of each $\mathcal{O}_n$ consists of 5 atoms, and the Hasse diagram for $(\mathcal{A}(\mathcal{O}_n), \succ)$ is represented in Figure 5.2.

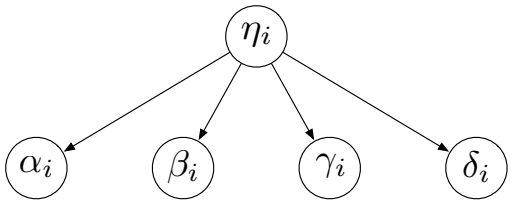


Figure 5.2: $\bot$-AD of the ontologies $\mathcal{O}_n$

Let $\mathfrak{e}$ be the atom containing the axioms of kind η. Now, any axiom η_i is non-local w.r.t. a signature Σ whenever Σ consists of one term from $\{\mathtt{A}_i, \mathtt{B}_i\}$ and one term from $\{\mathtt{C}_i, \mathtt{D}_i\}$. In particular, such a signature Σ is also a minimal seed signature for $\downarrow \mathfrak{e}$. Moreover, we obtain another minimal seed signature for $\downarrow \mathfrak{e}$ by replacing in Σ a term by on of the two terms defining it (for example, $\mathtt{A}_i$ can be replaced by $\mathtt{A}_{i-1}, \mathtt{A}'_{i-1}$). Since this procedure can be recursively applied, the module $\downarrow \mathfrak{e}$ has at least 4^n minimal seed signatures.

Even though an exponential number of minimal seed signatures for a genuine module can occur, we expect this event to be quite rare in practice, and we describe the results of an experiment that involves testing this hypothesis in Section 6.2. The algorithm described in Theorem 5.2.3 has been implemented, and an offline module extractor is currently available as a service at `http://sswap.info/modularize`. A discussion of the technical details behind the implementation is described in Section 6.2.

5.3 LADs based on Atoms' Signatures

In Subsection 4.3.2 we have introduced the chains of CEs in an ontology, and we have mentioned that the intuition behind this notion consists of the possibility of defining (many) total orderings of sets of axioms in $\mathcal{O}$ in such a way as to ease the construction of a model $\mathcal{I}$ of some genuine module $\mathcal{M} \subseteq \mathcal{O}$. Specifically, the stepwise construction consists of identifying genuine modules $\mathcal{M}_i \subseteq \mathcal{M}$ such that $\mathcal{M}_i \subsetneq \mathcal{M}_{i+1}$. At each step, then, a model $\mathcal{I}_i$ is built to interpret the terms in $\Sigma_i = \widetilde{\mathcal{M}_i}$. Please note that at each step the signature Σ_i is strictly larger than Σ_{i-1}. Hence, the total ordering of sets of axioms induces a total ordering of sets of terms as well.

A suitable LAD can represent both the total orderings (of sets of axioms vs. of sets of terms) when we use signatures as a base for the atoms' labels. In particular, we define the following labelling function:

$$\mathsf{Lab}_{sig}(\mathfrak{a}) := \widetilde{\mathfrak{a}}.$$

Please note that in this definition we abuse the notation of a labelling function since a label in this case is a signature, and not a set of signatures.

Using the atom's signature as a label helps to identify *where* in an AD the terms of $\widetilde{\mathcal{O}}$ occur. However, the intuition of using a LAD to reveal a logical dependence between *sets of terms* is hindered by the terms redundancies, as the following example shows.

Example 5.3.1. Let us consider again the ontology Chain_n for a fixed n, defined in Example ex:ModsContainHierarchies and used in Example 5.2.4. Then, its LAD $(\mathcal{A}^{\perp}(\mathsf{Chain}_n), \succ, \mathsf{Lab}_{sig})$ is represented in Figure 5.3. We can see that in this

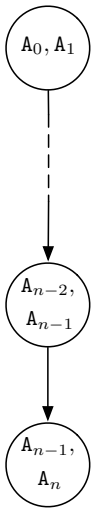


Figure 5.3: $(\mathcal{A}^{\perp}(\mathsf{Chain}_n), \succ, \mathsf{Lab}_{sig})$ of the ontology Chain_n

case any two directly comparable atoms $\mathfrak{a}_i, \mathfrak{a}_{i+1}$ contain the same term $\mathtt{A}_i$ in their labels. However, since by using the notion of chains of CEs we are "reading" the LAD from below, intuitively we see that $\mathtt{A}_i$ is *introduced* in $\mathfrak{a}_{i+1}$, whilst $\mathtt{A}_i$ is *used* in $\mathfrak{a}_i$.

The $\perp$-AD for Chain_n provides an order to the axioms such that the most general concepts $\mathtt{A}_n$ and $\mathtt{A}_{n-1}$ are introduced first, and the most specific concept $\mathtt{A}_1$ is introduced last. Following this ordering, we would like to say that the term $\mathtt{A}_i$

is defined in terms of $\mathtt{A}_{i+1}$ since it occurs on the left hand side of the subsumption axiom $\mathtt{A}_i \sqsubseteq \mathtt{A}_{i+1}$. We obtain this kind of dependency by recursively defining the following notion of labelling function applied to the $\bot$-AD.

$$\mathsf{Lab}^{\#}_{sig}(\mathfrak{a}) := \widetilde{\mathfrak{a}} \qquad \text{if } \mathfrak{a} \text{ is a minimal element in } (\mathcal{A}(\mathcal{O}), \succ)$$
$$\mathsf{Lab}^{\#}_{sig}(\mathfrak{a}) := \widetilde{\mathfrak{a}} \setminus \bigcup_{\mathfrak{b} \prec \mathfrak{a}} \mathsf{Lab}^{\#}_{sig}\mathfrak{b}.$$

Given an atom $\mathfrak{a}$, the deletion from $\mathsf{Lab}_{sig}(\mathfrak{a})$ of all the terms belonging to the labels of the atoms that $\mathfrak{a}$ depends on will generate what we call a *flat* label. The corresponding flat LAD for Chain_n is represented in Figure 5.4.

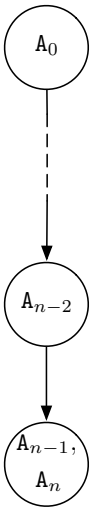


Figure 5.4: $(\mathcal{A}^{\bot}(\mathsf{Chain}_n), \succ, \mathsf{Lab}^{\#}_{sig})$ of the ontology Chain_n

Clearly, for each term $\mathtt{t}$ in the signature of a genuine module $\downarrow \mathfrak{a}$ (and hence for a general module), there is at least one atom $\mathfrak{b} \subseteq \downarrow \mathfrak{a}$ such that $\mathtt{t} \in \mathsf{Lab}^{\#}_{sig}(\mathfrak{b})$. In general, though, there can be more than one atom that contain $\mathtt{t}$. Moreover, this kind of labelling can generate also empty labels. The following is an example of an ontology where both cases just discussed occur.

Example 5.3.2. Let us consider the ontology Split defined as follows:

$$\alpha_1 : \mathtt{A} \sqcap \mathtt{B} \sqcap \mathtt{C} \sqsubseteq \mathtt{E}$$
$$\alpha_2 : \mathtt{A} \sqcap \mathtt{B} \sqsubseteq \mathtt{D}$$
$$\alpha_3 : \mathtt{A} \sqcap \mathtt{C} \sqsubseteq \mathtt{D}$$
$$\alpha_4 : \mathtt{D} \sqsubseteq \mathtt{E}$$

Then, the corresponding $\bot$-LAD is represented in Figure 5.5.

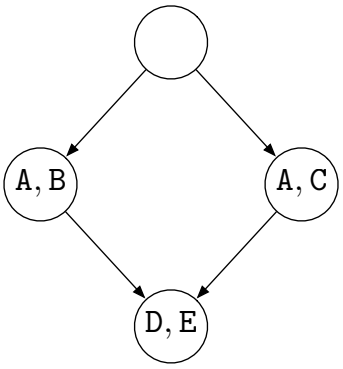


Figure 5.5: $(\mathcal{A}^{\bot}(\mathsf{Split}), \succ, \mathsf{Lab}^{\#}_{sig})$ of the ontology Split

Recall the notion of chain of CEs described in Subsection 4.3.2: we mentioned that this notion allows a stepwise construction of a model for a genuine module $\mathcal{M}$ by extending an interpretation $\mathcal{I}_0$ over a sequence of subontologies of $\mathcal{M}$. Then, a LAD provides information on the signatures over which the interpretations $\mathcal{I}_1, \mathcal{I}_2, \ldots$ are defined.

As an example, we build a model for the ontology Split defined in Example 5.3.2. We start from the minimal element in the Hasse diagram represented in Figure 5.5, and consider the chain of CEs consisting of the following modules: $\{\alpha_4\} \subsetneq \{\alpha_4, \alpha_2\} \subsetneq \mathsf{Split}$. Please note that chain of CEs is built according to Corollary 4.3.16. In particular, the chain of subsets $\{\alpha_4\} \subsetneq \{\alpha_4, \alpha_2, \alpha_3\} \subsetneq \mathsf{Split}$ is not a chain of CEs since we know that at each step in a chain of CEs the signature Σ_i increases, and in this case it does not.

First we build a model $\mathcal{I}_0$ for α_4 over the signature $\{\mathtt{D}, \mathtt{E}\}$; for example we define $\mathtt{D}^{\mathcal{I}_0} = \{d\}$ and $\mathtt{E}^{\mathcal{I}_0} = \Delta^{\mathcal{I}_0} = \{a, b, c, d, e\}$. Then, we extend $\mathcal{I}_0$ to a model $\mathcal{I}_1$ of $\{\alpha_4, \alpha_2\}$ by setting $\mathcal{I}_1|_{\{\mathtt{D},\mathtt{E}\}} := \mathcal{I}_0$, and by defining $\mathtt{A}^{\mathcal{I}_1} = \{a, d, e\}$ and $\mathtt{B}^{\mathcal{I}_1} = \{b, d\}$. Please note that since $\mathtt{D}^{\mathcal{I}_0} = \{d\}$ we have to pay attention that $\mathtt{A}^{\mathcal{I}_1}$ and $\mathtt{B}^{\mathcal{I}_1}$ do not contain both a same element beside d. Hence, we have the guarantee that at least one interpretation $\mathcal{I}'_1$ of $\{\mathtt{A}, \mathtt{B}\}$ exists such that $\mathcal{I}_1 = \mathcal{I}_0 \times \mathcal{I}'_1 \models \{\alpha_4, \alpha_2\}$. However, all such interpretations $\mathcal{I}'_1$ are *constrained* by the interpretations we have fixed in the previous step, i.e., there exists a model $\mathcal{J}$ of Split such that $\mathcal{I}_0 \times \mathcal{J}|_{\{\mathtt{A},\mathtt{B}\}}$ is not a model of $\{\alpha_4, \alpha_2\}$.

Finally, we complete the model construction for Split as follows: we see that the last extension in the chain of CEs pulls two more atoms in the module, i.e., $\mathfrak{a}_1 = \{\alpha_1\}$ and $\mathfrak{a}_3 = \{\alpha_3\}$, and that $\mathsf{Lab}^{\#}_{sig}(\mathfrak{a}_1) = \emptyset$. This means that the

interpretation of the terms in $\mathfrak{a}_1$ is induced by defining a model for the subsets of $\downarrow \mathfrak{a}_1$. Going down to the "unexplored" atom $\mathfrak{a}_3 = \{\alpha_3\}$ we see that $\mathfrak{a}_3$'s label contains $\mathtt{A}$, that has already been interpreted in $\mathcal{I}_1$, and $\mathtt{C}$, which is a fresh term. We can interpret $\mathtt{C}$ by defining a model $\mathcal{I}_2$ of $\{\alpha_4, \alpha_3\}$ *that expands* $\mathcal{I}_1$: for example, we set $\mathcal{I}_2|_{\{\mathtt{A},\mathtt{B},\mathtt{D},\mathtt{E}\}} := \mathcal{I}_1$, and then define $\mathtt{C}^{\mathcal{I}_2} = \{c, d\}$ such that the resulting interpretation $\mathcal{I}_2$ is a model for the whole ontology Split.

The model construction just described shows that a chain of CEs, which is a total orderings of *sets of axioms* of an ontology $\mathcal{O}$, induces also a total ordering of *set of terms* of $\mathcal{O}$. This observation has motivated us to explore what it means, in a chain of CEs, for a term to be defined in terms [*sic*] of other terms. Intuitively, this notion can be used to support the understanding of $\mathcal{O}$, as shown in the following example.

Example 5.3.3. Let us consider the ontology Teetotaller defined as follows:

$$
\begin{aligned}
\{\alpha_1 &: \mathtt{Student} \sqsubseteq \mathtt{Person} \sqcap \exists \mathtt{hasHabitat}.\mathtt{University},\\
\alpha_2 &: \mathtt{Person} \sqsubseteq \mathtt{Animal},\\
\alpha_3 &: \mathtt{Vegan} \equiv \mathtt{Person} \sqcap \forall \mathtt{eats}.(\mathtt{Vegetable} \sqcup \mathtt{Mushroom}),\\
\alpha_4 &: \mathtt{Animal} \sqsubseteq (= 1 \mathtt{hasGender}.\top),\\
\alpha_5 &: \mathtt{TeeTotaller} \equiv \mathtt{Person} \sqcap \forall \mathtt{drinks}.\mathtt{NonAlcoholicThing},\\
\alpha_6 &: \mathtt{GraduateStudent} \equiv \mathtt{Student} \sqcap \exists \mathtt{hasDegree}.\{BA, BS\},\\
\alpha_7 &: \mathtt{Animal} \sqsubseteq (\geq 1 \mathtt{hasHabitat}.\top),\\
\alpha_8 &: \mathtt{Car} \sqsubseteq \mathtt{Vehicle}\}
\end{aligned}
$$

Then, the $\bot$-LAD of Teetotaller defined under the labelling function $\mathtt{Lab}^{\#}_{sig}$ is represented in Figure 5.6.

Hence, we can first fix the meaning for `Animal` and `hasHabitat`, and then progressively the meaning for `Person` in terms of `Animal` and `hasHabitat`, the meaning for `Student` and `University` in terms of `Person`, and so on.

In the remainder of this chapter we will discuss some other notions that can be exploited to support ontology comprehension.

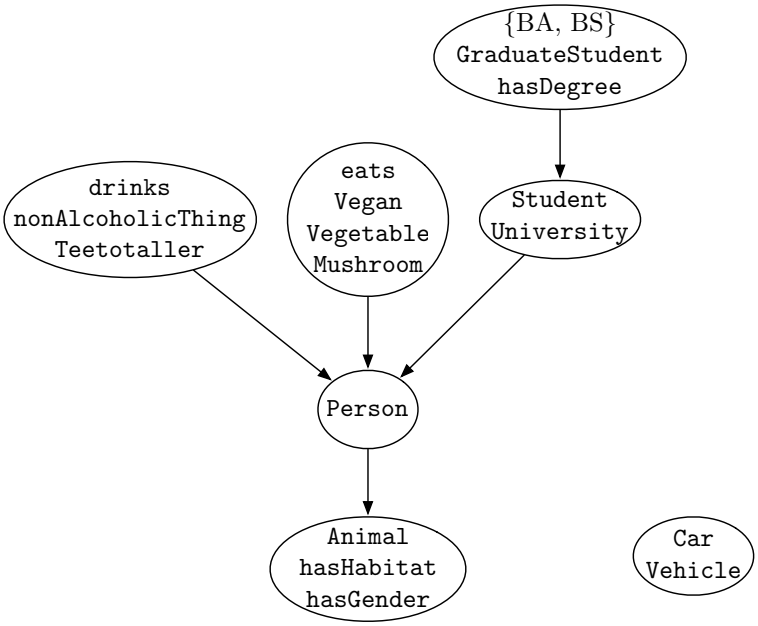


Figure 5.6: $\bot$-LAD of the ontology Teetotaller

5.4 Comparing LADs

In the previous two sections we have introduced two substantially different labelling functions for the AD of an ontology: the first one reveals the signatures whose terms are non trivially interrelated by the ontology $\mathcal{O}$. The second, instead, determines an order in which the terms in $\widetilde{\mathcal{O}}$ can be fixed (please note that in general the terms' order is not uniquely determined). These two notions, i.e., interrelating vs. mentioning, can be, in general, orthogonal to each other, as described in the two following examples: in the first, an axiom mentions a term $\mathtt{t}$, but it is irrelevant for $\mathtt{t}$'s meaning. In the second, an axiom β does not mention a term $\mathtt{t}$ at all, but it clearly says something about $\mathtt{t}$.

Example 5.4.1. Let us consider the ontology $\mathcal{O}$ consisting of the following axiom alone:

$$\alpha : \mathtt{A} \sqsubseteq \mathtt{B} \sqcap (\mathtt{C} \sqcup \neg \mathtt{C}).$$

Clearly, α does not say anything about $\mathtt{C}$, despite the fact that $\mathtt{C} \in \widetilde{\alpha}$. Hence a model for $\mathcal{O}$ will need to interpret $\mathtt{C}$, and $\mathsf{Lab}^{\#}_{sig}(\{\alpha\})$ contains the term $\mathtt{C}$, although under any interpretation of $\mathtt{C}$ the concept $\mathtt{C} \sqcup \neg \mathtt{C}$ has the same extension of $\top$. Quite differently, for each notion of locality x, there is no minimal signature Σ such that α is non x-local w.r.t. Σ with $\mathtt{C} \in \Sigma$. In particular, $\mathtt{C} \notin \mathsf{Lab}_{mss}(\{\alpha\})$ for

every notion x of locality.

Example 5.4.2. Let us consider the ontology Lambda defined as follows:

$$\alpha_1 : \mathtt{A} \sqsubseteq \neg \mathtt{B},$$
$$\alpha_2 : \mathtt{A} \equiv \mathtt{C},$$
$$\alpha_3 : \mathtt{B} \equiv \mathtt{D}.$$

Each atom consists of a single axiom, and Lambda equals the principal ideal of the atom $\mathfrak{a}_1$ consisting of the axiom α_1 that depends on both the atom $\mathfrak{a}_2 = \{\alpha_2\}$ and the atom $\mathfrak{a}_3 = \{\alpha_3\}$. In Figure 5.7 we show the two $\bot$-LADs of Lambda, on the left the one whose labels are defined by the labelling function $\mathsf{Lab}^{\#}_{sig}$, and on the right the one whose labels are defined by the labelling function Lab_{mss}.

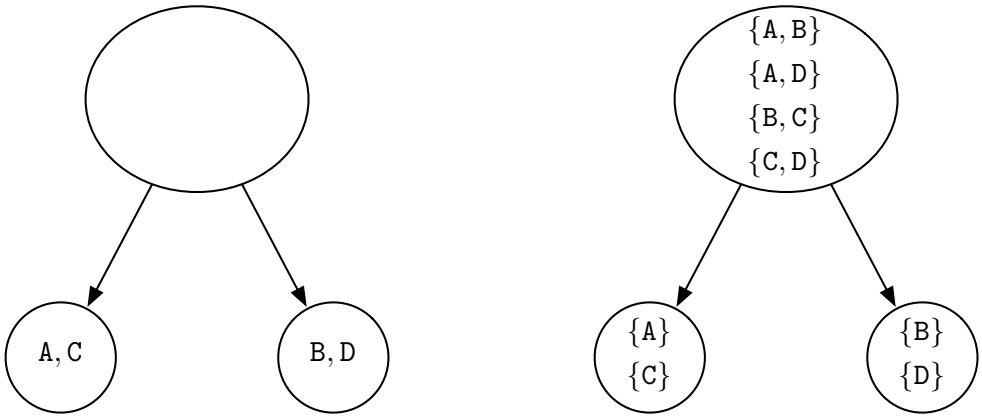


Figure 5.7: $(\mathcal{A}(\mathcal{O}), \succ, \mathsf{Lab}^{\#}_{sig})$ vs. $(\mathcal{A}(\mathcal{O}), \succ, \mathsf{Lab}_{mss})$ of the ontology Lambda

We notice that $\mathsf{Lab}^{\#}_{sig}(\mathfrak{a}_1)$ is empty, whilst $\mathsf{Lab}_{mss}(\mathfrak{a}_1)$ is made of signatures that cover the ontology's signature, in particular also the terms C and D that do not occur at all in $\widetilde{\mathfrak{a}}_1$. Hence, we realize that the atom $\mathfrak{a}_1$ has a strong logical influence on the terms C and D even though α_1 does not mention them. The need of this axiom for the signature $\Sigma = \{\mathtt{C}, \mathtt{D}\}$ is not evident from $\mathsf{Lab}^{\#}_{sig}$.

The intuition is clear: the minimal seed signatures reveal which terms are really constrained by the set of axioms in an atom, and the comparison between the two LADs $(\mathcal{A}(\mathcal{O}), \succ, \mathsf{Lab}_{mss})$ and $(\mathcal{A}(\mathcal{O}), \succ, \mathsf{Lab}^{\#}_{sig})$ can highlight the differences in the occurring terms vs. the constrained terms, hence it has promising properties to be used for detecting modelling errors. In the next section we are going to formalize the notion of constrained term either by a single axiom, or by a set of axioms, i.e. an ontology.

5.5 Model-theoretic Relevance

While the signature of an ontology describes the entities that the ontology deals with, it does not say how these objects are related. The relationship between terms is defined by the *axioms* of the ontology, that constrain which interpretations are allowed, and which are not. In particular, the interpretations of two distinct terms can be conflicting only if there is a set of axioms that does not allow any model to coincide with those interpretations over the two terms. From this perspective, the natural choice is to investigate the notion of relevance of an axiom to a term, rather than relevance between terms.

First, we are going to introduce some notions that we are going to use throughout the investigation of logical relevance.

Definition 5.5.1. Given a consistent ontology $\mathcal{O}$, a set Δ, and a signature $\Sigma \subseteq \widetilde{\mathcal{O}}$, we define a Σ-*model over* Δ *for* $\mathcal{O}$ to be an interpretation $(\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ over Σ such that there exists a model $(\Delta^{\mathcal{I}}, \cdot^{\mathcal{J}})$ for $\mathcal{O}$ whose Σ-reduct $\mathcal{J}|_{\Sigma} = \mathcal{I}$. In this case, we say that $\mathcal{I}$ *is expandable* to a model $\mathcal{J}$ for $\mathcal{O}$, and any such $\mathcal{J}$ is called an $\mathcal{O}$-*expansion of* $\mathcal{I}$.

If $\mathcal{O}$ is clear from the context, we simply drop it and say Σ-model. Please note that Definition 5.5.1 is also valid in the case of $\mathcal{O}$ being a single axiom.

Example 5.5.2. Let Δ be the set $\{a, m, f\}$, and $\mathcal{O}$ be the ontology:

$$\{\texttt{Child} \equiv (=1\,\texttt{hasMother}.\texttt{Mother}),$$
$$\texttt{Child} \equiv (=1\,\texttt{hasFather}.\texttt{Father})\}.$$

Please note that $\mathcal{O}$ is a subset of the ontology Child as in Example 3.1.4. Then, any interpretation of the singleton signature $\{\texttt{Child}\}$ is a Σ-model for $\mathcal{O}$. In contrast, the interpretation function $\mathcal{I}$ over the set $\Sigma' = \{\texttt{Child}, \texttt{hasMother}\}$ defined as $\texttt{Child}^{\mathcal{I}} = \{a\}$ and $\texttt{hasMother}^{\mathcal{I}} = \{(a, m), (a, f)\}$ is not a Σ'-model for $\mathcal{O}$ since a Mother is required to be unique for each Child.

Another classical notion that we will make use of is defined in what follows.

Definition 5.5.3. Let $\mathcal{I}$ be an interpretation, $\texttt{t}$ be a term, and Σ be a signature. A $\{\texttt{t}\}$-*variant* of $\mathcal{I}$ is an interpretation $\mathcal{J}$ such that, for each symbol $\texttt{s} \in \widetilde{\mathcal{O}} \setminus \{\texttt{t}\}$, we have $\texttt{s}^{\mathcal{I}} = \texttt{s}^{\mathcal{J}}$ and $\texttt{t}^{\mathcal{I}} \neq \texttt{t}^{\mathcal{J}}$. A Σ-*variant* of $\mathcal{I}$ is an interpretation $\mathcal{J}$ such that, for each symbol $\texttt{s} \in \widetilde{\mathcal{O}} \setminus \Sigma$, we have that $\texttt{s}^{\mathcal{I}} = \texttt{s}^{\mathcal{J}}$ and that there exists at least one term $\texttt{t} \in \Sigma$ such that $\texttt{t}^{\mathcal{I}} \neq \texttt{t}^{\mathcal{J}}$.

A first very basic notion of relevance has been introduced in Example 5.4.1 where we consider the axiom $\alpha = \mathtt{A} \sqsubseteq \mathtt{B} \sqcap (\mathtt{C} \sqcup \neg\mathtt{C})$. Then, in order for an interpretation $\mathcal{I}$ to be a model of α, it needs to satisfy the relation $\mathtt{A}^{\mathcal{I}} \subseteq \mathtt{B}^{\mathcal{I}}$. In other words, the acceptable interpretations of both $\mathtt{A}$ and of $\mathtt{B}$ are constrained. On the contrary, the interpretation of $\mathtt{C}$ is not constrained by α, and we can even rewrite the axiom into the equivalent $\mathtt{A} \sqsubseteq \mathtt{B}$ that does not even mention $\mathtt{C}$. This case of relevance provided by a single axiom is described in the next definition.

Definition 5.5.4. An axiom α is *directly relevant* to a term $\mathtt{t}$ if there exists a model $\mathcal{I}$ of α and a $\{\mathtt{t}\}$-variant $\mathcal{I}'$ of $\mathcal{I}$ such that $\mathcal{I}' \not\models \alpha$. We define also:
- $\gamma(\alpha)$ to be the *set of terms directly constrained by* α, that is all terms $\mathtt{t} \in \widetilde{\alpha}$ such that α is directly relevant to $\mathtt{t}$
- $\varphi(\alpha)$ to be the *set of free terms in* α, that is the complement of $\gamma(\alpha)$ in $\widetilde{\alpha}$.

As a straightforward consequence of Definition 5.5.4 we have that, if an axiom α does not contain a term $\mathtt{t}$ in its signature, then α is not directly relevant to it. This is true because we look at models defined over the minimum signature, as discussed in Section 2.1.

An interesting property of the terms in $\varphi(\alpha)$ is that, given any interpretation $\mathcal{I}$, then we can use interchangeably $\mathcal{I}$ or any $\varphi(\alpha)$-variant $\mathcal{I}'$ of $\mathcal{I}$: if $\mathcal{I}$ is a model for α, then $\mathcal{I}'$ is; dually, if $\mathcal{I}$ is not a model for α, then $\mathcal{I}'$ is also not a model for α.

Lemma 5.5.5. *Let α be an axiom and $\varphi(\alpha)$ be the set of all free terms in α. Then, for any model $\mathcal{I}$ for α and any set $\Sigma \subseteq \varphi(\alpha)$, we have that any Σ-variant $\mathcal{J}$ of $\mathcal{I}$ is a model for α.*

Proof. We prove this lemma by induction on the number n of the terms in Σ.
Base of the induction: $n = 1$. In this case, Σ is a singleton, so there exists $\mathtt{t} \in \varphi(\alpha)$ such that $\Sigma = \{\mathtt{t}\}$. In particular, a Σ-variant $\mathcal{J}$ of $\mathcal{I}$ is a $\{\mathtt{t}\}$-variant. Now, if $\mathcal{J}$ were not a model for α, this would contradict the assumption of $\mathtt{t}$ belonging to $\varphi(\alpha)$. Hence, $\mathcal{J} \models \alpha$.
Induction hypothesis: let us assume that if $\#\Sigma = n - 1$, then every Σ-variant $\mathcal{I}$ is still a model for α.
Induction step: let $\mathtt{t} \in \Sigma$ and $T = \Sigma \setminus \{\mathtt{t}\}$. Then, by induction hypothesis we have that any T-variant $\mathcal{J}$ of $\mathcal{I}$ is a model for α. Hence, a Σ-variant $\mathcal{I}'$ of $\mathcal{I}$ is a $\{\mathtt{t}\}$-variant of $\mathcal{J}$, and since $\mathtt{t} \in \varphi(\alpha)$ we have that $\mathcal{I}'$ is also a model for α. □

Lemma 5.5.6. *Let α be an axiom and $\varphi(\alpha)$ be the set of all free terms in α. Then, for any interpretation $\mathcal{I}$ over $\widetilde{\alpha}$ that is not a model for α and any set $\Sigma \subseteq \varphi(\alpha)$, we have that any Σ-variant $\mathcal{J}$ of $\mathcal{I}$ is not a model for α.*

Proof. Similarly to the proof for Lemma 5.5.5, we prove this lemma by induction on the number n of the terms in Σ.
Base of the induction: $n = 1$. In this case, Σ is a singleton, so there exists $\mathtt{t} \in \varphi(\alpha)$ such that $\Sigma = \{\mathtt{t}\}$. In particular, a Σ-variant $\mathcal{J}$ of $\mathcal{I}$ is a $\{\mathtt{t}\}$-variant. Now, if $\mathcal{J}$ were a model for α, this would contradict the assumption of $\mathtt{t}$ belonging to $\varphi(\alpha)$. Hence, $\mathcal{J} \models \alpha$.
Induction hypothesis: let us assume that if $\#\Sigma = n - 1$, then every Σ-variant $\mathcal{I}$ is still not a model for α.
Induction step: let $\mathtt{t} \in \Sigma$ and $T = \Sigma \setminus \{\mathtt{t}\}$. Then, by induction hypothesis we have that any T-variant $\mathcal{J}$ of $\mathcal{I}$ is not a model for α. Hence, a Σ-variant $\mathcal{I}'$ of $\mathcal{I}$ is a $\{\mathtt{t}\}$-variant of $\mathcal{J}$, and since $\mathtt{t} \in \varphi(\alpha)$ we have that $\mathcal{I}'$ is also not a model for α. □

Next, we want to find connections between the notion of direct relevance and the notion of model conservativity.

Proposition 5.5.7. *Let α be an axiom and $\mathtt{t} \in \widetilde{\alpha}$ a term. If there exists a signature $\Sigma \subseteq \widetilde{\alpha}$ such that $\Sigma \ni \mathtt{t}$, $\alpha \not\equiv^{mCE}_{\Sigma} \emptyset$, and $\alpha \equiv^{mCE}_{\Sigma\setminus\{\mathtt{t}\}} \emptyset$, then α is directly relevant to $\mathtt{t}$.*

Proof. Let α be an axiom and $\mathtt{t}$ be a term such that there exists a signature $\Sigma \subseteq \widetilde{\alpha}$ satisfying the hypothesis. Since $\alpha \not\equiv^{\mathrm{mCE}}_{\Sigma} \emptyset$, we know that there exists an interpretation $\mathcal{J}$ over Σ that cannot be expanded to a model for α, i.e., it is not an $\{\alpha\}$-model. In contrast, since $\alpha \equiv^{\mathrm{mCE}}_{\Sigma\setminus\{\mathtt{t}\}} \emptyset$, we have that $\mathcal{J}|_{\Sigma\setminus\{\mathtt{t}\}}$ can be expanded to a model $\mathcal{I}$ for α, and is indeed an $\{\alpha\}$-model.

Let us now consider the interpretation $\mathcal{J}'$ that interprets all symbols in $\alpha \setminus \{\mathtt{t}\}$ as $\mathcal{I}$ does, whilst it interprets $\mathtt{t}$ as $\mathcal{J}$ does. Then, $\mathcal{J}'$ is not a model for α, and it is a $\{\mathtt{t}\}$-variant for $\mathcal{I}$. Hence, α is directly relevant to $\mathtt{t}$. □

The inverse implication in Proposition 5.5.7 does not hold in general. For example, let us consider the $\mathcal{ALC}$ axiom $\top \sqsubseteq (\exists \mathtt{r}.\mathtt{A} \sqcap \exists \mathtt{r}.\neg \mathtt{A}) \sqcap \mathtt{B}$. Then, $\alpha \not\equiv^{mCE}_{\{\mathtt{B}\}} \emptyset$ since in any model $\mathcal{I}$ for α, the set $\mathtt{B}^{\mathcal{I}}$ contains at least two distinct elements, hence also the domain $\Delta^{\mathcal{I}}$ is forced to have at least two distinct elements. On the

contrary, if the ontology is empty, then there are models $\mathcal{J}$ such that the domain $\Delta^{\mathcal{J}}$ is a singleton, hence $\alpha \not\equiv_{\emptyset}^{mCE} \emptyset$ too.

Direct relevance of an axiom is not the only way for a term to be affected. The interpretations allowed for a term can also depend on axioms that do not even mention it, as discussed in the following example.

Example 5.5.8. Let us consider the ontology $\mathcal{O} = \{\alpha_i\}_{i=1\ldots n}$ where the i-th axiom is $\mathtt{A}_{i-1} \sqsubseteq \mathtt{A}_i$. If $n \geq 2$ then the axiom α_n does not contain $\mathtt{A}_0$. However, α_n does *indirectly* constrain $\mathtt{A}_0$ because in any model $\mathcal{I}$ of $\mathcal{O}$ where $\mathtt{A}_{n-1}^{\mathcal{I}} = \emptyset$ (and α_n is then satisfied for any interpretation of $\mathtt{A}_n$) we have that $\mathtt{A}_0^{\mathcal{I}}$ is forced to be empty.

In other words, in order to define relevance between a term and an axiom we need then to look at their interpretations *in the context of the ontology*. Intuitively, an axiom α is relevant to $\mathtt{t}$ w.r.t. the ontology $\mathcal{O}$ if the interpretation of $\mathtt{t}$ and the interpretation of the constrained terms in $\widetilde{\alpha} \setminus \mathtt{t}$ cannot be chosen independently from each other. In what follows we formalise and discuss this idea.

Definition 5.5.9. Let Δ be a set, $\mathcal{O}$ an ontology, $\mathtt{t} \in \widetilde{\mathcal{O}}$ a term, and $\alpha \in \mathcal{O}$ an axiom. We say that α is *relevant to* $\mathtt{t}$ *in* $\mathcal{O}$ if there exist:
- a subset $\mathcal{O}' \subseteq \mathcal{O}$ such that $\alpha \in \mathcal{O}'$,
- a $\{\mathtt{t}\}$-model $\mathcal{I}_1$ over Δ for $\mathcal{O}' \setminus \alpha$
- a $\{\gamma(\alpha) \setminus \{\mathtt{t}\}\}$-model $\mathcal{I}_2$ for $\mathcal{O}'$

such that there is no model $\mathcal{J} \models \mathcal{O}'$ over Δ with:
- $\mathcal{J}|_{\mathtt{t}} = \mathcal{I}_1$, and
- $\mathcal{J}|_{\gamma(\alpha)\setminus\{\mathtt{t}\}} = \mathcal{I}_2$.

Example 5.5.10. Let us consider the ontology Apart defined in Example 5.2.2. The term $\mathtt{A}$ does not occur in $\alpha_3 : \mathtt{X} \sqsubseteq \mathtt{Y} \sqcap \mathtt{Z}$. Clearly, $\gamma(\alpha_3) = \widetilde{\alpha}$, and $\widetilde{\alpha} \setminus \mathtt{A} = \widetilde{\alpha}$.

Now, the interpretation $\mathcal{I}_1$ defined as $\mathtt{A}^{\mathcal{I}_1} = \Delta^{\mathcal{I}_1}$ is an $\{\mathtt{A}\}$-model for Apart. However, any Apart-expansion $\mathcal{J}$ of $\mathcal{I}_1$ is such that $\mathtt{X}^{\mathcal{J}} = \mathtt{Z}^{\mathcal{J}} = \emptyset$. Moreover, the interpretation $\mathcal{I}_2$ defined as $\mathtt{X}^{\mathcal{I}_2} = \mathtt{Y}^{\mathcal{I}_2} = \mathtt{Z}^{\mathcal{I}_2} = \Delta^{\mathcal{I}_2}$ is also an $\widetilde{\alpha}$-model. Hence, there is no model $\mathcal{J}$ for Apart such that $\mathcal{J}|_{\{\mathtt{A}\}} = \mathcal{I}_1$ and $\mathcal{J}|_{\widetilde{\alpha}} = \mathcal{I}_2$. Hence, α_3 is relevant to $\mathtt{A}$.

The following example shows that requiring $\mathcal{I}_2$ to be defined over $\gamma(\alpha) \setminus \mathtt{t}$ rather than over $\widetilde{\alpha} \setminus \mathtt{t}$ is fundamental for the notion of relevance to capture the model-theoretic relation between the axiom α and the term $\mathtt{t}$.

Example 5.5.11. Let us consider the following ontology:

$$\mathcal{O} = \{\alpha_1 : \mathtt{A} \sqsubseteq \mathtt{B} \sqcap (\mathtt{C} \sqcup \neg \mathtt{C})$$
$$\alpha_2 : \mathtt{C} \sqsubseteq \mathtt{D}\}.$$

Please note that $\gamma(\alpha_1) = \{\mathtt{A}, \mathtt{B}\}$. Now, for any domain Δ, any $\{\mathtt{D}\}$-model $\mathcal{I}_1$ over Δ for α_2, and any $\{\mathtt{A}, \mathtt{B}\}$-model $\mathcal{I}_2$ over Δ for $\mathcal{O}$, the interpretation $\mathcal{J} = \mathcal{I}_1 + \mathcal{I}_2$ is a model for $\mathcal{O}$ over Δ that expands both $\mathcal{I}_1$ and $\mathcal{I}_2$. Hence, α_1 is irrelevant for $\mathtt{D}$.

In contrast, for any domain Δ let us now consider the $\{\mathtt{D}\}$-model $\mathcal{I}_1$ for α_2 defined as $\mathtt{D}^{\mathcal{I}_1} = \emptyset$, and the $\widetilde{\alpha}$-model $\mathcal{I}_2$ for $\mathcal{O}$ defined as $\mathtt{A}^{\mathcal{I}_2} = \mathtt{B}^{\mathcal{I}_2} = \mathtt{C}^{\mathcal{I}_2} = \Delta^{\mathcal{I}_2}$. Clearly, $\mathcal{I}_1$ and $\mathcal{I}_2$ cannot be expanded to a same model $\mathcal{J}$ for $\mathcal{O}$. In other words, the purely syntactic relation between $\mathtt{C}$ and α_1 hinders the semantic independence of $\mathtt{D}$ from α_1.

We can equivalently define the *irrelevance*, dual to relevance, of an axiom α to a term $\mathtt{t}$, to occur when, for any subontology $\mathcal{O}'$ containing α, for any $\{\mathtt{t}\}$-model $\mathcal{I}_1$ w.r.t. $\mathcal{O}' \setminus \alpha$ and any $\{\gamma(\alpha) \setminus \mathtt{t}\}$-model $\mathcal{I}_2$ w.r.t. $\mathcal{O}'$, there exists a model $\mathcal{J}$ which is an $\mathcal{O}'$-expansion of both $\mathcal{I}_1$ and $\mathcal{I}_2$.

The notions of relevance/irrelevance described above are clearly dependent also on the syntax of the ontology since all our definitions take into account axioms as they are.

Example 5.5.12. Let us consider again the ontology One and its rewriting All defined in Example 3.3.2. Then, the unique axiom $\alpha \in \mathsf{One}$ is relevant to each term in $\widetilde{\alpha}$. In contrast, for each term $\mathtt{t} \in \widetilde{\mathsf{All}}$ there is only one axiom out of n to be relevant to $\mathtt{t}$. In particular, One does not split into unrelated fragments as it does in Parikh's approach, whilst All does, despite the fact that $\mathsf{One} \equiv \mathsf{All}$.

The notions of direct relevance and relevance are strongly connected: if an axiom $\alpha \in \mathcal{O}$ is directly relevant to a term $\mathtt{t}$, then α is also relevant to $\mathtt{t}$ in $\mathcal{O}$.

Proposition 5.5.13. *Let $\alpha \in \mathcal{O}$ be an axiom, and let $\mathtt{t} \in \gamma(\alpha)$ be a term that α is directly relevant to. Then, α is also relevant to $\mathtt{t}$ in $\mathcal{O}$.*

Proof. Let us consider the subontology $\mathcal{O}' = \{\alpha\}$. By definition of direct relevance, we have that there is a model $\mathcal{I}$ for $\mathcal{O}' = \{\alpha\}$ and a $\{\mathtt{t}\}$-variant $\mathcal{J}$ of $\mathcal{I}$ such that $\mathcal{I} \not\models \mathcal{O}'$. Then, $\mathcal{J}|_{\mathtt{t}}$ is expandable to a model for $\mathcal{O}' \setminus \{\alpha\}$, since this ontology is empty. Similarly, the interpretation $\mathcal{I}|_{\gamma(\alpha)\setminus\{\mathtt{t}\}}$ is also expandable to

the model $\mathcal{J}$ for α. However, $\mathcal{J}$ is not a model for α. Moreover, since α is irrelevant to the terms in $\varphi(\alpha)$, we know by Lemma 5.5.6 that any interpretation $\mathcal{J}'$ obtained by changing the interpretation of the terms from $\varphi(\alpha)$ in $\mathcal{J}$ we still get a non model for α. Hence, there is no model $\mathcal{J}'$ of $\mathcal{O}'$ which is an $\mathcal{O}'$-expansion of both $\mathcal{I}$ and $\mathcal{J}|_{\mathtt{t}}$, that is, α is relevant to $\mathtt{t}$. □

The analogous for Proposition 5.5.7 for the general case of a set of axioms is expressed in the following proposition.

Proposition 5.5.14. *Let $\mathcal{O}$ be a consistent ontology, $\alpha \in \mathcal{O}$, and $\mathtt{t} \in \widetilde{\mathcal{O}}$ a term. If there exists a subset Σ of $\widetilde{\alpha}$ such that $\alpha \not\equiv^{mCE}_{\Sigma} \emptyset$ and $\alpha \equiv^{mCE}_{\Sigma\setminus\mathtt{t}} \emptyset$ then α is relevant to $\mathtt{t}$ in $\mathcal{O}$.*

Proof. Let α be an axiom and $\mathtt{t}$ be a term such that there exists a signature $\Sigma \subseteq \widetilde{\alpha}$ satisfying the hypothesis. Then, each possible interpretation $\mathcal{I}$ over $\Sigma \setminus \mathtt{t}$ is expandable to a model for α. In contrast, not all interpretations over Σ are expandable to a model for α. Let $\mathcal{I}'$ be such an interpretation. Since every possible interpretations over $\Sigma \setminus \alpha$ are expandable to a model for α, so it is for $\mathcal{I}'|_{\Sigma\setminus\alpha}$. Let $\mathcal{J}'$ be an α-expansion for $\mathcal{I}'|_{\Sigma\setminus\alpha}$. Then, $\mathcal{J}'$ is a model for α such that its $\mathtt{t}$-variant $\mathcal{I}'$ is not a model for α. Hence, α is directly relevant to $\mathtt{t}$. □

We know that in general the model-theoretic notions are hard to decide, so we want to go back to define a notion of relevance based on locality. An investigation of how these two notions are related is part of our future work.

Chapter 6

Applications

In the previous chapters we have introduced the Atomic Decomposition of an ontology $\mathcal{O}$, and the enriched versions called Labelled Atomic Decompositions. Throughout this chapter we describe those applications of (L)ADs that have been already investigated.

Since everything started from the question, raised in [PS10, DPSS10], of counting how many Σ-module an ontology has, we show how the AD of an ontology can be applied to provide a better estimation of this number than the trivial one 2^m, m being the minimum between the number of axioms in $\mathcal{O}$ and the number of entities in $\widetilde{\mathcal{O}}$.

We described in Section 5.2 the LAD based on the labelling function Lab_{mss}, and how to use it for Offline Module Extraction (OME). Here we present the experimental results investigating the practicality of this approach, especially with respect to the labelling of the AD since we know that in principle the label λ of an atom $\mathfrak{a}$ can be of exponential size in the number of terms in $\downarrow \mathfrak{a}$. In particular we show that the exponential blowup does not occur often, and introduce a way to cope with such an eventuality in a practical way.

The efficiency in extracting modules based on syntactic locality, together with their logical property of coverage, has recently inspired a new generation of reasoners [TP12, ACH12] that aim at exploiting the decomposition of an ontology into modules to speed up the reasoning over complex ontologies. We will discuss how the AD can be used to drive the selection of suitable modules to be fed to these modular reasoners.

The AD of an ontology has also been recently used as an external criterion for evaluating the quality of the design patterns identified with the Regularity

Inspector for Ontologies (RIO) framework for automatically identifying regularities and for clustering the terms in an ontology according to these regularities [MISR11, Mik13].

Finally, we present DeMoSt, a prototype for a tool to explore the AD of an ontology $\mathcal{O}$ in order to discover the modular structure of $\mathcal{O}$ and the logical dependencies between the axioms of $\mathcal{O}$. Ideally, this tool can be used for gaining an insight on what it is actually said in $\mathcal{O}$, enhancing and supporting the understanding of $\mathcal{O}$.

The applications described here show that the ADs and LADs can be used in several scenarios as a technique to perform engineering tasks, to improve reasoning performance, and to support the comprehension of ontologies.

6.1 Module Count

As we already mentioned, the investigation for finding a modular structure induced by locality-based modules started in [PS10, DPSS10]. The authors aim at defining a notion of "interesting" modules from the set of modules of an ontology $\mathcal{O}$, and then to reveal their interrelations. Since the authors do not have a definite notion of what an "interesting" module is, they aim at finding *all* the modules in $\mathcal{O}$, and then to extrapolate a good notion of interestingness for a module.

Let n be the number of axioms of $\mathcal{O}$, and m be the signature size of $\mathcal{O}$. Recall that a module $\mathcal{M}$ is a subset of $\mathcal{O}$, and that $\mathcal{M}$ is extracted from the selection of a seed signature Σ. Then, in principle an ontology can contain up to $2^{\min\{n,m\}}$ modules. Hence, both the search space given by the set of all seed signatures, and the output space consisting of all modules of $\mathcal{O}$, can in principle be exponential in the size of $\mathcal{O}$. The exponential blowup in the number of modules of $\mathcal{O}$, then, would make the search for a modular structure via the full modularisation of $\mathcal{O}$ infeasible. Moreover, even an only polynomial amount of modules can be too high for direct user inspection.

In order to understand whether this kind of analysis is practically feasible, the authors define as an intermediate task the investigation on the number of modules of an ontology $\mathcal{O}$. We will refer to this problem as *Module Count* (*MC*).

Approach (without ADs)

In [PS10, DPSS10] the authors try to extrapolate the behaviour of the number of modules of an ontology by fully modularising a small corpus of eight ontologies that they evaluate to be well designed and sufficiently diverse. By "well designed" they mean that these ontologies cover a specific domain to a certain level of detail; they are axiomatically rich, for example, they do not only connect terms via atomic subsumptions, which would make module extraction rather uninteresting because the terms in the signature of a module would hardly cause other terms to be included in the module. By "diverse" they mean that these ontologies have different sizes, expressivities, ratios of axiom and term numbers, and cover different domains. Given the task to be performed, they focus on small ontologies for practical reasons.

Despite the existence of ontologies such as All (defined in Example 3.3.2) with exponentially many modules, the authors provide examples of purpose-built less disconnected ontologies with a reasonable amount of modules, and test the hypothesis that "coherent" ontologies have less modules. Unfortunately, empirically this does not seem to be the case. In particular, the full modularisation was successfully completed for two small ontologies only, namely Koala and Mereology. For the other 6, the authors sampled subontologies $\mathcal{O}_i$ of increasing sizes n_i, extracted all of their modules, and tried to extrapolate the behaviour of the number f of modules of $\mathcal{O}$ from the trendline of the fitting curve for the points (n_i, f_i) where f_i is the number of modules of the subontology $\mathcal{O}_i$.

The fundamental conclusion the authors draw is that the number of modules is often exponential in the size of the ontology. The most reasonable estimates of the total number of modules in small to midsize ontologies (i.e., anything over 100 axioms) show that full modularization is practically impossible.

How ADs can help

From Proposition 4.3.10 we know that each module can be decomposed in a unique way as the ideal of a set S of incomparable atoms, i.e., an antichain. Since all modules are represented in $(\mathcal{A}(\mathcal{O}), \succ)$, the idea arises of counting modules via the AD of $\mathcal{O}$ rather than enumerating all of them. In particular, two estimates could be provided:

Upper Bound: we saw in Example 4.3.13 that not every antichain A in $(\mathcal{A}(\mathcal{O}), \succ)$

is such that $\downarrow A$ is a module. However, by estimating the number of antichains in $(\mathcal{A}(\mathcal{O}), \succ)$ we clearly have an upper bound for the number of modules in $\mathcal{O}$. Such an estimate can be computed by applying the algebraic result known as Dilworth's theorem:

Theorem 6.1.1 (Dilworth, [Dil50])**.** *Given a finite poset $(O, >)$, there exists an antichain A, and a partition of the order into a family P of chains, such that the number of chains in the partition equals the cardinality of A. When this occurs, A is largest antichain in the order, and P is smallest family of chains into which the order can be partitioned.*

Corollary 6.1.2. *For any poset $(O, \leq)$, let $P := \{A_1, \ldots, A_w\}$ be a decomposition as in Dilworth's theorem into w chains, and let d be the depth of the poset O; set $c_i := \#A_i + 1$ for all $i \in \{1, \ldots, w\}$. Then, the number of antichains in O is bounded by the quantity $c_1 \cdot \ldots \cdot c_w \leq (d+1)^w$.*

Please note that Corollary 6.1.2 provides an upper bound of the actual number a of antichains in a poset, and that in our case a is also an upper bound of the number of modules.

Lower Bound: The *rationale* for looking for a lower bound starts from the observation that, if all antichains of an AD generate distinct modules, then an efficient way to find a lower bound of the number of antichains of a poset is simply extracting the size w of the maximal antichain and compute 2^w.

Unfortunately, the measure 2^w is not always a lower bound of the actual number of modules, as the following examples show.

Example 6.1.3. Let us recall the ontology Apart defined in Example 5.2.2. Then, the $\bot$-AD of Apart consists of 3 independent atoms, so we would expect to have $2^3 = 8$ distinct $\bot$-modules. However, the set $S = \{\alpha_1, \alpha_3\}$ is not a module since the smallest module containing S is the whole ontology. Hence, Apart contains only 7 modules, and this technique fails in providing a lower bound for the number of modules.

Example 6.1.4. Let us now consider the ontologies Chain_n as defined in Example 5.2.4. The $\top\bot^*$-AD of Chain_n consists of n independent atoms containing one axiom α_i each, for every $i = 1, \ldots, n$. Hence, the maximal antichain is of size n, and we would estimate that $\mathcal{O}$ has 2^n $\top\bot^*$-modules, hence every subset of Chain_n would be a $\top\bot^*$-module.

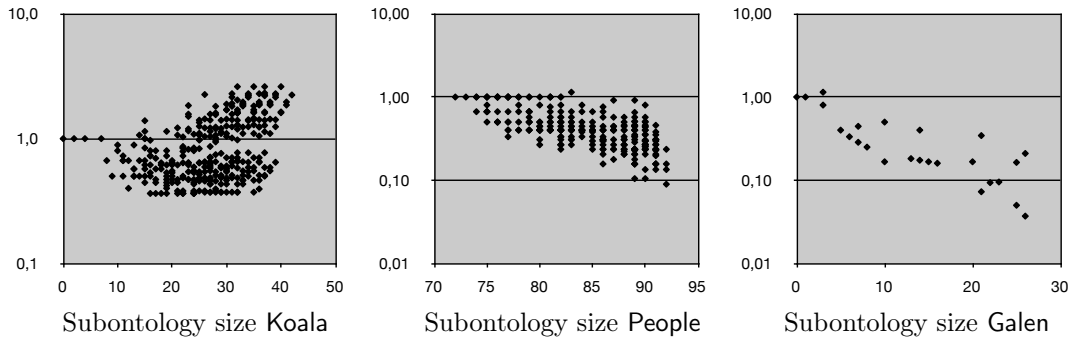
However, this is not the case: any non empty $\top\bot^*$-module $\mathcal{M}$ is such that its signature $\widetilde{\mathcal{M}}$ contains at least two concept names $\mathtt{A}_i, \mathtt{A}_j$. In particular, let i be such that $i = \min\{k \mid \mathtt{A}_k \in \widetilde{\mathcal{M}}\}$, and let j be such that $i = \max\{k \mid \mathtt{A}_k \in \widetilde{\mathcal{M}}\}$. Then, $\widetilde{\mathcal{M}}$ contains also all the terms $\mathtt{A}_k$ with $i < k < j$, and $\mathcal{M}$ is made of all the axioms α_k whose signature $\{\mathtt{A}_{k-1}, \mathtt{A}_k\}$ is contained in the set $\widetilde{\mathcal{M}}$ just described. This means that the actual module number is therefore only $\frac{n(n-1)}{2}$.

The explanation for the difference lies in the fact that atoms are not really independent since the minimal seed signatures of the corresponding genuine modules share terms with some other modules' signatures.

Discussion

The approach exploiting the ADs of an ontology has been preliminarily investigated in [DPSS11b] for what concerns a lower bound to estimate the number of modules of an ontology $\mathcal{O}$, that could help in understanding whether $\mathcal{O}$ generates exponentially many modules w.r.t. $\mathcal{O}$'s number of axioms.

Based on the module numbers from the experiment carried out in [DPSS10], we took into have considered the three ontologies People, Koala, and Galen. We have computed the $\top\bot^*$-AD of the subontologies for which we have extracted all their modules, computed the length w of the maximal antichain as well as the ratio between 2^w and the number of $\top\bot^*$-modules for the respective ontology. If that ratio is greater (less) than 1, then the value 2^w overestimates (underestimates) the module number. The picture below contains plots of the measured ratios against the subontology size for these 3 ontologies. The y-axis is scaled logarithmically, ensuring that ratios r and $1/r$ have the same vertical distance from the value 1.



To interpret the plots for every ontology $\mathcal{O}$ and its collections of subsets, the following observations are of interest.

How much does the maximal, minimal, or average ratio differ from 1?
If it tends to differ much in one direction, the estimate needs to be scaled. If it differs erratically, then the estimate will not be useful.

Does the maximal (minimal) ratio grow (shrink) when the size of $\mathcal{O}$ grows?
If it does, the the growth (shrinkage) function needs to be qualified for the estimate to be useful. It is problematic to predict the function if it differs between ontologies.

Are the differences to the "ideal" ratio 1 the same for the ratios >1 and those <1?
If they are not and if such an imbalance only occurs for some ontologies, then we should ask the question what property of the ontology is responsible for it. The degree of imbalance could then serve as gauge for that property.

How much do the maximal and the minimal ratio differ?
Their quotient represents a margin for the estimate. E.g., if the maximal and minimal ratio are 3.0 and 0.5, then we can conclude from the measured value $x = 2^w$ that $\mathcal{O}$ has between $0.333x$ and $2x$ modules. The quotient is 6; therefore we can estimate the $\top\bot^*$-module number up to one order of magnitude. Quotients > 10 decrease precision to more orders of magnitude.

We made the following observations for the ontologies we examined.

Koala. The ratio ranges from 0.36 to 2.61. For example, if we measure a maximal antichain of length 10 for any subontology of Koala, then we can estimate that the module number is between $\frac{2^{10}}{2.61} \approx 392$ and $\frac{2^{10}}{0.36} \approx 2,844$. The plot shows an even balance between "> 1" and "< 1" ratios. The minimal ratio seems to be constant with growing subontology size, but the maximal ratio seems to grow slightly. The quotient between max and min is 7.25.

People. The ratio is almost always < 1; it ranges from 0.09 to 1.14. This yields a quotient of 12.67, i.e., the prediction of the $\top\bot^*$-module number is only up to two orders of magnitude. For example, for a maximal antichain of length 10, the number of $\top\bot^*$-modules can now be between 898 and 11,378. Furthermore, the underestimation appears to grow with the ontology size.

Galen. There is almost always a ratio < 1, and the underestimation appears to grow with the subontology size. For the first 28 subontologies of very small size (up to 26 out of Galen's 4,528 axioms), we already obtain a quotient of $1.14/0.04 = 28.5$.

In summary, the ratio behaves quite differently for these five ontologies, and this restricts its use as an estimate of the $\top\bot^*$-module number. For some ontologies, the measured value 2^w tends to underestimate the $\top\bot^*$-module numbers, for others, there is no tendency. For some ontologies, the margin for the estimate obtained from 2^w is simply too large.

Overall, the AD of an ontology has clearly improved our understanding in the investigation on the number of modules of an ontology, and we have presented some interesting preliminary results. Future work to extend the investigation on the number of modules of an ontology is three-fold:

1. By exploiting the AD of $\mathcal{O}$ we plan to drive the search for distinct modules by combining atoms, and atoms signature, in order to be able to fully modularize ontologies with a number of axioms higher than 50.

2. We have so far analysed the lower bound only for $\top\bot^*$-modules, and we discussed in Subsection 4.4.2 that this kind of AD is in general the most disconnected. We have provided in Example 6.1.3 a case for undetected logical dependencies even for $\bot$-ADs. However, we expect this case to occur less often in $\bot$-ADs and $\top$-ADs rather than in $\top\bot^*$-ADs, so that the estimates provided by the AD for the number of modules of an ontology are expected to be closer to the actual number. This estimate can be further refined by designing suitable LADs to reveal this hidden dependencies to be exploited, and hence to determine a proper lower bound for the number of modules of an ontology.

3. We still need to empirically evaluate how well the estimate of the number of antichains of a poset provided by Corollary 6.1.2 approximates the actual number of modules of an ontology.

6.2 LADs for Offline Extraction of Modules

Ontologies are often maintained as monolithic collections of axioms in single files or in a few files. This is not ideal for applications which require access to individual fragments of ontologies, for example, axioms relevant for a particular term. One example is use of ontology terms in descriptions of Semantic Web services or requests for their discovery. In such cases it is undesirable to load the

entire ontology into memory (or transfer it over the network) in order to reason about a limited signature. Therefore it is important to investigate the possibility of maintaining ontologies in a more flexible form which supports reasoning over small (from the network's or the reasoner's viewpoint) fragments.

We investigated in Section 4.4 on the decomposability and modular structure of a corpus of ontologies including the NCBO BioPortal repository, and we proved that in general these ontologies can be split into *small* atoms. Now we show how modules can be assembled from their ADs before reasoning. This investigation aims at demonstrating a way to maintain and develop ontologies in a modular fashion, with the guarantee that no information get lost or corrupted since our approach generate correct and complete fragments of an ontology w.r.t. a given signature Σ. We will refer to this problem as *Offline Module Extraction* (*OME*).

How LADs can help

We have extensively discussed in Section 5.2 the notion of Labelled Atomic Decompositions with Minimal Seed Signatures (MSSs). In particular, we have described a procedure for *offline* extracting a module from such a LAD. However, this kind of structure can in principle generate labels containing exponentially many signatures w.r.t. the ontology's signature size. Hence, the main challenge to OME consists of computing *all* the MSSs for each genuine module in an ontology $\mathcal{O}$. To avoid infeasibility, an approximated version of the LAD with MSSs is generated: when the computation of the MSSs for a genuine module $\downarrow \mathfrak{a}$ takes too long, the procedure is interrupted and the atom is flagged as *dirty*. The inclusion of these atoms in a module is then handled with care when it comes to perform the OME.

Once the LAD is generated, the OME can be performed by checking whether the extended seed signature obtained by recursively enlarging a given seed signature Σ is a subset of one of the MSSs of the genuine module $\downarrow \mathfrak{a}$ for each atom $\mathfrak{a} \in \mathcal{A}(\mathcal{O})$, as described in Section 5.2.

In the remainder of this section the practicality of computing MSSs lasbels is empirically evaluated over a large corpus of ontologies from the NCBO BioPortal repository. From this analysis we can conclude that maintaining ontologies decomposed into their LADs is indeed practical.

Labelling Algorithm and Evaluation We now describe an AD-driven algorithm for computing, for each atom $\mathfrak{a}$ in the decomposition, the set of its minimal seed signatures $\mathsf{MSS}(\downarrow \mathfrak{a})$.

Algorithm 3 first computes the set $\mathsf{MGS}(\mathfrak{a})$ (minimal globalizing signatures) for all axioms in $\mathfrak{a}$ (Line 4). For an axiom α and a given notion of locality x, $\mathsf{MGS}(\alpha)$ is the set of all $\Sigma \subseteq \widetilde{\alpha}$ such that α is non-x-local w.r.t. Σ and α is x-local w.r.t. all proper subsets of Σ. For bottom atoms $\mathfrak{a}$ (i.e., atoms which do not depend on other atoms), we have that $\mathfrak{a} = \downarrow \mathfrak{a}$, and the sets $\mathsf{MSS}(\mathfrak{a})$ and $\mathsf{MGS}(\mathfrak{a})$ coincide.

Now, every signature $\Sigma \in \mathsf{MGS}(\mathfrak{a})$ is necessarily a seed signature for $\downarrow \mathfrak{a}$ but, unless $\mathfrak{a}$ is a bottom atom, is not necessarily minimal. The reason is similar to the one described in Example 6.2.1, but the redundancies can arise also from the atoms that $\mathfrak{a}$ depends on: $\Sigma' \subset \Sigma$ could be a seed signature for a module $\downarrow \mathfrak{b}$, for some atom $\mathfrak{b} \prec \mathfrak{a}$ if $\Sigma \subseteq \Sigma' \cup \widetilde{\downarrow \mathfrak{b}}$. In that case, informally, Σ' first "pulls" $\downarrow \mathfrak{b}$ into the module (Σ' being a seed signature for $\downarrow \mathfrak{b}$) and then the extended seed signature $\Sigma' \cup \widetilde{\downarrow \mathfrak{b}}$ "pulls" the axioms of $\mathfrak{a}$ and the rest of $\downarrow \mathfrak{a}$. By "extended seed signature" we mean the seed signature against which locality is checked at some iteration of the standard ME algorithm. Currently, the algorithm is limited to $\top$ or $\bot$-locality.

Please note that in Algorithm 3 the symbol $\cup^*$ means "union and minimisation w.r.t. set inclusion". This operator guarantees that every set S of seed signatures does not contain Σ' if $\Sigma \subseteq \Sigma'$ for some $\Sigma \in S$, as shown in the following example.

Example 6.2.1. Let us consider the axioms:

$$\alpha_1 : \mathsf{A} \equiv \mathsf{B}$$
$$\alpha_2 : \mathsf{A} \equiv \mathsf{B} \sqcap \mathsf{C}$$

Then, both axioms belong to the same $\bot$-atom $\mathfrak{a}$. Now, $\mathsf{MGS}(\alpha_1) = \{\{\mathsf{A}\}, \{\mathsf{B}\}\}$ and $\mathsf{MGS}(\alpha_2) = \{\{\mathsf{A}\}, \{\mathsf{B}, \mathsf{C}\}\}$. Thus, to get the MSSs for $\mathfrak{a}$ we have to exclude the signature $\{\mathsf{B}, \mathsf{C}\}$ since it is a proper superset of $\{\mathsf{B}\}$ which occurs among the MSSs of α_1. Hence, $\mathsf{MSS}(\mathfrak{a}) = \{\{\mathsf{A}\}, \{\mathsf{B}\}\}$.

A counterintuitive and surprising fact of MSSs for a genuine module $\downarrow \mathfrak{a}$ is that there can be some which are not subsets of $\widetilde{\mathfrak{a}}$, and hence that belong to the atoms that $\mathfrak{a}$ depends to. This clearly poses a computational challenge in finding them since we have to explore the whole module $\downarrow \mathfrak{a}$. An ontology showing this

Algorithm 3 Computing MSSs for a principal ideal

Input: Ontology $\mathcal{O}$; $x \in \{\top, \bot\}$; $\mathcal{O}$'s x-AD; an atom $\mathfrak{a}$
Output: $\mathsf{MSS}(\mathfrak{a})$, the set of all MSSs for $\downarrow \mathfrak{a}$
$\mathsf{MSS}(\mathfrak{a}), \mathsf{PreMSS}(\mathfrak{a}) \leftarrow \emptyset$
$\mathsf{MGS}(\mathfrak{a}) \leftarrow \bigcup^{*}_{\alpha \in \mathfrak{a}} \mathsf{MGS}(\alpha)$
$\mathsf{DD}(\mathfrak{a}) \leftarrow$ the set of atoms that $\mathfrak{a}$ non-transitively depends on
if $\mathsf{DD}(\mathfrak{a}) = \emptyset$ **then**
  return $\mathsf{MGS}(\mathfrak{a})$
end if
for each $\mathfrak{b} \in \mathsf{DD}(\mathfrak{a})$ **do**
  $\mathsf{MSS}(\mathfrak{b}) \leftarrow$ recursively compute MSSs for $\downarrow \mathfrak{b}$
end for
for each $\Sigma \in \mathsf{MGS}(\mathfrak{a})$ **do**
  $\mathsf{RC}_{\Sigma}(\mathfrak{a}) \leftarrow \{\mathfrak{b} \in \mathsf{DD}(\mathfrak{a}) \mid \Sigma \cap \widetilde{\downarrow \mathfrak{b}} \neq \emptyset\}$
  for each $\{\mathfrak{b}_1, \ldots, \mathfrak{b}_n\} \in \wp(\mathsf{RC}_{\Sigma}(\mathfrak{a}))$ **do**
    $\Sigma_{\mathfrak{a}} \leftarrow \Sigma \setminus \bigcup_{i=1,\ldots,n} \widetilde{\downarrow \mathfrak{b}_i}$
    for each $X \in \mathsf{MSS}(\mathfrak{b}_1) \times \cdots \times \mathsf{MSS}(\mathfrak{b}_n)$ **do**
      $\mathsf{PreMSS}(\mathfrak{a}) \leftarrow \mathsf{PreMSS}(\mathfrak{a}) \cup^{*} \{\Sigma_{\mathfrak{a}} \cup X\}$
    end for
  end for
end for
for each $\Sigma \in \mathsf{PreMSS}(\mathfrak{a})$ **do**
  $\mathsf{MSS}(\mathfrak{a}) \leftarrow \mathsf{MSS}(\mathfrak{a}) \cup^{*} \{\{\Sigma'\} \mid \Sigma' \subseteq \Sigma \text{ and } x\text{-mod}(\Sigma', \mathcal{O}) = \downarrow \mathfrak{a}\}$
end for
return $\mathsf{MSS}(\mathfrak{a})$

behaviour is described in what follows.

Example 6.2.2. Let us consider the following ontology:

$$\begin{aligned}\mathcal{O} = \{&\alpha : \mathtt{A} \equiv \mathtt{B} \sqcap \mathtt{C},\\ &\beta : \mathtt{B} \equiv \mathtt{D} \sqcup \mathtt{E},\\ &\gamma : \mathtt{C} \equiv \mathtt{F} \sqcup \mathtt{G}\}\end{aligned}$$

The $\bot$-genuine modules in $\mathcal{O}$ are described in what follows:

$$\begin{aligned}&\bot\text{-mod}(\{\mathtt{A}\}, \mathcal{O}) = \bot\text{-mod}(\{\mathtt{B}, \mathtt{C}\}, \mathcal{O}) &&= \{\alpha, \beta, \gamma\}\\ &\bot\text{-mod}(\{\mathtt{B}\}, \mathcal{O}) = \bot\text{-mod}(\{\mathtt{D}\}, \mathcal{O}) = \bot\text{-mod}(\{\mathtt{E}\}, \mathcal{O}) &&= \{\beta\}\\ &\bot\text{-mod}(\{\mathtt{C}\}, \mathcal{O}) = \bot\text{-mod}(\{\mathtt{F}\}, \mathcal{O}) = \bot\text{-mod}(\{\mathtt{G}\}, \mathcal{O}) &&= \{\gamma\}\end{aligned}$$

Therefore, there are three atoms $\mathfrak{a} = \{\alpha\}$, $\mathfrak{b} = \{\beta\}$, $\mathfrak{c} = \{\gamma\}$ with the dependencies $\mathfrak{b} \prec \mathfrak{a}$ and $\mathfrak{c} \prec \mathfrak{a}$. Now take the MSS $\{\mathtt{B}, \mathtt{C}\}$ for $\mathfrak{a}$ and replace $\mathtt{B}$ and $\mathtt{C}$, which occur in $\mathfrak{b}$ and $\mathfrak{c}$, with the MSSs $\{\mathtt{D}\}$ and $\{\mathtt{F}\}$ for $\mathfrak{b}$ and $\mathfrak{c}$. Then $\{\mathtt{D}, \mathtt{F}\}$ is an MSS for $\downarrow \mathfrak{a}$, although clearly $\{\mathtt{D}, \mathtt{F}\}$ is disjoint with $\widetilde{\mathfrak{a}}$ and with any member of $\mathsf{MGS}(\mathfrak{a})$.

Despite these complications, axioms of $\mathfrak{a}$ can only be pulled into the module once the extended seed signature includes at least one of the members of $\mathsf{MGS}(\mathfrak{a})$. The algorithm next recursively computes MSS for all direct children of $\mathfrak{a}$ (Line 10) and then proceeds to discover other MSSs of $\downarrow \mathfrak{a}$ by combining the sets MSS for direct children of $\mathfrak{a}$ with the set $\mathsf{MGS}(\mathfrak{a})$ (Lines 12–20). It does so by "elaborating" each $\Sigma \in \mathsf{MGS}(\mathfrak{a})$. It selects those atoms $\mathfrak{b} \prec \mathfrak{a}$ which behave as described above, i.e., $\widetilde{\downarrow \mathfrak{b}}$ overlaps with Σ. The set of all such direct children of $\mathfrak{a}$ w.r.t. Σ is stored as $\mathsf{RC}_\Sigma(\mathfrak{a})$ (Line 13). Then the algorithm removes from Σ (the signature being "elaborated") the terms in the "lower" atoms ($\bigcup_{i=1,\ldots,n} \widetilde{\downarrow \mathfrak{b}_i}$) and stores the result in $\Sigma_\mathfrak{a}$ (Line 15). Lines 16–18 go through all seed signatures X which are guaranteed to pull every atom in $\mathsf{RC}_\Sigma(\mathfrak{a})$. Then, $X \cup \Sigma_\mathfrak{a}$ is a seed signature (not necessarily minimal) for $\downarrow \mathfrak{a}$, as explained above. All such $X \cup \Sigma_\mathfrak{a}$ are collected in $\mathsf{PreMSS}(\mathfrak{a})$.

The members $\Sigma \in \mathsf{PreMSS}(\mathfrak{a})$ are not guaranteed to be *minimal* seed signatures for $\downarrow \mathfrak{a}$ because of possible weak dependencies between direct children of $\mathfrak{a}$. Informally, there could be a subset of Σ which first pulls some $\mathfrak{b}_i$, then some child of $\mathfrak{b}_i$ and only then $\mathfrak{b}_j$. Therefore, the algorithm has to "minimize" every $\Sigma \in \mathsf{PreMSS}(\mathfrak{a})$ by checking whether any of its subsets are, by themselves, already seed signatures of $\downarrow \mathfrak{a}$ (Lines 21–23). However, entries of $\mathsf{PreMSS}(\mathfrak{a})$ are usually good approximations of truly minimal seed signatures; in particular, they are much better approximations than just the signature of $\downarrow \mathfrak{a}$.

Properties of the Labelling Algorithm The correctness of Algorithm 3 can be seen as follows. For soundness, we need to prove that every Σ that is added to $\mathsf{MSS}(\mathfrak{a})$ is an MSS for $\downarrow \mathfrak{a}$. In case $\mathfrak{a}$ is not dependent on any atoms, $\mathsf{MSS}(\mathfrak{a}) = \mathsf{MGS}(\mathfrak{a})$ due to lines 6–8. Due to line 4, Σ is a minimal element of the set of all MGSs $\Sigma \subseteq \widetilde{\alpha}$ for some $\alpha \in \mathfrak{a}$ and therefore a seed signature for $\downarrow \mathfrak{a}$. Then, all $\alpha \in \mathfrak{a}$ are local w.r.t. every subset of Σ; hence Σ is a *minimal* seed signature for $\downarrow \mathfrak{a}$.

In case $\mathfrak{a}$ depends on other atoms, $\mathsf{MSS}(\mathfrak{a})$ is constructed solely in line 22. Only seed signatures for $\downarrow \mathfrak{a}$ enter $\mathsf{MSS}(\mathfrak{a})$. Since all subsets thereof are considered in this line too, all signatures remaining in $\mathsf{MSS}(\mathfrak{a})$ must be MSSs.

For completeness, we need to argue that every MSS eventually enters $\mathsf{MSS}(\mathfrak{a})$. In case $\mathfrak{a}$ is not dependent on any atoms, the MSSs for $\mathfrak{a}$ are exactly the MGSs for axioms in $\mathfrak{a}$. All these are captured in Line 7.

In case $\mathfrak{a}$ depends on other atoms, say $\mathsf{DD}(\mathfrak{a}) = \{\mathfrak{b}_1, \ldots, \mathfrak{b}_n\}$, an MSS Σ for $\mathfrak{a}$ can be smaller than an MGS for an axiom in $\mathfrak{a}$, as explained above: the terms in Σ may contain MSSs for some atoms in $\mathsf{DD}(\mathfrak{a})$ that contain the terms needed to make Σ an MGS for an axiom in $\mathfrak{a}$. To sum up, the following proposition holds.

Proposition 6.2.3. *Let $\mathfrak{a}$ be an atom and $\mathsf{DD}(\mathfrak{a})$ the set of atoms on which $\mathfrak{a}$ directly depends. For every seed signature Σ of $\downarrow\mathfrak{a}$, there exists some $\Sigma' \in \mathsf{MGS}(\mathfrak{a})$, some $\mathfrak{b}_1, \ldots, \mathfrak{b}_n \in \mathsf{DD}(\mathfrak{a})$ with $n > 0$, and $\Sigma_i \in \mathsf{MSS}(\mathfrak{b}_i)$ for every $i = 1, \ldots, n$, such that $\Sigma \subseteq \left(\Sigma' \setminus \bigcup_{i=1,\ldots,n} \widetilde{\downarrow \mathfrak{b}_i}\right) \cup \bigcup_{i=1,\ldots,n} \Sigma_i$.*

This means that every MSS Σ of $\mathfrak{a}$ is contained in some Σ', with Σ' obtained from some MGS for an axiom of $\mathfrak{a}$ by replacing the signature of the principal ideal for any atom $\mathfrak{b}_i \in \mathsf{DD}(\mathfrak{a})$ whose signature overlaps with Σ' with some MSS for $\mathfrak{b}_i$. This set Σ' is computed as PreMSS in Lines 12–20 exactly as described here, and is filtered for *minimal* seed signatures Σ in Line 22.

We will now analyze the complexity of Algorithm 3. It is clear that, in the worst case, there can be exponentially many MSSs for modules of an ontology $\mathcal{O}$. Since any MSS is added to $\mathsf{MSS}(\mathfrak{a})$ in Line 22, the worst-case runtime and memory consumption of the algorithm both have to be exponential in the size of $\mathcal{O}$.

Even when the modules of $\mathcal{O}$ have relatively few MSSs, there still needs to be an exponential amount of operations carried out in the for-loop in Lines 12–20. The loop in Lines 14–19 cycles through a potentially exponential subset of the powerset of all atoms on which $\mathfrak{a}$ depends. Hence $\mathsf{PreMSS}(\mathfrak{a})$ may become of exponential size. Furthermore, the for-loop in Lines 21–23 may cycle through all subsets of every member of this large set $\mathsf{PreMSS}(\mathfrak{a})$.

The remaining data structures, such as $\mathsf{MGS}(\mathfrak{a})$ and $\mathsf{DD}(\mathfrak{a})$ are at most exponential in the size of *axioms* in $\mathcal{O}$ or linear in the size of $\mathcal{O}$, respectively. Therefore, the outer for-loop in Lines 12–20 only squares the exponent of the exponential in the worst case. To see that the recursive call does not lead to a runtime with a tower of exponentials, it suffices to see that the sets $\mathsf{MSS}(\mathfrak{a})$ can be computed bottom-up w.r.t. the dependency relation, and there are only linearly many atoms.

Despite the worst-case intractability the algorithm has the *anytime* property: the

loops for elaborating (Lines 14–21) and minimizing (Line 21–23) a seed signature could be interrupted upon time-out, which will result in computing some subset of the MSSs set for an atom.[1] This allows for practical approximations in the case when computing all MSSs takes too long. We call atoms whose labels do not contain all MSSs *dirty* (other atoms are called *clean*).

Dirty atoms require special handling during module extraction because their relevance may not be determinable due to missing of some MSSs. In other words, if a dirty atom $\mathfrak{a}$ is not relevant to a signature, it could mean two things: first, the atom is not a part of the module or, second, the atom *is* part of the module but a seed signature, which would indicate the relevance of $\mathfrak{a}$, has not been computed due to the time-out. Therefore, in order for an OME algorithm to remain correct, we need to force it to include dirty atoms into the module even though they may be irrelevant. This means, in particular, that the performance of the OME algorithm directly depends on whether the MSSs algorithm has been able to compute all MSSs for every atom.

Algorithm 4 shows the pseudocode for performing the OME given the LAD $(\mathcal{A}(\mathcal{O}), \succ, \mathsf{Lab}_{mss})$. As for the Labelling Algorithm 3, also in this case the choice of the notion of locality is restricted to $s \in \{\bot, \top\}$.

Algorithm 4 AD-based module extraction algorithm (OME)

1: **Input:** LAD for OME of an ontology $\mathcal{O}$, a seed signature Σ
2: **Output:** The module x-mod$(\Sigma, \mathcal{O})$, where $x \in \{\top, \bot\}$
3: $\mathcal{M} \leftarrow \emptyset$
4: **repeat**
5: enlarged $\leftarrow$ `false`
6: $\mathcal{M} \leftarrow \mathcal{M} \cup$ "all atoms that are *possibly* relevant to Σ"
7: **if** $\widetilde{\mathcal{M}} \setminus \Sigma \neq \emptyset$ **then**
8: enlarged $\leftarrow$ `true`
9: **end if**
10: $\Sigma \leftarrow \Sigma \cup \widetilde{\mathcal{M}}$
11: **until** enlarged $=$ `false`
12: **return** $\mathcal{M}$

The relevance check at line 6 takes into account the possible dirtiness of an atom. In other words, the atom is *possibly relevant* to Σ if it is clean and there exists $\Sigma' \in \mathsf{MSS}(\mathfrak{a})$ such that $\Sigma' \subseteq \Sigma$ or it is dirty and $\widetilde{\downarrow \mathfrak{a}} \cap \Sigma \neq \emptyset$ and there is

[1] Minimization has to be interrupted carefully to make sure that all produced signatures are minimal w.r.t. inclusion even though some signatures could be missing.

no $\Sigma' \in \mathsf{MSS}(\mathfrak{a})$ such that $\Sigma \subset \Sigma'$.[2]

The OME algorithm has two important advantages over the standard module extraction algorithm. First, it should be faster for most of ontologies because it benefits from the labeled AD in two ways: i) it exploits labels to quickly detect relevant atoms, ii) once an atom $\mathfrak{a}$ is established to be relevant the corresponding module $\downarrow \mathfrak{a}$ is added to the module without further checks. Second, it consumes substantially less memory since only relevant atoms (and their principal ideals) need to be loaded.

Discussion

We follow [DGK+11] and discuss the evaluation of the labelling algorithm on a large portion (181 ontologies) of the NCBO BioPortal repository which is part of our corpus. The main goal of the evaluation is to assess the practical feasibility of computing all MSSs for atoms in the BioPortal ontologies. The time-out for computing labels for every atom is set to be 5 seconds, so the algorithm is guaranteed to finish in 5 times the number of atoms in seconds. The results are presented in Table 6.1, where the second column reports on the number of MSSs per genuine module, the third on the number of different terms in a label; both values are first averaged within an ontology, and the results are then averaged across all ontologies. The remaining columns report on the maximal number of MSSs reported for one atom, the number of ontologies with dirty axioms, and the maximal number of dirty atoms in an ontology.

Total no. of ont.s	Avg. size of $\mathsf{MSS}(\mathfrak{a})$	Avg. number of terms in all $\mathsf{MSS}(\mathfrak{a})$	Max. size of $\mathsf{MSS}(\mathfrak{a})$	Number of ont. with dirty atoms	Max. number of dirty atoms
181	1.4	2.1	4,252	5	554

Table 6.1: Experimental results of the labelling algorithm

For the vast majority of ontologies (176 out of 181) the algorithm was able to compute all MSSs for all atoms. Also, the average label size and the average number of terms in all MSSs per atom are small: 1.4 and 2.1, respectively. In contrast, we see that the few ontologies with dirty atoms either do not decompose well or have an interesting property of the AD graph: certain atoms depend on

[2]Observe that if a subset of $\mathsf{MSS}(\mathfrak{a})$ contains a proper superset of Σ, then, since all seed signatures are minimal, the full set $\mathsf{MSS}(\mathfrak{a})$ cannot contain a subset of Σ.

a high number of other atoms. As an example, we mention that both reasons are true for the International Classification for Nursing Practice ontology CNP for which the MSSs algorithm left 72 dirty atoms and and managed to compute $4,252$ MSSs sets for one atom.

Similar reasons can be found in Example 5.2.5, where the ontologies $\mathcal{O}_n$ defined have labels of exponential size in $\#\widetilde{\mathcal{O}}_n$. On the one hand, the atom $\mathfrak{e}$ contains n axioms which is a fifth of the overall number of axioms of $\mathcal{O}_n$ and indicates a poor decomposability. On the other hand, $\mathfrak{e}$ depends on 4 atoms only, but this kind of dependency is "well entangled" in the following sense: please note that $\#\widetilde{\mathfrak{e}} = 4n + 1$, and that all terms in $\widetilde{\mathfrak{e}}$ but $\mathtt{X}$ already appear in one of the 4 atoms that $\mathfrak{e}$ depends on. In particular, many signatures Σ obtained as combinations of terms from at least two of these 4 atoms make the module for Σ to include $\mathfrak{e}$ as well. To sum up, the exponential size of the labels seems to depend on the strong interrelations between the terms established by the ontology.

To come back to BioPortal, then, this remark suggests that in general for these ontologies the strict entanglement between terms does not occur. At this stage we cannot say whether this behaviour can be predicted, for example whether specific domains, consisting of highly interrelated concepts, need to be described into highly tangled ontologies. We leave it for future research to investigate such cases, where a subset of an ontology turns out to be relevant for such a high number of distinct, but overlapping, seed signatures.

6.3 Modular Reasoning

As we mentioned in Chapter 1, ontologies tend to grow in size, expressivity, and reasoning time. The increasing complexity of ontologies makes up a challenge for reasoner developers, and new approaches are under investigation. Recently, the idea has emerged of dealing with reasoning tasks for complex ontologies in a modular way, i.e., by decomposing an ontology $\mathcal{O}$ into supposedly more manageable fragments while still preserving all the entailments derivable from $\mathcal{O}$, following a general principle that smaller and less expressive ontologies should be easier to deal with [TP12, ACH12]. Clearly, modules based on syntactic locality are a quite interesting candidate to be used in modular reasoning since they provide coverage as required, are generally small, and can be efficiently extracted.

The supposed easier manageability comes from two factors: expressivity and

number of axioms. For expressivity, we know that ontologies in languages like $\mathcal{EL}++$ can be reasoned over in polynomial time on the input, whilst ontologies encoded in $\mathcal{SROIQ}$ require up to N2ExpTime on the input to be reasoned over. Chopping ontologies into modules means that each module $\mathcal{M}$ could be assigned to a suitable reasoner for dealing with $\mathcal{M}$. For example, in [ACH12] the authors describe the reasoner MORe and the experimental speed up compared to HermiT [MSH07] for the classification of ontologies. MORe searches in an ontology $\mathcal{O}$ the largest module $\mathcal{M} \subseteq \mathcal{O}$ encoded in $\mathcal{EL}++$ whose reasoning can be delegated to ELK [KKS11], a Java-based reasoner able to classify the SNOMED CT ontology in 5 seconds. The remainder of the ontology is then classified by using HermiT [MSH07].

For the number of axioms, as a word of caution, we have to point out that the rule "small fragment, hence little reasoning time" is not always true since the complexity of an ontology is not, in general, proportional to the number of its axioms. A comparison of different reasoners performances over a selection of biomedical ontologies is reported in [GPS12] where the number of axioms clearly is not the main factor to increase the reasoning time. The authors then define *hot spots*, which are small amounts of axioms that, removed from an ontology $\mathcal{O}$, decrease the reasoning time by several orders of magnitude. The detection of hot spots is primarily intended to be used by ontology developers who might want to understand which axioms cause an ontology to degrade its reasoning performance. However, some possibilities are under investigation for optimizing the detection of hot spots, and to use this technique to implement an approximated reasoner. Clearly, this implies that the ontology gets modified since the hot spots need to be removed. The purpose of modular reasoning instead is to make possible, or at least to speed up, reasoning on very hard ontologies without changing them.

How LADs can help

The LAD of an ontology has properties useful for modular reasoning. The idea is that it could help in driving the choice for the modules to be extracted and delegated to different reasoners, as the following results show.

Proposition 6.3.1. *Let $\mathcal{O}$ be an ontology, $\mathsf{Lab}^{\#}_{sig}$ the labelling function based on flattened atoms' signatures as defined in Section 5.3, and $(\mathcal{A}^{\perp}(\mathcal{O}), \succ, \mathsf{Lab}^{\#}_{sig})$ the corresponding LAD. Let $\mathtt{A} \in \widetilde{\mathcal{O}}$ be a concept name such that there exists another atomic concept name $\mathtt{B}$ with $\mathcal{O} \models \mathtt{A} \sqsubseteq \mathtt{B}$. Then, there is a unique atom*

$\mathfrak{a} \in \mathcal{A}(\mathcal{O}) \setminus \{\mathfrak{t}\}$, *where* $\mathfrak{t}$ *is the atom for the syntactic tautologies, such that* $\mathtt{A} \in \mathsf{Lab}^{\#}_{sig}(\mathfrak{a})$. *Equivalently, if there is a concept name* $\mathtt{A}$ *that occurs in more than one label, then there is no concept name* $\mathtt{B}$ *such that* $\mathcal{O} \models \mathtt{A} \sqsubseteq \mathtt{B}$.

Proof. Let us consider the module $\mathcal{M}_{\mathtt{A}} = \bot\text{-mod}(\{\mathtt{A}\}, \mathcal{O})$. Then, by hypothesis we know that there exists a concept name $\mathtt{B}$ such that $\mathcal{O} \models \mathtt{A} \sqsubseteq \mathtt{B}$, and then by Proposition 2.3.15 we know that $\mathcal{M}_{\mathtt{A}}$ is non empty. In particular, $\mathtt{A}, \mathtt{B} \in \widetilde{\mathcal{M}_{\mathtt{A}}}$.

We claim that $\mathcal{M}_{\mathtt{A}}$ is the unique, smallest module among the modules $\mathcal{M} \subseteq \mathcal{O}$ such that $\mathtt{A} \in \widetilde{\mathcal{M}}$. This follows from Definition 2.3.1 and from Algorithm 1 describing the procedure to extract a Σ-module: by enlarging the seed signature of a module $\mathcal{M} = \bot\text{-mod}(\Sigma, \mathcal{O})$ we obtain a superset of $\mathcal{M}$, hence:

$$\mathcal{M} = \bot\text{-mod}(\widetilde{\mathcal{M}}, \mathcal{O}) \supseteq \bot\text{-mod}(\{\mathtt{A}\}, \mathcal{O}) = \mathcal{M}_{\mathtt{A}}. \tag{6.1}$$

We want now to prove that there exists a unique atom $\mathfrak{a}$ such that $\mathcal{M}_{\mathtt{A}} = \downarrow \mathfrak{a}$ with $\mathtt{A} \in \mathsf{Lab}^{\#}_{sig}(\mathfrak{a})$. By contraposition, let us now suppose that there exists at least two atoms $\mathfrak{a}_1, \mathfrak{a}_2$ such that $\mathtt{A} \in \mathsf{Lab}^{\#}_{sig}(\mathfrak{a}_i)$, and that $\mathcal{M}_{\mathtt{A}} = \downarrow \mathfrak{a}_1 \cup \mathcal{M}'_{\mathtt{A}}$ with $\mathcal{M}'_{\mathtt{A}} \supseteq \downarrow \mathfrak{a}_2$. Please note that by the definition of the labelling function $\mathsf{Lab}^{\#}_{sig}$ we have that $\mathfrak{a}_1$ and $\mathfrak{a}_2$ must be incomparable. Now, let us consider the module $\downarrow \mathfrak{a}_1$. Then, $\downarrow \mathfrak{a}_1$ is a proper subset of $\mathcal{M}_{\mathtt{A}}$. Moreover, $\mathtt{A} \in \widetilde{\downarrow \mathfrak{a}_1}$ since $\mathtt{A} \in \mathsf{Lab}^{\#}_{sig}(\mathfrak{a})$. Hence, $\downarrow \mathfrak{a}_1$ is a smaller module than $\mathcal{M}_{\mathtt{A}}$ that $\mathtt{A} \in \widetilde{\mathcal{M}}$. This contradicts (6.1). □

Corollary 6.3.2. *Let* $\mathtt{A} \in \widetilde{\mathcal{O}}$ *be a concept name as in Proposition 6.3.1. Then, the module* $\mathcal{M}_{\mathtt{A}} = \bot\text{-mod}(\{\mathtt{A}\}, \mathcal{O})$ *is genuine.*

Proof. It follows immediately from Proposition 6.3.1. □

Properties of LADs are exploited by CHAINSAW, a wrapper of reasoners described in [TP12]. The main idea behind CHAINSAW consists of dealing with some expensive reasoning tasks as classification in a lazy way, and to initiate the reasoning procedure to decide over a query q by computing the LAD $(\mathcal{A}(\mathcal{O}), \succ, \mathsf{Lab}^{\#}_{sig})$ as a first step. The authors describe two main bottlenecks in using a modular approach to reasoning about q. The first consist of the need for recomputing a suitable module each time a new query q is asked, whilst it can easily be the case that the same module is suitable for more than one query. The second consists on the time needed to delegate and initialize a reasoner for each query. The solution adopted consists of using a cache for modules and reasoners, which can strike a tradeoff between performances and memory footprint according to the

available resources. The use of separate, independent reasoners also suggests the possibility of using multiple threads, even when the delegate reasoners do not themselves take advantage of multicore processors. This feature, however, is not implemented in CHAINSAW yet.

Discussion

Exploiting LADs to be used for modular reasoning is at its dawn, but the very preliminary results seem quite promising in future application in this field. CHAINSAW, for example, was designed to meet a specific need in the use of the interface `OWLKnowledgeExplorerReasoner`, only implemented by the reasoner FACT++, applied to the Ontology for Biomedical Investigations OBI.[3]. Unfortunately, FACT++ does not easily manage OBI: for classifying the ontology, in fact, FACT++ takes four hours and approximatively six gigabytes of memory to reach 97%. CHAINSAW instead, by computing modules and never classifying the whole ontology, allows to complete this task in less than 4 hours, and it uses considerably less resources, i.e., approximatively only three gigabytes of RAM.

The reasoner MORe described in [ACH12] uses highly optimized heuristics to find for the largest module of an ontology encoded in $\mathcal{EL}++$. In the future we plan to investigate on the performance of using an AD-driven approach to find such a module. However, it could be the case that computing the whole AD is a too long procedure to be performed only to select the module. Other lazy techniques, as the computation of an approximate AD, could be explored.

6.4 Patterns Evaluation in Ontologies

In [MISR11] the authors study the problem of recognizing an emerging *design pattern* within an ontology $\mathcal{O}$. Please note that a pattern here is not intended as one expressing a best practice to describe a domain, rather it is intended as a design template adopted by the developers of an ontology. A consequence of applying one such design pattern consists of the presence of repetitive structures in an ontology. The regularities are then provided as *generalised axioms*, also called *generalisations*, i.e., axioms in which some of the entities are replaced with variables.

[3] `http://obi-ontology.org/`

The authors propose the Regularity Inspector for Ontologies (RIO) framework for automatically identifying regularities and for clustering the terms in an ontology according to these regularities. We here disregard a discussion of the different methods that can be used to identify clusters, and present the results obtained for the socalled "popularity replacement function" method. We refer the interested reader to [Mik13] for more information.

Example 6.4.1. Let us consider the ontology Hobbies defined as follows:

$$\alpha_1 : \texttt{Cycling} \sqsubseteq \texttt{Sport}$$
$$\alpha_2 : \texttt{Swimming} \sqsubseteq \texttt{Sport}$$
$$\alpha_3 : \texttt{Sightseeing} \sqsubseteq \texttt{Hobby}$$
$$\alpha_4 : \texttt{Painting} \sqsubseteq \texttt{Hobby}$$
$$\alpha_5 : \texttt{Freestyle} \sqsubseteq \texttt{Swimming}$$
$$\alpha_6 : \texttt{Breaststroke} \sqsubseteq \texttt{Swimming}$$
$$\alpha_7 : \texttt{OilPainting} \sqsubseteq \texttt{Painting}$$
$$\alpha_8 : \texttt{WatercolorPainting} \sqsubseteq \texttt{Painting}$$

The following are generalisations obtained by applying the RIO framework to the ontology Hobbies.

$$g_1 : ?\texttt{Hobby} \sqsubseteq \texttt{Hobby} \qquad ?\texttt{Hobby} \in \{\texttt{Sightseeing}, \texttt{Painting}\}$$

$$g_2 : ?\texttt{Painting} \sqsubseteq ?\texttt{Hobby} \quad ?\texttt{Painting} \in \{\texttt{OilPainting}, \texttt{WatercolorPainting}\}$$
$$?\texttt{Hobby} \in \{\texttt{Painting}\}$$

$$g_3 : ?\texttt{Sport} \sqsubseteq \texttt{Sport} \qquad ?\texttt{Sport} \in \{\texttt{Cycling}, \texttt{Swimming}\}$$

$$g_4 : ?\texttt{Swimming} \sqsubseteq ?\texttt{Sport} \quad ?\texttt{Swimming} \in \{\texttt{Freestyle}, \texttt{Breaststroke}\}$$
$$?\texttt{Sport} \in \{\texttt{Swimming}\}$$

Detecting these regularities means the possibility of identifying parts of the schemas and guidelines upon which ontologies are often built, especially in the absence of explicit documentation. In contrast, detecting deviations from a given

regular design might either be of help in identifying defects in the ontology modelling.

However, evaluating the goodness of the clusters obtained is not an easy task. We mention two possible approaches: the first is based on internal criteria based on measurements that involve the data set themselves. The second approach is based on external criteria, i.e., the clustering results are assessed based on a pre-specified structure, which is imposed on a data set and reflects an intuition about the clustering structure of the data set. Among the external criteria, an interesting choice consists of checking the results against the schema and design documentation provided to the ontology developers. Unfortunately, this information is not always available.

How ADs can help

The AD of an ontology is sensitive to both its semantics, thanks to the strong logical properties that Σ-modules satisfy, and to its syntax since the logical axioms are untouched. From this observation the idea arises of using the AD of an ontology as an external criterion for evaluating the goodness of the clusters obtained [Mik13]. Of course, the input in both structures is the same ontology but the methods for partitioning this input are different.

The basic idea consists of *projecting* the original AD into the AD for the generalised axioms that we will call GAD, and of measuring the *compression* obtained. In fact, we expect that since a generalisation represents more than one axiom, then the GAD's atoms will result as the merging of the AD's atoms. In particular, we expect the GAD structure to be different from the AD, but not orthogonal. This intuition is captured by the following example.

Example 6.4.2. Let us consider again the ontology Hobbies. In Figure 6.1 we show the original AD and its compression after the detection of the regularities as listed in Example 6.4.1.

We see that the GAD contains only 4 atoms, hence the generalisation causes a 50% compression of the AD. Moreover, we also see that, if two axioms α_i, α_j belong to two comparable atoms, then this relation is preserved for their generalisation in the GAD. The compression indicates that indeed the generalised axioms are able to capture the design patterns in Hobbies that the AD shows as spread across the ontology.

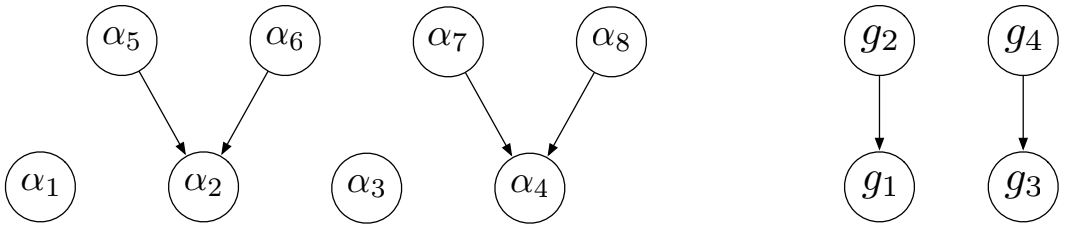


Figure 6.1: AD vs. GAD of the ontology Hobbies

Discussion

In [Mik13] the percentage compression GAD/AD is used as an indication of good quality of generalisations for 93 ontologies from BioPortal that have been inspected to detect regularities with the RIO framework. Of the 93 ontologies used in the experiment, 33 of them exhibit less than 50% GAD/AD compression, and 9 have this metric below 30%. 5 of the ontologies have a 0% compression, but all of these have at least a huge atom, with more than 50 axioms per atom.

To sum up, the high correlation between the AD structure and the generalisations found by RIO shows that the detected generalisations are not random, rather they correspond to repetitive structures also reflected in the AD structure.

6.5 DeMoSt

Finally, we want to describe the preliminary results of applying the AD in the field for supporting the comprehension of an ontology $\mathcal{O}$. The idea is to provide the user with a tool to explore the AD of $\mathcal{O}$, in order to discover the modular structure of $\mathcal{O}$, and the logical dependencies between the axioms of $\mathcal{O}$. The prototype is implemented as a plugin for Protégé, a popular editor of ontologies, and it is called DeMoSt (Decomposition & Modular Structure).

We want to show how DeMoSt can be used by working out a cognitive walkthrough for an ontology engineer who designed the ontology Teetotaller defined in Example 5.3.3. We assume them to have knowledge and experience with OWL ontologies, and with their features and representations, like concept- and role-hierarchies. Moreover, we assume them to have some basic notions of posets, as, for example, how to represent them via Hasse diagrams. This last requirement enables the users to understand the meaning of the representation shown, but it is not necessary in order to navigate the tool.

DeMoSt takes an OWL ontology $\mathcal{O}$ as input and it outputs an interactive window divided into 3 panes:
- a *terminology* pane, showing the concept-hierarchy (or, optionally, the role-hierarchy) of $\mathcal{O}$
- a *navigation* pane, showing the AD of $\mathcal{O}$ over a selected area, with a control bar to zoom in/out and to change the centre of the picture
- an *info* pane, either summarizing statistics about $\mathcal{O}$ or showing information concerning single terms or atoms whenever one of these is selected.

Now, let us suppose that the user wants to verify if the modular structure of Teetotaller reflects her intentions. In Figure 6.2 we show the screenshot of DeMoSt when we open the ontology Teetotaller.

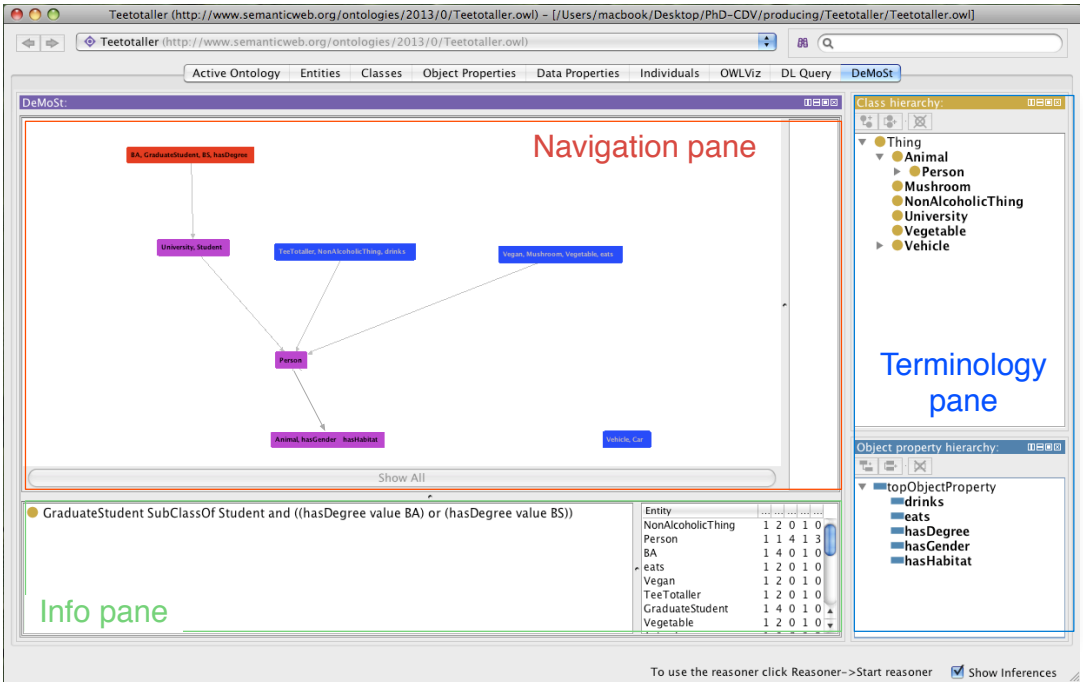


Figure 6.2: Screenshot of DeMoSt for the ontology Teetotaller

We see:
- in the navigation pane, a view of the whole ontology AD at the default zoom rate
- the concept- and the role-hierarchy in the terminology pane
- in the info pane, some data concerning occurrence frequency of terms w.r.t. atoms.

Next, let us suppose that the user wants to look closer at the term `Vegan`. She selects it in the terminology pane; then the minimap pane highlights all atoms where this term occurs (in our case, only in atom $\mathfrak{a}_3$), whilst the info pane shows the atoms' IDs (this function is useful for the case that the change in highlighting cannot be observed in a big graph). So she selects the term `Vegan` in the terminology pane, and DeMoSt shows in the navigation pane that this term occurs only in $\mathfrak{a}_3$ which depends on $\mathfrak{a}_2$. She selects the atom $\mathfrak{a}_3$, and sees in the info pane that it contains only axiom α_4.

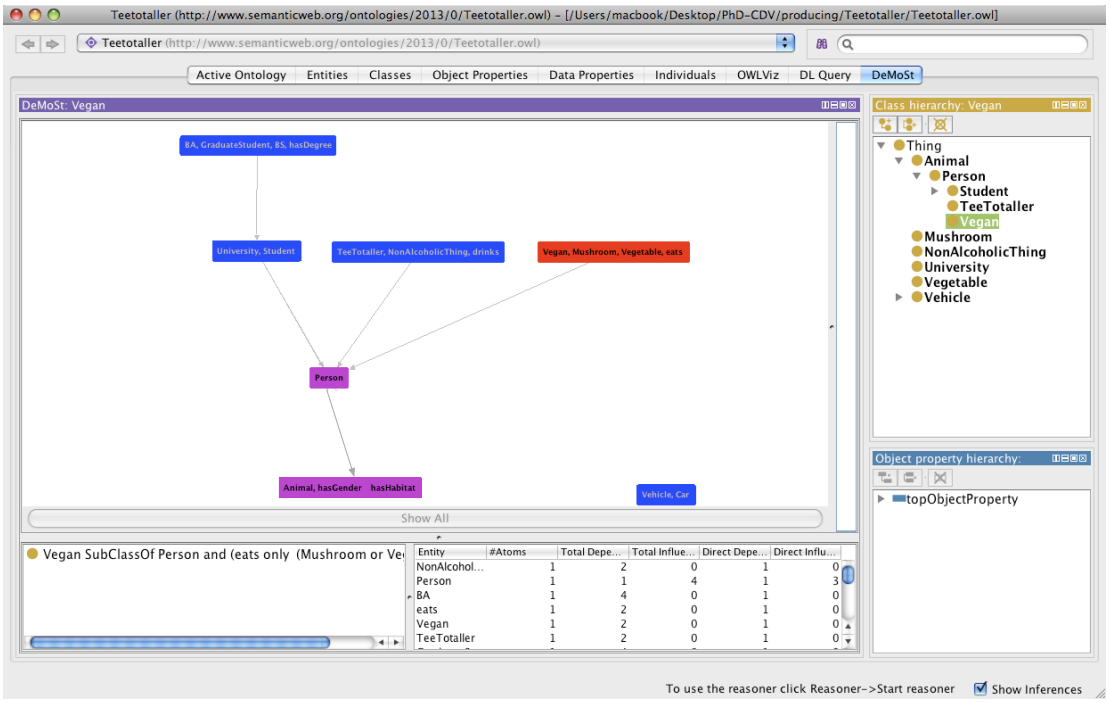


Figure 6.3: Screenshot of DeMoSt showing the selection of the term `Vegan`

However, she knows that a vegan has limitations also in what to drink (no milk, for example). She selects the term `drinks` in the terminology pane, and sees that the only atom where it occurs is $\mathfrak{a}_4$. So she understands that the ontology is not well modelled since it does not state any logical connection between α_4 and α_5, and that at least it should include the axiom $\alpha_9 =$ '$\texttt{Vegan} \sqsubseteq \forall \texttt{drinks}.\neg\texttt{Milk}$'.

Chapter 7

Conclusions

In this thesis we have defined the (L)AD which we show to be a reasonable and useful notion of locality based modular structure. It is reasonable in that:

1. it allows us to give a principle distinction between genuine (i.e., distinctively interesting) modules and "fake" modules (i.e., those were are mere unions of other modules),

2. it is easily computable,

3. it is terse, and

4. it allows us to capture other critical notions such as relevance.

It is useful in that it supports several critical applications such as offline module extraction and modular reasoning and seems promising for several others.

7.1 Summary

The focus of this thesis was advancing the state of the study of logic-based modularity in ontologies, with a look towards the modular structures that different notions of modules induce, and towards the applications that modular structures aim at in the fields of performance optimization for standard and non-standard reasoning tasks, and of the comprehensibility of an ontology.

We have discussed why some logical properties of modules are desirable by stating an analogy with the notion of modularity in Software Engineering, and analysed similarities and differences. In particular, we have argued that modules

should satisfy the following 4 properties: (1) *coverage*, i.e., modules that preserve the knowledge of $\mathcal{O}$ over a set of terms $\Sigma \subseteq \widetilde{\mathcal{O}}$; (2) *self-containment*, i.e., modules $\mathcal{M}$ that can be used instead of $\mathcal{O}$ for all the terms in $\widetilde{\mathcal{M}}$; (3) *depletion*, i.e., modules that contain all the knowledge of $\mathcal{O}$ over a signature Σ; (4) *uniqueness*, i.e., modules that are uniquely determined by the signature Σ they provide coverage for.

A modular structure is a pair $(\mathcal{F}(\mathcal{O}), \rightarrow)$, where $\mathcal{F}(\mathcal{O})$ is the set of modules of $\mathcal{O}$, or a representative subset thereof, and $\rightarrow$ is a logical relation between the elements of $\mathcal{F}(\mathcal{O})$ that generally means that, whenever $\mathcal{M}_1 \rightarrow \mathcal{M}_2$, then $\mathcal{M}_1$ needs to import $\mathcal{M}_2$ in order to preserve certain logical properties. We have described a framework to evaluate the modular structure of an ontology based on three features: (1) the *coherence* of the modules identified, i.e., if further refinements of an element of $\mathcal{F}(\mathcal{O})$ can be identified that the notion of module used does not capture; (2) the interestingness of the structure identified, i.e., which kind of logical *dependence/independence* reveals; (3) the *granularity* of the modules, i.e., how scattered the elements of $\mathcal{F}(\mathcal{O})$ are in the ontology $\mathcal{O}$. We have used this framework to evaluate several notions of modular structures can be found in the literature applied to a number of toy ontologies designed to highlight pros and cons of the different approaches.

We have defined the class of Σ-modules, i.e., a broad class of modules designed for ontology reuse and that satisfy the 4 properties described above. These modules were not provided with any interesting notion of a modular structure. The main issue with Σ-modules is that their number can be exponential in the number of axioms of the ontology $\mathcal{O}$, and extracting all of them is in practice infeasible even for the most efficiently computable notions of modules over ontologies of size greater than 100.

Despite these complications, we have been able to define the *Atomic Decomposition* $(\mathcal{A}(\mathcal{O}), \succ)$ of an ontology $\mathcal{O}$, i.e., an interesting modular structure induced on $\mathcal{O}$ by Σ-modules. In essence, $(\mathcal{A}(\mathcal{O}), \succ)$ is a dependency graph between *atoms*, defined as maximal sets of axioms such that for each module $\mathcal{M}$ either they all occur in $\mathcal{M}$, or none of them does. The atoms of $\mathcal{O}$ form a partitioning of $\mathcal{O}$, hence there can be only linearly many atoms in the number of axioms of $\mathcal{O}$.

The AD of an ontology is also a succinct representation of all the *genuine modules* of an ontology, i.e., those modules that constitute a base for all the other modules of $\mathcal{O}$. Genuine modules are interesting for themselves since they

are indecomposable w.r.t. the module notion; this feature shows a strong logical coherence of genuine modules. There is a 1-1 correspondence between genuine modules and atoms, so, in surprising contrast with all modules that can be exponentially many, the number of genuine modules is linear in the number of axioms of $\mathcal{O}$. Moreover, the problem of computing $(\mathcal{A}(\mathcal{O}), \succ)$ and of computing the set $\mathfrak{G}(\mathcal{O})$ of all genuine modules are interreducible, and the poset structure $(\mathfrak{G}(\mathcal{O}), \subseteq)$ is dual to the AD of an ontology.

A key result about genuine module is that filtering them out of the set of all modules of an ontology does not require the computation of all the modules: in fact we prove that genuine modules are also *α-modules*, i.e., modules $\mathcal{M}$ for which there exists an axiom $\alpha \in \mathcal{O}$ such that $\mathcal{M}$ is the module extracted for the signature $\widetilde{\alpha}$. This observation in particular means that we only need to perform as many module extractions as the number of axioms in $\mathcal{O}$. Hence, this task can be performed in polynomial time w.r.t. size of the ontology whenever we consider efficiently computable notions of modules, such as those based on syntactic locality. Because of the duality between $(\mathfrak{G}(\mathcal{O}), \subseteq)$ and $(\mathcal{A}(\mathcal{O}), \succ)$, this means that the AD of an ontology can be computed also in polynomial time for the same notions of modules.

The efficient computation of ADs is validated by empirical results: we have computed three kinds of ADs for a corpus of 230 ontologies including BioPortal, which is a repository of ontologies used in the biomedical domain and publicly available on the Web. Overall our corpus includes more than 500,000 axioms. The whole experiment took less than one hour on a standard machine, and only in 2 cases it exceeded $2mins$.

The AD of an ontology allows the identification of a logic-based reading order for the axioms of an ontology: we have defined *chains of Conservative Extensions*, i.e., sequences of sets of axioms $\mathcal{M}_1 \subsetneq \mathcal{M}_2 \subsetneq \ldots \subsetneq \mathcal{M}_\kappa$ in $\mathcal{O}$ such that by adding axioms at each step the meaning of the terms is preserved. This construction can be done by using modules, however the AD allows for the identification of the longest sequences of such chains, so that the complexity of understanding each enlargement is minimal.

What the AD $(\mathcal{A}(\mathcal{O}), \succ)$ of an ontology $\mathcal{O}$ does not show are the complex relationships between modules and signatures: for example, while all the modules, also the non genuine ones, are represented in $\mathcal{A}(\mathcal{O})$, the AD alone does not

provide any means to support modules identification. We have solved this problem by enriching the AD of an ontology with suitable labels for each atom, i.e., by defining a *labelling function* Lab and hence the corresponding *Labelled Atomic Decomposition* (LAD). ADs and LADs can be of help in many application scenarios; some of them have been preliminarily addressed and described, but many other remain to be discovered and put to use.

7.2 Future Work

While the investigation of the (Labelled) Atomic Decomposition of an ontology is at an advanced stage, and our understanding and exploitation of (L)ADs has already lead this notion to be used for several applications, a number of challenges remain.

Theoretical/conceptual ADs, as we define them, are intimately tied to locality based modules and thus suffer from their limitations. Most notably, we are bound by axiom structure. Since most ontologies in principle can be rewritten as a single axiom, we necessarily miss some aspects of the semantic structure of the ontology. It may be possible to produce normalizations of the ontology that would allow for canonical ADs, though there is a risk of making the ontology incomprehensive to its users. The investigation of the geometry of the AD structure when an ontology gets normalized or modified is a research line that could shade some light on a number of conceptual issues, from the unification of ontologies to the definition of design patterns.

For what concerns the labelled versions, we can see from the problems with module counting that our current labelling schemes still fail to capture some information about the set of modules. While module counting is perhaps itself uninteresting now that we have a good notion of genuine modules, it is still a sign of the information content of a LAD. An investigation in this field is needed to deepen the understanding of the way the LAD represents the modules of an ontology, so that LADs can be used to execute and optimize engineering procedures.

Empirical Although the corpus of ontologies used for the experimental results of this thesis shows some nice features, like diversity of sizes and expressivities, it

has to be noted that most of it is based on ontologies in the biomedical domain. An interesting investigation would be to compare the repositories of ontologies from other communities, in order to analyse whether and how different modelling styles impact on the structure of ADs.

Applications The (L)ADs of an ontology clearly has a number of interesting features that can be exploited in other applications beside those already investigated. We highlight the following two features: (1) the definition of a logical ordering of axioms, that allows the reading of the axioms of an ontology in a principled way, and can be further explored for supporting the development, maintainance, and repair of ontologies; (2) the guidance through the selection of the modules of an ontology; this feature could be of help in scenarios that include the selection of the right seed signature: in fact, the users of ontologies are often interested in extracting a (possibly minimal) set of axioms that "know everything" about a term `t`, for example "everything about `Bones`". However, modules based on locality are designed to solve the related, but different problem of preserving the logical relations occurring between the terms in the seed signature Σ, hence defining a singleton seed signature is likely to lead to a modules that leaves out relevant axioms describing the relations of Σ with other terms. A suitable LAD of an ontology could be of help in driving the user to select the right seed signature by showing how Σ is related to the other terms.

Appendix A

Ontologies Corpus

For our experiments, we have built a corpus containing: (1) all the ontologies belonging to the NCBO BioPortal ontology repository[1] in November 2012; (2) those ontologies from the TONES repository[2] which have already been studied in previous work on modularity [DPSS10]: Koala, Mereology, University, People, miniTambis, OWL-S, Tambis, Galen From this corpus, we have removed those ontologies with at least one of the following problems: the ontology is impossible to download; the .owl file is corrupted when downloaded; the file is not parsable; the ontology is inconsistent; the ontology contains some constructors that the implementation provided by Tsarkov does not yet support, e.g., some kinds of datatypes, the ontologies with more than 20K axioms, with the exclusion of the NCI thesaurus ontology (NCI). Please note that the ontology NCI has been modified to exclude class assertions (more than 5K axioms) since the analysis conducted in [GPS11] has detected that the presence of a huge amount of individuals is likely to be due to either a modelling or a tooling error: NCI contains both an annotation property and a data property named "code". This presumably results in the parsing of data property assertions rather than annotations, therefore leading to the mass-creation of individuals.

This selection results in a corpus of 253 ontologies designed and built by users (domain experts) for application purposes. These ontologies greatly differ in sizes, expressivities, ratios of axiom and term numbers, classification times, number of non-trivial entailments, and number of justifications per entailment [HPS11].

We want to give an idea of how diverse this corpus is at least w.r.t. ontologies'

[1] `http://bioportal.bioontology.org`
[2] `http://owl.cs.manchester.ac.uk/repository/`

sizes and expressivities. We can *prima facie* suppose that the length of an average axiom in a user designed ontology does not exceed a certain threshold c, and hence it can be considered as a multiplicative constant across all ontologies in the calculation of an ontology's size. Hence, we can discard this parameter, so that the significant factor in identifying the size of an ontology is, simply, the number of its axioms.

The analysis of expressivities requires some attention: the family of DL dialects consists of the combinations of the possible constructors as defined in Section 2.1, and even if some of these languages coincide, we still have a large number of different dialects. Moreover, two different languages can still be too similar to belong to two different bins. Hence, we follow a similar approach to the one described in [WPH06], and group the ontologies into bins defined upon families of languages.

We can easily identify three expressivity bins upon the main OWL and OWL 2 Profiles available: OWL Lite (called OWL QL in OWL 2), whose underpinning DL is a DL-lite language, hence in $\mathcal{SHIF}(\mathcal{D})$; OWL (1) DL, whose underpinning DL is $\mathcal{SHOIN}(\mathcal{D})$, and OWL (2) DL, whose underpinning DL is $\mathcal{SROIQ}(\mathcal{D})$. These DL languages can be totally ordered by their expressivities since $\mathcal{SHIF} \subsetneq \mathcal{SHOIN} \subsetneq \mathcal{SROIQ}$. Note that this is unusual since DL languages can be incomparable.

Although OWL Profiles provides us with a rough idea of the distribution of expressivities among ontologies, we can still see that $\mathcal{SHIF}$ has a large number of sublanguages, hence we want to refine this bin further. The smallest language that can be detected by the OWL API is $\mathcal{AL}$,[3] hence it is the natural candidate for being the smallest language bin. Moreover, we split this bin further by identifying $\mathcal{S}$ as an intermediate language between $\mathcal{AL}$ and $\mathcal{SHIF}$. Finally, for simplicity we ignore the presence of datatypes, hence $\mathcal{AL}(\mathcal{D})$ is considered the same as $\mathcal{AL}$.

To sum up, we have identified five bins: the first contains all the ontologies in $\mathcal{AL}/\mathcal{EL}$; the second contains those ontologies in $\mathcal{S}$ that are strictly more expressive than $\mathcal{AL}$; the third contains all ontologies in $\mathcal{SHIF}$ that need more than $\mathcal{S}$ to be expressed; the fourth contains those ontologies that are strictly more expressive than $\mathcal{SHIF}$ and can be expressed in $\mathcal{SHOIN}$; the fifth bin contains the remaining ontologies.

[3]Please note that the $\mathcal{EL}$ language is labelled under $\mathcal{AL}$

In Table A.1 we show how our corpus gets subdivided by following this expressivity binning.

Bin	$\mathcal{AL}$	$\mathcal{S}$	$\mathcal{SHIF}$	$\mathcal{SHOIN}$	$\mathcal{SROIQ}$
Count	56	63	51	36	47

Table A.1: Expressivity bin numbers for our corpus

Figure A.1 aims at giving an idea of the diversity of the ontologies w.r.t. their sizes and expressivity: each ontology is represented by a point (E, A), where the abscissa E corresponds to the expressivity bin that $\mathcal{O}$ belongs to, and the ordinate A is the number of axioms in $\mathcal{O}$. Please note that the ordinates follow a logarithmic scale.

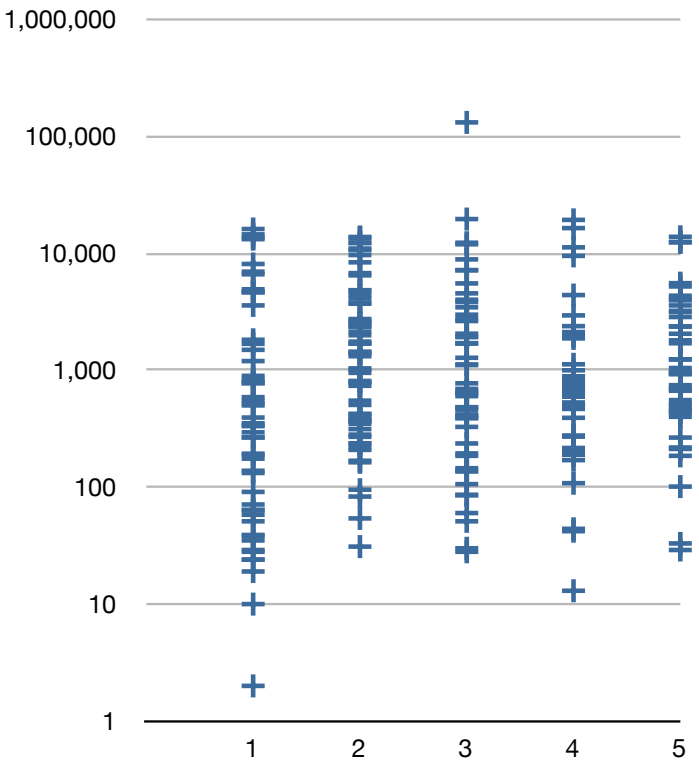


Figure A.1: Diversity of our corpus: Expressivity vs. Number of axioms

In what follows, we list the whole set of ontologies considered in our experiments.

Table A.2: Corpus of ontologies included in experiments

Ontology	Abbr.	DL express.	#axs
aba-adult-mouse-brain	ABA	$\mathcal{ALCI}$	3,441
adverse-event-reporting-ontology	AER	$\mathcal{SHOIQ}(\mathcal{D})$	873
african-traditional-medicine	ATM	$\mathcal{ALE}$	208
amino-acid	AMA	$\mathcal{ALCF}(\mathcal{D})$	477
amphibian-gross-anatomy	AGA	$\mathcal{ALE}$	2,673
amphibian-taxonomy	AMT	$\mathcal{ALE}$	12,163
anatomical-entity-ontology	AEO	$\mathcal{ALE}$	368
animal-natural-history-and-life-history	ALH	$\mathcal{ALCOF}(\mathcal{D})$	638
apollo-akesios	AAK	$\mathcal{AL}$	2
ascomycete-phenotype-ontology	APO	$\mathcal{AL}$	294
basic-formal-ontology	BFO	$\mathcal{ALC}$	95
basic-vertebrate-anatomy	BVA	$\mathcal{SHIF}$	388
bilateria-anatomy	BIA	$\mathcal{ALEH}+$	138
bio-health-ontological-knowledge-base-cystic--fibrosis	BCF	$\mathcal{ALCHIF}$	660
bioassay-ontology	BAO	$\mathcal{SROIQ}(\mathcal{D})$	1,797
bioinformatics-operations-types-of-data--formats-and-topics	BDT	$\mathcal{ALEH}$	3,814
bioinformatics-web-service-ontology	BWS	$\mathcal{SROIQ}(\mathcal{D})$	430
biological-imaging-methods	BIM	$\mathcal{S}$	548
biomedical-resource-ontology	BRO	$\mathcal{SHIF}(\mathcal{D})$	634
biopax	BPX	$\mathcal{SHIN}(\mathcal{D})$	391
bioportal-metadata	BPM	$\mathcal{ALUHIN}(\mathcal{D})$	822
biotop	BTP	$\mathcal{SRI}$	922
birnlex	BLX	$\mathcal{AL}$	3,572
bleeding-history-phenotype	BHP	$\mathcal{ALCIF}(\mathcal{D})$	1,925
body-system	BOS	$\mathcal{AL}$	28
book	BOO	$\mathcal{ALCHOIN}(\mathcal{D})$	529
breast-cancer-grading-ontology	BCG	$\mathcal{SHOIN}(\mathcal{D})$	690

Continued on next page

Table A.2 – Continued from previous page

Ontology	Abbr.	DL express.	#axs
breast-tissue-cell-lines	BTC	$\mathcal{ALCH}(\mathcal{D})$	2,734
brenda-tissue-enzyme-source	BTE	$\mathcal{ALE}$	6,451
c-elegans-development	CED	$\mathcal{AL}$	71
c-elegans-gross-anatomy	CEG	$\mathcal{ALE}$	12,341
c-elegans-phenotype	CEP	$\mathcal{AL}+$	2,366
cao	CAO	$\mathcal{SHIQ}(\mathcal{D})$	442
carelex	CAX	$\mathcal{ALH}(\mathcal{D})$	327
cell-behavior-ontology	CBO	$\mathcal{ALUO}$	13
cell-culture-ontology	CCO	$\mathcal{SHOIF}$	9,467
cell-line-ontology	CLO	$\mathcal{ALCH}(\mathcal{D})$	3,996
cell-type	CTY	$\mathcal{SH}$	2,975
cereal-plant-development	CPD	$\mathcal{ALE}$	235
cerebrotendinous-xanthomatosis	CTX	$\mathcal{ALCOIN}(\mathcal{D})$	1,969
chemical-information-ontology	CIO	$\mathcal{SROIN}(\mathcal{D})$	1,237
clinical-measurement-ontology	CMO	$\mathcal{ALE}+$	1,382
cognitive-atlas	COA	$\mathcal{ALC}$	4,100
common-anatomy-reference-ontology	CAR	$\mathcal{ALE}+$	54
common-terminology-criteria-for-adverse--events	CTC	$\mathcal{AL}(\mathcal{D})$	6,940
comparative-data-analysis-ontology	CDA	$\mathcal{SROIQ}(\mathcal{D})$	462
computational-neuroscience-ontology	CNO	$\mathcal{SHOIF}$	1,121
dendritic-cell	DEC	$\mathcal{ALC}$	313
dengue-fever-ontology	DEN	$\mathcal{SROIF}$	5,534
dictyostelium-discoideum-anatomy	DDA	$\mathcal{ALE}+$	379
dikb-evidence-ontology	DEO	$\mathcal{ALCHOIN}(\mathcal{D})$	660
drosophila-development	DRD	$\mathcal{ALEH}+$	410
drosophila-gross-anatomy	DGA	$\mathcal{SH}$	19,538
eagle-i-research-resource-ontology	ERR	$\mathcal{SHOIF}(\mathcal{D})$	4,378
electrocardiography-ontology	ECO	$\mathcal{ALCIF}(\mathcal{D})$	1,274
emotion-ontology	EMO	$\mathcal{SROIQ}$	728
environment-ontology	ENO	$\mathcal{S}$	1,752

Continued on next page

Table A.2 – Continued from previous page

Ontology	Abbr.	DL express.	#axs
enzyme-mechanism-ontology	ENM	$\mathcal{ALCRQ}(\mathcal{D})$	931
epilepsy	EPI	$\mathcal{ALH}(\mathcal{D})$	145
event-inoh-pathway-ontology-	EIP	$\mathcal{ALEH}+$	7,131
evidence-codes	EVC	$\mathcal{ALE}$	363
experimental-conditions-ontology	EXC	$\mathcal{ALE}+$	269
experimental-factor-ontology	EXF	$\mathcal{ALHIF}+$	7,156
exposure-ontology	EXP	$\mathcal{ALER}+$	101
family-health-history-ontology	FHH	$\mathcal{ALCHIF}(\mathcal{D})$	1,103
fda-medical-devices-2010-	FDA	$\mathcal{AL}$	4,907
fission-yeast-phenotype-ontology	FYP	$\mathcal{SH}$	12,265
fly-taxonomy	FLY	$\mathcal{AL}$	6,587
flybase-controlled-vocabulary	FCV	$\mathcal{ALE}+$	793
fungal-gross-anatomy	FGA	$\mathcal{ALEI}+$	106
gene-regulation-ontology	GR1	$\mathcal{ALCHIQ}(\mathcal{D})$	962
gene-regulation-ontology	GR2	$\mathcal{ALCHIQ}(\mathcal{D})$	962
general-formal-ontology-biology	GFB	$\mathcal{SHIN}$	466
general-formal-ontology	GFO	$\mathcal{SHIQ}$	212
genomic-clinical-decision-support-genomic-cds	GCD	$\mathcal{ALCQ}$	4,322
health_indicators	HEI	$\mathcal{AL}$	548
health-level-seven	HEL	$\mathcal{AL}$	8,072
hom_elixhauserscores	HEC	$\mathcal{AL}$	29
hom_mdcs-drgs	HMD	$\mathcal{AL}$	774
hom-datasource_oshpd	HDD	$\mathcal{AL}$	351
hom-datasource_oshpdsc	HDC	$\mathcal{AL}$	351
hom-dxprocs_mdcdrg	HDM	$\mathcal{AL}$	774
hom-dxvcodes2_oshpd	HD2	$\mathcal{AL}$	16,064
hom-harvard	HOH	$\mathcal{AL}$	189
hom-icd9_dxandvcodes_oshpd	HID	$\mathcal{AL}$	16,066
hom-icd9_procs_oshpd	HIP	$\mathcal{AL}$	4,642
hom-icd9cm-ecodes	HIE	$\mathcal{AL}$	1,490
hom-icd9pcs	HI9	$\mathcal{AL}$	4,643

Continued on next page

Table A.2 – Continued from previous page

Ontology	Abbr.	DL express.	#axs
hom-mdcdrg	HMO	$\mathcal{AL}$	790
hom-oshpd_usecase	HOU	$\mathcal{AL}$	393
hom-oshpd-sc	HOS	$\mathcal{AL}$	266
hom-procs2_oshpd	HPO	$\mathcal{AL}$	4,642
hom-ucare	HUC	$\mathcal{AL}$	64
homerun-ontology	HOO	$\mathcal{AL}$	1,194
host-pathogen-interactions-ontology	HPI	$\mathcal{SHI}$	403
human-developmental-anatomy-abstract--version-v2	HD1	$\mathcal{ALE}+$	13,495
human-developmental-anatomy-abstract--version	HDA	$\mathcal{ALE}$	2,336
human-developmental-anatomy-timed-version	HDT	$\mathcal{ALE}$	8,339
human-disease-ontology	HDO	$\mathcal{ALE}$	6,743
human-phenotype-ontology	HPH	$\mathcal{AL}$	13,153
hymenoptera-anatomy-ontology	HAO	$\mathcal{SR}$	3,944
icps-network	ICP	$\mathcal{AL}$	24
imgt-ontology	IMG	$\mathcal{SHIN}(\mathcal{D})$	2,114
immune-disorder-ontology	IDO	$\mathcal{AL}$	1,676
infectious-disease-ontology	INF	$\mathcal{SROIF}$	1,233
influenza-ontology	INO	$\mathcal{SROIN}(\mathcal{D})$	1,696
information-artifact-ontology	IAO	$\mathcal{ALRIF}+(\mathcal{D})$	554
interaction-network-ontology	INN	$\mathcal{ALC}$	1,034
interaction-ontology	ION	$\mathcal{AL}$	39
international-classification-for-nursing-practice	CNP	$\mathcal{SHIF}$	11,891
international-classification-of-external-causes-of-injuries	CCJ	$\mathcal{AL}(\mathcal{D})$	13,612
international-classification-of-functioning--disability-and-health-icf-	IFD	$\mathcal{ALCHOIF}(\mathcal{D})$	19,223
kinetic-simulation-algorithm-ontology	KSA	$\mathcal{ALCRIQ}(\mathcal{D})$	710
linkingkin2pep	L2P	$\mathcal{SHIF}(\mathcal{D})$	30
lipid-ontology	LIO	$\mathcal{ALCHIN}$	2,375
loggerhead-nesting	LON	$\mathcal{ALE}$	347

Continued on next page

Table A.2 – Continued from previous page

Ontology	Abbr.	DL express.	#axs
mahco-an-mhc-ontology	MAM	$\mathcal{ALCIQ}(\mathcal{D})$	13,781
maize-gross-anatomy	MGO	$\mathcal{ALE}$	217
malaria-ontology	MAL	$\mathcal{ALER}+$	3,212
mammalian-phenotype	MAP	$\mathcal{AL}+$	10,934
mass-spectrometry	MAS	$\mathcal{ALE}+$	2,518
measurement-method-ontology	MMO	$\mathcal{AL}$	332
medaka-fish-anatomy-and-development	MFA	$\mathcal{ALE}$	4,402
mego	MEG	$\mathcal{ALE}+$	421
mental-functioning-ontology	MFO	$\mathcal{SROIQ}$	514
microrna-ontology	MRN	$\mathcal{ALEI}$	638
minimal-anatomical-terminology	MAT	$\mathcal{ALE}$	504
mixs-controlled-vocabularies	MIX	$\mathcal{AL}$	518
molecule-role-inoh-protein-name-family-name-ontology-	MPF	$\mathcal{ALE}+$	9,629
mosquito-gross-anatomy	MGA	$\mathcal{ALE}+$	2,733
mosquito-insecticide-resistance	MIR	$\mathcal{ALE}+$	4,413
mouse-adult-gross-anatomy	MAA	$\mathcal{ALE}+$	3,776
mouse-experimental-design-ontology	MED	$\mathcal{ALH}$	86
mouse-gross-anatomy-and-development	MGD	$\mathcal{ALE}$	13,730
mouse-pathology	MOP	$\mathcal{ALE}+$	808
multiple-alignment	MUA	$\mathcal{ALE}+$	168
nanoparticle-ontology	NAN	$\mathcal{SHIN}(\mathcal{D})$	16,353
nci-thesaurus-repaired	NCI	$\mathcal{SH}(\mathcal{D})$	130,945
neglected-tropical-disease-ontology-ntdo-	NTD	$\mathcal{SRIQ}$	1,237
neomark-oral-cancer-ontology	NOC	$\mathcal{SHIQ}$	399
neural-electromagnetic-ontologies	NEO	$\mathcal{SHIQ}(\mathcal{D})$	2,843
neural-immune-gene-ontology	NIG	$\mathcal{SH}$	8,835
neuro-behavior-ontology	NBO	$\mathcal{ALE}$	1,314
nif-cell	NIC	$\mathcal{SROIF}(\mathcal{D})$	3,570
nif-subcellular	NIS	$\mathcal{SROIF}(\mathcal{D})$	4,061
nmr-instrument-specific-component-of-metabolomics-investigations	NMR	$\mathcal{SH}(\mathcal{D})$	599

Continued on next page

Table A.2 – Continued from previous page

Ontology	Abbr.	DL express.	#axs
obo-relationship-types	ORT	$\mathcal{ALR}+$	33
oboe	OBE	$\mathcal{SRIQ}(\mathcal{D})$	265
ontologia-proj-alternativa	OPA	$\mathcal{ALUIN}+(\mathcal{D})$	270
ontology-for-disease-genetic-investigation	ODG	$\mathcal{SHIN}(\mathcal{D})$	1,867
ontology-for-drug-discovery-investigations	DDI	$\mathcal{SHOIN}(\mathcal{D})$	996
ontology-for-general-medical-science	GMS	$\mathcal{ALCO}$	216
ontology-for-genetic-interval	OGI	$\mathcal{SHIN}(\mathcal{D})$	509
ontology-for-microrna-target-prediction	MTP	$\mathcal{ALCHIQ}(\mathcal{D})$	2,364
ontology-for-parasite-lifecycle	OPL	$\mathcal{SHOIF}$	885
ontology-of-clinical-research-ocre-	OCR	$\mathcal{ALCHIF}(\mathcal{D})$	51
ontology-of-data-mining	ODM	$\mathcal{SHOIQ}(\mathcal{D})$	2,353
ontology-of-experimental-variables-and-values	EVV	$\mathcal{ALCO}(\mathcal{D})$	189
ontology-of-general-purpose-datatypes	GPD	$\mathcal{SHOI}$	773
ontology-of-geographical-region	OGR	$\mathcal{AL}$	38
ontology-of-glucose-metabolism-disorder	GMD	$\mathcal{AL}$	132
ontology-of-homology-and-related-concepts--in-biology	OHC	$\mathcal{ALC}$	83
ontology-of-language-disorder-in-autism	LDA	$\mathcal{AL}$	35
ontology-of-medically-related-social-entities	MSE	$\mathcal{ALCHOIQ}$	218
ontology-of-physics-for-biology	OPB	$\mathcal{ALCHIQ}(\mathcal{D})$	954
pathogen-transmission	PAT	$\mathcal{AL}$	24
pathway-ontology	PAO	$\mathcal{ALE}$	1,432
pediatric-terminology	PET	$\mathcal{AL}$	893
phare	PHA	$\mathcal{ALCHIF}(\mathcal{D})$	459
phenotype-fragment-ontology	PFO	$\mathcal{ALUHI}+$	28
phenotypic-quality	PHQ	$\mathcal{SH}$	1,916
phenx-terms	PHT	$\mathcal{AL}$	339
phylogenetic-ontology	PHY	$\mathcal{ALCH}(\mathcal{D})$	194
physical-medicine-and-rehabilitation	PMR	$\mathcal{ALU}$	163
physicalfields	PHF	$\mathcal{ALI}$	136
physico-chemical-methods-and-properties	PCM	$\mathcal{ALE}$	1,684

Continued on next page

Table A.2 – Continued from previous page

Ontology	Abbr.	DL express.	#axs
physico-chemical-process	PCP	$\mathcal{ALE}$	734
pilot-ontology	PIL	$\mathcal{ALCIF}(\mathcal{D})$	85
pko_re	PKO	$\mathcal{ALCF}$	771
plant-anatomy	PLA	$\mathcal{ALE}+$	2,128
plant-environmental-conditions	PEC	$\mathcal{AL}$	499
plant-ontology	PLO	$\mathcal{S}$	2,545
plant-structure-development-stage	PDS	$\mathcal{ALE}+$	281
plant-trait-ontology	PTO	$\mathcal{ALE}$	1,429
platynereis-stage-ontology	PSO	$\mathcal{ALE}$	31
pma-2010	PMA	$\mathcal{AL}(\mathcal{D})$	10
protein-modification	PMO	$\mathcal{ALE}+$	1,986
protein-ontology	PRO	$\mathcal{ALCF}(\mathcal{D})$	691
protein-protein-interaction	PPI	$\mathcal{ALE}+$	1,007
proteomics-data-and-process-provenance	PDP	$\mathcal{SHOIN}(\mathcal{D})$	732
proteomics-pipeline-infrastructure-for-cptac	PIC	$\mathcal{ALCF}(\mathcal{D})$	1,118
prov-o	PRV	$\mathcal{ALCRIN}(\mathcal{D})$	185
pseudogene	PSG	$\mathcal{AL}$	19
quantitative-imaging-biomarker-ontology	QIB	$\mathcal{ALUIF}(\mathcal{D})$	1,699
rapid-phenotype-ontology	RPO	$\mathcal{ALF}(\mathcal{D})$	2,047
rat-strain-ontology	RAS	$\mathcal{ALE}$	4,739
reproductive-trait-and-phenotype-ontology	RTP	$\mathcal{AL}$	91
rna-ontology	RNA	$\mathcal{SRIQ}$	666
sample-processing-and-separation-techniques	SPS	$\mathcal{AL}$	193
sanou	SA1	$\mathcal{ALC}$	400
sanou	SA2	$\mathcal{ALC}$	400
semanticscience-integrated-ontology	SSI	$\mathcal{SRIQ}(\mathcal{D})$	2,044
sequence-types-and-features	STF	$\mathcal{SHI}$	2,627
skin-physiology-ontology	SPO	$\mathcal{ALERIF}+$	678
sleep-domain-ontology	SDO	$\mathcal{ALCO}$	204
smoking-behavior-risk-ontology	SBR	$\mathcal{ALEI}+$	185
snp-ontology	SNP	$\mathcal{SHOIN}(\mathcal{D})$	11,199

Continued on next page

Table A.2 – Continued from previous page

Ontology	Abbr.	DL express.	#axs
software-ontology	SOO	$\mathcal{ALHI}+(\mathcal{D})$	5,507
solanaceae-phenotype-ontology	SOP	$\mathcal{ALE}$	422
soyontology	SOY	$\mathcal{AL}$	1,816
spatial-ontology	SPA	$\mathcal{ALEHI}+$	236
spider-ontology	SPI	$\mathcal{ALE}+$	778
student-health-record	STU	$\mathcal{ALH}(\mathcal{D})$	418
subcellular-anatomy-ontology-sao-	SAO	$\mathcal{SHIN}(\mathcal{D})$	2,935
symptom-ontology	SYM	$\mathcal{AL}$	839
synapse-ontology	SYN	$\mathcal{AL}$	14,458
syndromic-surveillance-ontology	SSO	$\mathcal{ALIF}(\mathcal{D})$	1,684
sysmo-jerm	SYG	$\mathcal{SHI}(\mathcal{D})$	482
systems-biology	SYB	$\mathcal{AL}$	587
systems-chemical-biology-chemogenomics	SCC	$\mathcal{SHIN}(\mathcal{D})$	489
taxonomic-rank-vocabulary	TRV	$\mathcal{AL}$	58
teleost-anatomy-ontology	TAO	$\mathcal{ALERI}+$	5,188
terminological-and-ontological-knowledge--resources	TOK	$\mathcal{SRIQ}(\mathcal{D})$	466
terminology-for-the-description-of-dynamics	TDD	$\mathcal{SRIQ}(\mathcal{D})$	12,344
thesaurus-alternativa	THA	$\mathcal{AL}(\mathcal{D})$	138
thesaurus	THE	$\mathcal{AL}(\mathcal{D})$	138
thomcan-upper-level-cancer-ontology	TUL	$\mathcal{AL}$	51
tick-gross-anatomy	TGA	$\mathcal{ALE}+$	948
time-event-ontology	TEO	$\mathcal{SROIQ}(\mathcal{D})$	1,042
tissue-microarray-ontology	TMA	$\mathcal{ALI}(\mathcal{D})$	60
translational-medicine-ontology	TMO	$\mathcal{SRIN}(\mathcal{D})$	502
uni-ece	UNI	$\mathcal{SOIF}(\mathcal{D})$	3,133
units-of-measurement	UOM	$\mathcal{ALE}$	371
units-ontology	UNO	$\mathcal{AL}$	63
vaccine-ontology	VAC	$\mathcal{SROIQ}$	12,451
variation-ontology	VAO	$\mathcal{ALEI}+$	390
vertebrate-homologous-organ-groups	VHO	$\mathcal{ALE}+$	1,688

Continued on next page

Table A.2 – Continued from previous page

Ontology	Abbr.	DL express.	#axs
vertebrate-skeletal-anatomy-ontology	VSA	$\mathcal{ALER}+$	455
vertebrate-trait-ontology	VTO	$\mathcal{AL}+$	3,691
vital-sign-ontology	VSI	$\mathcal{SROIQ}$	743
web-service-interaction-ontology	WSI	$\mathcal{ALER}+$	29
wheat-trait	WHT	$\mathcal{AL}$	175
xeml-environment-ontology	XEO	$\mathcal{ALE}$	237
xenopus-anatomy-and-development	XAD	$\mathcal{ALE}+$	4,819
yeast-phenotypes	YEP	$\mathcal{AL}$	266
zebrafish-anatomy-and-development	ZAD	$\mathcal{ALE}+$	10,600
Galen	GAL	$\mathcal{ALEHF}+$	4,528
Koala	KOA	$\mathcal{ALCON}(\mathcal{D})$	42
Mereology	MER	$\mathcal{SHIN}$	44
MiniTambis-repaired	MNT	$\mathcal{ALCN}$	170
OWL-S	OWL	$\mathcal{ALCHOIN}(\mathcal{D})$	276
People	PEO	$\mathcal{ALCHOIN}$	108
tambis-patched-repaired	TPR	$\mathcal{SHIN}(\mathcal{D})$	592
University	UNV	$\mathcal{SOIN}(\mathcal{D})$	52

Bibliography

[ACH12] 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-12)*, volume 7649 of *Lecture Notes in Computer Science*, pages 1–16, 2012.

[Baa03] Franz Baader. Terminological cycles in a Description Logic with existential restrictions. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI-03)*, pages 325–330, 2003.

[BBL05] Franz Baader, Sebastian Brandt, and Carsten Lutz. Pushing the $\mathcal{EL}$ envelope. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI-05)*, pages 364–369, 2005.

[BCH06] Jie Bao, Doina Caragea, and Vasant G. Honavar. On the semantics of linking and importing in modular ontologies. In *Proceedings of the 5th International Semantic Web Conference (ISWC-06)*, volume 4273 of *Lecture Notes in Computer Science*, pages 72–86. Springer-Verlag, 2006.

[BCNP03] Franz Baader, Deborah Calvanese, Diego andMcGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications.* Cambridge University Press, 2003.

[BLS06] Franz Baader, Carsten Lutz, and Boontawee Suntisrivaraporn. `CEL`—a polynomial-time reasoner for life science ontologies. In *Proceedings of the 3rd International Joint Conference on Automated Reasoning*

(IJCAR-06), volume 4130 of *Lecture Notes in Computer Science*, pages 287–291. Springer-Verlag, 2006.

[BLSW02] Franz Baader, Carsten Lutz, Holger Sturm, and Frank Wolter. Fusions of description logics and abstract description systems. *Journal of Artificial Intelligence Research*, 16:1–58, 2002.

[BPS07] Franz Baader, Rafael Peñaloza, and Boontawee Suntisrivaraporn. Pinpointing in the description logic $\mathcal{EL}^+$. In *Proceedings of the 30th Annual German Conference on Artificial Intelligence (KI-07)*, pages 52–67, 2007.

[BS03] Alexander Borgida and Luciano Serafini. Distributed Description Logics: Assimilating information from peer sources. *Journal on Data Semantics*, 2800:153–184, 2003.

[CDL+05] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. DL-Lite: Tractable description logics for ontologies. In *Proceedings of the 20th National Conference on Artificial Intelligence (AAAI-05)*, pages 602–607, 2005.

[CDL+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. *Journal of Automated Reasoning*, 39(3):385–429, 2007.

[CHKS07] Bernardo Cuenca Grau, Ian Horrocks, Yevgeny Kazakov, and Ulrike Sattler. Just the right amount: Extracting modules from ontologies. In *Proceedings of the 16th International World Wide Web Conference (WWW-07)*, pages 717–726, 2007.

[CHKS08] Bernardo Cuenca Grau, Ian Horrocks, Yevgeny Kazakov, and Ulrike Sattler. Modular reuse of ontologies: Theory and practice. *Journal of Artificial Intelligence Research*, 31(1):273–318, 2008.

[CK07] Bernardo Cuenca Grau and Oliver Kutz. Modular ontology languages revisited. In *Proceedings of the Workshop on Semantic Web for Collaborative Knowledge Acquisition (SWeCKa-07)*, 2007.

[CPS06] Bernardo Cuenca Grau, Bijan Parsia, and Evren Sirin. Combining OWL ontologies using $\mathcal{E}$-connections. *Journal of Web Semantics*, 4(1):40–59, 2006.

[CPSK06] Bernardo Cuenca Grau, Bijan Parsia, Evren Sirin, and Aditya Kalyanpur. Modularity and Web ontologies. In *Proceedings of the 10th International Conference on the Principles of Knowledge Representation and Reasoning (KR-06)*. AAAI Press/The MIT Press, 2006.

[Del11] Chiara Del Vescovo. The modular structure of an ontology: Atomic decomposition towards applications. In *Proceedings of the 24th International Workshop on Description Logics (DL-11)*, volume 745 of *CEUR (`http://ceur-ws.org/`)*, 2011.

[DGK+11] Chiara Del Vescovo, Damian Gessler, Pavel Klinov, Bijan Parsia, Ulrike Sattler, Thomas Schneider, and Andrew Winget. Decomposition and modular structure of BioPortal ontologies. In *Proceedings of the 10th International Semantic Web Conference (ISWC-11)*, volume 7031 of *Lecture Notes in Computer Science*, pages 130–145, 2011.

[Dil50] Robert P. Dilworth. A decomposition theorem for partially ordered sets. *Annals of Mathematics*, 51(1):161–166, 1950.

[DKP+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 (WoMO-12)*, volume 875 of *CEUR (`http://ceur-ws.org/`)*, 2012.

[DPS11] Chiara Del Vescovo, Bijan Parsia, and Ulrike Sattler. Topicality in logic-based ontologies. In *Proceedings of the 19th International Conference on Conceptual Structures (ICCS-11)*, pages 187–200, 2011.

[DPS12] Chiara Del Vescovo, Bijan Parsia, and Ulrike Sattler. Logical relevance in ontologies. In *Proceedings of the 25th International Workshop on Description Logics (DL-12)*, volume 846 of *CEUR (`http://ceur-ws.org/`)*, 2012.

[DPSS10] Chiara Del Vescovo, Bijan Parsia, Ulrike Sattler, and Thomas Schneider. The modular structure of an ontology: an empirical study. In *Proceedings of the 23rd International Workshop on Description Logics (DL-10)*, volume 573 of *CEUR (`http://ceur-ws.org/`)*, 2010.

[DPSS11a] Chiara Del Vescovo, Bijan Parsia, Ulrike Sattler, and Thomas Schneider. The modular structure of an ontology: Atomic decomposition. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI-11)*, pages 2232–2237, 2011.

[DPSS11b] Chiara Del Vescovo, Bijan Parsia, Ulrike Sattler, and Thomas Schneider. The modular structure of an ontology: Atomic decomposition and module count. In *Proceedings of the 5th International Workshop on Modular Ontologies (WoMO-11)*, volume 230 of *Frontiers in Artificial Intelligence and Applications*, pages 25–39, 2011.

[GFH+11] Jennifer Golbeck, Gilberto Fragoso, Frank Hartel, Jim Hendler, Jim Oberthaler, and Bijan Parsia. The National Cancer Institute's Thesaurus and Ontology. *Web Semantics: Science, Services and Agents on the World Wide Web*, 1(1), 2011.

[GLW06] Silvio Ghilardi, Carsten Lutz, and Frank Wolter. Did I damage my ontology? A case for conservative extensions in Description Logics. In *Proceedings of the 10th International Conference on the Principles of Knowledge Representation and Reasoning (KR-06)*, pages 187–197. AAAI Press/The MIT Press, 2006.

[GPS11] Rafael S. Gonçalves, Bijan Parsia, and Ulrike Sattler. Analysing the evolution of the NCI thesaurus. In *Proceedings of the 24th IEEE International Symposium on Computer-Based Medical Systems (CBMS-11)*, page 76. IEEE Xplore Digital Library, 2011.

[GPS12] Rafael S. Gonçalves, Bijan Parsia, and Ulrike Sattler. Performance heterogeneity and approximate reasoning in description logic ontologies. In *Proceedings of the 11th International Semantic Web Conference (ISWC-12)*, volume 7649 of *Lecture Notes in Computer Science*, pages 82–98, 2012.

[HM01] Volker Haarslev and Ralf Möller. RACER system description. In *Proceedings of the 1st International Joint Conference on Automated Reasoning (IJCAR-01)*, volume 2083 of *Lecture Notes in Computer Science*, pages 701–705. Springer-Verlag, 2001.

[Hor11] Matthew Horridge. *Justification based Explanation in Ontologies*. PhD thesis, The University of Maryland, 2011.

[HPS11] Matthew Horridge, Bijan Parsia, and Ulrike Sattler. The state of bio-medical ontologies. In *Proceedings of the Bio-Ontologies SIG at ISMB 2011*, 2011.

[HPSvH03] Ian Horrocks, Peter F. Patel-Schneider, and Frank van Harmelen. From SHIQ and RDF to OWL: The making of a web ontology language. *Journal of Web Semantics*, 1(1):7–26, 2003.

[JCS+08] Ernesto Jiménez-Ruiz, Bernardo Cuenca Grau, Ulrike Sattler, Thomas Schneider, and Rafael Berlanga Llavori. Safe and economic re-use of ontologies: A logic-based methodology and tool support. In *Proceedings of the 5th European Semantic Web Conference (ESWC-08)*, volume 5021 of *Lecture Notes in Computer Science*, pages 185–199, 2008.

[Kal06] Aditya Kalyanpur. *Debugging and Repair of OWL ontologies*. PhD thesis, The University of Maryland, 2006.

[Kaz08] Yevgeny Kazakov. $\mathcal{RIQ}$ and $\mathcal{SROIQ}$ are harder than $\mathcal{SHOIQ}$. In *Proceedings of the 11th International Conference on the Principles of Knowledge Representation and Reasoning (KR-08)*, pages 274–284. AAAI Press, 2008.

[Kaz09] Yevgeny Kazakov. Consequence-driven reasoning for Horn $\mathcal{SHIQ}$ ontologies. In *Proceedings of the 22nd International Workshop on Description Logics (DL-09)*, volume 477 of *CEUR (`http://ceur-ws.org/`)*, 2009.

[KKS11] Yevgeny Kazakov, Markus Krötzsch, and Frantisek Simancik. Concurrent classification of $\mathcal{EL}$ ontologies. In *Proceedings of the 10th International Semantic Web Conference (ISWC-11)*, volume 7031 of *Lecture Notes in Computer Science*, pages 305–320, 2011.

[KLPW10] Boris Konev, Carsten Lutz, Denis Ponomaryov, and Frank Wolter. Decomposing Description Logic ontologies. In *Proceedings of the 12th International Conference on the Principles of Knowledge Representation and Reasoning (KR-10)*, pages 236–246. AAAI Press/The MIT Press, 2010.

[KLWW08] Boris Konev, Carsten Lutz, Dirk Walther, and Frank Wolter. Semantic modularity and module extraction in description logics. In *Proceedings of the 18th European Conference on Artificial Intelligence (ECAI-08)*, pages 55–59, 2008.

[KLWW09] Boris Konev, Carsten Lutz, Dirk Walther, and Frank Wolter. Formal properties of modularization. In Heiner Stuckenschmidt, Christine Parent, and Stefano Spaccapietra, editors, *Modular Ontologies: Concepts, Theories and Techniques for Knowledge Modularization*, volume 5445 of *Lecture Notes in Computer Science*, pages 25–66. Springer-Verlag, 2009.

[KLWZ04] Oliver Kutz, Carsten Lutz, Frank Wolter, and Michael Zakharyaschev. $\mathcal{E}$-connections of abstract description systems. *Artificial Intelligence*, 156(1):1–73, 2004.

[KPS+09] Roman Kontchakov, Luca Pulina, Ulrike Sattler, Thomas Schneider, Petra Selmer, Frank Wolter, and Michael Zakharyaschev. Minimal module extraction from DL-Lite ontologies using QBF solvers. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI-09)*, pages 836–841, 2009.

[KWZ10] Roman Kontchakov, Frank Wolter, and Michael Zakharyaschev. Logic-based ontology comparison and module extraction, with an application to DL-Lite. *Artificial Intelligence*, 174(15):1093–1141, 2010.

[LW10] Carsten Lutz and Frank Wolter. Deciding inseparability and conservative extensions in the description logic $\mathcal{EL}$. *Journal of Symbolic Computation*, 45(2):194–228, 2010.

[LWW07] Carsten Lutz, Dirk Walther, and Frank Wolter. Conservative extensions in expressive Description Logics. In *Proceedings of the 20th*

International Joint Conference on Artificial Intelligence (IJCAI-07), pages 453–458, 2007.

[Mik13] Eleni Mikroyannidi. *Detection of syntactic and semantic regularities in ontologies*. PhD thesis, The University of Manchester, 2013. To appear.

[MISR11] Eleni Mikroyannidi, Luigi Iannone, Robert Stevens, and Alan L. Rector. Inspecting regularities in ontology design using clustering. In *Proceedings of the 10th International Semantic Web Conference (ISWC-11)*, volume 7031 of *Lecture Notes in Computer Science*, pages 438–453, 2011.

[MMV11] Kodylan Moodley, Thomas A. Meyer, and Ivan José Varzinczak. Root justifications for ontology repair. In *Proceedings of the 5th International Conference on Web Reasoning and Rule Systems (RR-11)*, volume 6902 of *Lecture Notes in Computer Science*, pages 275–280. Springer-Verlag, 2011.

[MSH07] Boris Motik, Rob Shearer, and Ian Horrocks. A hypertableau calculus for $\mathcal{SHIQ}$. In *Proceedings of the 20th International Workshop on Description Logics (DL-07)*, pages 419–426. Bozen/Bolzano University Press, 2007.

[NM03] Natalya F. Noy and Mark A. Musen. The prompt suite: interactive tools for ontology merging and mapping. *International Journal of Human-Computer Studies*, 59(6):983–1024, 2003.

[NRG12] Nadeschda Nikitina, Sebastian Rudolph, and Birte Glimm. Interactive ontology revision. *Journal of Web Semantics*, 12:118–130, 2012.

[Par99] Rohit Parikh. Beliefs, belief revision, and splitting languages. In Lawrence S. Moss, Jonathan Ginzburg, and Maarten de Rijke, editors, *Logic, Language and Computation*, volume 2, pages 266–278. CSLI Publication, Cambridge University Press, 1999.

[Pon08] Denis Ponomaryov. On decomposability in logical calculi. *Bulletin of the Novosibirsk Computing Center*, 28:111–120, 2008.

[PS10] Bijan Parsia and Thomas Schneider. The modular structure of an ontology: an empirical study. In *Proceedings of the 12th International Conference on the Principles of Knowledge Representation and Reasoning (KR-10)*, pages 584–586. AAAI Press/The MIT Press, 2010.

[SCC97] Kent A. Spackman, Keith E. Campbell, and Roger A. Côté. SNOMED RT: A reference terminology for health care. *Journal of the American Medical Informatics Association*, pages 640–644, 1997.

[Sch03] Bernd S. W. Schröder. *Ordered Sets: An Introduction*. Birkhäuser, 2003.

[SMC74] Wayne P. Stevens, Glenford J. Myers, and Larry L. Constantine. Structured system. *IBM Systems Journal*, 13(2):115–139, 1974.

[SPC+07] Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur, and Yarden Katz. Pellet: A practical OWL-DL reasoner. *Journal of Web Semantics*, 5(2):51–53, 2007.

[SR06] Julian Seidenberg and Alan L. Rector. Web ontology segmentation: analysis, classification and use. In *Proceedings of the 15th International World Wide Web Conference (WWW-06)*, pages 13–22, 2006.

[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-09)*, volume 477 of *CEUR (`http://ceur-ws.org/`)*, 2009.

[TH06] Dmitry Tsarkov and Ian Horrock. FaCT++ Description Logic reasoner: System description. In *Proceedings of the 3rd International Joint Conference on Automated Reasoning (IJCAR-06)*, volume 4130 of *Lecture Notes in Computer Science*, pages 292–297. Springer-Verlag, 2006.

[TP12] Dmitry Tsarkov and Ignazio Palmisano. Divide et impera: Metareasoning for large ontologies. In *Proceedings of the 9th International Workshop on OWL: Experiences and Directions (OWLED-12)*, volume 849 of *CEUR (`http://ceur-ws.org/`)*, 2012.

[Tsa12] Dmitry Tsarkov. Improved algorithms for module extraction and atomic decomposition. In *Proceedings of the 25th International Workshop on Description Logics (DL-12)*, volume 846. CEUR (`http://ceur-ws.org/`), 2012.

[WPH06] Taowei D. Wang, Bijan Parsia, and James Hendler. A survey of the web ontology landscape. In *Proceedings of the 5th International Semantic Web Conference (ISWC-06)*, volume 4273 of *Lecture Notes in Computer Science*, pages 682–694. Springer-Verlag, 2006.