The Modular Structure of an Ontology: Atomic Decomposition and Module Count

Chiara DEL VESCOVO [a] Bijan PARSIA [a] Uli SATTLER [a] and Thomas SCHNEIDER [b]
[a] *University of Manchester, UK* {delvescc,bparsia,sattler}@cs.man.ac.uk
[b] *Universität Bremen, Germany* tschneider@informatik.uni-bremen.de

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.

Keywords. locality-based modules, decomposition, ontology comprehension

1. Introduction

Why modularize an ontology? Modern ontologies can get quite large as well as complex, which poses challenges to tools and users when it comes to processing, editing, analyzing them, or reusing their parts. This suggests that exploiting modularity of ontologies might be fruitful, and research into this topic has been an active area for ontology engineering. Much recent effort has gone into developing *logically sensible* modules, that is, parts of an ontology which offer strong logical guarantees for intuitive modular properties. One such guarantee is called *coverage*. It means that a module captures all the ontology's knowledge about a given set of terms (signature)—a kind of dependency isolation. A module in this sense is a subset of an ontology's axioms that provides coverage for a signature, and each possible signature determines such a module. Coverage is provided by modules based on conservative extensions, but also by efficiently computable approximations, such as modules based on syntactic locality [5].

We call the task of extracting one module given a signature GetOne; it is well understood and starting to be deployed in standard ontology development environments, such as Protégé 4,[1] and online.[2] Locality-based modules have already been effectively used for ontology reuse [14] and as a subservice for incremental reasoning [4].

[1] http://www.co-ode.org/downloads/protege-x
[2] http://owl.cs.manchester.ac.uk/modularity

Despite its usefulness, the service GetOne does not provide information about the ontology as a whole. It cannot help us to exploit an ontology as a one-piece of software, and understand its topicality, connectedness, structure, superfluous parts, or agreement between actual and intended modeling. To gain that understanding, we aim at revealing an ontology's modular structure, a task that we call GetStruct. That structure is determined by the set of *all* modules and their inter-relations, or at least a suitable subset thereof.

From a naïve point of view, a necessary step to achieve GetStruct is GetAll, the task of extracting *all* modules. This is the case as long as we have not specified what a "suitable subset of all modules" is, or do not know how to obtain such a subset efficiently. It might well be that GetAll is feasible and yields a small enough structure, in which case it would solve GetStruct. While GetOne is well-understood and often computationally cheap, GetAll has hardly been examined for module notions with strong logical guarantees, with the works described in [7, 8] being promising exceptions. GetOne also requires the user to know in advance the right signature to input to the extractor: we call this a *seed* signature for the module and note that each 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. If they had access to the overall modular structure of the ontology determined by GetStruct, they could use it to guide their extraction choices.

While GetAll seems to be a necessary step to perform GetStruct, we note that in the worst case, the number of all modules of an ontology is exponential in the number of terms or axioms in the ontology, in fact in the minimum of these numbers. In [20], we have shown cases of (artificial) ontologies with exponentially many modules w.r.t. their sizes, and obtained empirical results confirming that in general ontologies have too many modules to extract all of them, even with an optimized extraction methodology. Then, some other form of analysis would have to be designed.

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 use this decomposition to give an estimate of the number of modules of an ontology, and compare these numbers with the real number of modules (when possible), obtained following the same approach as in [20]. For full proofs and more details, the reader is referred to [9].

Related work. One solution to GetStruct is described in [7, 6] via partitions related to $\mathcal{E}$-connections. When this technique succeeds, it divides an ontology into three kinds of disjoint modules: (A) those which import vocabulary from others, (B) those whose vocabulary is imported, and (C) isolated parts. In experiments and user experience, the numbers of parts extracted were quite low and 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 [7] that certain combinations of these parts provide coverage. Partitions can be computed efficiently.

Ontology partitions based on $\mathcal{E}$-connections require rather strong conditions to ensure modular separation. However, it has been observed that ontologies with fairly elaborate modular structure have impoverished $\mathcal{E}$-connections based structures. For the onto-

logy Periodic,[3] for example, such a combination is still the whole ontology, even though Periodic seems well structured. Furthermore, the robustness properties of the parts (e.g., under vocabulary extension [17]) are not as well-understood as those of locality-based modules. Finally, there is only a preliminary implementation of the partition algorithm.

Among the other approaches to GetStruct we find the tool ModOnto [2], which aims at providing support for working with ontology modules, that borrows intuitions from software modules. This approach is logic-based and a-posteriori but, to the best of our knowledge, it has not been examined whether such modules provide coverage. Another procedure to partition an ontology is described in [22]. However, this method only takes the concept hierarchy into account, therefore it cannot guarantee to provide coverage.

In [15], it was shown how to decompose the *signature* of an ontology to obtain the dependencies between its terms. In contrast to the previous ones, this approach 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, only some provide logical guarantees. Those are usually restricted to "small" DLs where deciding conservative extensions—which underly coverage—is tractable. Examples are the module extraction feature of CEL [25] and the system MEX [16]. However, we want to cover DLs up to OWL 2.

There are several logic-based approaches to modularity that function a-priori, i.e., the modules of an ontology have to be specified in advance using 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 [24], $\mathcal{E}$-connections [19], Distributed Description Logics [3], and Package-Based Description Logics [1]. Even in these cases, however, we may want to understand the modular structure of the syntactically delineated parts (modules), because decisions about modular structure have to be taken early in the modeling which may enshrine misunderstandings. Currently there is no requirement that these modules provide coverage, so GetStruct can be useful to verify the imposed structure throughout the development process. Examples were reported in [7], where user attempts to capture the modular structure of their ontology by separating the axioms into separate files were totally at odds with the analyzed structure.

2. Preliminaries

Underlying description logics. We assume the reader to be familiar with OWL and the underlying description logics (DLs) [12, 11]. We consider an ontology to be a finite set of axioms, which are of the form $C \sqsubseteq D$ or $C \equiv D$, where C, D are (possibly complex) concepts, or $R \sqsubseteq S$, where R, S are (possibly inverse) roles. Since we are interested in the logical part of an ontology, we disregard non-logical axioms such as annotation and declaration axioms. However, it is easy to add those in retrospect once the logical part of a module has been extracted. This is included in the publicly available implementation of locality-based module extraction in the OWL API.[4]

Let $\mathsf{N_C}$ be a set of concept names, and $\mathsf{N_R}$ a set of role names. A *signature* Σ is a set of terms, i.e., $\Sigma \subseteq \mathsf{N_C} \cup \mathsf{N_R}$. We can think of a signature as specifying a topic of interest. Axioms using only terms from Σ are "on-topic". For instance, if $\Sigma = \{\mathsf{Animal}, \mathsf{Duck}, \mathsf{Grass}, \mathsf{eats}\}$, then $\mathsf{Duck} \sqsubseteq \exists \mathsf{eats}.\mathsf{Grass}$ is on-topic, while $\mathsf{Duck} \sqsubseteq \mathsf{Bird}$ is off-topic. Given an ontology $\mathcal{O}$ (axiom α), its signature is denoted with $\widetilde{\mathcal{O}}$ ($\widetilde{\alpha}$).

[3] http://www.cs.man.ac.uk/~stevensr/ontology/periodic.zip
[4] http://owlapi.sourceforge.net

Conservative extensions and locality. Conservative extensions (CEs) capture the above described encapsulation of knowledge. They are defined in [5] as follows.

Definition 2.1. *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 [10, 17]. Therefore, approximations have been devised. We focus on *syntactic locality* [21] (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 [5, 13]. Although locality is defined for the DL $\mathcal{SHIQ}$, an extension to $\mathcal{SHOIQ}(D)$ is straightforward [5, 13] and has been implemented in the OWL API. For the sake of completeness, we define locality and locality-based modules below. However, the atomic decomposition introduced later does not rely on locality because it will work for almost every notion of a "module for a signature".

Definition 2.2. *An axiom α is called* syntactically $\bot$-local ($\top$-local) w.r.t. signature Σ *if it is of the form $C^{\bot} \sqsubseteq C$, $C \sqsubseteq C^{\top}$, $R^{\bot} \sqsubseteq R$ ($R \sqsubseteq R^{\top}$), or $\mathsf{Trans}(R^{\bot})$ ($\mathsf{Trans}(R^{\top})$), where C is an arbitrary concept, R is an arbitrary role name, $R^{\bot} \notin \Sigma$ ($R^{\top} \notin \Sigma$), and $C^{\bot}$ and $C^{\top}$ are from* $\mathrm{Bot}(\Sigma)$ *and* $\mathrm{Top}(\Sigma)$ *as defined in Table (a) (Table (b)) below.*

(a) $\bot$-Locality — *Let* $A^{\bot}, R^{\bot} \notin \Sigma$, $C^{\bot} \in \mathrm{Bot}(\Sigma)$, $C^{\top}_{(i)} \in \mathrm{Top}(\Sigma)$, $\bar{n} \in \mathbb{N} \setminus \{0\}$

$$\mathrm{Bot}(\Sigma) ::= A^{\bot} \mid \bot \mid \neg C^{\top} \mid C \sqcap C^{\bot} \mid C^{\bot} \sqcap C \mid \exists R.C^{\bot} \mid \geqslant \bar{n} R.C^{\bot} \mid \exists R^{\bot}.C \mid \geqslant \bar{n} R^{\bot}.C$$
$$\mathrm{Top}(\Sigma) ::= \top \mid \neg C^{\bot} \mid C_1^{\top} \sqcap C_2^{\top} \mid \geqslant 0 R.C$$

(b) $\top$-Locality — *Let* $A^{\top}, R^{\top} \notin \Sigma$, $C^{\bot} \in \mathrm{Bot}(\Sigma)$, $C^{\top}_{(i)} \in \mathrm{Top}(\Sigma)$, $\bar{n} \in \mathbb{N} \setminus \{0\}$

$$\mathrm{Bot}(\Sigma) ::= \bot \mid \neg C^{\top} \mid C \sqcap C^{\bot} \mid C^{\bot} \sqcap C \mid \exists R.C^{\bot} \mid \geqslant \bar{n} R.C^{\bot}$$
$$\mathrm{Top}(\Sigma) ::= A^{\top} \mid \top \mid \neg C^{\bot} \mid C_1^{\top} \sqcap C_2^{\top} \mid \exists R^{\top}.C^{\top} \mid \geqslant \bar{n} R^{\top}.C^{\top} \mid \geqslant 0 R.C$$

It has been shown in [5] that $\mathcal{M} \subseteq \mathcal{O}$ and all axioms in $\mathcal{O} \setminus \mathcal{M}$ being $\bot$-local (or all axioms being $\top$-local) w.r.t. $\Sigma \cup \widetilde{M}$ is sufficient for $\mathcal{O}$ to be a Σ-dCE of $\mathcal{M}$. The converse does not hold: e.g., the axiom $A \equiv B$ is neither $\bot$- nor $\top$-local w.r.t. $\{A\}$, but the ontology $\{A \equiv B\}$ is an $\{A\}$-dCE of the empty ontology.

A locality-based module is computed as follows [5]: given an ontology $\mathcal{O}$, a seed signature $\Sigma \subseteq \widetilde{\mathcal{O}}$ and an empty set $\mathcal{M}$, each axiom $\alpha \in \mathcal{O}$ is tested whether it is local with respect to Σ; if not, α is added to $\mathcal{M}$, the signature Σ is extended with all terms in $\widetilde{\alpha}$, and the test is re-run against the extended signature until $\mathcal{M}$ is stable. $\mathcal{M}$ is denoted as $\top$-mod$(\Sigma, \mathcal{O})$ or $\bot$-mod$(\Sigma, \mathcal{O})$, respectively.

Sometimes the resulting modules are quite large; for example, given the ontology $\mathcal{O} = \{\mathtt{C}_i \sqsubseteq \mathtt{D} \mid 1 \leq i \leq n\}$, the module $\top$-mod$(\{\mathtt{D}\}, \mathcal{O})$ contains the whole ontology. In order to make modules smaller, we will nest alternatively $\bot$- and $\top$-module extraction. The resulting sets are again dCE-based modules, denoted $\bot\top$-mod$(\Sigma, \mathcal{O})$ or $\top\bot$-mod$(\Sigma, \mathcal{O})$, depending on the type of the first extraction [21]. We can keep nesting the extraction until a fixpoint is reached. The number of steps needed to reach it can be at most as big as the number of axioms in $\mathcal{O}$ [21].

The fixpoint, denoted as $\top\bot^*$-mod$(\Sigma,\mathcal{O})$, does not depend on the type of the first extraction [9]. In contrast, $\top$- and $\bot$-modules do not have to be equal—in fact, the former are usually larger than the latter. Through the nesting, $\top\bot^*$-mod$(\Sigma,\mathcal{O})$ is always contained in $\top$-mod$(\Sigma,\mathcal{O})$ and $\bot$-mod$(\Sigma,\mathcal{O})$.

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 $x \in \{\top,\bot,\bot\top,\top\bot,\ldots,\top\bot^*\}$, including any alternate nesting of these symbols. Finally, we want to point out that, for $\mathcal{M} = x$-mod$(\Sigma,\mathcal{O})$, neither $\Sigma \subseteq \widetilde{\mathcal{M}}$ nor $\widetilde{\mathcal{M}} \subseteq \Sigma$ needs to hold.

Properties of locality-based modules. We list in this paragraph the properties of locality-based modules of interest for this paper. Proofs can be found in the papers cited.

Proposition 2.3. *Let $\mathcal{O}$ be an ontology, Σ be a signature, $x \in \{\bot,\top,\top\bot^*\}$; let $\mathcal{M} = x$-mod$(\Sigma,\mathcal{O})$ and Σ' be a signature with $\Sigma \subseteq \Sigma' \subseteq \Sigma \cup \widetilde{\mathcal{M}}$. Then x-mod$(\Sigma',\mathcal{O}) = \mathcal{M}$.*
(For $x \in \{\bot,\top\}$, see [5]; the transfer to nested modules is straightforward).

Locality is *anti-monotonic*: a growing seed signature makes no more axioms local.

Corollary 2.4. *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) (see [5]).*

Remark 2.5. 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 $\mathcal{O}$.

Proposition 2.6. *In general, the following are not modules (see [9]): the union, intersection or complement of modules.*

Definition 2.7. *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 $\mathcal{O}$ is a $(\Sigma \cup \widetilde{\mathcal{M}})$-dCE of $\mathcal{M}$.*
$\mathcal{M}$ is called depleting *if $\mathcal{O} \setminus \mathcal{M}$ is a $(\Sigma \cup \widetilde{\mathcal{M}})$-dCE of the empty ontology.*

Proposition 2.8. *If S is an inseparability relation that is robust under replacement, then every depleting S_Σ-module is a self-contained S_Σ-module (see [18]).*

Theorem 2.9. *Let S be a monotonic 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 $\mathcal{T}$ (see [18]).*

Remark 2.10. From now on, we use the notion of $\top\bot^*$-locality from [21]. However, the results we obtain can be generalized to every notion of module that guarantees 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.

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

Definition 2.11. *A* field of sets *is a pair (O,F), where O is a set and F is an algebra over O i.e., set of subsets of O that is closed under intersection, union and complement. Elements of O are called* points*, while those of F are called* complexes*.*

Given a finite set O and a family F of subsets of O, we can build the set $B(O,F) = (O,F')$, where F' is the closure of F under union, intersection and complement. Then $B(O,F)$ is clearly a field of sets, as well as a partial order w.r.t. the inclusion relation "$\subseteq$", because $\subseteq$ is reflexive, transitive and antisymmetric. We focus on the *minimal* elements of $B(O,F)$, i.e., elements $a \in B(O,F)$ such that there exists no non-empty element b of $B(O,F) \setminus \{a\}$ with $b \subset a$.

Definition 2.12. *The minimal elements of $B(O,F) \setminus \{\emptyset\}$ with respect to "$\subseteq$" are called* atoms.[5] *The* principal ideal *of an element $a \in B(O,F)$ is the set*

$$(a] := \{x \in B(O,F) \mid x \subseteq a\}.$$

3. The Atomic Decomposition

Modules and atoms. In what follows, we are using the notion of $\top\bot^*$-locality from [21]. However, the approach we present can be applied to any notion of a module that is monotonic, self-contained, and depleting. These properties have a deep impact on the modules generated, as described in Proposition 3.1. See [18] for more details.

Proposition 3.1. *Any module notion that satisfies monotonicity, self-containedness, and depletingness is such that any given signature generates a unique module.*

We are going to define a correspondence among ontologies with relative families of modules and fields of sets as defined in Definition 2.11. Axioms correspond to points. Let then $\mathfrak{F}(\mathcal{O})$ denote the family of $\top\bot^*$-modules of $\mathcal{O}$ (or let $\mathfrak{F}_x(\mathcal{O})$ be such family for each corresponding notion x of module if not univocally specified). Then $\mathfrak{F}(\mathcal{O})$ 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, only some complexes correspond to modules. Next, we introduce the (induced) field of modules, that is the field of sets over $\mathfrak{F}(\mathcal{O})$. This enables us to use properties of fields of sets also for ontologies.

Definition 3.2. *Given an ontology $\mathcal{O}$ and the family $\mathfrak{F}(\mathcal{O})$ of $\top\bot^*$-modules of $\mathcal{O}$, we define the* (induced) field of modules $\mathcal{B}(\mathfrak{F}(\mathcal{O}))$ *as the closure of the set $\mathfrak{F}(\mathcal{O})$ under union, intersection and complement.*

Definition 3.3. *A* syntactic tautology *is an axiom that does not occur in any module and hence belongs to $\mathcal{O} \setminus \top\bot^*\text{-}mod(\emptyset,\mathcal{O})$. A* global axiom *is an axiom that occurs in each module, in particular in $\top\bot^*\text{-}mod(\emptyset,\mathcal{O})$.*

Remark 3.4. To make the presentation easier, we assume that $\mathcal{O}$ contains no syntactic tautologies or global axioms. This is no real restriction: we can always remove those unwanted axioms that occur in either all or no module, and consider them separately.

An (induced) field of modules is, by construction, a field of sets. It is partially ordered by $\subseteq$ and, due to the finiteness of $\mathcal{O}$, can thus be represented via its Hasse diagram. Next, we define *atoms* of our field of modules as building blocks of modules of an ontology; recall that these are the $\subseteq$-minimal complexes of $\mathcal{B}(\mathfrak{F}(\mathcal{O})) \setminus \{\emptyset\}$.

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

Definition 3.5. *The family of atoms from $\mathcal{B}(\mathfrak{F}(\mathcal{O}))$ is denoted by $\mathcal{A}(\mathfrak{F}(\mathcal{O}))$ and is called* atomic decomposition.

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 the following table the four structures introduced so far and, for each, its elements, source, maximal size, and mathematical structure.

Structure	$\mathcal{O}$	$\mathfrak{F}(\mathcal{O})$	$\mathcal{B}(\mathfrak{F}(\mathcal{O}))$	$\mathcal{A}(\mathfrak{F}(\mathcal{O}))$
Elements	axioms α	modules $\mathcal{M}$	complexes	atoms $\mathfrak{a}, \mathfrak{b}, \ldots$
Source	ontology engineers	module extractor	closure of $\mathfrak{F}(\mathcal{O})$	atoms of $\mathcal{B}(\mathfrak{F}(\mathcal{O}))$
Maximal size	baseline	exponential	exponential	linear
Mathem. object	set	family of sets	complete lattice	poset

Atoms and their structure. The family $\mathcal{A}(\mathfrak{F}(\mathcal{O}))$ of atoms of an ontology, as in Definition 3.5, has many properties of interest for us.

Lemma 3.6. *The family $\mathcal{A}(\mathfrak{F}(\mathcal{O}))$ of atoms of an ontology $\mathcal{O}$ is a partition of $\mathcal{O}$, and thus $\#\mathcal{A}(\mathfrak{F}(\mathcal{O})) \leq \#\mathcal{O}$.*

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

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 3.7. ***(Relations between atoms)*** *Let $\mathfrak{a} \neq \mathfrak{b}$ be atoms of an ontology $\mathcal{O}$. Then:*

- $\mathfrak{a}$ *is* dependent *on* $\mathfrak{b}$ *(written $\mathfrak{a} \succeq \mathfrak{b}$) if, for every module $\mathcal{M} \in \mathfrak{F}(\mathcal{O})$ 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 $\mathcal{M}_1, \mathcal{M}_2 \in \mathfrak{F}(\mathcal{O})$ 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; in this case, there exists an atom $\mathfrak{c} \in \mathcal{A}(\mathfrak{F}(\mathcal{O}))$ which both $\mathfrak{a}$ and $\mathfrak{b}$ are dependent on.*

We also define the relation "$\preceq$" to be the inverse of "$\succeq$", i.e., $\mathfrak{b} \preceq \mathfrak{a} \Leftrightarrow \mathfrak{a} \succeq \mathfrak{b}$.

Proposition 3.8. *For every pair of distinct atoms exactly one of the relations in Definition 3.7 applies.*

The logical dependence between atoms can, in general, be incomplete: for example, consider the following (hypothetical) family of modules: $\mathfrak{F}(\mathcal{O}) = \{\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3, \mathcal{M}_4\}$ where $\mathcal{M}_1 = \mathfrak{a} \cup \mathfrak{b}$, $\mathcal{M}_2 = \mathfrak{a} \cup \mathfrak{c}$, $\mathcal{M}_3 = \mathfrak{a} \cup \mathfrak{b} \cup \mathfrak{d}$ and $\mathcal{M}_4 = \mathfrak{a} \cup \mathfrak{c} \cup \mathfrak{d}$. Following Definition 3.7, the atoms $\mathfrak{b}$, $\mathfrak{c}$ and $\mathfrak{d}$ depend on $\mathfrak{a}$. However, we want our structure to reflect that $\mathfrak{b}$ and $\mathfrak{c}$ act as "intermediates" in the dependency of $\mathfrak{d}$ on $\mathfrak{a}$, i.e., that $\mathfrak{d}$ depends via "$\mathfrak{c}$ or $\mathfrak{b}$" on $\mathfrak{a}$. Since in Def. 3.7 we do not capture disjunctions of occurrences of atoms, we call the pairs $(\mathfrak{d}, \mathfrak{b})$ and $(\mathfrak{d}, \mathfrak{c})$ *problematic*. Fortunately, problematic atom pairs do not exist

in an atomic decomposition obtained via locality-based modules, as Lemma 3.9 shows. Its consequences on the dependency relation on atoms are captured by Proposition 3.12.

Lemma 3.9. *Since the $\top\bot^*$ notion of module is monotonic, self-contained, and depleting, there are no problematic pairs in the set $\mathcal{A}(\mathfrak{F}(\mathcal{O}))$ of atoms over $\mathcal{O}$.*

The key to proving Lemma 3.9 is the following result.

Proposition 3.10. *The module $\top\bot^*$-mod$(\widetilde{\alpha}, \mathcal{O})$ is the smallest containing α.*

Proof. We recall $\top\bot^*$-mod satisfies the properties as in Prop. 2.3. Then:
(i) $\mathcal{M}_\alpha$ is not empty since it contains α (recall that $\mathcal{O}$ does not contain syntactic tautologies)
(ii) $\mathcal{M}_\alpha$ is the unique and thus smallest module for the seed signature $\widetilde{\alpha}$
(iii) by monotonicity, enlarging the seed signature $\widetilde{\alpha}$ results in a superset of $\mathcal{M}_\alpha$
(iv) $\mathcal{M}' = \top\bot^*\text{-mod}(\widetilde{\mathcal{M}'}, \mathcal{O}) = \top\bot^*\text{-mod}(\widetilde{\mathcal{M}'} \cup \widetilde{\alpha}, \mathcal{O}) \supseteq \top\bot^*\text{-mod}(\widetilde{\alpha}, \mathcal{O})$ by self-containedness and monotonicity, thus any module $\mathcal{M}'$ that contains α needs to contain also $\mathcal{M}_\alpha$. □

Corollary 3.11. *Given an atom $\mathfrak{a}$, for every axiom $\alpha \in \mathfrak{a}$ we have that $\mathcal{M}_\alpha = \top\bot^*$-mod$(\widetilde{\mathfrak{a}}, \mathcal{O})$. Moreover, $\mathfrak{a}$ is dependent on all atoms belonging to $\mathcal{M}_\alpha \setminus \mathfrak{a}$.*

Proposition 3.12. *The binary relations "$\succeq$" and "$\preceq$" are partial orders over the set $\mathcal{A}(\mathfrak{F}(\mathcal{O}))$ of atoms of an ontology $\mathcal{O}$.*

Definition 3.7 and Proposition 3.12 allow us to draw a Hasse diagram also for the atomic decomposition $\mathcal{A}(\mathfrak{F}(\mathcal{O}))$, where independent atoms belong to different chains, see Figure 1 for the Hasse diagram of Koala. The edges in this diagram denote dependency: an edge from node $\mathfrak{a}$ to $\mathfrak{b}$ means that $\mathfrak{b} \preceq \mathfrak{a}$, i.e., atom $\mathfrak{a}$ depends on $\mathfrak{b}$. Some atoms depend on more than one atom. Their nodes have more than one outgoing edge.

Atoms as a module base. As an immediate consequence of our observations so far, a module is a disjoint finite union of atoms. Conversely, it is not true that arbitrary unions of atoms are modules. However, the atomic decomposition satisfies another interesting property: from each atom, it is straightforward to identify the smallest module containing it.

Definition 3.13. *The* principal ideal of an atom $\mathfrak{a}$ *is the set* $(\mathfrak{a}] = \{\alpha \in \mathfrak{b} \mid \mathfrak{b} \preceq \mathfrak{a}\} \subseteq \mathcal{O}$.

Proposition 3.14. *For every atom $\mathfrak{a}$, $(\mathfrak{a}]$ is a module.*

Definition 3.15. *A module is called* compact *if there exists an atom $\mathfrak{a}$ in $\mathcal{A}(\mathfrak{F}(\mathcal{O}))$ such that $\mathcal{M} = (\mathfrak{a}]$.*

Given the (possibly exponential w.r.t. the ontology size) family $\mathfrak{F}(\mathcal{O})$ of all modules of an ontology $\mathcal{O}$, there is a well-defined injection that maps every module $\mathcal{M}$ to the set of atoms in $\wp(\mathcal{A}(\mathfrak{F}(\mathcal{O})))$ whose union is $\mathcal{M}$: given the module signature $\widetilde{\mathcal{M}}$, its image is the set of all atoms that are *relevant w.r.t. $\mathcal{M}$'s terminology*, defined in the following. Hence, $\mathcal{A}(\mathfrak{F}(\mathcal{O}))$ is indeed a succinct *representation* of all modules.

Definition 3.16. *We say that an atom* $\mathfrak{a}$ *is* relevant w.r.t. its terminology *for a module* $\mathcal{M}$ *if its signature* $\widetilde{\mathfrak{a}}$ *is contained in the module's signature* $\widetilde{\mathcal{M}}$.

The well-definedness of Def. 3.16 follows from the properties of depletingness and self-containedness that locality-based modules satisfy. We can however restrict our attention to just some relevant (w.r.t. its terminology) atoms to identify our module within the atomic decomposition.

Definition 3.17. *Let* $(P,\geq)$ *be a poset, and* $(P,\leq)$ *its dual. Then, an* antichain *is a set of pairwise incomparable elements* $A \subseteq P$*, i.e. such that for each* $a,b \in A$*, neither* $a \geq b$ *nor* $b \geq a$ *(dually, neither* $a \leq b$ *nor* $b \leq a$*).*

Proposition 3.18. *Let* $\mathcal{M} \subseteq \mathcal{O}$ *be a module. Then, there exists an antichain of atoms* $\mathfrak{a}_1,\ldots,\mathfrak{a}_\kappa$ *such that* $\mathcal{M} = \bigcup_{i=1}^{\kappa} (\mathfrak{a}_i]$.

In particular, the set of compact modules is a base for the set $\mathfrak{F}(\mathcal{O})$ of all modules.

4. 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 4.1. *Given an ontology* $\mathcal{O}$ *and decomposition* $\mathcal{A}(\mathfrak{F}(\mathcal{O}))$*, we call module* $\mathcal{M}$*:*

1) α-module *if there is an axiom* $\alpha \in \mathcal{O}$ *such that* $\mathcal{M} = \top\bot^*\text{-}mod(\widetilde{\alpha},\mathcal{O})$.
2) fake *if there exist two incomparable (w.r.t. set inclusion) 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 [20], where the incomparable "building" modules were also required to be disjoint.

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

Lemma 4.2. *The notions of compact (as in Def. 3.15),* α *and genuine modules coincide.*

Algorithm 1 gives our procedure for computing atomic decompositions that runs in time polynomial 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 `ToDoAx`, all axioms that are neither tautologies nor global, see Remark 3.4, and computes all genuine modules, all atoms with their dependency relation and the cardinalities of all modules and atoms. For each axiom α "generating" a module, that module is stored in $\texttt{Mod}(\alpha)$ and the corresponding atom is constructed in $\texttt{At}(\alpha)$; those functions are undefined for axioms outside `GenAx`. We prove the correctness of Algorithm 1 in [9].

Algorithm 1 Atomic decomposition

Input: An ontology $\mathcal{O}$.
Output: The set $\mathfrak{G}$ of genuine $\top\bot^*$-modules; the poset of atoms $(\mathcal{A}(\mathfrak{F}(\mathcal{O})), \succeq)$; the set of generating axioms $\mathtt{GenAx}$; for $\alpha \in \mathtt{GenAx}$, the cardinality $\mathtt{CardAt}(\alpha)$ of its atom.

$\mathtt{ToDoAx} \leftarrow \top\bot^*\text{-mod}(\widetilde{\mathcal{O}}, \mathcal{O}) \setminus \top\bot^*\text{-mod}(\emptyset, \mathcal{O})$
$\mathtt{GenAx} \leftarrow \emptyset$
for each $\alpha \in \mathtt{ToDoAx}$ **do**
  $\mathtt{Mod}(\alpha) \leftarrow \top\bot^*\text{-mod}(\widetilde{\alpha}, \mathcal{O})$ $\{\neq \emptyset\}$
  $new \leftarrow true$
  for each $\beta \in \mathtt{GenAx}$ **do**
    if $\mathtt{Mod}(\alpha) = \mathtt{Mod}(\beta)$ **then**
      $\mathtt{At}(\beta) \leftarrow \mathtt{At}(\beta) \cup \{\alpha\}$
      $\mathtt{CardAt}(\beta) \leftarrow \mathtt{CardAt}(\beta) + 1$
      $new \leftarrow false$
    end if
  end for
  if $new = true$ **then**
    $\mathtt{At}(\alpha) \leftarrow \{\alpha\}$
    $\mathtt{CardAt}(\alpha) \leftarrow 1$
    $\mathtt{GenAx} \leftarrow \mathtt{GenAx} \cup \{\alpha\}$
  end if
end for
for each $\alpha \in \mathtt{GenAx}$ **do**
  for each $\beta \in \mathtt{GenAx}$ **do**
    if $\beta \in \mathtt{Mod}(\alpha)$ **then**
      $\mathtt{At}(\beta) \succeq \mathtt{At}(\alpha)$
    end if
  end for
end for
$\mathcal{A}(\mathfrak{F}(\mathcal{O})) \leftarrow \{\mathtt{At}(\alpha) \mid \alpha \in \mathtt{GenAx}\}$
$\mathfrak{G} \leftarrow \{\mathtt{Mod}(\alpha) \mid \alpha \in \mathtt{GenAx}\}$
return $[(\mathcal{A}(\mathfrak{F}(\mathcal{O})), \succeq), \mathfrak{G}, \mathtt{GenAx}, \mathtt{CardAt}(\cdot)]$

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

Table 1. Experiments summary

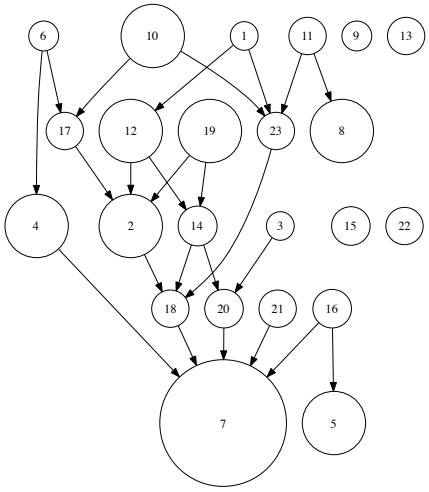


Figure 1. The atomic decomposition of Koala

5. Empirical evaluation

We ran Algorithm 1 on a selection of ontologies[6], including those used in [8, 20], and indeed managed to compute the atomic decomposition in all cases, even for ontologies where a complete modularization was previously impossible. Table 1 summarizes ontology data: 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, (resp. 3 minutes for Galen).

We have also generated a graphical representation using GraphViz[7]. Our atomic decompositions show atom size as node size, see e.g. Fig. 1. It shows four isolated atoms,

[6] Ontologies and their decompositions can be found at `http://bit.ly/i4olY0`.
[7] `http://www.graphviz.org/About.php`

e.g., Atom 22, consisting of the axiom $\texttt{DryEucalyptForest} \sqsubseteq \texttt{Forest}$. This means that, although other modules may use 22's terms, they do not "need" 22's axioms for any entailment. Hence, removing (the axioms in) isolated atoms from the ontology would not result in the loss of any entailments regarding other modules or terms. Of course, for entailments involving both DryEucalyptForest and Forest and possibly other terms, axioms in isolated atoms may be needed. A similar structure is observable in all ontologies considered, see the graphs at http://bit.ly/i4olY0 .

6. Labelling

The atomic decomposition partitions the ontology into highly coherent fragments. However, we still need to understand their structure and access their content. To this aim, it can be useful to label an atom with the terms that we find relevant. An obvious candidate is simply the signature of the corresponding genuine module. However, genuine modules, and hence their signatures, can be too numerous, as well as unstructured. Another candidate is suggested by Proposition 3.18: we could label an atom $\mathfrak{a}$ with the set of all its minimal seed signatures for which $\mathfrak{a}$ is relevant. As before, a genuine module can have in principle a large number of such signatures, even more numerous than the number of axioms it contains. So, we suggest here different candidates for a labelling and discuss them; but we leave applying them for future work.

Definition 6.1. *Given: an ontology $\mathcal{O}$; the atomic decomposition of the ontology $\mathcal{A}(\mathfrak{F}(\mathcal{O})) = \{\mathfrak{a}_1, \mathfrak{a}_2, \ldots, \mathfrak{a}_n\}$; the set of genuine modules $\mathfrak{G} = \{\mathcal{M}_i \mid \mathcal{M}_i = (\mathfrak{a}_i], 1 \leq i \leq n\}$. We define the following* labelling functions $\mathsf{Lab}_j(.)$ *from $\mathcal{A}(\mathfrak{F}(\mathcal{O}))$ to $\widetilde{\mathcal{O}}$:*

$\mathsf{Lab}_1(\mathfrak{a}_i) := \widetilde{\mathfrak{a}_i}$ $\qquad$ $\mathsf{Lab}_3(\mathfrak{a}_i) := \bigcup_{\Sigma \in \mathtt{mssig}(\mathcal{M}_i)} \Sigma$

$\mathsf{Lab}_2(\mathfrak{a}_i) := \widetilde{\mathfrak{a}_i} \setminus \bigcup_{\mathfrak{b} \prec \mathfrak{a}_i} \mathsf{Lab}_2(\mathfrak{b})$ $\qquad$ $\mathsf{Lab}_4(\mathfrak{a}_i) := \bigcup_{\Sigma \in \mathtt{mssig}(\mathcal{M}_i)} \Sigma \setminus \bigcup_{\mathfrak{b} \prec \mathfrak{a}_i} \mathsf{Lab}_4((\mathfrak{b}])$

Lab_1 is defined to label each atom with the vocabulary used in its axioms. However, an atom $\mathfrak{a}$ can be large and reuse terms already introduced in the atoms that $\mathfrak{a}$ is dependent on. To better represent the "logical dependency" between terms, we recursively define Lab_2 to label an atom only with the "new terms" introduced.

We want to note that such label can be empty, as in the following example: let us consider the ontology $\mathcal{O} = \{\mathtt{A} \sqsubseteq \mathtt{B}, \mathtt{C} \sqsubseteq \mathtt{D}, \mathtt{A} \sqcap \mathtt{C} \sqsubseteq \mathtt{B} \sqcup \mathtt{D}\}$. This ontology generates 3 atoms, one for each axiom, such that the atom $\mathfrak{a}_3 = \mathtt{A} \sqcap \mathtt{C} \sqsubseteq \mathtt{B} \sqcup \mathtt{D}$ is dependent on both the other 2, which are independent of each other. Clearly, $\mathsf{Lab}_2(\mathfrak{a}_3)$ is empty, because $(\mathfrak{a}_3]$ reuses terms from the other atoms. Moreover, let us consider the axiom $\mathtt{A} \sqsubseteq \mathtt{B} \sqcap (\mathtt{C} \sqcup \neg \mathtt{C})$. Then, all the labelling defined so far will include the term C in the label for the atom containing this axiom, even if this axiom does not say anything about it.

This behaviour does not occur for labellings Lab_3 and Lab_4, because C is not necessary in any of the minimal seed signatures for $(\mathfrak{a}_3]$. Moreover, these labellings are also useful to discover "hidden relations" between an atom and terms that do not occur in it. For example, let us consider the ontology $\mathcal{O} = \{\mathtt{A} \equiv \mathtt{B}, \mathtt{B} \sqsubseteq \mathtt{C}, \mathtt{B} \sqcap \mathtt{D} \sqsubseteq \mathtt{C} \sqcup \mathtt{E}, \mathtt{D} \sqsubseteq \mathtt{E}, \mathtt{E} \equiv \mathtt{F}\}$. Then, each axiom identifies an atom, and $\mathcal{O}$ equals the principal ideal of the atom $\mathfrak{a}_3$ containing the axiom $\mathtt{B} \sqcap \mathtt{D} \sqsubseteq \mathtt{C} \sqcup \mathtt{E}$. Although the signature of $\mathfrak{a}_3$ contains neither A nor F, the set $\Sigma = \{\mathtt{A}, \mathtt{F}\}$ is indeed a minimal seed signature of the genuine module $(\mathfrak{a}_3]$. The need of this axiom for the signature Σ is not evident at first sight. However, the set of all minimal seed signatures of a module $\mathcal{M}$ is in principle exponential in the size of $\widetilde{\mathcal{M}}$.

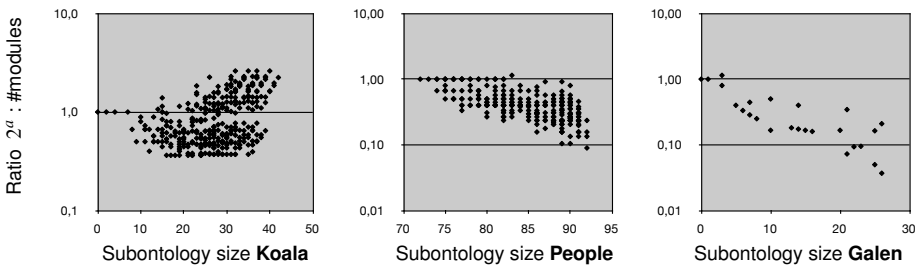
7. Module number estimation via atomic decomposition

In order to test the hypothesis that the number of modules does not grow exponentially with the size of the ontology, in [8] we tried to compute a full modularization for the ontologies of different size listed in Table 1 but managed to compute all modules for two ontologies only, namely Koala and Mereology. Then, we sampled subontologies of these ontologies, and extracted all of their modules. The results we obtained made us tend towards rejecting the hypothesis, but they were not strong enough for a clear rejection.

One plausible application of the atomic decomposition is an estimate of the number of modules of an ontology: Proposition 3.18 implies that a module is the union of principal ideals of the atoms over an antichain. In general, the converse does not hold, but *prima facie* this seems to be a reasonable approximation, and can help us in understanding whether or not the number of modules is exponential w.r.t. the size of the ontology: as a matter of fact, if all antichains of an atomic decomposition 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 a of the maximal antichain and compute 2^a.

Unfortunately, the measure 2^a is not always a lower bound of the actual number of modules. For example, consider the ontology $\mathcal{O} = \{\mathtt{A}_i \sqsubseteq \mathtt{A}_{i+1} \mid i = 0,\ldots,n-1\}$, which consists of a single subsumption path p. The atomic decomposition of $\mathcal{O}$ consists of n independent atoms: $\top\bot^*\text{-mod}(\{\mathtt{A}_i,\mathtt{A}_{i+1}\},\mathcal{O}) = \{\mathtt{A}_i \sqsubseteq \mathtt{A}_{i+1}\}$, for every $i = 0,\ldots,n-1$. Hence, the maximal antichain is of size n, and we would estimate that $\mathcal{O}$ has 2^n modules. However, the modules of $\mathcal{O}$ are all subpaths of p: for seed signatures Σ of size < 2, $\top\bot^*\text{-mod}(\Sigma,\mathcal{O}) = \emptyset$; for all other Σ, $\top\bot^*\text{-mod}(\Sigma,\mathcal{O})$ is the smallest subpath of p containing all concepts in Σ. 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 they share parts of the minimal seed signatures of their induced modules.

Based on the module numbers from that previous experiment, we have now performed an atomic decomposition of all the subontologies, computed the length a of the maximal antichain as well as the ratio between 2^a and the number of modules for the respective ontology. If that ratio is greater (less) than 1, then the value 2^a overestimates (underestimates) the module number. The picture below contains plots of the measured ratios against the subontology size for 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 $x = 2^a$ that $\mathcal{O}$ has between $0.333x$ and $2x$ modules. The quotient is 6; therefore we can estimate the 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.

Mereology. The observations are similar, with a slight imbalance towards ratios < 1. The min and max ratio are 0.40 and 1.42, yielding a quotient of only 3.55.

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 module number is only up to two orders of magnitude. For example, for a maximal antichain of length 10, the number of modules can now be between 898 and 11,378. Furthermore, the underestimation appears to grow with the ontology size.

University. The ratio is evenly distributed and ranges from 0.25 to 5.35. The quotient of 21.4 is even larger than for People.

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 module number. For some ontologies, the measured value 2^a tends to underestimate the module numbers, for others, there is no tendency. For some ontologies, the margin for the estimate obtained from 2^a is simply too large.

8. 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), 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, more precise upper and lower bounds for the number of all modules to answer an open question from [20]. Second, we will continue to 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 engineers' intuitive understanding of the domain and thus detect modelling errors, and to identify suitable modules for reuse. Third, we will investigate when module extraction using the atomic decomposition is faster than using a module extractor.

References

[1] J. Bao, G. Voutsadakis, G. Slutzki, and V. Honavar. Package-based description logics. In [23], pp. 349–371.

[2] C. Bezerra, F. Freitas, A. Zimmermann, and J. Euzenat. ModOnto: A tool for modularizing ontologies. In *Proc. WONTO-08*, vol. 427 of `ceur-ws.org`, 2008.

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

[4] B. Cuenca Grau, C. Halaschek-Wiener, and Y. Kazakov. History matters: Incremental ontology reasoning using modules. In *Proc. ISWC-07*, vol. 4825 of *LNCS*, pp. 183–196, 2007.

[5] B. Cuenca Grau, I. Horrocks, Y. Kazakov, and U. Sattler. Modular reuse of ontologies: Theory and practice. *J. of Artif. Intell. Research*, 31:273–318, 2008.

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

[7] B. Cuenca Grau, B. Parsia, E. Sirin, and A. Kalyanpur. Modularity and web ontologies. In *Proc. of KR-06*, pp. 198–209. AAAI Press, 2006.

[8] C. Del Vescovo, B. Parsia, U. Sattler, and T. Schneider. The modular structure of an ontology: an empirical study. In *Proc. of WoMO-10*, vol. 211 of *FAIA*, pp. 11–24. IOS Press, 2010.

[9] C. Del Vescovo, B. Parsia, U. Sattler, and T. Schneider. The modular structure of an ontology: atomic decomposition. Technical report, University of Manchester, 2011. Available at `http://bit.ly/i4olY0`.

[10] S. Ghilardi, C. Lutz, and F. Wolter. Did I damage my ontology? A case for conservative extensions in description logics. In *Proc. of KR-06*, pp. 187–197, 2006.

[11] I. Horrocks, O. Kutz, and U. Sattler. The even more irresistible $\mathcal{SROIQ}$. In *Proc. of KR-06*, pp. 57–67, 2006.

[12] I. Horrocks, P. F. Patel-Schneider, and F. van Harmelen. From $\mathcal{SHIQ}$ and RDF to OWL: The making of a web ontology language. *J. of Web Sem.*, 1(1):7–26, 2003.

[13] 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 *Proc. of ESWC-08*, vol. 5021 of *LNCS*, pp. 185–199, 2008.

[14] A. Jimeno, E. Jiménez-Ruiz, R. Berlanga, and D. Rebholz-Schuhmann. Use of shared lexical resources for efficient ontological engineering. In *SWAT4LS-08*, `ceur-ws.org`, 2008.

[15] B. Konev, C. Lutz, D. Ponomaryov, and F. Wolter. Decomposing description logic ontologies. In *Proc. of KR-10*, pp. 236–246. AAAI Press, 2010.

[16] B. Konev, C. Lutz, D. Walther, and F. Wolter. Logical difference and module extraction with CEX and MEX. In *Proc. of DL 2008*, vol. 353 of `ceur-ws.org`, 2008.

[17] B. Konev, C. Lutz, D. Walther, and F. Wolter. Formal properties of modularization. In [23], pp. 25–66.

[18] 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 IJCAI-09*, pp. 836–841, 2009.

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

[20] B. Parsia and T. Schneider. The modular structure of an ontology: an empirical study. In *Proc. of KR-10*, pp. 584–586. AAAI Press, 2010.

[21] U. Sattler, T. Schneider, and M. Zakharyaschev. Which kind of module should I extract? In *DL 2009*, vol. 477 of `ceur-ws.org`, 2009.

[22] H. Stuckenschmidt and M. Klein. Structure-based partitioning of large concept hierarchies. In *Proc. of ISWC-04*, vol. 3298 of *LNCS*, pp. 289–303. Springer-Verlag, 2004.

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

[24] H. Stuckenschmidt, F. van Harmelen, P. Bouquet, F. Giunchiglia, and L. Serafini. Using C-OWL for the alignment and merging of medical ontologies. In *Proc. KR-MED*, `ceur-ws.org`, pp. 88–101, 2004.

[25] B. Suntisrivaraporn. Module extraction and incremental classification: A pragmatic approach for $\mathcal{EL}^+$ ontologies. In *Proc. of ESWC-08*, vol. 5021 of *LNCS*, pp. 230–244, 2008.