The modular structure of an ontology: atomic decomposition

February 5, 2011

Abstract

Extracting a subset of a given ontology that captures all the ontology's knowledge about a specified set of terms is a well-understood task. This task can be based, for instance, on locality-based modules. However, a single module does not allow us to understand neither topicality, connectedness, structure, or superfluous parts of an ontology, nor agreement between actual and intended modeling.

The strong logical properties of locality-based modules suggest that the family of all such modules of an ontology can support comprehension of the ontology as a whole. However, extracting that family is not feasible, since the number of locality-based modules of an ontology can be exponential w.r.t. its size.

In this paper we report on a new approach that enables us to efficiently extract a polynomial representation of the family of all locality-based modules of an ontology. We also describe the fundamental algorithm to pursue this task, and report on experiments carried out and results obtained.

1 Introduction

In software engineering, modularly structured systems are desirable, all other things being equal. Given a well-designed modular program, it is generally easier to process, modify, and analyze it and to reuse parts by exploiting the modular structure. As a result, support for modules (or components, classes, objects, packages, aspects) is a commonplace feature in programming languages.

Ontologies are computational artefacts akin to programs and, in notable examples, can get quite large and complex, which suggests that exploiting modularity might be fruitful. Research into modularity for ontologies has been an active area for ontology engineering. Recently, a lot of effort has gone into developing *logically sensible* modules, that is, modules which offer strong logical guarantees for intuitive modular properties [CHKS08]. One such guarantee is called *coverage* and means that the module captures all the ontology's knowledge about a given set of terms (signature)—a kind of dependancy isolation or encapsulation. This guarantee is provided by modules based on conservative extensions, but also by efficiently computable approximations, such as locality-based modules.

The task of extracting one such module given a signature (GetOne) is well understood and starting to be deployed in standard ontology development environments, such as Protégé 4,[1] and online.[2] The extraction of locality-based modules has already been effectively used in the field for ontology reuse [JJBR08] as well as a subservice for incremental reasoning [CHKS10]. Now GetOne requires the user to know in advance the right set of terms to input to the extractor: we call this a *seed* signature for the module and note that one module can have several such seed signatures. Since there are non-obvious relations between the final signature of a module and its seed signature, users are often unsure how to generate a proper request and confused by the results.

While GetOne is an important and useful service, it, by itself, tells us nothing about the modular structure of the ontology as a whole. The modular structure is determined by the set of *all* modules and their inter-relations, or at least a suitable subset thereof. The task of a-posteriori determining the modular structure of an ontology is called GetStruct, and the task of extracting all modules is called GetAll. While GetOne is well-understood and often computationally cheap, GetAll and GetStruct have not been examined for module notions with strong logical guarantees, with a few preliminary exceptions [CPSK06, DPSS10]. If ontology engineers had access to the overall modular structure of the ontology determined by GetStruct, they might be able to use it to guide their extraction choices and, supported by the experience described in [CPSK06], to understand its topicality, connectedness, structure, superfluous parts, or agreement between actual and intended modeling. For example, by inspecting the modular structure and observing un-connected parts that are intended to be connected, ontology designers could learn of weakly modeled parts of their ontology.

In the worst case, however, the number of all modules of an ontology is exponential in the number of terms or axioms in the ontology [DPSS10].

[1] `http://www.co-ode.org/downloads/protege-x`

[2] `http://owl.cs.manchester.ac.uk/modularity`

More importantly, even for very small ontologies, the number of all modules is far too large for them to be inspected by a user or even computed; e.g., Koala is an ontology with 42 axioms that has 3,660 modules, and GetAll fails on many ontologies less than one hundred axioms.

In this paper, we report on new insights regarding the modular structure of ontologies which leads to a new, polynomial algorithm for GetStruct (provided that module extraction is polynomial) that generates a linear (in the size of the ontology), partially ordered set of modules and *atoms* which succinctly represent *all* (potentially exponentially many) modules of an ontology. We also report on some experiments carried out with an implementation of this algorithm.

2 Related work

One solution to GetStruct is described in [CPSK06] via partitions related to $\mathcal{E}$-connections. When it succeeds, it divides an ontology into three kinds of modules: (A) those which import vocabulary (and axioms) from others, (B) those whose vocabulary (and axiom set) is imported, and (C) isolated parts. In various experiments, the numbers of parts extracted were quite low and the structure often corresponded usefully to user understanding. For instance, the tutorial ontology Koala, consisting of 42 logical axioms, is partitioned into one A-module about animals and three B-modules about genders, degrees and habitats.

It has also been shown in [CPSK06] that certain combinations of these parts provide coverage. For Koala, such a combination would still be the whole ontology (though smaller parts have coverage as well).

$\mathcal{E}$-connections are require rather strong conditions to ensure modular separation and have been observed to force together axioms and terms which are logically separable. As a consequence, it has been observed that ontologies with fairly elaborate modular structure have improverished $\mathcal{E}$-connections based structures: sometimes extraction resulted in a single partition even though the ontology seemed well structured. Furthermore, the robustness properties of the parts (e.g., under vocabulary extension) are not as well-understood as those of locality-based modules. Partitions ensure, however, a linear upper bound on the number of parts.

Differently, locality-based modules are finer-grained and can overlap. In principle this could lead to an exponential number (w.r.t. the size of the ontology) of modules; in [DPSS10] the trendline of the number of modules has been empirically studied for a selection of different ontologies, and the exponentiality of the family of modules seems to be confirmed by the experiments. However, the strong logical properties described in the introduction suggest that it is worth investigating if, similarly to $\mathcal{E}$-connections, a partition-based approach can be applied by using locality-based modules. The answer is positive under some mild requirements on properties of modules, as we are going to show throughout this paper.

Another approach to $\mathsf{GetStruct}$ is described in [BFZE08]. It underlies the tool ModOnto, which aims at providing support for working with ontology modules that borrows intuitions from software modules. To the best of our knowledge, however, it has not been examined whether such modules provide coverage in the above sense. Furthermore, ModOnto does not aim at obtaining *all* modules.

Another procedure for partitioning an ontology is described in [SK04]. However, this method only takes the concept hierarchy of the ontology into account and can therefore not provide the strong logical guarantee of coverage.

In [KLPW10], it was shown how to decompose the *signature* of an ontology to obtain the dependencies between its terms. In contrast to previous such approaches, this one is syntax-independent. While gaining information about term dependencies is one goal of our approach, we are also interested in the *modules* of the ontology.

Among the a-posteriori approaches to GetOne, some provide logical guarantees such as coverage, and others do not. The latter are not of interest for this paper. The former are usually restricted to DLs of low expressivity, where deciding conservative extensions—which underly coverage—is tractable. Prominent examples are the module extraction feature of CEL [Sun08] and the system MEX [KLWW08]. However, we aim at an approach that covers DLs up to OWL 2.

There are a number of logic-based approaches to modularity that function a-priori, i.e., the modules of an ontology have to be specified in advance by features that are added to the underlying (description) logic and whose semantics is well-defined. These approaches often support distributed reasoning; they include C-OWL [SvB$^+$04], $\mathcal{E}$-connections [KLWZ04], Distributed Description Logics [BS03], and Package-Based Description Logics [BVSH09]. Even in these cases, however, we may want to understand the modular structure of the syntactically delineated parts. Furthermore, with imposed structure, it is not always clear that that structure is correct. Decisions about modular structure have to be taken early in the modeling which may enshrine misunderstandings. Examples were reported in [CPSK06], where user attempts to capture the modular structure of their ontology by separating the axioms into separate files were totally at odds with the analysed structure.

3 Modules

We assume the reader to be familiar with Description Logics [BCM+03], and only briefly sketch here some of the central notions around locality-based modularity. We use $\mathcal{L}$ for a Description Logic, e.g., $\mathcal{SHIQ}$, and $\mathcal{O}, \mathcal{M}$, etc., for a knowledge base, i.e., a finite set of axioms. Moreover, we use $\widetilde{\mathcal{O}}$ for the signature of $\mathcal{O}$, i.e., the set of concept, role, and individual names used in $\mathcal{O}$.

Conservative extensions (CEs) are designed to capture the above described encapsulation of knowledge. They are defined as follows.

Definition 3.1. Let $\mathcal{L}$ be a DL, $\mathcal{M} \subseteq \mathcal{O}$ be $\mathcal{L}$-ontologies, and Σ be a signature.

1. $\mathcal{O}$ is a *deductive* Σ*-conservative extension* (Σ*-dCE*) of $\mathcal{M}$ w.r.t. $\mathcal{L}$ if for all axioms α over $\mathcal{L}$ with $\widetilde{\alpha} \subseteq \Sigma$, it holds that $\mathcal{M} \models \alpha$ if and only if $\mathcal{O} \models \alpha$.

2. $\mathcal{M}$ is a *dCE-based module for* Σ of $\mathcal{O}$ if $\mathcal{O}$ is a Σ-dCE of $\mathcal{M}$ w.r.t. $\mathcal{L}$.

Unfortunately, CEs are hard or even impossible to decide for many DLs, see [GLW06, KLWW09, SSZ09]. Therefore, approximations have been devised. We focus on *syntactic locality* (here for short: locality). Locality-based modules can be efficiently computed and provide coverage; that is, they capture *all* the relevant entailments, but not necessarily *only* those [CHKS08, JRCS+08]. Although locality is defined for the DL $\mathcal{SHIQ}$, it is straightforward to extend it to $\mathcal{SHOIQ}(D)$ (see [CHKS08, JRCS+08]), and a locality-based module extractor is implemented in the OWL API.[3] In what follows we introduce locality-based modules, and some of their properties.

Given an ontology $\mathcal{O}$ and a set of terms Σ we are interested in, we want to select from $\mathcal{O}$ the set of all axioms that "say something" about Σ. To efficiently extract such axioms, we define when an axiom is irrelevant (*local*) w.r.t. a signature Σ: this happens because there is a way to satisfy it trivially, i.e. an interpretation function that provides terms not in Σ with a "trivial" meaning [CHKS08].

Definition 3.2. An axiom α is called:

[3] http://owlapi.sourceforge.net/

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

$$C^{\bot} \sqsubseteq \mathtt{C},\ \mathtt{C} \sqsubseteq C^{\top},\ C^{\bot} \equiv C^{\bot},\ C^{\top} \equiv C^{\top},\ R^{\bot} \sqsubseteq \mathtt{R},\ \mathtt{Trans}(R^{\bot}),$$

where $\mathtt{C}$ means "arbitrary concept", $\mathtt{R}$ means "arbitrary role name", $R^{\bot}$ is a role name such that $R^{\bot} \notin \Sigma$, while $C^{\bot} \in \mathrm{Bot}(\Sigma)$ and $C^{\top} \in \mathrm{Top}(\Sigma)$ as defined in Table 1(a);

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

$$C^{\bot} \sqsubseteq \mathtt{C},\ \mathtt{C} \sqsubseteq C^{\top},\ C^{\bot} \equiv C^{\bot},\ C^{\top} \equiv C^{\top},\ \mathtt{R} \sqsubseteq R^{\top},\ \mathtt{Trans}(R^{\top}),$$

where $\mathtt{C}$ means "arbitrary concept", $\mathtt{R}$ means "arbitrary role name", $R^{\top}$ is a role name such that $R^{\top} \notin \Sigma$, while $C^{\bot} \in \mathrm{Bot}(\Sigma)$ and $C^{\top} \in \mathrm{Top}(\Sigma)$ as defined in Table 1(b).

(a) $\bot$-locality	Let $\overline{n} \in \mathbb{N} \setminus \{0\}$
$\mathrm{Bot}(\Sigma) ::= A^{\bot} \mid \bot \mid \neg C^{\top} \mid \mathtt{C} \sqcap C^{\bot} \mid C^{\bot} \sqcap \mathtt{C} \mid \exists \mathtt{R}.C^{\bot} \mid\ \leq \overline{n}\ \mathtt{R}.C^{\bot} \mid \exists R^{\bot}.\mathtt{C} \mid\ \leq \overline{n}\ R^{\bot}.\mathtt{C}$	
$\mathrm{Top}(\Sigma) ::= \top \mid \neg C^{\bot} \mid C_1^{\top} \sqcap C_2^{\top} \mid\ \leq 0\ \mathtt{R}.\mathtt{C}$	

(b) $\top$-locality	Let $\overline{n} \in \mathbb{N} \setminus \{0\}$
$\mathrm{Bot}(\Sigma) ::= \bot \mid \neg C^{\top} \mid \mathtt{C} \sqcap C^{\bot} \mid C^{\bot} \sqcap \mathtt{C} \mid \exists \mathtt{R}.C^{\bot} \mid\ \leq \overline{n}\ \mathtt{R}.C^{\bot}$	
$\mathrm{Top}(\Sigma) ::= A^{\top} \mid \top \mid \neg C^{\bot} \mid C_1^{\top} \sqcap C_2^{\top} \mid \exists R^{\top}.C^{\top} \mid\ \leq \overline{n}\ R^{\top}.C^{\top} \mid\ \leq 0\ \mathtt{R}.\mathtt{C}$	

Table 1: Syntactic locality conditions

The set of $\top$ (or $\bot$) locality-based axioms w.r.t. a set Σ of terms from $\widetilde{\mathcal{O}}$ is denoted by x-*local*$(\Sigma, \mathcal{O})$, where $x \in \{\top, \bot\}$.

Proposition 3.3. *Let $\mathcal{M} \subseteq \mathcal{O}$ two ontologies such that all axioms in $\mathcal{O} \setminus \mathcal{M}$ are $\bot$-local (or $\top$-local) w.r.t. $\Sigma \cup \widetilde{\mathcal{M}}$. Then, $\mathcal{O}$ is a Σ-mCE of $\mathcal{M}$.*

Proof. See [CHKS08]. □

A simple but useful property of locality is *anti-monotonicity*: the larger a seed signature Σ is, the smaller the set of local axioms is.

Corollary 3.4. *Let Σ_1 and Σ_2 be two sets of terms, and let $x \in \{\top, \bot\}$. Then, $\Sigma_1 \subseteq \Sigma_2$ implies x-local$(\Sigma_2) \subseteq x$-local(Σ_1).*

Proof. See [CHKS08]. □

Remark 3.5. Some obvious tautologies are always local axioms, for any choice of a seed signature Σ. Hence, they will not appear in locality-based modules. Anyway, they do not add any knowledge to an ontology $\mathcal{O}$.

A locality-based module can be computed as follows [CHKS08]: given an ontology $\mathcal{O}$, a signature $\Sigma \subseteq \widetilde{\mathcal{O}}$ and an empty set $\mathcal{M}$, every axiom $\alpha \in \mathcal{O}$ is tested whether it is Σ-local; if not, α is added to $\mathcal{M}$, and the signature Σ is extended with all terms in $\widetilde{\alpha}$, and the test is re-run against the extended signature. However, the resulting modules are sometimes quite large; for example, given the ontology $\mathcal{O} = \{\mathtt{C}_i \sqsubseteq \mathtt{D} \,|\, 1 \leq i \leq n\}$, the $\top$-module $\top$-mod$(\mathcal{O}, \Sigma)$ contains the whole ontology. In order to make modules smaller, we will nest alternatively $\bot$- and $\top$-locality on the previously extracted module: the resulting sets are again mCE-based modules, called $\bot\top$- or $\top\bot$-modules, depending on the type of the first extraction, $\bot$ in the first case, $\top$ in the second [SSZ09]. Moreover, 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 $\mathcal{O}$.

Lemma 3.6. *For every integer $n \leq 1$ and order of extraction $x \in \{\top\bot, \bot\top\}$, there exists an $\mathcal{ALC}$-module $\mathcal{M}_0$ of size $\mathcal{O}(n)$ and a signature Σ of size $\mathcal{O}(n)$ such that $\mathcal{M}_{i+1} = x$-mod$(\Sigma, \mathcal{M}_i)$, for each $i = 0, \ldots, n-1$, and $\mathcal{M}_0 \supset \cdots \supset \mathcal{M}_n$.*

Proof. See [SSZ09]. □

Notation 3.7. From now on, we will denote by x-mod$(\Sigma, \mathcal{O})$ the x-module $\mathcal{M}$ extracted from an ontology $\mathcal{O}$ by using the notion of x-locality w.r.t. Σ, where $x \in \{\top, \bot, \bot\top, \top\bot, \ldots, \top\bot^*, \bot\top^*\}$, including any alternate nesting of these symbols.

We list here some results about modules that will be used later.

Proposition 3.8. *The union of modules is not, in general, a module,*

Proof. Consider, for example, the ontology

$$\mathcal{O} = \{\mathtt{A} \sqsubseteq \mathtt{B}, \mathtt{B} \sqsubseteq \mathtt{C}, \mathtt{B} \sqsubseteq \mathtt{D}, \mathtt{C} \sqcap \mathtt{D} \sqsubseteq \mathtt{E}\}.$$

Then,

$$\begin{aligned} \top\bot^*\text{-mod}(\{\mathtt{A}, \mathtt{C}\}) &= \{\mathtt{A} \sqsubseteq \mathtt{B}, \mathtt{B} \sqsubseteq \mathtt{C}\} \\ \top\bot^*\text{-mod}(\{\mathtt{A}, \mathtt{D}\}) &= \{\mathtt{A} \sqsubseteq \mathtt{B}, \mathtt{B} \sqsubseteq \mathtt{D}\} \end{aligned}$$

but their union is not a module, because whenever we have both $\mathtt{C}$ and $\mathtt{D}$ in a seed signature, we get into the module also the axiom $\mathtt{C} \sqcap \mathtt{D} \sqsubseteq \mathtt{E}$. □

Proposition 3.9. *The intersection of modules is not, in general, a module.*

Proof. Consider, for example, the ontology

$$\mathcal{O} = \{\mathtt{A} \sqsubseteq \mathtt{B}, \mathtt{B} \sqcap \mathtt{C} \sqsubseteq \mathtt{D}, \mathtt{A} \sqsubseteq \mathtt{C}, \mathtt{A} \sqsubseteq \mathtt{D}\}.$$

Then,

$$\begin{aligned} \top\bot^*\text{-mod}(\{\mathtt{A}, \mathtt{B}, \mathtt{C}\}) &= \{\mathtt{A} \sqsubseteq \mathtt{B}, \mathtt{B} \sqcap \mathtt{C} \sqsubseteq \mathtt{D}, \mathtt{A} \sqsubseteq \mathtt{C}\} \\ \top\bot^*\text{-mod}(\{\mathtt{A}, \mathtt{B}, \mathtt{D}\}) &= \{\mathtt{A} \sqsubseteq \mathtt{B}, \mathtt{B} \sqcap \mathtt{C} \sqsubseteq \mathtt{D}, \mathtt{A} \sqsubseteq \mathtt{D}\} \end{aligned}$$

but their intersection is not a module, because both axioms $\mathtt{A} \sqsubseteq \mathtt{B}, \mathtt{B} \sqcap \mathtt{C} \sqsubseteq \mathtt{D}$ are in a module if, and only if, at least one of the axioms $\mathtt{A} \sqsubseteq \mathtt{C}, \mathtt{A} \sqsubseteq \mathtt{D}$ is in the module. □

Proposition 3.10. *The complement of a module is not, in general, a module.*

Proof. Consider, for example, the ontology

$$\mathcal{O} = \{\mathtt{A} \sqsubseteq \mathtt{B}, \mathtt{B} \sqcap \mathtt{C} \sqsubseteq \mathtt{A} \sqcup \mathtt{D}\}.$$

Then,

$$\top\bot^*\text{-mod}(\{\mathtt{A}, \mathtt{B}\}) = \{\mathtt{A} \sqsubseteq \mathtt{B}\}$$

But the set $\mathcal{O} \setminus \{\mathtt{A} \sqsubseteq \mathtt{B}\}$ made by the axiom $\mathtt{B} \sqcap \mathtt{C} \sqsubseteq \mathtt{A} \sqcup \mathtt{D}$ is not a module by itself. □

The following properties of locality-based modules will be of interest for our modularization.

Definition 3.11. Let $\mathcal{O}$ be an ontology, $\mathcal{M} \subseteq \mathcal{O}$ a module, and $\Sigma \subseteq \widetilde{\mathcal{O}}$ a signature.

- $\mathcal{M}$ is called *self-contained* if it is indistinguishable from $\mathcal{O}$ w.r.t. the set of terms in $\Sigma \cup \widetilde{\mathcal{M}}$.
- $\mathcal{M}$ is called *depleting* if the set of axioms in $\mathcal{O} \setminus \mathcal{M}$ is indistinguishable from the empty set w.r.t. Σ.

Proposition 3.12. *If S is an inseparability relation that is robust under replacement, then every depleting S_Σ-module is a self-contained S_Σ-module.*

Proof. See Proposition 4, [KPS+09]. □

Theorem 3.13. *Let S be a monotone inseparability relation that is robust under replacement, $\mathcal{T}$ a TBox, and Σ a signature. Then there is a unique minimal depleting S_Σ-module of $\mathcal{T}$.*

Proof. See Theorem 5, [KPS$^+$09]. □

Remark 3.14. From now on, we use the notion of $\top\bot^*$-locality from [SSZ09]. However, the results we obtain can be generalized to every notions of modules that guarantee the existence of a unique and depleting module for each signature Σ. In particular, the same conditions guarantee also that such notions of modules satisfy self-containedness.

4 Algebraic background

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

We will make use of *partial order relations*, defined as follows.

Definition 4.1. A *partial order relation* $\leq$ defined on a set X is a binary relation over the elements $x_1, x_2, \ldots$ of X satisfying 3 properties:

- $x_1 \leq x_2$ (reflexivity)
- $x_1 \leq x_2$ and $x_1 \neq x_2 \implies x_2 \not\leq x_1$ (antisymmetry)
- $x_1 \leq x_2$ and $x_2 \leq x_3 \implies x_1 \leq x_3$ (transitivity)

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

Definition 4.2. Two elements x, y of a poset $(X, \leq)$ are called *comparable* if $x \leq y$ or $y \leq x$. Otherwise they are *incomparable*.

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

Definition 4.3. Given a poset $(X, \leq)$, an element $x \in X$ is called *minimal* if there exists no element y of X with $y \leq x$.

Definition 4.4. For an element $x \in X$, the set $(x] := \{y \in X \mid y \leq x\}$ is called the *principal ideal* of x.

A less standard mathematical structure is the *field of sets*, defined as follows.

Definition 4.5. A *field of sets* is a pair (O, F), where O is a set and F is an algebra over O i.e., a non-empty subset of the power set of O closed under intersection, union and complement of sets. Elements of O are called *points* and are denoted by small letters as a, b, and those of F are called *complexes* and are denoted by capital letters such as M, N.

Definition 4.6. Given a finite set O and a family F of subsets of O, we can build the *induced field of sets* $B(O, F)$ by closing the family under union, intersection and complement.

Remark 4.7. $B(O, F)$ is obviously a field of sets as in Definition 4.5 and its elements are called *induced complexes*; $B(O, F)$ inherits a partial order relation defined by the inclusion relation "$\subseteq$", that satisfies the properties listed in Definition 4.1.

Definition 4.8. A field of sets $B(O, F)$ where minimal elements can not be defined is called *atomless*. Otherwise, the minimal elements of $B(O, F)$ with respect to the inclusion relation "$\subseteq$" are called *atoms*.

The minimal elements of the $B(O, F) \setminus \emptyset$ with respect to the inclusion relation "$\subseteq$" are called *atoms*.[4]

5 Modules and atoms

As we already say in Remark 3.14, we consider $\top\bot^*$-locality based modules, but the approach we present can be applied to any notion of a module that is monotonic, self-contained, and depleting, and we know from [KPS$^+$09] that robustness under replacement and depletingness implies self-containedness. In particular, these properties guarantee the uniqueness of a depleting module for any given signature [KPS$^+$09].

Moreover, to make the presentation easier, we assume, without loss of generality,[5] that

$$\mathcal{O}' = x\text{-mod}(\widetilde{\mathcal{O}}, \mathcal{O}) \setminus x\text{-mod}(\emptyset, \mathcal{O}). \tag{1}$$

Next, we are going to define a correspondence among ontologies with relative families of modules and fields of sets as defined in Definition 4.5. Axioms correspond to points; however, only some complexes correspond

[4]Slightly abusing notation, we use $B(O, F)$ here for the set of complexes in $B(O, F)$.

[5]We can always remove those unwanted axioms that occur in either all or no module, and consider them separately.

to modules since the family $\mathfrak{F}_x(\mathcal{O})$ of modules is not, in general, closed under union, intersection and complement: given two modules, neither their union nor their intersection nor the complement of a module is, in general, a module. Hence, we introduce the (induced) field of modules, that is the field of sets of the family $\mathfrak{F}_x(\mathcal{O})$ of modules. This enables us to use properties of fields of sets also for ontologies.

Definition 5.1. Given an ontology $\mathcal{O}$ and a notion of module x-mod$(_, \mathcal{O})$, we use $\mathfrak{F}_x(\mathcal{O})$ for the set of all x-modules of $\mathcal{O}$, and the *(induced) field of modules* $\mathcal{B}(\mathfrak{F}_x(\mathcal{O}))$ is the closure of the set $\mathfrak{F}_x(\mathcal{O})$ under union, intersection and complement.

We define *atoms* of our field of modules as blocks of modules of an ontology; recall that these are the $\subseteq$-minimal complexes of $\mathcal{B}(\mathfrak{F}_x(\mathcal{O})) \setminus \{\emptyset\}$.

Definition 5.2. Given a field of modules $\mathcal{B}(\mathfrak{F}_x(\mathcal{O}))$ over an ontology $\mathcal{O}$, its *atoms* $\{\mathfrak{a}, \mathfrak{b}, \ldots\}$ are the minimal complexes in the set $\mathcal{B}(\mathfrak{F}_x(\mathcal{O})) \setminus \{\emptyset\}$ w.r.t. the partial order relation "$\subseteq$". The family of atoms from $\mathcal{B}(\mathfrak{F}_x(\mathcal{O}))$ is denoted by $\mathcal{A}(\mathfrak{F}_x(\mathcal{O}))$ and is called *atomic decomposition*.

Remark 5.3. The existence of atoms is guaranteed by the finiteness of $\mathcal{O}$, because then the algebra $\mathcal{B}(\mathfrak{F}_x(\mathcal{O}))$ is finite and consequently not atomless. However, it is still possible that infinite ontologies induce a field of sets with atoms.

An atom is a set of axioms such that, for any module, it either contains all axioms in the atom or none of them. Moreover, every module is the union of atoms. Next, we show how atoms can provide a succinct representation of the family of modules. Before proceeding further, we summarize in Table 2 the four structures introduced so far and, for each, its elements, source, maximal size, and structure.

	$\mathcal{O}$	$\mathfrak{F}_x(\mathcal{O})$	$\mathcal{B}(\mathfrak{F}_x(\mathcal{O}))$	$\mathcal{A}(\mathfrak{F}_x(\mathcal{O}))$
elements	axioms	modules $\mathcal{M}_i$	complexes $\mathcal{K}_{i,j}$	atoms $\mathfrak{a}, \mathfrak{b}, \ldots$
source	ontology engineers	locality check	closure of $\mathfrak{F}_x(\mathcal{O})$	atoms of $\mathcal{B}(\mathfrak{F}_x(\mathcal{O}))$
maximal size	baseline	exponential	exponential	linear
structure	set	family of sets	complete lattice	poset

Table 2: 4 ways for looking at ontologies fragments

5.1 Atoms and their structure

The family $\mathcal{A}(\mathfrak{F}_x(\mathcal{O}))$ of atoms of an ontology, as in Definition 5.2, has many properties of interest for us.

Lemma 5.4. *The family $\mathcal{A}(\mathfrak{F}_x(\mathcal{O}))$ of atoms of an ontology $\mathcal{O}$ is a partition of $\mathcal{O}$, and thus $\#\mathcal{A}(\mathfrak{F}_x(\mathcal{O})) \leq \#\mathcal{O}$.*

Proof. Trivial, by Definition 4.8, and because atoms are disjoint by construction. □

Hence the atomic decomposition is *succinct*; we will see next whether its computation is tractable and whether it is indeed a representation of $\mathfrak{F}_x(\mathcal{O})$.

The following definition aims at defining a notion of "logical dependence" between axioms: the idea is that an axiom α depends on another axiom β if, whenever α occurs in a module $\mathcal{M}$ then β also belongs to $\mathcal{M}$. A slight extension of this argument allows us to generalize this idea because, by definition of atoms, whenever α occurs in a module, all axioms belonging to α's atom $\mathfrak{a}$ occur. Hence, we can formalize this idea by defining a relation between atoms.

Definition 5.5. (Relations between atoms) Let $\mathfrak{a}$ and $\mathfrak{b}$ be two distinct atoms of an ontology $\mathcal{O}$. Then:

- $\mathfrak{a}$ is *dependent* on $\mathfrak{b}$ (written $\mathfrak{a} \succeq \mathfrak{b}$) if, for every module $\mathcal{M} \in \mathfrak{F}_x(\mathcal{O})$ such that $\mathfrak{a} \subseteq \mathcal{M}$, we have $\mathfrak{b} \subseteq \mathcal{M}$.

- $\mathfrak{a}$ and $\mathfrak{b}$ are *independent* if there exist two disjoint modules $\mathcal{M}_1, \mathcal{M}_2 \in \mathfrak{F}_x(\mathcal{O})$ such that $\mathfrak{a} \subseteq \mathcal{M}_1$ and $\mathfrak{b} \subseteq \mathcal{M}_2$.

- $\mathfrak{a}$ and $\mathfrak{b}$ are *weakly dependent* if, they are neither independent, nor dependent, and if there exists an atom $\mathfrak{c} \in \mathcal{A}(\mathfrak{F}_x(\mathcal{O}))$ which both $\mathfrak{a}$ and $\mathfrak{b}$ are dependent on.

- $\mathfrak{a}, \mathfrak{b}$ form a *problematic pair* if none of the previous cases holds for $\mathfrak{a}$ and $\mathfrak{b}$.

Problematic pairs are undesirable because they can hide the logical dependence described above between atoms. As an example, let us consider a notion of modules that could determine the following family of modules: $\mathfrak{F}_x(\mathcal{O}) = \{\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3, \mathcal{M}_4\}$ where $\mathcal{M}_1 = \{\mathfrak{a}, \mathfrak{b}\}$, $\mathcal{M}_2 = \{\mathfrak{a}, \mathfrak{c}\}$, $\mathcal{M}_3 = \{\mathfrak{a}, \mathfrak{b}, \mathfrak{d}\}$ and $\mathcal{M}_4 = \{\mathfrak{a}, \mathfrak{c}, \mathfrak{d}\}$. Every module that contains $\mathfrak{d}$ contains also $\mathfrak{a}$, so we want to infer that $\mathfrak{d}$ depends on $\mathfrak{a}$. However, $\mathfrak{d}$ does not directly depends on $\mathfrak{a}$, since there is no module containing $\mathfrak{a}$ and $\mathfrak{d}$ but not $\mathfrak{b}$ or $\mathfrak{c}$. The underlying idea is that, in order to deal with the content of $\mathfrak{d}$, we need $\mathfrak{a}$ and something more, but we can not uniquely determine what this "something more" means. Fortunately, this does not happen to our modules.

Theorem 5.6. *If a notion of module x-mod is monotonic, self-contained, and depleting, there are no problematic pairs of atoms in the set $\mathcal{A}(\mathfrak{F}_x(\mathcal{O}))$ of atoms induced over an ontology $\mathcal{O}$ by x-mod.*

Key point for proving Theorem 5.6 is the following lemma.

Lemma 5.7. *Let x-mod be a notion of module as in Theorem 5.6 and $\mathfrak{a} \in \mathcal{A}(\mathfrak{F}_x(\mathcal{O}))$ an atom induced by x-mod. Then, for every nonempty set of axioms $\{\alpha_1, \ldots, \alpha_k\} \subseteq \mathfrak{a}$ we have that $x\text{-mod}(\{\widetilde{\alpha}_1, \ldots, \widetilde{\alpha}_k\}, \mathcal{O})$ is the smallest module containing $\mathfrak{a}$.*

Proof. Let $\alpha \in \mathfrak{a}$ be an axiom, and let us consider the module $\mathcal{M}_\alpha := x\text{-mod}(\widetilde{\alpha}, \mathcal{O})$. We recall x-mod is self-contained and monotonic. Then:

(1) $\mathcal{M}_\alpha$ is not empty since it contains α (recall Equation 1).

(2) $\mathcal{M}_\alpha \supseteq \mathfrak{a}$, by the definition of atoms.

(3) $\mathcal{M}_\alpha$ is the unique and thus smallest module for the seed signature $\widetilde{\alpha}$.

(4) by monotonicity, enlarging the seed signature $\widetilde{\alpha}$ results in a superset of $\mathcal{M}_\alpha$.

(5) by self-containedness and monotonicity, any module $\mathcal{M}'$ that contains α needs to contain also $\mathcal{M}_\alpha$, because

$$\mathcal{M}' = x\text{-mod}(\widetilde{\mathcal{M}'}, \mathcal{O}) = x\text{-mod}(\widetilde{\mathcal{M}'} \cup \widetilde{\alpha}, \mathcal{O}) \supseteq x\text{-mod}(\widetilde{\alpha}, \mathcal{O}).$$

(6) because of (2), we have that $\mathcal{M}_\alpha \supseteq x\text{-mod}(\widetilde{S}, \mathcal{O})$ for every non empty set of axioms $S = \{\alpha_1, \ldots, \alpha_k\} \subseteq \mathfrak{a}$; in particular, this holds if $S = \{\alpha_i\}$ for any $\alpha_i \in \mathfrak{a}$.

(7) by the arbitrarity of choice of α in $\mathfrak{a}$, we have that also the inverted inclusion $x\text{-mod}(\widetilde{\alpha}_i, \mathcal{O}) \supseteq \mathcal{M}_\alpha$ holds. □

Corollary 5.8. *Given an atom $\mathfrak{a}$, for any axiom $\alpha \in \mathfrak{a}$ we have $\mathcal{M}_\alpha = x\text{-mod}(\widetilde{\mathfrak{a}}, \mathcal{O})$. Moreover, $\mathfrak{a}$ is dependent on all atoms belonging to $\mathcal{M}_\alpha \setminus \mathfrak{a}$.*

Proof. *(of Theorem 5.6)*

Let $\mathfrak{a}$ and $\mathfrak{b}$ be two distinct atoms in $\mathcal{A}(\mathfrak{F}_x(\mathcal{O}))$; then, there exist two distinct axioms $\alpha \in \mathfrak{a}$ and $\beta \in \mathfrak{b}$. Let us consider the modules $\mathcal{M}_\alpha := X\text{-mod}(\widetilde{\alpha}, \mathcal{O})$ and $\mathcal{M}_\beta := X\text{-mod}(\widetilde{\beta}, \mathcal{O})$, with $\#\mathcal{M}_\alpha \leq \#\mathcal{M}_\beta$ without loss of generality. Then, by Lemma 5.7 $\mathcal{M}_\alpha$ (resp. $\mathcal{M}_\beta$) is the smallest module containing $\mathfrak{a}$ (resp. $\mathfrak{b}$).

Since $\mathfrak{a}$ and $\mathfrak{b}$ are distinct, we have that also $\mathcal{M}_\alpha$ and $\mathcal{M}_\beta$ are distinct. Then there are 3 possibilities:

- $\mathcal{M}_\alpha$ and $\mathcal{M}_\beta$ are disjoint. In this case, $\mathfrak{a}$ and $\mathfrak{b}$ are independent by definition.

- $\mathcal{M}_\alpha \subseteq \mathcal{M}_\beta$; then, since $\mathcal{M}_\beta$ is the smallest containing $\mathfrak{b}$, by monotonicity all modules containing $\mathfrak{b}$ will contain $\mathfrak{a}$, in which case $\mathfrak{b}$ is dependent on $\mathfrak{a}$ by definition.

- otherwise, the set $\mathcal{C} = \mathcal{M}_\alpha \cap \mathcal{M}_\beta$ is non empty. Then, by Lemma 5.7 $\mathfrak{a}$ and $\mathfrak{b}$ are dependent on the atoms belonging to $\mathcal{C}$, and consequently weakly dependent. □

Theorem 5.6 has interesting consequences on the dependency relation on atoms.

Proposition 5.9. *The binary relation "$\succeq$" as in Definition 5.5 is a partial order on the set $\mathcal{A}(\mathfrak{F}_x(\mathcal{O}))$ of atoms induced by a family of modules $\mathfrak{F}_x(\mathcal{O})$ of an ontology $\mathcal{O}$.*

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

- reflexivity: trivial;

- antisymmetry: let $\mathfrak{a}$ be dependent on $\mathfrak{b}$; that is, every module containing $\mathfrak{a}$, contains also $\mathfrak{b}$; now, we set also the inverse to hold, that is, every module containing $\mathfrak{b}$ contains also $\mathfrak{a}$. This means that it does not exist any module containing only one among $\mathfrak{a}$ and $\mathfrak{b}$; by construction of atoms, then, we have that $\mathfrak{a} \equiv \mathfrak{b}$.

- 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 module contains $\mathfrak{b}$, then it contains also $\mathfrak{c}$. Hence, $\mathfrak{a}$ is dependent on $\mathfrak{c}$.

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

Definition 5.10. Given a notion x-mod of module as above, we call *ordered atomic decomposition* the poset $(\mathcal{A}(\mathfrak{F}_x(\mathcal{O})), \succeq)$. Slightly abusing notation, the term "ordered" will be omitted when it will clear from the context that we are referring to the poset structure.

Definition 5.5 and Proposition 5.9 allow us to draw a Hasse diagram also for the atomic decomposition $\mathcal{A}(\mathfrak{F}_x(\mathcal{O}))$, where independent atoms belong to different chains, see Figure 1 for the Hasse diagramm of Koala. As an atom can be dependent on more that one atom; hence, we will have some nodes with more than one outgoing edge.

6 Atoms as a module base

Given an atomic decomposition as in Definition 5.10, we want to be able to recognize modules in it.

Lemma 6.1. *A module is a disjoint finite union of atoms.*

Proof. From construction of atoms as in Definition 5.2, we have that for any atom $\mathfrak{a} \in \mathcal{A}(\mathfrak{F}_x(\mathcal{O}))$, 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 6.2. *Every module $\mathcal{M}$ is determined by selecting in the atomic Hasse diagram one suitable antichain $\mathfrak{a}_1, \dots \mathfrak{a}_k$, $k \in \mathbb{N}$, and by taking the union of principal ideals of these atoms:*

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

Proof. From Lemma 6.1, 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 $(\mathfrak{a}]$. □

Proposition 6.3. *Arbitrary unions of atoms are not in general modules.*

Proof. Trivial, because arbitrary union of modules is not a module, as stated in Proposition 3.8. □

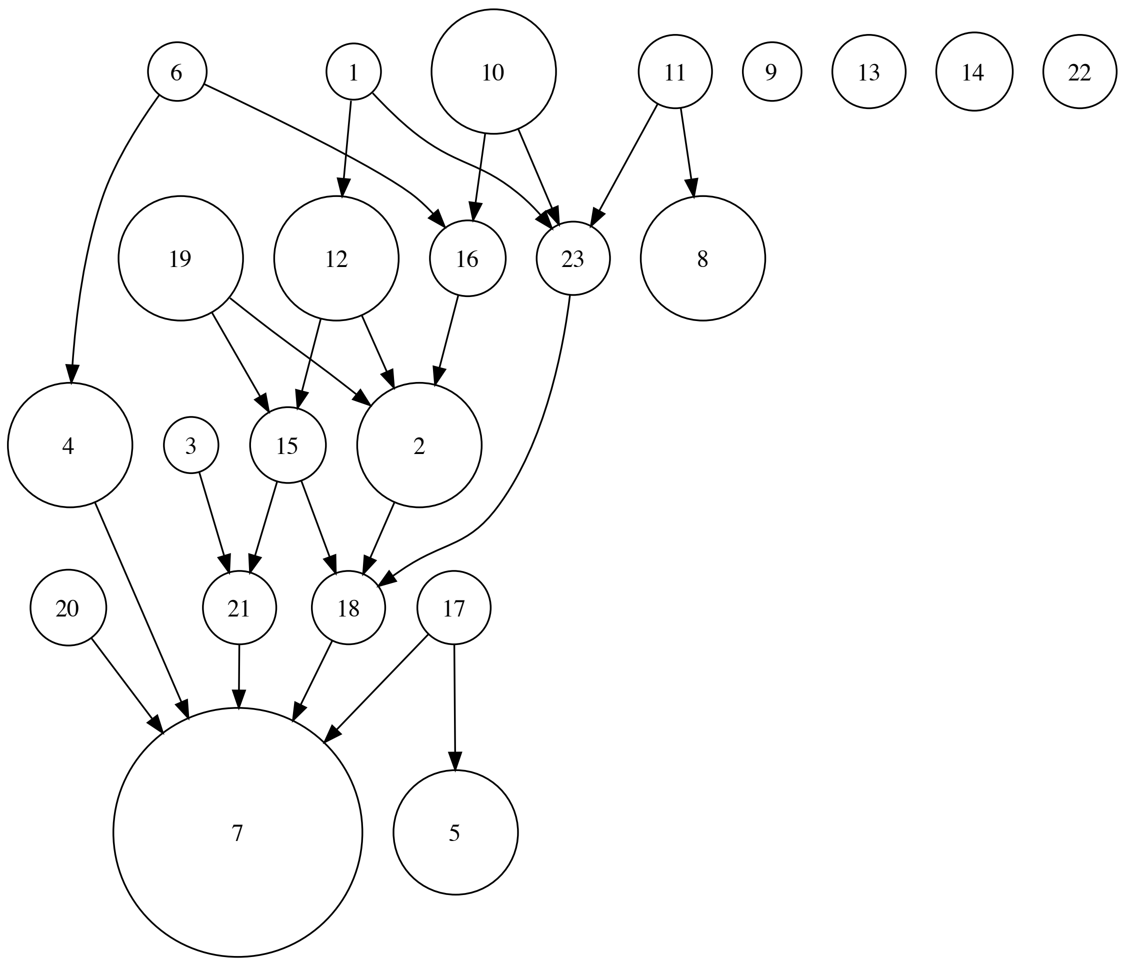

Figure 1: The atomic decomposition of Koala

Principal ideals of modules have also another nice property, that is, they are modules theirself; this is non trivial, since arbitrary intersections of modules are non in general modules, as seen in Proposition 3.9.

Proposition 6.4. *Principal ideals of atoms are modules.*

Proof. Given an atom $\mathfrak{a} \in \mathcal{A}(\mathfrak{F}_x(\mathcal{O}))$, we want to compare its principal ideal $(\mathfrak{a}] = \{\mathfrak{b} \mid \mathfrak{a} \succeq \mathfrak{b}\}$ with the module $\mathcal{M}_\alpha$. By the definition of atoms, $\mathcal{M}_\alpha \supseteq (\mathfrak{a}]$. We still need to prove that the equality holds. By contraposition, let $\mathcal{M}_\alpha$ be a proper superset of $(\mathfrak{a}]$. 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 Theorem 5.7, $\mathcal{M}_\beta$ is the smallest module containing $\mathfrak{b}$. Then, $\mathcal{M}_\beta$ is contained in $\mathcal{M}_\alpha$, 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. □

To sum up, on the one hand modules are built as union of suitable principal ideals of atoms in $\mathcal{A}(\mathfrak{F}_x(\mathcal{O}))$, and for every atom $\mathfrak{a}$, $(\mathfrak{a}]$ is a module. On the other hand, however, the converse of what stated in Proposition 6.5 is false: not all such unions of atoms are modules. We can, however, compute each module x-mod$(\Sigma, \mathcal{O})$ from $\mathcal{A}(\mathfrak{F}_x(\mathcal{O}))$, and thus the latter is indeed a succinct *representation* of all modules. For this computation, we need to store, with each atom $\mathfrak{a}$, the $\subseteq$-minimal seed signatures that lead to $(\mathfrak{a}]$: we say that an atom $\mathfrak{a}$ is *relevant* for Σ if there is a seed signature Σ' for $(\mathfrak{a}]$ with $\Sigma' \subseteq \Sigma$.

Proposition 6.5. *Let* $\mathfrak{a}_1, \dots \mathfrak{a}_k$, $k \in \mathbb{N}$, *be all atoms that are relevant for* Σ. *Then the module* x-mod$(\Sigma, \mathcal{O})$ *is the union of principal ideals of these atoms:* $x\text{-mod}(\Sigma, \mathcal{O}) = \bigcup_{i=1}^{k} (\mathfrak{a}_i]$.

7 Computing the atomic decomposition

As we have seen, the atomic decomposition is a succinct representation of all modules of an ontology: its linearly many atoms represent all its worst case exponentially many modules. Next, we will show how we can compute the atomic decomposition in polynomial time, i.e., without computing all modules, provided that module extraction is polynomial (which is the case, e.g., for syntactic locality-based modules). Our approach relies on modules "generated" by a single axioms, which can be used to generate all others.

Definition 7.1. A module $\mathcal{M}$ is called:

1) *compact* if there exists an atom $\mathfrak{a}$ in the atomic decomposition $\mathcal{A}(\mathfrak{F}_x(\mathcal{O}))$ such that $\mathcal{M} = (\mathfrak{a}]$.

2) *α-module* if there is an axiom $\alpha \in \mathcal{O}$ such that $\mathcal{M} = x\text{-mod}(\widetilde{\alpha}, \mathcal{O})$.

3) *fake* if there exist two uncomparable modules $\mathcal{M}_1 \neq \mathcal{M}_2$ with $\mathcal{M}_1 \cup \mathcal{M}_2 = \mathcal{M}$; a module is called *genuine* if it is not fake.

Please note that our notion of genuinity is different from the one in [PS10], where the uncomparable "building block" modules were also required to be disjoint.

The following lemma provides the basis for our polynomial algorithm for the computation of the atomic decomposition since it allows us to construct $\mathcal{A}(\mathfrak{F}_x(\mathcal{O}))$ via α-modules only.

Lemma 7.2. *The notions of compact, α and genuine modules coincide.*

Proof. We will prove that compact modules coincide both with α-modules and with genuine module.

1) $\Leftrightarrow$ 2) The equivalence has been already proven, and it follows from Corollary 5.8.

1) $\Rightarrow$ 3) By contraposition, let $\mathcal{M}$ be a fake module. Then there are two uncomparable modules $\mathcal{M}_1$ and $\mathcal{M}_2$ such that $\mathcal{M} = \mathcal{M}_1 \cup \mathcal{M}_2$. From Lemma 6.1, 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 uncomparable, 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 $\mathcal{M}_2$; that is, $\mathcal{M}$ is not compact.

3) $\Rightarrow$ 1) By contraposition, let $\mathcal{M}$ be a non compact module. Consequently, there exist atoms $\mathfrak{a}_1, \ldots, \mathfrak{a}_\kappa$ such that $\mathcal{M} = (\mathfrak{a}_1] \cup \ldots \cup (\mathfrak{a}_\kappa]$, 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 Lemma 6.4 we have that the principal ideal of every atom is a module. Hence $\mathcal{M} = (\mathfrak{a}_1] \cup \ldots \cup (\mathfrak{a}_\kappa]$ is a union of uncomparable modules, and more in specific, fake. $\square$

Algorithm 1 sketches our algorithm for computing atomic decompositions that runs in polynomial time in the size of $\mathcal{O}$ (provided that module extraction is polynomial), and calls a module extractor as many times as there are axioms in $\mathcal{O}$. It considers, in `ToDoAxioms`, all axioms that are neither tautologies nor global (as in Equation 1), and computes all genuine modules, all atoms with their dependency relation and, for each module and atom, their cardinality. For each axiom α "generating" a module, the algorithm stores that module in $\texttt{Module}(\alpha)$ and the corresponding atom is constructed in $\texttt{Atom}(\alpha)$; those functions are undefined for axioms outside `GeneratingAxioms`.

8 Empirical evaluation

We have run the atomic decomposition algorithm on a selection of ontologies, including those that were used in [DPSS10, PS10], and indeed managed to compute the atomic decomposition of all ontologies, even for ontologies for which a complete modularisation was not possible so far. Table 3 shows summary data for each ontology: size, expressivity, number of genuine modules, number of connected components, size of largest module and of largest atom. Our tests were obtained on a 2.16 GHz Intel Core 2 Duo Macbook with 2 GB of memory running Mac OS X 10.5.8; each atomic decomposition was computed within a couple of seconds, apart from that for Galen, which took less than 3 minutes.

We have also generated a graphical representation of our atomic decompositions which show atom size as node size, see Figure 1 for example. We notice that it shows four isolated atoms, e.g., Atom 22 in the top right corner, which consists of the axiom $\texttt{DryEucalyptForest} \sqsubseteq \texttt{Forest}$. This means that, even though other modules may use terms from Atom 22, they do not "need" the axioms in Atom 22 for any entailments; i.e., removing (the axioms in) these isolated atoms from the ontology would not result in the loss of any entailments regarding other modules or terms. Of course, for entailments regarding both `DryEucalyptForest` and `Forest` and possibly other terms, this axiom is required. A similar structure is observable in all ontologies considered apart from People and OWL-S: this indicates a greater cohesion and richness, which remains to be confirmed. The material on `http://bit.ly/i4olY0` includes graphs for all ontologies considered.

9 Conclusion and outlook

We have presented the *atomic decomposition* of an ontology, and shown how it is a succinct, tractable representation of the modular structure of an ontology: it is of polynomial size and can be computed in polynomial time in the size of the ontology (provided module extraction is polynomial),

Name	#Axioms	DL	#Gen. mods	#Con. comp.	#max. mod.	#max. atom
Koala	42	$\mathcal{ALCON}(\mathcal{D})$	23	5	18	7
Mereology	44	$\mathcal{SHIN}$	17	2	11	4
University	52	$\mathcal{SOIN}(\mathcal{D})$	31	11	20	11
People	108	$\mathcal{ALCHOIN}$	26	1	77	77
miniTambis	173	$\mathcal{ALCN}(\mathcal{D})$	129	85	16	8
OWL-S	277	$\mathcal{SHOIN}(\mathcal{D})$	114	1	57	38
Tambis	595	$\mathcal{ALCN}(\mathcal{D})$	369	119	236	61
Galen	4,528	$\mathcal{ALEHF}+$	3,340	807	458	29

Table 3: Experiments summary; only logical axioms are counted

whereas the number of modules of an ontology is exponential in the worst case and prohibitely large in cases so far investigated. Moreover, it can be used to assemble all other modules without touching the whole ontology and without invoking a direct module extractor.

Future work is three-fold: first, we will try to compute, from the atomic decomposition, good upper and lower bounds for the number of all modules to answer an open question from [PS10]. Second, we will investigate suitable labels for atoms, e.g., suitable representation of seed and module signatures, and how to employ the atomic decomposition for ontology engineering, e.g., to compare the modular structure with their intuitive understanding of the domain and thus detect modelling errors, and to identify suitable modules for reuse. Third, we will investigate when module extraction from the atomic decomposition is faster than extracting it using a module extractor.

Algorithm 1 Atomic decomposition algorithm

1: **Input:** Ontology $\mathcal{O}$ and suitable x-mod$(\cdot,\cdot)$

2: **Output:** The set $\mathfrak{G}$ of genuine modules; the poset of atoms $(\mathcal{A}(\mathfrak{F}_x(\mathcal{O})), \succeq)$; the set of generating axioms `GeneratingAxioms`; for $\alpha \in$ `GeneratingAxioms`, the cardinality `CardOfAtom`(α) of its atom.

3: `ToDoAxioms` $\leftarrow$ x-mod$(\widetilde{\mathcal{O}}, \mathcal{O}) \setminus x$-mod$(\emptyset, \mathcal{O})$
4: `GeneratingAxioms` $\leftarrow \emptyset$
5: **for** each $\alpha \in$ `ToDoAxioms` **do**
6:   `Module`(α) $\leftarrow$ x-mod$(\widetilde{\alpha}, \mathcal{O})$ % $\neq \emptyset$ due to line 3
7:   *new* $\leftarrow$ *true*
8:   **for** each $\beta \in$ `GeneratingAxioms` **do**
9:     **if** `Module`$(\alpha) =$ `Module`(β) **then**
10:       `Atom`(β) $\leftarrow$ `Atom`$(\beta) \cup \{\alpha\}$
11:       `CardOfAtom`(β) $\leftarrow$ `CardOfAtom`$(\beta) + 1$
12:       *new* $\leftarrow$ *false*
13:     **end if**
14:   **end for**
15:   **if** *new* $=$ *true* **then**
16:     `Atom`(α) $\leftarrow \{\alpha\}$
17:     `CardOfAtom`(α) $\leftarrow 1$
18:     `GeneratingAxioms` $\leftarrow$ `GeneratingAxioms` $\cup \{\alpha\}$
19:   **end if**
20: **end for**
21: **for** each $\alpha \in$ `GeneratingAxioms` **do**
22:   **for** each $\beta \in$ `GeneratingAxioms` **do**
23:     **if** $\beta \in$ `Module`(α) **then**
24:       `Atom`$(\beta) \succeq$ `Atom`(α)
25:     **end if**
26:   **end for**
27: **end for**
28: $\mathcal{A}(\mathfrak{F}_x(\mathcal{O}))$ $\leftarrow$ $\{$`Atom`$(\alpha) \mid \alpha \in$ `GeneratingAxioms`$\}$
29: $\mathfrak{G}$ $\leftarrow$ $\{$`Module`$(\alpha) \mid \alpha \in$ `GeneratingAxioms`$\}$
30: **return** $[(\mathcal{A}(\mathfrak{F}_x(\mathcal{O})), \succeq)$, $\mathfrak{G}$, `GeneratingAxioms`, `CardOfAtom`$(\cdot)]$

References

[AK00] G. Antoniou and K. Kehagias. A note on the refinement of ontologies. *Int. J. of Intelligent Systems*, 15, 2000.

[BCM+03] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.

[BFZE08] C. Bezerra, F. Freitas, A. Zimmermann, and J. Euzenat. ModOnto: A tool for modularizing ontologies. In F. Freitas, H. Stuckenschmidt, H. Pinto, A. Malucelli, and Ó. Corcho, editors, *Proc. WONTO-08*, volume 427 of *CEUR*. `CEUR-WS.org/Vol-427/`, 2008.

[BS03] A. Borgida and L. Serafini. Distributed description logics: Assimilating information from peer sources. *J. Data Semantics*, 1:153–184, 2003.

[BVSH09] J. Bao, G. Voutsadakis, G. Slutzki, and V. Honavar. Package-based description logics. In *Modular Ontologies*, volume 5445 of *Lecture Notes in Computer Science*, pages 349–371. Springer-Verlag, 2009.

[CHKS07a] B. Cuenca Grau, I. Horrocks, Y. Kazakov, and U. Sattler. Just the right amount: Extracting modules from ontologies. In C. Williamson, M. Zurko, P. Patel-Schneider, and P. Shenoy, editors, *Proc. of the 16th Int. WWW Conf. (WWW-07)*, pages 717–726. ACM, 2007.

[CHKS07b] B. Cuenca Grau, I. Horrocks, Y. Kazakov, and U. Sattler. A logical framework for modularity of ontologies. In *Proc. of the 20th Int. Joint Conf. on Artificial Intelligence (IJCAI-07)*, pages 298–304, 2007.

[CHKS08] B. Cuenca Grau, I. Horrocks, Y. Kazakov, and U. Sattler. Modular reuse of ontologies: Theory and practice. *Journal of Artificial Intelligence Research*, 31:273–318, 2008.

[CHKS09] B. Cuenca Grau, I. Horrocks, Y. Kazakov, and U. Sattler. Extracting modules from ontologies: A logic-based approach. In

H. Stuckenschmidt, C. Parent, and S. Spaccapietra, editors, *Modular Ontologies*, volume 5445 of *Lecture Notes in Computer Science*, pages 159–186. Springer-Verlag, 2009.

[CHKS10] B. Cuenca Grau, C. Halaschek-Wiener, Y. Kazakov, and B. Suntisrivaraporn. Incremental classification of description logics ontologies. *Journal of Automated Reasoning*, 44(4):337–369, 2010.

[CPS06] B. Cuenca Grau, B. Parsia, and E. Sirin. Combining OWL ontologies using $\mathcal{E}$-connections. *Journal of Web Semantics*, 4(1):40–59, 2006.

[CPSK06] B. Cuenca Grau, B. Parsia, E. Sirin, and A. Kalyanpur. Modularity and web ontologies. In P. Doherty, J. Mylopoulos, and C. A. Welty, editors, *Proc. of the 10th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR-06)*, pages 198–209. AAAI Press, 2006.

[DPSS10] C. Del Vescovo, B. Parsia, U. Sattler, and T. Schneider. The modular structure of an ontology: an empirical study. In *Proc. of the 2010 Description Logic Workshop (DL 2010)*. `CEUR-WS.org/Vol-573/`, 2010.

[GLW06] S. Ghilardi, C. Lutz, and F. Wolter. Did I damage my ontology? A case for conservative extensions in description logics. In P. Doherty, J. Mylopoulos, and C. A. Welty, editors, *Proc. of the 10th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR-06)*, pages 187–197. AAAI Press, 2006.

[Gru93] T. R. Gruber. Towards principles for the design of ontologies used for knowledge sharing. In N. Guarino and R. Poli, editors, *Formal Ontology in Conceptual Analysis and Knowledge Representation*. Kluwer Academic Publishers, 1993.

[GW98] B. Ganter and R. Wille. *Formal Concept Analysis: Mathematical Foundations*. Springer-Verlag, 1st edition, 1998.

[HKS06] I. Horrocks, O. Kutz, and U. Sattler. The even more irresistible $\mathcal{SROIQ}$. In *Proc. of the 10th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR-06)*, pages 57–67, 2006.

[HPS03] I. Horrocks and P. F. Patel-Schneider. Reducing owl entailment to description logic satisfiability. In *Journal of Web Semantics*, pages 17–29. Springer-Verlag, 2003.

[JJBR08] A. Jimeno, E. Jiménez-Ruiz, R. Berlanga, and D. Rebholz-Schuhmann. Use of shared lexical resources for efficient ontological engineering. In *Proc. of the Int. Workshop on Semantic Web Applications and Tools for Life Sciences (SWAT4LS-08)*, volume 435 of *CEUR*, 2008.

[JRCS+08] E. Jiménez-Ruiz, B. Cuenca Grau, U. Sattler, T. Schneider, and R. Berlanga Llavori. Safe and economic re-use of ontologies: A logic-based methodology and tool support. In S. Bechhofer, M. Hauswirth, J. Hoffmann, and M. Koubarakis, editors, *Proc. of the 5th Europ. Semantic Web Conf. (ESWC-08)*, volume 5021 of *Lecture Notes in Computer Science*, pages 185–199. Springer-Verlag, 2008.

[KLPW10] B. Konev, C. Lutz, D. Ponomaryov, and F. Wolter. Decomposing description logic ontologies. In F. Lin, U. Sattler, and M. Truszczynski, editors, *Proc. of the 12th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR-10)*, pages 236–246. AAAI Press, 2010.

[KLWW08] B. Konev, C. Lutz, D. Walther, and F. Wolter. Logical difference and module extraction with CEX and MEX. In *Proc. of the 2008 Description Logic Workshop (DL 2008)*, volume 353 of *CEUR*, 2008.

[KLWW09] B. Konev, C. Lutz, D. Walther, and F. Wolter. Formal properties of modularization. In H. Stuckenschmidt, S. Spacciapietra, and C. Parent, editors, *Ontology modularization*, volume 5445 of *Lecture Notes in Computer Science*, pages 25–66. 2009.

[KLWZ04] O. Kutz, C. Lutz, F. Wolter, and M. Zakharyaschev. $\mathcal{E}$-connections of abstract description systems. *Artificial Intelligence*, 156(1):1–73, 2004.

[KPS+09] R. Kontchakov, L. Pulina, U. Sattler, T. Schneider, P. Selmer, F. Wolter, and M. Zakharyaschev. Minimal module extraction from DL-Lite ontologies using QBF solvers. In *Proc. of the 21st Int. Joint Conf. on Artificial Intelligence (IJCAI-09)*, pages 836–841, 2009.

[LWW07] C. Lutz, D. Walther, and F. Wolter. Conservative extensions in expressive description logics. In M. M. Veloso, editor, *Proc. of the 20th Int. Joint Conf. on Artificial Intelligence (IJCAI-07)*, pages 453–458, 2007.

[Par72] D. L. Parnas. On the criteria to be used in decomposing systems into modules. *Communications of the ACM*, 15(12):1053–1058, 1972.

[Par99] R. Parikh. Beliefs, belief revision, and splitting languages. 2:266–278, 1999.

[PS10] B. Parsia and T. Schneider. The modular structure of an ontology: an empirical study. In F. Lin, U. Sattler, and M. Truszczynski, editors, *Proc. of the 12th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR-10)*, pages 584–586. AAAI Press, 2010.

[SG09] P. R. Halmos S. R. Givant. *Introduction to Boolean Algebras.* Undergr. texts in Math. Springer-Verlag, 1st edition, 2009.

[SK04] H. Stuckenschmidt and M. Klein. Structure-based partitioning of large concept hierarchies. In S. McIlraith, D. Plexousakis, and F. van Harmelen, editors, *Proc. of the 3rd Int. Semantic Web Conf. (ISWC-04)*, volume 3298 of *Lecture Notes in Computer Science*, pages 289–303. Springer-Verlag, 2004.

[SS91] M. Schmidt-Schauß and G. Smolka. Attributive concept descriptions with complements. *Artificial Intelligence*, 48(1):1–26, 1991.

[SSZ09] U. Sattler, T. Schneider, and M. Zakharyaschev. Which kind of module should I extract? In B. Cuenca Grau, I. Horrocks, B. Motik, and U. Sattler, editors, *Proc. of the 2009 Description Logic Workshop (DL 2009)*, volume 477 of *CEUR*. `CEUR-WS.org/Vol-477/`, 2009.

[Sun08] B. Suntisrivaraporn. Module extraction and incremental classification: A pragmatic approach for $\mathcal{EL}^+$ ontologies. In S. Bechhofer, M. Hauswirth, J. Hoffmann, and M. Koubarakis, editors, *Proc. of the 5th Europ. Semantic Web Conf. (ESWC-08)*, volume 5021 of *Lecture Notes in Computer Science*, pages 230–244. Springer-Verlag, 2008.

[SvB+04] H. Stuckenschmidt, F. van Harmelen, P. Bouquet, F. Giunchiglia, and L. Serafini. Using C-OWL for the alignment and merging of medical ontologies. In U. Hahn, editor, *Proc. KR-MED*, volume 102 of *CEUR*, pages 88–101. `CEUR-WS.org/Vol-102/`, 2004.