

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

Chiara Del Vescovo¹, Bijan Parsia¹, Uli Sattler¹, and Thomas Schneider²

¹ School of Computer Science, University of Manchester, UK
{delvescc, bparsia, sattler}@cs.man.ac.uk

² Department of Computer Science, 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.

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 4,³ and online.⁴ Locality-based modules have already been effectively used for ontology reuse [13] and as a subservice for incremental reasoning [4].

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 [19], 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 [19]. For data generated during the experiments, the reader is referred to <http://bit.ly/i4o1Y0> .

³ <http://www.co-ode.org/downloads/protege-x>

⁴ <http://owl.cs.manchester.ac.uk/modularity>

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 ontology Periodic,⁵ for instance, such a combination is still the whole ontology, even though the ontology seems 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. 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 [21]. However, this method only takes the concept hierarchy into account, therefore it cannot guarantee to provide coverage.

In [14], 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 [24] and the system MEX [15]. 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 [23], \mathcal{E} -connections [18], 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 require-

⁵ <http://www.cs.man.ac.uk/~stevensr/ontology/periodic.zip>

ment 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) [11, 10]. 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. However, it is easy to add the corresponding annotation and declaration axioms 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.⁶

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

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

Definition 1. *Let \mathcal{L} be a DL, $\mathcal{M} \sqsubseteq \mathcal{O}$ 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 GCI axioms α over \mathcal{L} with $\tilde{\alpha} \subseteq \Sigma$, 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 [9, 16]. Therefore, approximations have been devised. We focus on *syntactic locality* [20] (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, 12]. Although locality is defined for the DL \mathcal{SHIQ} , an extension to $\mathcal{SHOIQ}(D)$ is straightforward [5, 12] and has been implemented in the OWL API.

Definition 2. *An axiom α is called syntactically \perp -local (\top -local) w.r.t. signature Σ if it is of the form $C^\perp \sqsubseteq C$, $C \sqsubseteq C^\top$, $R^\perp \sqsubseteq R$ ($R \sqsubseteq R^\top$), or $\text{Trans}(R^\perp)$ ($\text{Trans}(R^\top)$), where C is an arbitrary concept, R is an arbitrary role name, $R^\perp \notin \Sigma$ ($R^\top \notin \Sigma$), and C^\perp and C^\top are from $\text{Bot}(\Sigma)$ and $\text{Top}(\Sigma)$ as defined in Table (a) (Table (b)) below.*

⁶ <http://owlapi.sourceforge.net>

$$\begin{array}{l}
\text{(a) } \perp\text{-Locality} \quad \text{Let } A^\perp, R^\perp \notin \Sigma, C^\perp \in \text{Bot}(\Sigma), C_{(i)}^\top \in \text{Top}(\Sigma), \bar{n} \in \mathbb{N} \setminus \{0\} \\
\text{Bot}(\Sigma) ::= A^\perp \mid \perp \mid \neg C^\top \mid C \sqcap C^\perp \mid C^\perp \sqcap C \mid \exists R.C^\perp \mid \geq \bar{n} R.C^\perp \mid \geq \bar{n} R^\perp.C \\
\text{Top}(\Sigma) ::= \top \mid \neg C^\perp \mid C_1^\top \sqcap C_2^\top \mid \geq 0 R.C \\
\hline
\text{(b) } \top\text{-Locality} \quad \text{Let } A^\top, R^\top \notin \Sigma, C^\perp \in \text{Bot}(\Sigma), C_{(i)}^\top \in \text{Top}(\Sigma), \bar{n} \in \mathbb{N} \setminus \{0\} \\
\text{Bot}(\Sigma) ::= \perp \mid \neg C^\top \mid C \sqcap C^\perp \mid C^\perp \sqcap C \mid \geq \bar{n} R.C^\perp \\
\text{Top}(\Sigma) ::= A^\top \mid \top \mid \neg C^\perp \mid C_1^\top \sqcap C_2^\top \mid \geq \bar{n} R^\top.C^\top \mid \geq 0 R.C \\
\hline
\end{array}$$

It has been shown in [5] that $\mathcal{M} \subseteq \mathcal{O}$ and all axioms in $\mathcal{O} \setminus \mathcal{M}$ being \perp -local (or all axioms being \top -local) w.r.t. $\Sigma \cup \widetilde{\mathcal{M}}$ 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 \perp - 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; 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. Sometimes the resulting modules are quite large; for example, given the ontology $\mathcal{O} = \{C_i \sqsubseteq D \mid 1 \leq i \leq n\}$, the module $\top\text{-mod}(D, \mathcal{O})$ contains the whole ontology. In order to make modules smaller, we will nest alternatively \perp - and \top -locality on the previously extracted module: the resulting sets are again mCE-based modules, called $\perp\top$ - or $\top\perp$ -modules, depending on the type of the first extraction [20]. 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} [20].

The fixpoint, denoted by the symbol $\top\perp^*$, does not depend on the type of the first extraction.

Lemma 3. $\top\perp^*\text{-mod}(\Sigma, \mathcal{O}) = \perp\top^*\text{-mod}(\Sigma, \mathcal{O})$.

Proof. We remember that the notation $\top\perp\text{-mod}(\Sigma, \mathcal{O})$ means that the first extraction starts with \top -locality. Because of the fixpoint hypothesis, we have:

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

Now, from anti-monotonicity of local axioms, we have that $\perp\text{-mod}(\Sigma, \mathcal{O}) \subseteq \mathcal{O}$; hence,

$$\begin{array}{c}
\top\perp^*\top\text{-mod}(\Sigma, \perp\text{-mod}(\Sigma, \mathcal{O})) \subseteq \top\perp^*\top\text{-mod}(\Sigma, \mathcal{O}) = \top\perp^*\text{-mod}(\Sigma, \mathcal{O}) \\
\parallel \\
\perp\top^*\text{-mod}(\Sigma, \mathcal{O})
\end{array}$$

This proves the inclusion “ \supseteq ” of the statement. The converse is analogous. \square

In contrast, \top - and \perp -modules do not have to be equal—in fact, the former are usually larger than the latter. Through the nesting, $\top\perp^*\text{-mod}(\Sigma, \mathcal{O})$ is always contained in $\top\text{-mod}(\Sigma, \mathcal{O})$ and $\perp\text{-mod}(\Sigma, \mathcal{O})$.

From now on, we will denote by $x\text{-mod}(\Sigma, \mathcal{O})$ the x -module \mathcal{M} extracted from an ontology \mathcal{O} by using the notion of x -locality w.r.t. Σ , where $x \in$

$\{\top, \perp, \perp\top, \top\perp, \dots, \top\perp^*\}$, including any alternate nesting of these symbols. Finally, we want to point out that, for $\mathcal{M} = x\text{-mod}(\Sigma, \mathcal{O})$, 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 4. *Let \mathcal{O} be an ontology, Σ be a signature, $x \in \{\perp, \top, \top\perp^*\}$; let $\mathcal{M} = x\text{-mod}(\Sigma, \mathcal{O})$ and Σ' be a signature with $\Sigma \subseteq \Sigma' \subseteq \Sigma \cup \widetilde{\mathcal{M}}$. Then $x\text{-mod}(\Sigma', \mathcal{O}) = \mathcal{M}$.*

(For $x \in \{\perp, \top\}$, see [5]; the transfer to nested modules is straightforward).

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

Corollary 5. *Let Σ_1 and Σ_2 be two sets of terms, and let $x \in \{\top, \perp\}$. Then, $\Sigma_1 \subseteq \Sigma_2$ implies $x\text{-local}(\Sigma_2) \subseteq x\text{-local}(\Sigma_1)$ (see [5]).*

Remark 6. 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} .

In general, the following are not modules: the union, intersection or complement of modules; see the following propositions.

Proposition 7. *The union of modules is not, in general, a module,*

Proof. Consider, for example, the ontology

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

Then,

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

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

Proposition 8. *The intersection of modules is not, in general, a module.*

Proof. Consider, for example, the ontology

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

Then,

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

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

Proposition 9. *The complement of a module is not, in general, a module.*

Proof. Consider, for example, the ontology

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

Then,

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

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

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

Definition 10. *Let \mathcal{O} be an ontology, $\mathcal{M} \subseteq \mathcal{O}$ a module, and $\Sigma \subseteq \tilde{\mathcal{O}}$ a signature.*

\mathcal{M} is called self-contained if it is indistinguishable from \mathcal{O} w.r.t. $\Sigma \cup \tilde{\mathcal{M}}$.

\mathcal{M} is called depleting if $\mathcal{O} \setminus \mathcal{M}$ is indistinguishable from the empty set w.r.t. Σ .

Proposition 11. *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 [17]).*

Theorem 12. *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 [17]).*

Remark 13. From now on, we use the notion of $\top\perp^*$ -locality from [20]. 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.

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 14. *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.*

We will make use of a *partial order* \leq , i.e., a reflexive, transitive, and anti-symmetric binary relation. Two elements a, b of a poset are called *comparable* if $a \leq b$ or $b \leq a$, otherwise they are *incomparable*. Given a finite set O and a family F of subsets of O , we can build the set $B(O, F)$ by closing the family 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 $a \in O$ such that if there exists no element b of $B(O, F) \setminus a$ with $b \subset a$.

Definition 15. *The minimal elements of the $B(O, F) \setminus \emptyset$ with respect to “ \subseteq ” are called atoms.⁷ The principal ideal of an element $a \in O$ is the set $(a] := \{x \in O \mid x \leq a\}$.*

Every finite poset (O, \leq) (and every lattice) can be depicted in a graph, called *Hasse diagrams*, where nodes are elements of O and edges connect two elements $a \leq b$ if there is no element c distinct from a and b such that $a \leq c \leq b$; for $a \leq b$, we will draw b in a position higher than a 's.

3 The Atomic Decomposition

Modules and atoms. In what follows, we are using the notion of $\top\perp^*$ -locality from [20]. 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 16. See [17] for more details.

Proposition 16. *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 14. Axioms correspond to points. Let then $\mathfrak{F}(\mathcal{O})$ denote the family of $\top\perp^*$ -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 17. *Given an ontology \mathcal{O} and the family $\mathfrak{F}(\mathcal{O})$ of $\top\perp^*$ -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 18. *A syntactic tautology is an axiom that does not occur in any module and hence belongs to $\mathcal{O} \setminus \top\perp^*\text{-mod}(\tilde{\mathcal{O}}, \mathcal{O})$. A global axiom is an axiom that occurs in each module, in particular in $\top\perp^*\text{-mod}(\emptyset, \mathcal{O})$.*

Remark 19. 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.

⁷ Slightly abusing notation, we use $B(O, F)$ here for the set of complexes in $B(O, F)$.

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} , and 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 $\mathcal{B}(\mathfrak{F}(\mathcal{O})) \setminus \{\emptyset\}$.

Definition 20. *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 $\mathfrak{a}, \mathfrak{b}, \dots$
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 20, has many properties of interest for us.

Lemma 21. *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 22. (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.

Proposition 23. *Def. 22 describes the all and only relations between atoms.*

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 $\mathcal{M}_1 = \{\mathbf{a}, \mathbf{b}\}$, $\mathcal{M}_2 = \{\mathbf{a}, \mathbf{c}\}$, $\mathcal{M}_3 = \{\mathbf{a}, \mathbf{b}, \mathbf{d}\}$ and $\mathcal{M}_4 = \{\mathbf{a}, \mathbf{c}, \mathbf{d}\}$. Following Definition 22, the atoms \mathbf{b} , \mathbf{c} and \mathbf{d} depend on \mathbf{a} . However, we want our structure to reflect that \mathbf{b} and \mathbf{c} act as “intermediates” in the dependency of \mathbf{d} on \mathbf{a} , i.e., that \mathbf{d} depends via “ \mathbf{c} or \mathbf{b} ” on \mathbf{a} . Since our definition does not capture disjunctions of occurrences of atoms, we call the pairs (\mathbf{d}, \mathbf{b}) and (\mathbf{d}, \mathbf{c}) *problematic*. Fortunately, problematic atom pairs do not exist in an atomic decomposition obtained via locality-based modules, as Lemma 24 shows. Its consequences on the dependency relation on atoms are captured by Proposition 27.

Lemma 24. *Since the $\top\perp^*$ notion of module is monotonic, self-contained, and deleting, there are no problematic pairs in the set $\mathcal{A}(\mathfrak{F}(\mathcal{O}))$ of atoms over \mathcal{O} .*

The key to proving Lemma 24 is the following remark:

Remark 25. Let $\mathbf{a} \in \mathcal{A}(\mathfrak{F}(\mathcal{O}))$ be an atom induced over \mathcal{O} by $\top\perp^*$ -mod. Then, for every nonempty set of axioms $\{\alpha_1, \dots, \alpha_k\} \subseteq \mathbf{a} : \top\perp^*$ -mod($\{\tilde{\alpha}_1, \dots, \tilde{\alpha}_k\}, \mathcal{O}$) is the smallest module containing \mathbf{a} .

Proof. Let $\alpha \in \mathbf{a}$ be an axiom, and consider the module $\mathcal{M}_\alpha := \top\perp^*$ -mod($\tilde{\alpha}, \mathcal{O}$). We recall $\top\perp^*$ -mod is self-contained and monotonic. Then:

- (1) \mathcal{M}_α is not empty since it contains α (recall Remark 19).
- (2) $\mathcal{M}_\alpha \supseteq \mathbf{a}$, by the definition of atoms.
- (3) \mathcal{M}_α is the unique and thus smallest module for the seed signature $\tilde{\alpha}$.
- (4) by monotonicity, enlarging the seed signature $\tilde{\alpha}$ results in a superset of \mathcal{M}_α .
- (5) by self-containedness and monotonicity, any module \mathcal{M}' that contains \mathbf{a} needs to contain also \mathcal{M}_α : $\mathcal{M}' = \top\perp^*$ -mod($\widetilde{\mathcal{M}'}, \mathcal{O}$) = $\top\perp^*$ -mod($\widetilde{\mathcal{M}'} \cup \tilde{\alpha}, \mathcal{O}$) $\supseteq \top\perp^*$ -mod($\tilde{\alpha}, \mathcal{O}$).
- (6) because of (2), we have that $\mathcal{M}_\alpha \supseteq \top\perp^*$ -mod(\tilde{S}, \mathcal{O}) for every non empty set of axioms $S = \{\alpha_1, \dots, \alpha_k\} \subseteq \mathbf{a}$; in particular, this holds if $S = \{\alpha_i\}$ for any $\alpha_i \in \mathbf{a}$.
- (7) the inverted inclusion $\top\perp^*$ -mod($\tilde{\alpha}_i, \mathcal{O}$) $\supseteq \mathcal{M}_\alpha$ also holds by the arbitrariness of choice of α in \mathbf{a} .

□

Corollary 26. *Given an atom \mathbf{a} , for every axiom $\alpha \in \mathbf{a}$ we have that $\mathcal{M}_\alpha = \top\perp^*$ -mod($\tilde{\alpha}, \mathcal{O}$). Moreover, \mathbf{a} is dependent on all atoms belonging to $\mathcal{M}_\alpha \setminus \mathbf{a}$.*

Proposition 27. *The binary relation \succeq is a partial order over the set $\mathcal{A}(\mathfrak{F}(\mathcal{O}))$ of atoms of an ontology \mathcal{O} .*

Definition 22 and Proposition 27 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**. As an atom can be dependent on more than one atom; hence, we will have some nodes with 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, we can compute the modules from $\mathcal{A}(\mathfrak{F}(\mathcal{O}))$, and thus the latter is indeed a succinct *representation* of all modules.

Definition 28. *The principal ideal of an atom \mathbf{a} is the set $(\mathbf{a}] = \{\alpha \in \mathfrak{b} \mid \mathfrak{b} \prec \alpha\} \subseteq \mathcal{O}$.*

Proposition 29. *For every atom \mathbf{a} , $(\mathbf{a}]$ is a module.*

To get modules from $\mathcal{A}(\mathfrak{F}(\mathcal{O}))$, we need, for each atom \mathbf{a} , to store the \subseteq -minimal seed signatures that lead to $(\mathbf{a}]$: we say that an atom \mathbf{a} is *relevant* for a module $\top\perp^*$ -mod(Σ, \mathcal{O}) if there is a seed signature Σ' for $(\mathbf{a}]$ with $\Sigma' \subseteq \Sigma$.

Proposition 30. *Let $\mathbf{a}_1, \dots, \mathbf{a}_k, k \in \mathbb{N}$, be all atoms that are relevant for Σ . Then the module for Σ is the union of principal ideals of these atoms:*

$$\top\perp^*\text{-mod}(\Sigma, \mathcal{O}) = \bigcup_{i=1}^k (\mathbf{a}_i].$$

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

- 1) compact if there exists an atom \mathbf{a} in $\mathcal{A}(\mathfrak{F}(\mathcal{O}))$ such that $\mathcal{M} = (\mathbf{a}]$.
- 2) α -module if there is an axiom $\alpha \in \mathcal{O}$ such that $\mathcal{M} = \top\perp^*\text{-mod}(\tilde{\alpha}, \mathcal{O})$.
- 3) 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 [19], 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 32. *The notions of compact, α - and genuine modules coincide.*

Algorithm 1 Atomic decomposition

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

3: **ToDoAx** $\leftarrow \top\perp^*$ -mod($\tilde{\mathcal{O}}, \mathcal{O}$) $\setminus \top\perp^*$ -mod(\emptyset, \mathcal{O})
4: **GenAx** $\leftarrow \emptyset$
5: **for** each $\alpha \in \mathbf{ToDoAx}$ **do**
6: **Mod**(α) $\leftarrow \top\perp^*$ -mod($\tilde{\alpha}, \mathcal{O}$) { $\neq \emptyset$ due to Line 3}
7: **isNew** $\leftarrow \mathbf{true}$
8: **for** each $\beta \in \mathbf{GenAx}$ **do**
9: **if** **Mod**(α) = **Mod**(β) **then**
10: **At**(β) $\leftarrow \mathbf{At}(\beta) \cup \{\alpha\}$
11: **CardAt**(β) $\leftarrow \mathbf{CardAt}(\beta) + 1$
12: **isNew** $\leftarrow \mathbf{false}$
13: **end if**
14: **end for**
15: **if** **isNew** **then**
16: **At**(α) $\leftarrow \{\alpha\}$
17: **CardAt**(α) $\leftarrow 1$
18: **GenAx** $\leftarrow \mathbf{GenAx} \cup \{\alpha\}$
19: **end if**
20: **end for**
21: **for** each $\alpha \in \mathbf{GenAx}$ **do**
22: **for** each $\beta \in \mathbf{GenAx}$ **do**
23: **if** $\beta \in \mathbf{Mod}(\alpha)$ **then**
24: **At**(β) $\succeq \mathbf{At}(\alpha)$
25: **end if**
26: **end for**
27: **end for**
28: $\mathcal{A}(\mathfrak{F}(\mathcal{O})) \leftarrow \{\mathbf{At}(\alpha) \mid \alpha \in \mathbf{GenAx}\}$
29: $\mathfrak{G} \leftarrow \{\mathbf{Mod}(\alpha) \mid \alpha \in \mathbf{GenAx}\}$
30: **return** $[(\mathcal{A}(\mathfrak{F}(\mathcal{O})), \succeq), \mathfrak{G}, \mathbf{GenAx}, \mathbf{CardAt}(\cdot)]$

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 19, 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 $\text{Mod}(\alpha)$ and the corresponding atom is constructed in $\text{At}(\alpha)$; those functions are undefined for axioms outside `GenAx`.

Proposition 33. *Algorithm 1 is correct.*

Proof. Since the algorithm returns four values in Line 30, we have to show that, before Line 30 is executed, the following four conditions hold.

1. *The set \mathfrak{G} consists of all genuine modules.*

Due to Line 29, it suffices to show that $\{\text{At}(\alpha) \mid \alpha \in \text{GenAx}\}$ is the set of all genuine modules.

For the inclusion “ \subseteq ”, let $\alpha \in \text{GenAx}$ and $\mathcal{M} = \text{Mod}(\alpha)$. Then, due to Line 6, \mathcal{M} is the α -module. Via Lemma 32, we conclude that \mathcal{M} is a genuine module.

For the inclusion “ \supseteq ”, suppose that \mathcal{M} is a genuine module. Then, due to Lemma 32, $\mathcal{M} = \top\perp^*\text{-mod}(\tilde{\alpha}, \mathcal{O})$, for some $\alpha \in \mathcal{O}$. Hence, \mathcal{M} is computed in Line 6 during at least one traversal of the for-loop in Lines 5–20. During the first such traversal, the variable `isNew` will never be set to `false` in Line 12; therefore, α is subsequently added to `GenAx` in Line 18.

2. *The computed structure $(\mathcal{A}(\mathfrak{F}(\mathcal{O})), \succeq)$ coincides with the poset of all atoms and the dependency relation.*

We need to show two things.

- (a) $\{\text{At}(\alpha) \mid \alpha \in \text{GenAx}\} = \mathcal{A}(\mathfrak{F}(\mathcal{O}))$

For the inclusion “ \subseteq ”, let $\alpha \in \text{GenAx}$. Due to Lines 16 and 10, we have that

$$\begin{aligned} \text{At}(\alpha) &= \{\alpha\} \cup \{\beta \neq \alpha \mid \text{Mod}(\alpha) = \text{Mod}(\beta)\} \\ &= \{\beta \mid \text{Mod}(\alpha) = \text{Mod}(\beta)\} \\ &= \{\beta \mid \mathcal{M}_\alpha = \mathcal{M}_\beta\}, \end{aligned}$$

with the last equality following from Line 6. We need to show that $\text{At}(\alpha)$ is the atom containing α . If $\beta \in \text{At}(\alpha)$, then $\mathcal{M}_\alpha = \mathcal{M}_\beta$. Hence, β is in the smallest module containing α (Remark 25) and therefore in the atom containing α . For the converse, if β is in the atom containing α , then $\mathcal{M}_\alpha = \mathcal{M}_\beta$ (Remark 25), and therefore $\beta \in \text{At}(\alpha)$.

For the inclusion “ \supseteq ”, let \mathfrak{a} be an atom. Let $\alpha \in \mathfrak{a}$ be that of \mathfrak{a} ’s axioms which is handled first in some traversal of the for-loop in Lines 5–20. Then, in that traversal, α enters $\text{At}(\alpha)$ in Line 16, and subsequent traversals add exactly those β to $\text{At}(\alpha)$ in Line 10 where $\mathcal{M}_\alpha = \mathcal{M}_\beta$. Therefore, $\text{At}(\alpha) = \mathfrak{a}$.

- (b) *The algorithm computes that $\text{At}(\beta) \succeq \text{At}(\alpha)$ if and only if—for the atoms \mathbf{b}, \mathbf{a} represented by $\text{At}(\beta), \text{At}(\alpha)$ —it holds that $\mathbf{b} \succeq \mathbf{a}$.*

Due to Line 24, the algorithm computes that $\text{At}(\beta) \succeq \text{At}(\alpha)$ if and only if $\beta \in \text{Mod}(\alpha)$. Via Item 1, this equivalent to $\beta \in \mathcal{M}_\alpha$. We want to show that this last condition is equivalent to the condition “for all modules $\mathcal{M} : \text{At}(\beta) \subseteq \mathcal{M}$ implies $\text{At}(\alpha) \subseteq \mathcal{M}$ ”. The direction “ \Leftarrow ” is obvious when we set $\mathcal{M} = \mathcal{M}_\alpha$. For the direction “ \Rightarrow ”, let \mathcal{M} be a module with $\text{At}(\alpha) \subseteq \mathcal{M}$. Then $\mathcal{M}_\alpha \subseteq \mathcal{M}$ (Remark 25). Since $\beta \in \mathcal{M}_\alpha$, we have that $\text{At}(\beta) \subseteq \mathcal{M}$.

3. *For every atom \mathbf{a} , the set GenAx contains some axiom $\alpha \in \mathbf{a}$.*

Let \mathbf{a} be an atom. Then, due to Item 2, $\mathbf{a} = \text{At}(\alpha)$ for some $\alpha \in \mathbf{a}$ with $\alpha \in \text{GenAx}$.

4. *For every $\alpha \in \text{GenAx}$, $\text{CardAt}(\alpha)$ is the cardinality of the atom containing α .*

This is ensured by Lines 17 and 11 in connection with Lines 16 and 10.

□

5 Empirical evaluation

We ran Algorithm 1 on a selection of ontologies, including those used in [8, 19], and indeed managed to compute the atomic decomposition in all cases, even for ontologies where a complete modularization was previously impossible.

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

Table 1. Experiments summary

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).

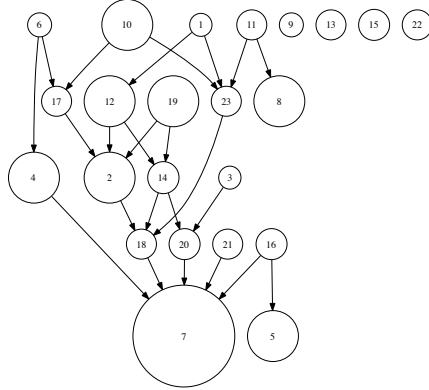


Figure 1. The atomic decomposition of Koala

We have also generated a graphical representation using GraphViz⁸. Our atomic decompositions show atom size as node size, see e.g. Fig. 1. It shows four isolated atoms, e.g., Atom 22, consisting of the axiom `DryEucalyptForest` \sqsubseteq `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/i4o1Y0>.

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 30: we could label an atom \mathbf{a} with the set of all its minimal seed signatures for which \mathbf{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 34. *Given: an ontology \mathcal{O} ; the atomic decomposition of the ontology $\mathcal{A}(\mathfrak{F}(\mathcal{O})) = \{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n\}$; the set of genuine modules $\mathfrak{G} = \{\mathcal{M}_i \mid \mathcal{M}_i = (\mathbf{a}_i], 1 \leq i \leq n\}$. We define the following labelling functions $\text{Lab}_j(\cdot)$ from*

⁸ <http://www.graphviz.org/About.php>

$\mathcal{A}(\mathfrak{F}(\mathcal{O}))$ to $\tilde{\mathcal{O}}$:

$$\begin{aligned}\text{Lab}_1(\mathfrak{a}_i) &:= \tilde{\mathfrak{a}}_i \\ \text{Lab}_2(\mathfrak{a}_i) &:= \tilde{\mathfrak{a}}_i \setminus \bigcup_{\mathfrak{b} \prec \mathfrak{a}_i} \text{Lab}_2(\mathfrak{b}) \\ \text{Lab}_3(\mathfrak{a}_i) &:= \bigcup_{\Sigma \in \text{mssig}(\mathcal{M}_i)} \Sigma \\ \text{Lab}_4(\mathfrak{a}_i) &:= \bigcup_{\Sigma \in \text{mssig}(\mathcal{M}_i)} \Sigma \setminus \bigcup_{\mathfrak{b} \prec \mathfrak{a}_i} \text{Lab}_4(\mathfrak{b})\end{aligned}$$

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 $\mathcal{O} = \{A \sqsubseteq B, C \sqsubseteq D, A \sqcap C \sqsubseteq B \sqcup D\}$. This ontology generates 3 atoms, one for each axiom, such that the atom $\mathfrak{a}_3 = A \sqcap C \sqsubseteq B \sqcup D$ is dependent on both the other 2, which are independent of each other. Clearly, $\text{Lab}_2(\mathfrak{a}_3)$ is empty, because $\{\mathfrak{a}_3\}$ reuses terms from the other atoms. Moreover, let us consider the axiom $A \sqsubseteq B \sqcap (C \sqcup \neg C)$. 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 $\mathcal{O} = \{A \equiv B, B \sqsubseteq C, B \sqcap D \sqsubseteq C \sqcup E, D \sqsubseteq E, E \equiv F\}$. Then, each axiom identifies an atom, and \mathcal{O} equals the principal ideal of the atom \mathfrak{a}_3 containing the axiom $B \sqcap D \sqsubseteq C \sqcup E$. Although the signature of \mathfrak{a}_3 contains neither A nor F , the set $\Sigma = \{A, F\}$ 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 $\hat{\mathcal{M}}$.

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 30 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 $\mathcal{O} = \{A_i \sqsubseteq A_{i+1} \mid i = 0, \dots, n-1\}$, which consists of a single subsumption path p . The atomic decomposition of \mathcal{O} consists of n independent atoms: $\top \perp^* \text{-mod}(\{A_i, A_{i+1}\}, \mathcal{O}) = \{A_i \sqsubseteq A_{i+1}\}$, for every $i = 0, \dots, n-1$. 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 , $\top \perp^* \text{-mod}(\Sigma, \mathcal{O}) = \emptyset$; for all other Σ , $\top \perp^* \text{-mod}(\Sigma, \mathcal{O})$ 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 <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 prohibitively 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 [19]. 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 Stuckenschmidt et al. [22], pages 349–371.
- [2] C. Bezerra, F. Freitas, A. Zimmermann, and J. Euzenat. ModOnto: A tool for modularizing ontologies. In *Proc. WONTO-08*, volume 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*, volume 4825 of *LNCS*, pages 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*, pages 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*, volume 211 of *FAIA*, pages 11–24. IOS Press, 2010.
- [9] 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*, pages 187–197, 2006.
- [10] I. Horrocks, O. Kutz, and U. Sattler. The even more irresistible *SR_{OL}IQ*. In *Proc. of KR-06*, pages 57–67, 2006.
- [11] I. Horrocks, P. F. Patel-Schneider, and F. van Harmelen. From *SHIQ* and RDF to OWL: The making of a web ontology language. *J. of Web Sem.*, 1(1):7–26, 2003.
- [12] 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*, volume 5021 of *LNCS*, pages 185–199, 2008.
- [13] A. Jimeno, E. Jiménez-Ruiz, R. Berlanga, and D. Rebolz-Schuhmann. Use of shared lexical resources for efficient ontological engineering. In *SWAT4LS-08*, *ceur-ws.org*, 2008.
- [14] B. Konev, C. Lutz, D. Ponomaryov, and F. Wolter. Decomposing description logic ontologies. In *Proc. of KR-10*, pages 236–246. AAAI Press, 2010.
- [15] B. Konev, C. Lutz, D. Walther, and F. Wolter. Logical difference and module extraction with CEX and MEX. In *Proc. of DL 2008*, volume 353 of *ceur-ws.org*, 2008.

- [16] B. Konev, C. Lutz, D. Walther, and F. Wolter. Formal properties of modularization. In Stuckenschmidt et al. [22], pages 25–66.
- [17] R. Kontchakov, L. Pulina, U. Sattler, T. Schneider, P. Selmer, F. Wolter, and M. Zakharyashev. Minimal module extraction from DL-Lite ontologies using QBF solvers. In *Proc. of IJCAI-09*, pages 836–841, 2009.
- [18] O. Kutz, C. Lutz, F. Wolter, and M. Zakharyashev. \mathcal{E} -connections of abstract description systems. *Artificial Intelligence*, 156(1):1–73, 2004.
- [19] B. Parsia and T. Schneider. The modular structure of an ontology: an empirical study. In *Proc. of KR-10*, pages 584–586. AAAI Press, 2010.
- [20] U. Sattler, T. Schneider, and M. Zakharyashev. Which kind of module should I extract? In *DL 2009*, volume 477 of *ceur-ws.org*, 2009.
- [21] H. Stuckenschmidt and M. Klein. Structure-based partitioning of large concept hierarchies. In *Proc. of ISWC-04*, volume 3298 of *LNCS*, pages 289–303. Springer-Verlag, 2004.
- [22] H. Stuckenschmidt, C. Parent, and S. Spaccapietra, editors. *Modular Ontologies: Concepts, Theories and Techniques for Knowledge Modularization*, volume 5445 of *LNCS*. Springer, 2009.
- [23] 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*, pages 88–101, 2004.
- [24] B. Suntisrivaraporn. Module extraction and incremental classification: A pragmatic approach for \mathcal{EL}^+ ontologies. In *Proc. of ESWC-08*, volume 5021 of *LNCS*, pages 230–244, 2008.