Fast Modularisation and Atomic Decomposition of Ontologies using Axiom Dependency Hypergraphs

Francisco Martín-Recuerda[1] and Dirk Walther[2]

[1] Universidad Politécnica de Madrid, Spain
fmartinrecuerda@fi.upm.es
[2] TU Dresden, Theoretical Computer Science
Center for Advancing Electronics Dresden, Germany
Dirk.Walther@tu-dresden.de

Abstract. In this paper we define the notion of an axiom dependency hypergraph, which explicitly represents how axioms are included into a module by the algorithm for computing locality-based modules. A locality-based module of an ontology corresponds to a set of connected nodes in the hypergraph, and atoms of an ontology to strongly connected components. Collapsing the strongly connected components into single nodes yields a condensed hypergraph that comprises a representation of the atomic decomposition of the ontology. To speed up the condensation of the hypergraph, we first reduce its size by collapsing the strongly connected components of its graph fragment employing a linear time graph algorithm. This approach helps to significantly reduce the time needed for computing the atomic decomposition of an ontology. We provide an experimental evaluation for computing the atomic decomposition of large biomedical ontologies. We also demonstrate a significant improvement in the time needed to extract locality-based modules from an axiom dependency hypergraph and its condensed version.

1 Introduction

A *module* is a subset of an ontology that includes all the axioms required to define a set of terms and the relationships between them. Computing minimal modules is very expensive (or even impossible) and cheap approximations have been developed based on the notion of *locality* [7]. Module extraction facilitates the reuse of existing ontologies. Moreover, some meta-reasoning systems such as MORe[1] and Chainsaw[2] also exploit module extraction techniques for improving the performance of some reasoning tasks.

The number of all possible modules of an ontology can be exponential wrt. the number of terms or axioms of the ontology [7]. *Atomic decomposition* was

Partially supported by the German Research Foundation (DFG) via the Cluster of Excellence 'Center for Advancing Electronics Dresden'.

[1] http://www.cs.ox.ac.uk/isg/tools/MORe/
[2] http://sourceforge.net/projects/chainsaw/

introduced as a succinct representation of all possible modules of an ontology [5]. Tractable algorithms for computing the atomic decomposition for locality-based modules have been defined [5], and subsequently improved further [14]. Moreover, it has been suggested that the atomic decomposition of an ontology can help to improve the performance of the locality-based module extraction algorithm [4].

In this paper we introduce the notion of an *axiom dependency hypergraph (ADH)* for OWL ontologies, which explicitly represents how axioms are included into a module by the locality-based module extraction algorithm [7]. This algorithm first identifies the axioms that are non-local wrt. a given signature Σ, and then it extends Σ with the symbols of the axioms selected. In this fashion, the algorithm iteratively includes in the module more axioms of the ontology that become non-local wrt. to the extended signature until no more axioms are added. The hyperedges of an ADH indicate which axioms become non-local wrt. a signature after one or more axioms of the ontology have been included in the module [9]. Unlike other hypergraph representations of ontologies [12, 10], the relationship between atoms of an ontology and the strongly connected components (SCCs) of the ADH becomes apparent. This allows us to employ standard algorithms from graph theory to compute atoms and locality-based modules.

To speed up the computation of SCCs in a directed hypergraph, we first compute the SCCs of its graph fragment (only directed edges are considered), and subsequently we collapse them into a single nodes. Note that in directed graphs, the SCCs can be computed in linear time wrt. the size of the graph [13], whereas in directed hypergraphs, this process is at least quadratic [1]. In this way, we manage to reduce the size of the original hypergraph significantly, in some cases, which then reduces the time needed for computing the SCCs in the hypergraph. The result of computing and collapsing all SCCs of an axiom dependency hypergraph yields its condensed version, a *condensed* axiom dependency hypergraph. The graph fragment of this hypergraph corresponds to the atomic decomposition of the ontology as introduced in [5]. From the condensed axiom dependency hypergraph, it is also possible to compute locality-based modules using an adapted version of the modularization algorithm discussed in [7]. In this case, a module correspond to a connected component in the hypergraph.

We implemented our method in a Java prototype named HyS. We compared our prototype against state-of-the-art implementations for computing locality-based modules and atomic decomposition [14, 15]. We confirm a significant improvement in running time for a selection of large biomedical ontologies from the NCBO Bioportal.[3]

The paper is organised as follows. In Section 2 we present relevant notions on syntactic locality, atomic decomposition, and hypergraphs. In Section 3 we introduce the notion of axiom dependency hypergraphs, and we use this notion to characterise locality-based modules and the atomic decomposition of any OWL ontology. We explain implementation details of HyS in Section 4, and we report on the result of the evaluation of our Java prototype in Section 5. We conclude this paper in a final section.

[3] `http://bioportal.bioontology.org/`

2 Preliminaries

We consider ontologies formulated in the expressive description logic $\mathcal{SROIQ}$ [8] which underlies the Web Ontology Language OWL 2.[4] For the evaluation of our algorithms for computing modules and the atomic decomposition as introduced in this paper, we consider prominent biomedical ontologies formulated in the light-weight description logic $\mathcal{EL}^{++}$ [2], which is at the core of the OWL 2 EL profile.[5] We refer to [3] for a detailed introduction to description logics.

2.1 Syntactic Locality-based Modules

For an ontology $\mathcal{O}$ and a signature Σ, a module $\mathcal{M}$ is a subset of $\mathcal{O}$ that preserves all entailments formulated using symbols from Σ only. A signature Σ is a finite set of symbols, and we denote with $\mathsf{sig}(X)$ the signature of X, where X ranges over any syntactic object.

Definition 1 (Module). *$\mathcal{M} \subseteq \mathcal{O}$ is a* module *of $\mathcal{O}$ wrt. a signature Σ if for all entailments α with $\mathsf{sig}(\alpha) \subseteq \Sigma$: $\mathcal{M} \models \alpha$ iff $\mathcal{O} \models \alpha$.* ⊣

Computing a minimal module is hard (or even impossible) for expressive fragments of OWL 2. The notion of *syntactic locality* was introduced to allow for efficient computation of approximations of minimal modules [7]. Intuitively, an axiom α is *local* wrt. Σ if it does not state anything about the symbols in Σ. In this case, an ontology can *safely* be extended with α, or it can *safely* import α, where 'safe' means not changing the meaning of terms in Σ. A locality-based module wrt. Σ of an ontology consists of the axioms that are non-local wrt. Σ and the axioms that become non-local wrt. Σ extended with the symbols in other non-local axioms. Typically the notions $\bot$-*locality* and $\top$-*locality* are considered [7]. We denote with $\mathsf{Mod}^x_{\mathcal{O}}(\Sigma)$ the x-local module of an ontology $\mathcal{O}$ wrt. Σ, where $x \in \{\bot,\top\}$.

Checking for syntactic locality involves checking that an axiom is of a certain form (syntax), no reasoning is needed, and it can be done in polynomial time [7]. However, the state of non-locality of an axiom can also be checked in terms of signature containment [12]. To this end, we introduce the notion of minimal non-locality signature for $\mathcal{SROIQ}$ axioms.

Definition 2 (Minimal non-Locality Signature). *Let $x \in \{\bot,\top\}$ denote a locality notion. A* Minimal non-x-Locality Signature *for an axiom α is a signature $\Sigma \subseteq \mathsf{sig}(\alpha)$ such that α is not x-local wrt. Σ, and Σ is minimal (wrt. set inclusion) with this property. The set of minimal non-x-locality signatures is denoted by $\mathsf{MLS}^x(\alpha)$.* ⊣

The notion of minimal non-locality signature turns out to be equivalent to the notion of minimal globalising signatures, which were introduced specifically for computing modules from an atomic decomposition [4].

[4] `http://www.w3.org/TR/owl2-overview/`
[5] `http://www.w3.org/TR/owl2-profiles/#OWL_2_EL`

The following example shows that there can be exponentially many minimal non-locality signatures for an axiom using merely conjunction and disjunction as logical operators.

Example 1. Let $\alpha = (X_{11} \sqcup X_{12} \sqcup \dots \sqcup X_{1m}) \sqcap \dots \sqcap (X_{n1} \sqcup X_{n2} \sqcup \dots \sqcup X_{nm}) \sqsubseteq Y$ be an axiom. The minimal non-$\bot$-locality signature $\mathsf{MLS}(\alpha)$ of α is as follows:

$$\begin{aligned}\mathsf{MLS}^{\bot}(\alpha) = \{\{&X_{1i_1}, X_{2i_2}, \dots, X_{ni_n}\} \mid \\ &i_1, i_2, \dots, i_n \in \{1, ..., m\}\}\end{aligned}$$

Then: $|\mathsf{MLS}^{\bot}(\alpha)| = m^n$. $\lhd$

However, exponentially many minimal non-locality signatures can be avoided if the axiom is normalised. An ontology $\mathcal{O}$ (that is formulated in the description logic $\mathcal{SRIQ}$) is normalised by applying the normalisation rules presented in [10], which are an extension of the normalisation for $\mathcal{EL}$ ontologies [12]. Axioms of a normalised ontology have one of the following forms, where $A_i \in \mathsf{N_C} \cup \{\top\}$, $B_i \in \mathsf{N_C} \cup \{\bot\}$, $R_i \in \mathsf{N_R} \cup \mathsf{inv}(\mathsf{N_R})$, $X, Y \in \{\exists R.B, (\geq n\, R.B), \exists R.Self \mid B \in \mathsf{N_C}, R \in \mathsf{N_R} \cup \mathsf{inv}(\mathsf{N_R}), n \geq 0\}$ and $\ell, m \geq 0$:

$$\begin{array}{ll}\alpha_1 : A_1 \sqcap \ldots \sqcap A_\ell \sqsubseteq B_1 \sqcup \ldots \sqcup B_m & \alpha_5 : X \sqsubseteq Y \\ \alpha_2 : X \sqsubseteq B_1 \sqcup \ldots \sqcup B_m & \alpha_6 : R_1 \sqsubseteq R_2 \\ \alpha_3 : A_1 \sqcap \ldots \sqcap A_\ell \sqsubseteq Y & \alpha_7 : Dis(R_1, R_2) \\ \alpha_4 : R_1 \circ \ldots \circ R_\ell \sqsubseteq R_{\ell+1} & \end{array}$$

where $\mathsf{inv}(\mathsf{N_R})$ is the set of inverse roles r^-, for $r \in \mathsf{N_R}$, and $\exists R.Self$ expresses the local reflexivity of R. The normalisation of an ontology $\mathcal{O}$ runs in linear time in the size of $\mathcal{O}$. The normalised ontology preserves Σ-entailments of $\mathcal{O}$ [10].[6] Notice that the normalisation rules can be applied backwards over normalised axioms to compute the original axioms of the ontology. However, denormalisation requires a careful application of the normalisation rules to ensure that we obtain the original axioms.

There are at most two minimal non-locality signatures for a normalised axiom.

Proposition 1. *Let α be a normalised axiom. Then: $|\mathsf{MLS}^{\bot}(\alpha)| = 1$ and $|\mathsf{MLS}^{\top}(\alpha)| \leq 2$.* $\dashv$

We can apply additional normalisation rules to reduce the number of symbols on the left- and right-hand side of normalised axioms [9]. Bounding the number of symbols in an axiom results in bounding the size of the minimal non-locality signatures of the axiom.

We now give simple conditions under which normalised axioms are not syntactic local. Similar non-locality conditions are presented in the notions of $\bot$- and $\top$-reachability in [10].

[6] The normalisation in [10] can straightforwardly be extended to $\mathcal{SROIQ}$-ontologies. Then a normalised axiom can be of the forms as described, where A_i and B_i additionally range over nominals. However, nominals are not contained in any minimal non-locality signature of a normalised axiom.

Proposition 2 (Non-locality via Signature Containment). *Let α be a normalised axiom, and denote with $\mathsf{LHS}(\alpha)$ and $\mathsf{RHS}(\alpha)$ the left- and the right-hand side of α, respectively. Let Σ be a signature. Then: α is not $\bot$-local wrt. Σ iff one of the following holds:*

- $\mathsf{sig}(\mathsf{LHS}(\alpha)) \subseteq \Sigma$ *if α is of the form α_1, α_2, α_3, α_4, α_5, α_6;*
- $\mathsf{sig}(\alpha) \subseteq \Sigma$ *if α is of the form α_7;*

Then: α is not $\top$-local wrt. Σ iff α is of the form α_7 or one of the following holds:

- $\mathsf{sig}(\mathsf{RHS}(\alpha)) \cap \Sigma \neq \emptyset$ *if α is of the form α_3, α_4, α_5, α_6;*
- $\mathsf{sig}(\mathsf{RHS}(\alpha)) \subseteq \Sigma$ *if α is of the form α_1, α_2.* ⊣

2.2 Atomic Decomposition

An atom is a set of highly related axioms of an ontology in the sense that they always co-occur in modules [5].

Definition 3 (Atom). *An atom $\mathfrak{a}$ is a* maximal *set of axioms of an ontology $\mathcal{O}$ such that for every module $\mathcal{M}$ of $\mathcal{O}$ either $\mathfrak{a} \cap \mathcal{M} = \mathfrak{a}$ or $\mathfrak{a} \cap \mathcal{M} = \emptyset$.* ⊣

Consequently, we have that two axioms α and β are contained in an atom $\mathfrak{a}$ iff $\mathsf{Mod}^x_{\mathcal{O}}(\mathsf{sig}(\alpha)) = \mathsf{Mod}^x_{\mathcal{O}}(\mathsf{sig}(\beta))$, where $\mathsf{sig}(\alpha)$ ($\mathsf{sig}(\beta)$) is the signature of the axiom α (β). We denote with $\mathsf{Atoms}^x_{\mathcal{O}}$ the set of all atoms of $\mathcal{O}$ wrt. syntactic x-locality modules, for $x \in \{\bot,\top\}$. The atoms of an ontology partition the ontology into pairwise disjoint subsets. All axioms of the ontology are distributed over atoms such that every axiom occurs in exactly one atom. A dependency relation between atoms can be established as follows [5].

Definition 4 (Dependency relation between atoms). *An atom $\mathfrak{a}_2$* depends on *an atom $\mathfrak{a}_1$ in an ontology $\mathcal{O}$ (written $\mathfrak{a}_1 \succcurlyeq_{\mathcal{O}} \mathfrak{a}_2$) if $\mathfrak{a}_2$ occurs in every module of $\mathcal{O}$ containing $\mathfrak{a}_1$. The binary relation $\succcurlyeq_{\mathcal{O}}$ is a partial order.* ⊣

In other words, an atom $\mathfrak{a}_2$ depends on an atom $\mathfrak{a}_1$ in an ontology $\mathcal{O}$ if the module $\mathsf{Mod}^x_{\mathcal{O}}(\mathsf{sig}(\beta))$ is contained in the module $\mathsf{Mod}^x_{\mathcal{O}}(\mathsf{sig}(\alpha))$, for some α, β with $\alpha \in \mathfrak{a}_1$ and $\beta \in \mathfrak{a}_2$. For a given ontology $\mathcal{O}$, the poset $\langle \mathsf{Atoms}^x_{\mathcal{O}}, \succcurlyeq_{\mathcal{O}} \rangle$ was introduced as the *Atomic Decomposition* (*AD*) of $\mathcal{O}$, and it represents the modular structure of the ontology [5].

2.3 Directed Hypergraphs

A *directed hypergraph* is a tuple $\mathcal{H} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a non-empty set of *nodes* (vertices), and $\mathcal{E}$ is a set of *hyperedges* (*hyperarcs*) [6]. A *hyperedge* e is a pair $(T(e), H(e))$, where $T(e)$ and $H(e)$ are non-empty disjoint subsets of $\mathcal{V}$. $H(e)$ ($T(e)$) is known as the *head* (*tail*) and represents a set of nodes where the hyperedge ends (starts). A *B-hyperedge* is a directed hyperedge with only one node in the head. We call a B-hyperedge e *simple* if $|T(e)| = 1$ (i.e., if e corresponds to a directed edge); otherwise, if $|T(e)| > 1$, e is called *complex*. Directed hypergraphs containing B-hyperedges only are called *directed B-hypergraphs*; these are the only type of hypergraphs considered in this paper.

A node v is *B-connected* (or forward reachable) from a set of nodes $\mathcal{V}'$ (written $\mathcal{V}' \geq_B v$) if (i) $v \in \mathcal{V}'$, or (ii) there is a B-hyperedge e such that $v \in H(e)$ and all tail nodes in $T(e)$ are B-connected from $\mathcal{V}'$. For a set of nodes $\mathcal{V}' \subseteq \mathcal{V}$, we denote with $\geq_B(\mathcal{V}')$ the set $\geq_B(\mathcal{V}') = \{v \in \mathcal{V} \mid \mathcal{V}' \geq_B v\}$ of B-connected nodes from $\mathcal{V}'$.

In a directed hypergraph $\mathcal{H}$, two nodes v_1 and v_2 are *strongly B-connected* if v_2 is B-connected to v_1 and *vice versa*. In other words, both nodes, v_1 and v_2, are *mutually reachable*. A *strongly B-connected component (SCC)* is a set of nodes from $\mathcal{H}$ that are all mutually reachable [1]. We allow an SCC to be a singleton set since the reachability relation is reflexive, i.e., any axiom is mutually reachable from itself.

3 Axiom Dependency Hypergraph

Directed B-hypergraphs can be used to explicitly represent the locality-based dependencies between axioms. *Axiom dependency hypergraphs* for ontologies wrt. the locality-based modularity notions are defined as follows.

Definition 5 (Axiom Dependency Hypergraph). *Let $\mathcal{O}$ be an ontology. Let $x \in \{\bot, \top\}$ denote a locality notion. The* Axiom Dependency Hypergraph $\mathcal{H}^x_\mathcal{O}$ for $\mathcal{O}$ wrt. x-locality (x-ADH) *is defined as the directed B-hypergraph $\mathcal{H}^x_\mathcal{O} = (\mathcal{V}^x, \mathcal{E}^x)$, where*

- *$\mathcal{V}^x = \mathcal{O}$; and*
- *$e = (T(e), H(e)) \in \mathcal{E}^x$ iff $T(e) \subseteq \mathcal{V}^x$ and $H(e) = \{\beta\}$, for some $\beta \in \mathcal{V}^x$, such that:*
  (i) $\beta \notin T(e)$, and
  (ii) β is not x-local wrt. $\mathsf{sig}(T(e))$. ⊣

The nodes of the axiom dependency hypergraph are the axioms in the ontology. Hyperedges are directed and they might connect many tail nodes with one head node. Note that a head node of a hyperedge is not allowed to occur in its tail. Intuitively, the tail nodes of an hyperedge e correspond to axioms that provide the signature symbols required by the axiom represented by the head node of e to be non-local. We can think on reaching B-connected nodes as how the module extraction algorithm computes a module by successively including axioms into the module [9].

The notion of ADH for ontologies depends on the notion of syntactic locality. Using Prop. 2, we can similarly define this notion using minimal non-locality signatures by replacing Item (ii) of Def. 5 with:

(iib) $\Sigma \subseteq \mathsf{sig}(T(e))$, for some $\Sigma \in \mathsf{MLS}(\beta)$.

An ADH $\mathcal{H}_\mathcal{O}$ contains all locality-based dependencies between different axioms of the ontology $\mathcal{O}$. These dependencies are represented by the hyperedges in

$\mathcal{H}_{\mathcal{O}}$. Note that $\mathcal{H}_{\mathcal{O}}$ may contain exponentially many hyperedges, many of which can be considered redundant in the following sense.

Definition 6. *A hyperedge e in a directed B-hypergraph $\mathcal{H}$ is called* redundant *if there is a hyperedge e' in $\mathcal{H}$ such that $H(e) = H(e')$ and $T(e') \subsetneq T(e)$.* $\dashv$

A *compact* version of a directed B-hypergraph $\mathcal{H}$ is obtained from $\mathcal{H}$ by removing all redundant hyperedges while the B-connectivity relation between axioms is preserved. In the remainder of the paper, we consider ADHs that are compact. Notice that compact ADHs are unique and they may still contain exponentially many hyperedges. The number of hyperedges can be reduced to polynomially many by applying extra-normalisation rules that restrict the amount of signature symbols in each side of the axiom up to 2 symbols.

Next, we characterise modules and atoms together with their dependencies in terms of ADHs for which B-reachability is crucial.

3.1 Locality-based modules in an ADH

B-connectivity in an ADH can be used to specify locality-based modules in the corresponding ontology. A locality-based module of an ontology $\mathcal{O}$ for the signature of an axiom α (or a subset of axioms $\mathcal{O}' \subseteq \mathcal{O}$) corresponds to the B-connected component in the ADH for $\mathcal{O}$ from α (or $\mathcal{O}'$) [9].

Proposition 3. *Let $\mathcal{O}$ be an ontology, $\mathcal{O}' \subseteq \mathcal{O}$ and $\Sigma = \mathsf{sig}(\mathcal{O}')$. Let $\geq_B$ be the B-connectivity relation of the x-ADH for $\mathcal{O}$, where $x \in \{\bot,\top\}$. Then: $\mathsf{Mod}^x_{\mathcal{O}}(\Sigma) = \geq_B(\mathcal{O}')$.* $\dashv$

However, ADHs do not contain sufficient information for computing a module for *any* signature as the following simple example shows.

Example 2. Let $\mathcal{O} = \{\alpha_1 = A \sqsubseteq C, \alpha_2 = C \sqcap B \sqsubseteq D, \alpha_3 = D \sqsubseteq A\}$ and $\Sigma = \{A, B\}$. We have that $\mathsf{Mod}^{\bot}(\Sigma) = \{\alpha_1, \alpha_2, \alpha_3\}$. The $\bot$-ADH for $\mathcal{O}$ contains no hyperedge e with $H(e) = \{\alpha_2\}$ and, consequently, α_2 cannot be reached via a hyperedge. $\lhd$

The problem can be solved by incorporating the signature Σ into the ADH. The *Σ-extension $\mathcal{H}^x_{\mathcal{O},\Sigma}$ of an x-ADH $\mathcal{H}^x_{\mathcal{O}}$ for an ontology $\mathcal{O}$ wrt. x-locality,* $x \in \{\bot,\top\}$*,* is defined as the ADH according to Def. 5 but with Item (*ii*) replaced with:

(iii) β is not x-local wrt. $\Sigma \cup \mathsf{sig}(T(e))$.

Intuitively, no symbol in Σ contributes to the dependencies between axioms. Consequently, less axioms in the tail are needed to provide the signature for non-locality of β. Note that non-redundant hyperedges in the original ADH may become redundant in the Σ-extended ADH. The remaining hyperedges represent the dependencies between axioms *modulo Σ*.

Example 3. Let $\mathcal{O}$ and Σ as in Ex. 2. The Σ-extension of $\bot$-ADH for $\mathcal{O}$ contains the edge $e = \{\{\alpha_1\}, \{\alpha_2\}\}$. Hence, α_2 can be reached via the hyperedge e. Axiom α_1 is the only axiom that is not-$\bot$ local wrt. Σ. The B-connected nodes from α_1 are the axioms in $\mathsf{Mod}^{\bot}(\Sigma)$. $\lhd$

Given the Σ-extension of an ADH for an ontology, B-connectivity can be used to determine the axioms that are not local wrt. to Σ and to compute the corresponding locality-based module.

Proposition 4. *Let $\mathcal{O}$ be an ontology, Σ a signature and $x \in \{\bot,\top\}$. Let $\mathcal{O}^x_\Sigma$ be the set of axioms from $\mathcal{O}$ that are not x-local wrt. Σ. Let $\geq_B$ be the B-connectivity relation of the Σ-extension of the x-ADH for $\mathcal{O}$. Then: $\mathsf{Mod}^x_\mathcal{O}(\Sigma) = \geq_B(\mathcal{O}^x_\Sigma)$.* $\dashv$

Proof. The algorithm for computing the locality-based module $\mathsf{Mod}^x_\mathcal{O}(\Sigma)$ (see [9]) computes a sequence $\mathcal{M}_0, ..., \mathcal{M}_n$ such that $\mathcal{M}_0 = \emptyset$, $\mathcal{M}_i \subseteq \mathcal{M}_{i+1}$, for $i \in \{0, ..., n-1\}$, and $\mathcal{M}_n = \mathsf{Mod}^x_\mathcal{O}(\Sigma)$. We show by induction on $n > 0$ that $\mathcal{M}_1 \geq_B \alpha$, for every axiom $\alpha \in \mathcal{M}_n$.

For the direction from right to left of the set inclusion, we show that $\mathcal{O}^x_\Sigma \geq_B \beta$ implies $\beta \in \mathsf{Mod}^x_\mathcal{O}(\Sigma)$ by induction on the maximal length $n = \mathsf{dist}_\mathcal{H}(\mathcal{O}^x_\Sigma, \beta)$ of an acyclic hyperpath from an axiom α in $\mathcal{O}^x_\Sigma$ to β. $\dashv$

3.2 ADH Atomic Decomposition

In the previous section, we have established that locality-based modules of an ontology $\mathcal{O}$ correspond to sets of B-connected nodes in the axiom dependency hypergraph for $\mathcal{O}$. An atom of $\mathcal{O}$ consists of axioms α that share the same modules wrt. the signature of α. It holds that for every x-local atom $\mathfrak{a} \subseteq \mathcal{O}$ with $x \in \{\bot,\top\}$: $\alpha, \beta \in \mathfrak{a}$ if, and only if, $\mathsf{Mod}^x_\mathcal{O}(\mathsf{sig}(\alpha)) = \mathsf{Mod}^x_\mathcal{O}(\mathsf{sig}(\beta))$ [5]. Together with Proposition 3, we can now characterise the notion of an atom with a corresponding notion in axiom dependency hypergraphs. We have that two nodes in an ADH represent axioms that are contained in the same atom if, and only if, the nodes agree on the set of nodes that are B-connected from them. Formally: $\alpha, \beta \in \mathfrak{a}$ if, and only if, $\geq_B(\alpha) = \geq_B(\beta)$, where $\geq_B$ be the B-connectivity relation of the ADH $\mathcal{H}_\mathcal{O}$ for $\mathcal{O}$. It follows that all axioms of an atom are mutually B-connected in $\mathcal{H}_\mathcal{O}$. Axioms that are mutually B-connected constitute strongly B-connected components of $\mathcal{H}_\mathcal{O}$. Consequently, the set of atoms for an ontology $\mathcal{O}$ corresponds to the set of strongly B-connected components in the axiom dependency hypergraph for $\mathcal{O}$. Let $\mathsf{SCCs}(\mathcal{H}^x_\mathcal{O})$ be the set of strongly connected components of the hypergraph $\mathcal{H}^x_\mathcal{O}$, where $x \in \{\bot,\top\}$.

Proposition 5. *Let $\mathcal{O}$ be an ontology and let $x \in \{\bot,\top\}$ denote a locality notion. Let $\mathcal{H}^x_\mathcal{O} = (\mathcal{V}^x_\mathcal{O}, \mathcal{E}^x_\mathcal{O})$ be the x-ADH for $\mathcal{O}$. Then: $\mathsf{Atoms}^x_\mathcal{O} = \mathsf{SCCs}(\mathcal{H}^x_\mathcal{O})$.* $\dashv$

The condensed ADH is formed by collapsing the strongly B-connected components into single nodes and turning hyperedges between axioms into hyperedges between sets of axioms. The condensed ADH corresponds to the quotient hypergraph $\mathcal{H}_\mathcal{O}/_{\simeq_B}$ of $\mathcal{H}_\mathcal{O}$ under the mutual B-connectivity relation $\simeq_B$ in $\mathcal{H}_\mathcal{O}$. The $\simeq_B$-equivalence classes are the strongly B-connected components of $\mathcal{H}_\mathcal{O}$. The partition of a hypergraph under an equivalence relation is defined as follows.

Definition 7 (Quotient Hypergraph). *Let $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ be a hypergraph. Let $\simeq$ be an equivalence relation over $\mathcal{V}$. The* quotient of $\mathcal{H}$ under $\simeq$, *written $\mathcal{H}/_{\simeq}$, is the graph $\mathcal{H}/_{\simeq} = (\mathcal{V}/_{\simeq}, \mathcal{E}_{\simeq})$, where*

- *$\mathcal{V}/_{\simeq} = \{[x]_{\simeq} \mid x \in \mathcal{V}\}$; and*
- *$e = (T(e), H(e)) \in \mathcal{E}_{\simeq}$ iff there is an $e' \in \mathcal{E}$ such that $T(e) = \{[x]_{\simeq} \mid x \in T(e')\}$, $H(e) = \{[x]_{\simeq} \mid x \in H(e')\}$ and $T(e) \cap H(e) = \emptyset$.* $\dashv$

We can now define the notion of a *condensed* ADH (cADH) as the partition of the ADH under the mutual B-reachability relation. The cADH is formed by collapsing the strongly B-connected components into single nodes and turning hyperedges between axioms into hyperedges between the newly formed nodes.

Definition 8 (Condensed Axiom Dependency Hypergraph). *Let $\mathcal{H}^x_{\mathcal{O}}$ be the x-ADH for an ontology $\mathcal{O}$, where $x \in \{\bot, \top\}$. Let $\simeq_B$ be the mutual B-connectivity relation in $\mathcal{H}^x_{\mathcal{O}}$. The* condensed axiom dependency hypergraph for $\mathcal{O}$ wrt. x-locality (x-cADH) *is defined as the quotient $\mathcal{H}^x_{\mathcal{O}}/_{\simeq_B}$ of $\mathcal{H}^x_{\mathcal{O}}$ under $\simeq_B$.* $\dashv$

Similarly, it is also possible to compute the *partially condensed* ADH (pcADH) of an ADH. The idea is to identify and collapse the strongly connected components of the graph fragment of the ADH (Axiom Dependency Graph) such that only simple B-hyperedges are considered ($|T(e)| = 1$). The hyperedges of the ADH are re-calculated to consider the newly formed nodes.

Definition 9 (Partially Condensed Axiom Dependency Hypergraph). *Let $\mathcal{H}^x_{\mathcal{O}} = (\mathcal{V}^x_{\mathcal{O}}, \mathcal{E}^x_{\mathcal{O}})$ be the x-ADH for an ontology $\mathcal{O}$, where $x \in \{\bot, \top\}$. Let $\mathcal{G}_{\mathcal{H}^x_{\mathcal{O}}} = (\mathcal{V}_{\mathcal{H}^x_{\mathcal{O}}}, \mathcal{E}_{\mathcal{H}^x_{\mathcal{O}}})$ be a directed graph such that $\mathcal{V}_{\mathcal{H}^x_{\mathcal{O}}} = \mathcal{V}^x_{\mathcal{O}}$ and $\mathcal{E}_{\mathcal{H}^x_{\mathcal{O}}} = \{(T(e), H(e)) \in \mathcal{E}^x_{\mathcal{O}} \mid \mid T(e) \mid = 1\}$.*

Let $\simeq_B$ be the mutual B-connectivity relation in $\mathcal{G}_{\mathcal{H}^x_{\mathcal{O}}}$. The partially condensed axiom dependency hypergraph for $\mathcal{O}$ wrt. x-locality (x-cADH) *is defined as the quotient $\mathcal{H}^x_{\mathcal{O}}/_{\simeq_B}$ of $\mathcal{H}^x_{\mathcal{O}}$ under $\simeq_B$.* $\dashv$

The dependency relation $\succcurlyeq^x_{\mathcal{O}}$ between x-local atoms of $\mathcal{O}$, for $x \in \{\bot, \top\}$, is defined as follows [5]. For atoms $\mathfrak{a}, \mathfrak{b} \in \mathsf{Atoms}^x_{\mathcal{O}}$ and axioms $\alpha \in \mathfrak{a}$ and $\beta \in \mathfrak{b}$: $\mathfrak{a} \succcurlyeq^x_{\mathcal{O}} \mathfrak{b}$ if, and only if, $\mathfrak{b} \subseteq \mathsf{Mod}^x_{\mathcal{O}}(\alpha)$ if, and only if, $\mathsf{Mod}^x_{\mathcal{O}}(\beta) \subseteq \mathsf{Mod}^x_{\mathcal{O}}(\alpha)$.

Proposition 6. *Let $\mathcal{O}$ be an ontology with $\alpha, \beta \in \mathcal{O}$. Let $\mathfrak{a}, \mathfrak{b} \in \mathsf{Atoms}^x_{\mathcal{O}}$ such that $\alpha \in \mathfrak{a}$ and $\beta \in \mathfrak{b}$, where $x \in \{\bot, \top\}$. Let $\simeq$ be the mutual B-connectivity relation in the x-locality ADH $\mathcal{H}$ for $\mathcal{O}$ and $\geq$ the B-connectivity relation in the x-cADH for $\mathcal{O}$. Then: $\mathfrak{a} \succcurlyeq^x_{\mathcal{O}} \mathfrak{b}$ iff $[\alpha]_{\simeq} \geq [\beta]_{\simeq}$.* $\dashv$

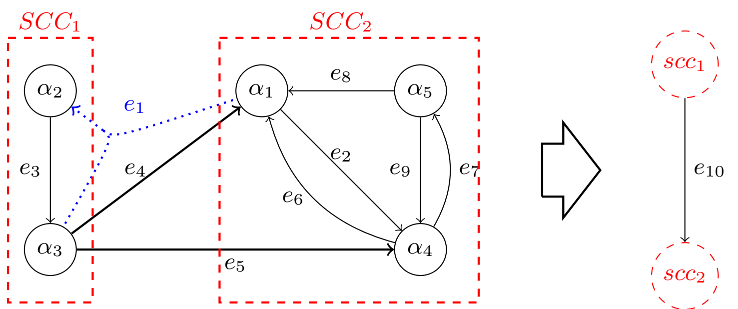
Example 4. Let $\mathcal{O} = \{\alpha_1, ..., \alpha_5\}$, where $\alpha_1 = A \sqsubseteq B$, $\alpha_2 = B \sqcap C \sqcap D \sqsubseteq E$, $\alpha_3 = E \sqsubseteq A \sqcap C \sqcap D$, $\alpha_4 = A \sqsubseteq X$, $\alpha_5 = X \sqsubseteq A$. The $\bot$-ADH $\mathcal{H}^{\bot}_{\mathcal{O}}$ contains the following hyperedges:

$$
\begin{array}{llll}
e_1 = (\{\alpha_1, \alpha_3\}, \{\alpha_2\}) & e_2 = (\{\alpha_1\}, \{\alpha_4\}) & e_3 = (\{\alpha_2\}, \{\alpha_3\}) & e_4 = (\{\alpha_3\}, \{\alpha_1\}) \\
e_5 = (\{\alpha_3\}, \{\alpha_4\}) & e_6 = (\{\alpha_4\}, \{\alpha_1\}) & e_7 = (\{\alpha_4\}, \{\alpha_5\}) & e_8 = (\{\alpha_5\}, \{\alpha_1\}) \\
e_9 = (\{\alpha_5\}, \{\alpha_4\}) & & &
\end{array}
$$

We obtain the following $\perp$-local modules for the axioms:

$$\begin{array}{ll} \mathsf{Mod}^{\perp}_{\mathcal{O}}(\alpha_1) = \{\alpha_1, \alpha_4, \alpha_5\} & \mathsf{Mod}^{\perp}_{\mathcal{O}}(\alpha_4) = \{\alpha_1, \alpha_4, \alpha_5\} \\ \mathsf{Mod}^{\perp}_{\mathcal{O}}(\alpha_2) = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5\} & \mathsf{Mod}^{\perp}_{\mathcal{O}}(\alpha_5) = \{\alpha_1, \alpha_4, \alpha_5\} \\ \mathsf{Mod}^{\perp}_{\mathcal{O}}(\alpha_3) = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5\} & \end{array}$$

The resulting atoms in $\mathsf{Atoms}^{\perp}_{\mathcal{O}}$ are $\mathfrak{a}_1 = \{\alpha_2, \alpha_3\}$ and $\mathfrak{a}_2 = \{\alpha_1, \alpha_4, \alpha_5\}$, where $\mathfrak{a}_1 \succcurlyeq \mathfrak{a}_2$, i.e. $\mathfrak{a}_2$ depends on $\mathfrak{a}_1$. The ADH $\mathcal{H}^{\perp}_{\mathcal{O}}$ with the SCCs and the condensed ADH $\mathcal{H}^{\perp}_{\mathcal{O}}/_{\simeq_B}$ is depicted in Figure 1.



(a) $\mathcal{H}^{\perp}_{\mathcal{O}}$ (b) $\mathcal{H}^{\perp}_{\mathcal{O}}/_{\simeq_B}$

Fig. 1. Example 4: From the $\perp$-ADH to the condensed $\perp$-ADH

Consider the strongly connected components of $\mathcal{H}^{\perp}_{\mathcal{O}}$. Axiom α_1 is B-connected with the axioms α_4 and α_5, α_4 is B-connected with α_1 and α_5, and α_5 is B-connected with α_1 and α_4. Axiom α_2 is B-connected with α_3 and *vice versa*. Axioms α_2, α_3 are each B-connected with α_1, α_4 and α_5, but not *vice versa*. Hence, $\{\alpha_1, \alpha_4, \alpha_5\}$ and $\{\alpha_2, \alpha_3\}$ are the strongly connected components of $\mathcal{H}^{\perp}_{\mathcal{O}}$. Moreover, we say that the former component *depends* on the latter as any two axioms contained in them are unilaterally and not mutually B-connected. Note that the atoms $\mathfrak{a}_1$ and $\mathfrak{a}_2$ of $\mathcal{O}$ and their dependency coincide with the strongly connected components of $\mathcal{H}^{\perp}_{\mathcal{O}}$. $\lhd$

Analogously to the previous section, we can characterise modules in terms of B-reachability in condensed axiom dependency hypergraphs. Proposition 4 can be lifted to cADHs as follows.

Proposition 7. *Let $\mathcal{O}$ be an ontology, Σ a signature and $x \in \{\perp, \top\}$. Let $\mathcal{O}^x_\Sigma$ be the set of axioms from $\mathcal{O}$ that are not x-local wrt. Σ. Let $\simeq$ be the mutual B-connectivity relation of the x-ADH for $\mathcal{O}$ and $\geq_B$ the B-connectivity relation of the Σ-extended x-cADH for $\mathcal{O}$. Then: $\mathsf{Mod}^x_{\mathcal{O}}(\Sigma) = \bigcup \geq_B(\{[\alpha]_\simeq \mid \alpha \in \mathcal{O}^x_\Sigma\})$.* $\dashv$

4 Implementation

The number of hyperedges of an ADH may be exponential in the size of the input ontology [9], which makes it impractical to represent the entire ADH explicitly. We implement an ADH $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ as a directed labelled graph $\mathcal{G}_{\mathcal{H}} = (\mathcal{V}, \mathcal{E}', \mathcal{L})$ containing the simple hyperedges of $\mathcal{H}$ and encoding the complex hyperedges in the node labels as follows. A node v_α in $\mathcal{G}$ for an axiom α is labelled with the pair $\mathcal{L}(v_\alpha) = (\mathsf{MLS}^x(\alpha), \mathsf{sig}(\alpha))$ consisting of the minimal non-x-locality signatures of α and the signature of α, where $x \in \{\bot, \top\}$. In fact, not all symbols of $\mathsf{sig}(\alpha)$ are needed in the second component, only those symbols that occur in the minimal non-locality signature of some axiom in the ontology. Condensed axiom dependency hypergraphs are implemented in a similar way with the difference that nodes represent sets of axioms. A node v_S for a set S of axioms is labelled with the pair $\mathcal{L}(v_S) = (\mathsf{MLS}^x(S), \mathsf{sig}(S))$, where $\mathsf{MLS}^x(S) = \bigcup_{\alpha \in S} \mathsf{MLS}^x(\alpha)$ and $\mathsf{sig}(S) = \bigcup_{\alpha \in S} \mathsf{sig}(\alpha)$.

We introduce the notion of a graph representation of an axiom dependency hypergraph that may be (partially) condensed.

Definition 10. *Let $\mathcal{H} = (\mathcal{V}_{\mathcal{H}}, \mathcal{E}_{\mathcal{H}})$ be an ADH, pcADH or cADH. Let $x \in \{\bot, \top\}$ be a syntactic locality notion. The* graph representation $\mathcal{G}_{\mathcal{H}}$ of $\mathcal{H}$ *is the directed labelled graph $\mathcal{G}_{\mathcal{H}} = (\mathcal{V}, \mathcal{E}, \mathcal{L})$, where*

- $\mathcal{V} := \mathcal{V}_{\mathcal{H}}$;
- $\mathcal{E} := \{(v, v') \mid \Sigma_{v'} \subseteq \mathsf{sig}(v),$ *for some* $\Sigma_{v'} \in MLS^x(v')\}$;
- $\mathcal{L}(v) := (MLS^x(v), \mathsf{sig}(v))$, *for every* $v \in \mathcal{V}$. $\dashv$

To define the graph representation $\mathcal{G}_{\mathcal{H}}$ of a hypergraph $\mathcal{H}$, we assume that every node v in $\mathcal{H}$ is associated with a set $\mathsf{MLS}^x(v)$ of minimal non-locality signatures, and a set $\mathsf{sig}(v)$ of signature symbols. Note that a node in $\mathcal{H}$ represents an axiom if $\mathcal{H}$ is an ADH, and a set of axioms if $\mathcal{H}$ is a pcADH or a cADH.

4.1 Atomic Decomposition

For a collection of well-known biomedical ontologies from the NCBO Bioportal, we observe that for many (if not all) axioms, the locality-based dependencies to other axioms can be represented using only *simple* directed hyperedges. For instance, the ADH for ontologies like CHEBI can be seen as a directed graph without *complex* hyperedges. Computing strongly connected components in a directed graph can be done in linear-time using standard algorithms from graph theory [11, 13]. That is, for ontologies like CHEBI we compute the strongly connected components of the respective ADH in linear time.

For ADHs of ontologies $\mathcal{O}$ like SNOMED CT that contain both, simple and complex hyperedges, we compute the strongly connected components in four steps. First, we build the axiom dependency graph $\mathcal{G}_{\mathcal{H}^x_{\mathcal{O}}}$, which is the fragment of the ADH $\mathcal{H}^x_{\mathcal{O}}$ for $\mathcal{O}$ without complex hyperedges. Second, we compute the strongly connected components of $\mathcal{G}_{\mathcal{H}^x_{\mathcal{O}}}$ using a linear-time algorithm [11, 13].

Note that the strongly connected components give rise to an equivalence relation $\simeq_{B_\mathcal{G}}$ on the nodes in $\mathcal{G}_{\mathcal{H}^x_\mathcal{O}}$. In the third step, we reduce the size of $\mathcal{H}^x_\mathcal{O}$ by computing the quotient graph $\mathcal{H}^x_\mathcal{O}/_{\simeq_{B_\mathcal{G}}}$ of $\mathcal{H}^x_\mathcal{O}$ using $\simeq_{B_\mathcal{G}}$ (cf. Def. 7). This corresponds to the computation of the pcADH, $\mathcal{H}^x_\mathcal{O}/_{\simeq_{B_\mathcal{G}}}$, for the ADH $\mathcal{H}^x_\mathcal{O}$. Finally, in step four, we obtain the strongly connected components of $\mathcal{H}^x_\mathcal{O}$ by determining for any two nodes in $\mathcal{H}^x_\mathcal{O}/_{\simeq_{B_\mathcal{G}}}$ whether they are mutually reachable. This last step produces the cADH, $\mathcal{H}^x_\mathcal{O}/_{\simeq_{B_\mathcal{H}}}$, where $\simeq_{B_\mathcal{H}}$ is the mutual B-connectivity relation in $\mathcal{H}^x_\mathcal{O}/_{\simeq_{B_\mathcal{G}}}$. Note that computing mutual reachability this way is a quadratic process [1]. However, using $\mathcal{H}^x_\mathcal{O}/_{\simeq_{B_\mathcal{G}}}$ instead of $\mathcal{H}^x_\mathcal{O}$ it is usually more efficient as the number of nodes is typically reduced.

The function $\mathsf{compute\_condensed\_hypergraph}(.)$ provides a more succinct description of the previous process.

---

function $\mathsf{compute\_condensed\_hypergraph}(\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{L}))$ returns $\mathcal{G}_c$

1: $\mathcal{G}_{pc} := \mathsf{collapse\_SCCs}(\mathcal{G}, \mathsf{Tarjan}((\mathcal{V}, \mathcal{E})))$
2: if ($\mathsf{contains\_complex\_Dependencies}(\mathcal{G}_{pc}) = \mathsf{false}$) then
3:     return $\mathcal{G}_c := \mathcal{G}_{pc}$
4: end if
5: $\mathcal{G}_c := \mathsf{collapse\_SCCs}(\mathcal{G}_{pc}, \mathsf{mutual\_reach}(\mathcal{G}_{pc}))$
6: return $\mathcal{G}_c$

Given the graph representation $\mathcal{G}$ of an ADH $\mathcal{H}^x_\mathcal{O}$, the function $\mathsf{compute\_condensed\_hypergraph}(\mathcal{G})$ computes the graph representation, denoted with $\mathcal{G}_c$, of the cADH of $\mathcal{H}^x_\mathcal{O}$ in two main steps. In the first step, the function computes the graph representation of the pcADH, which we denote with $\mathcal{G}_{pc}$ (Line 1). Only simple directed hyperedges ($\mathcal{E}$) of $\mathcal{G}$ are considered. The strongly connected components are determined in linear time using the Tarjan algorithm [13] (Line 2). The computation of the strongly connected components when complex directed hyperedges are considered is done in Line 5. After the strongly connected components are identified, the function $\mathsf{collapse\_SCCs}$ produces the graph representation $\mathcal{G}_c$ of the cADH for $\mathcal{H}^x_\mathcal{O}$.

4.2 Module Extraction

Modules correspond to connected components in the axiom dependency hypergraph or its (partially) condensed version. We now present the algorithm for computing the connected components in the graph representation of a directed hypergraph that can encode an ADH, pcADH or cADH for the input ontology.

The function $\mathsf{Mod}^x(\mathcal{G}, \Sigma)$ computes all Σ-reachable nodes in the labelled graph $\mathcal{G}$ and returns the axioms represented by these nodes. In Line 2, the algorithm determines the set $\mathcal{S}_1$ of initial nodes in $\mathcal{G}$. Every initial node $\mathcal{S}_1$ is associated with a minimal non-locality signature that is contained in Σ. In Line 5, the set of nodes is determined that are reachable via simple B-hyperedges that are explicitly given in $\mathcal{E}$. Note that $\mathcal{E}(v)$ denotes the set of nodes that are directly reachable in $\mathcal{G}$ from the node v using simple directed hyperedges.

function $\mathsf{Mod}^x(\mathcal{G}_{\mathcal{H}^x_{\mathcal{O}}} = (\mathcal{V}, \mathcal{E}, \mathcal{L}), \Sigma)$ returns x-local module of $\mathcal{O}$ wrt. Σ

1: $\Sigma_0 := \Sigma$, $m := 1$

2: $\mathcal{S}_0 := \emptyset$, $\mathcal{S}_1 := \{v \in \mathcal{V} \mid \Sigma_v \subseteq \Sigma_0$ for some $\Sigma_v \in \mathsf{MLS}^x(v)\}$

3: do

4: $\quad m := m + 1$

5: $\quad \mathcal{S}_m := \bigcup \{ \mathcal{E}(v) \mid v \in \mathcal{S}_{m-1} \setminus \mathcal{S}_{m-2} \} \cup \mathcal{S}_{m-1}$

6: $\quad \Sigma_m := (\bigcup_{s \in \mathcal{S}_m \setminus \mathcal{S}_{m-1}} \mathsf{sig}(s)) \cup \Sigma_{m-1}$

7: $\quad \mathcal{S}_m := \mathcal{S}_m \cup \{v \in \mathcal{V} \mid \Sigma_v \subseteq \Sigma_m$ for some $\Sigma_v \in \mathsf{MLS}^x(v)$ with $|\Sigma_v| > 1\}$

8: until $\mathcal{S}_m = \mathcal{S}_{m-1}$

9: return $\mathsf{get\_axioms}(\mathcal{S}_m)$

In Line 7, the input signature is extended with the symbols that are associated to the nodes reached so far. Using the extended signature Σ_m, the function $\mathsf{Mod}^x(\cdot, \cdot)$ computes the nodes that can be reached using complex B-hyperedges implicitly represented by the labels $\mathcal{L}(v)$ of the nodes v in $\mathcal{S}_m$. The algorithm iterates until a fix point is reached and no more new nodes are added (Lines 3−8). Finally, in Line 9, the function $\mathsf{get\_axioms}(\cdot)$ computes the set of axioms that correspond to the nodes in $\mathcal{S}_m$.

5 Evaluation

The system HyS is a Java implementation of the approach described in the previous section. HyS can compute syntactic locality-based modules for a given input signature and the atomic decomposition of an ontology defined in $\mathcal{EL}^{++}$ extended with inverse and functional role axioms.[7] In the current version of HyS only syntactic $\bot$-locality is supported. We plan to extend the implementation to support both $\top$-locality and full $\mathcal{SROIQ}$-ontologies in the future.

For the evaluation, we have selected nine well-known biomedical ontologies. Seven of them are available in the NCBO Bioportal. The version of Full-Galen that we used is available in the Oxford ontology repository.[8]

We divide the ontologies into two groups: a group consisting of CHEBI, FMA-lite, Gazetteer, GO, NCBI and RH-Mesh, and another group consisting of CPO, Full-Galen and SNOMED CT. Every ontology in the former group consist of axioms whose $\bot$-locality dependencies between axioms can be represented using simple directed hyperedges only. This means that the ADH can be represented using a direct graph. On the other hand, each of the latter three ontologies contain axioms that require complex hyperedges to represent the dependencies.

We compare HyS against two systems for computing the atomic decomposition of OWL 2 ontologies which implement the same algorithm from [14]:

[7] HyS supports all the constructors used in the ontology Full-Galen

[8] `http://www.cs.ox.ac.uk/isg/ontologies/`

FaCT++ v1.6.2, which is implemented in C++ [14][9], and OWLAPITOOLS v1.0.0 which is implemented in Java [15][10] as an extension of the OWLAPI.[11]

Ontology $\mathcal{O}$	Properties of $\mathcal{O}$				Time for Atomic Dec. of $\mathcal{O}$		
	Signature size	#axioms $A \sqsubseteq C$	#axioms $C \equiv D$	#role axioms	FaCT++	OWLAPI TOOLS	HyS
CHEBI	37 891	85 342	0	5	137 s	1 619 s	4 s
FMA-lite	75 168	119 558	0	3	18 481 s	13 258 s	17 s
Gazetteer	517 039	652 355	0	6	31 595 s	–	24 s
GO	36 945	72 667	0	2	47 s	1 489 s	4 s
NCBI	847 796	847 755	0	0	49 228 s	–	66 s
RH-Mesh	286 382	403 210	0	0	6 921 s	9 159 s	17 s
CPO	136 090	306 111	73 461	96	9 731 s	26 480 s	2 283 s
Full-Galen	24 088	25 563	9 968	2 165	640 s	781 s	115 s
SNOMED CT	291 207	227 698	63 446	12	16 081 s	57 282 s	2 540 s

All experiments were conducted on an Intel Xeon E5-2640 2.50GHz with 100GB RAM running Debian GNU/Linux 7.3. We use Java 1.7.0_51 and the OWLAPI version 3.5.0. The table lists the time needed for each system to compute the atomic decomposition of the ontologies. The time values are the average of at least 10 executions. We applied a timeout of 24h, which aborted the executions of the OWLAPITOOLS on the ontologies Gazetteer and NCBI. Moreover, the table contains, for each ontology, the size of the signature, the number of axioms of the form $A \sqsubseteq C$, where A is a concept name, the number of axioms of the form $C \equiv D$, the number of role axioms contained in the ontology.

HyS consistently outperforms FaCT++ which in turn (considerably) outperforms the OWLAPITOOLS, with the exception of FMA-lite. In the case of the first group of six ontologies, an over 1 000-fold speedup could be achieved compared to the performance of FaCT++ on FMA-lite and Gazetteer. For the smallest ontology in this group, which is GO, HyS is 13 times faster than FaCT++. HyS also scales better than the other systems. For the second group of three ontologies, the speedup is reduced but HyS is still considerably faster. HyS is 4–7 times faster than FaCT++ and 11–23 faster than the OWLAPITOOLS. The computation of the partially condensed ADH nearly decreases 50% the number of nodes in the ADH. The use of a tree datastructure to represent the set of reachable nodes computed for each node of the ADH reduces the time needed to identify mutually reachable nodes.

We compare the performance of HyS for extracting $\bot$-locality modules with the performance of FaCT++ and the OWLAPI. The following table presents for every method the time needed to extract a module from an ontology for a signature consisting of 500 symbols selected at random.

[9] `http://code.google.com/p/factplusplus/`
[10] `http://owlapitools.sourceforge.net/`
[11] `http://owlapi.sourceforge.net/`

Ontology $\mathcal{O}$	Time for Extraction of $\perp$-local Modules from $\mathcal{O}$				
	FacT++	OWLAPI	HyS		
			ADH	pcADH	cADH
CHEBI	38.6 ms	175.8 ms	3.9 ms	2.4 ms	2.1 ms
FMA-lite	326.9 ms	1 042.3 ms	55.3 ms	3.9 ms	3.4 ms
Gazetteer	177.9 ms	1 503.0 ms	27.3 ms	16.1 ms	15.9 ms
GO	512.2 ms	1 398.7 ms	8.1 ms	6.2 ms	6.1 ms
NCBI	236.2 ms	9 193.6 ms	22.7 ms	15.8 ms	16.3 ms
RH-Mesh	91.2 ms	1 811.3 ms	10.6 ms	9.1 ms	8.9 ms
CPO	564.7 ms	3 026.8 ms	84.3 ms	53.4 ms	51.6 ms
Full-Galen	75.2 ms	215.4 ms	13.2 ms	3.7 ms	2.9 ms
SNOMED CT	525.0 ms	2 841.3 ms	93.6 ms	88.4 ms	84.4 ms

HyS outperforms FaCT++ and the OWLAPITOOLS in all cases. For the first group of six ontologies, the best speedup of over 90 times was achieved in the case of FMA-lite. Notice that module extraction times using the pcADH and the cADH (last two columns) are nearly the same as the two graphs are equivalent. The small variation in extraction time is due to noise in the execution environment. The differences in the times values in the third column and the last two columns correspond to the differences in size of the ADH and the pcADH/cADH. For the second group of three ontologies, the best performance improvement was realised in the case of Full-Galen with a speedup of over 20-times. However, we note that using the cADH instead of the pcADH does not yield a large performance difference despite the fact that the cADH is slightly smaller than the pcADH. In the particular case of Full-Galen, there appears to be a trade-off between condensation and increased time needed to perform signature containment checks. Computing the partially condensed ADH (using a linear time algorithm) is generally much faster than computing the condensed ADH (which is done in quadratic time). Given that the module extraction times are similar when using the pcADH and the cADH (cf. the times in the last two columns), it seems more efficient to only compute modules using the partially condensed ADH.

6 Conclusion

We have introduced the notion of an axiom dependency hypergraph that represents explicitly the locality-based dependencies between axioms. We have shown that locality-based modules of an ontology correspond to a set of connected nodes in the hypergraph, and atoms of an ontology to strongly connected components. We have implemented a prototype in Java that computes, based on axiom dependency hypergraphs, the atomic decomposition of $\mathcal{EL}^{++}$-ontologies wrt. $\perp$-locality. Our prototype outperforms FaCT++ and the OWLAPITOOLS in computing the atomic decomposition of all biomedical ontologies tested. In some cases a staggering speedup of over 1 000 times could be realised. Moreover, the prototype significantly outperforms FaCT++ and the OWLAPI in extracting syntactic $\perp$-locality modules.

We plan to extend the prototype implementation to support both $\top$-locality and full $\mathcal{SROIQ}$-ontologies. Moreover, it would be interesting to investigate the possibility to compute strongly connected components in hypergraphs in less than quadratic time. Such a result would improve the performance of computing mutual reachability in the axiom dependency hypergraph for ontologies whose locality-based dependencies can only be represented by hyperedges with more than one node in the tail.

References

1. X. Allamigeon. On the complexity of strongly connected components in directed hypergraphs. *Algorithmica*, volume 69, issue 2, pages 335–369, June 2014.
2. F. Baader, S. Brandt, and C. Lutz. Pushing the $\mathcal{EL}$ envelope further. In *In Proc. of the OWLED'08 DC Workshop on OWL: Experiences and Directions*, 2008.
3. F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The description logic handbook: theory, implementation, and applications.* Cambridge University Press, 2007.
4. C. Del Vescovo, D. D. G. Gessler, P. Klinov, B. Parsia, U. Sattler, T. Schneider, and A. Winget. Decomposition and modular structure of bioportal ontologies. In *Proc. of ISWC'11*, pages 130–145. Springer-Verlag, 2011.
5. C. Del Vescovo, B. Parsia, U. Sattler, and T. Schneider. The modular structure of an ontology: Atomic decomposition. In *Proc. of IJCAI'11*, pages 2232–2237, 2011.
6. G. Gallo, G. Longo, S. Pallottino, and S. Nguyen. Directed hypergraphs and applications. *Discrete Applied Mathematics*, volume 42, issue 2–3, pages 177–201, 1993.
7. B. C. Grau, I. Horrocks, Y. Kazakov, and U. Sattler. Modular reuse of ontologies: theory and practice. *JAIR*, volume 31, pages 273–318, 2008.
8. I. Horrocks, O. Kutz, and U. Sattler. The even more irresistible SROIQ. In *Proc. of KR'06*, pages 57–67. AAAI Press, 2006.
9. F. Martín-Recuerda and D. Walther. Towards fast atomic decomposition using axiom dependency hypergraphs. In *Proc. of WoMO'13*, pages 61–72. CEUR-WS.org, 2013.
10. R. Nortje, A. Britz, and T. Meyer. Reachability modules for the description logic SRIQ. In *Proc. of LPAR-13*, volume 8312, pages 636–652, 2013.
11. M. Sharir. A strong connectivity algorithm and its applications to data flow analysis. *Computers & Mathematics with Applications*, volume 7, issue 1, pages 67–72, 1981.
12. B. Suntisrivaraporn. *Polynomial time reasoning support for design and maintenance of large-scale biomedical ontologies.* PhD thesis, TU Dresden, Germany, 2009.
13. R. E. Tarjan. Depth-first search and linear graph algorithms. *SIAM J. Computation*, volume 1, issue 2, pages 146–160, 1972.
14. D. Tsarkov. Improved algorithms for module extraction and atomic decomposition. In *Proc. of DL'12*, volume 846. CEUR-WS.org, 2012.
15. D. Tsarkov, C. D. Vescovo, and I. Palmisano. Instrumenting atomic decomposition: Software APIs for OWL. In *Proc. of OWLED-13*, volume 1080. CEUR-WS.org, 2013.
16. V. K. C. Turlapati and S. K. Puligundla. Efficient module extraction for large ontologies. In *Knowledge Engineering and the Semantic Web*, volume 394, pages 162–176. Springer Berlin Heidelberg, 2013.