Bidirectional reachability-based modules

Riku Nortje[1,2], Katarina Britz[1,2], and Thomas Meyer[1,3]

[1]CSIR Meraka Institute, Pretoria, South Africa
[2]University of South Africa, Pretoria, South Africa
[3]University of KwaZulu-Natal, Durban, South Africa
Email: nortjeriku@gmail.com; {arina.britz;tommie.meyer}@meraka.org.za

Abstract. We introduce an algorithm for MinA extraction in $\mathcal{EL}$ based on bidirectional reachability. We obtain a significant reduction in the size of modules extracted at almost no additional cost to that of extracting standard reachability-based modules. Bidirectional modules are related to nested locality modules, but are aimed specifically at MinA extraction and are generally smaller. For acyclic $\mathcal{EL}$ TBoxes consisting of only primitive concept inclusions, all MinAs can be extracted without the need for subsumption testing.

1 Introduction

Module extraction plays an important role in the design, reuse and maintenance of ontologies as well as aiding in the optimization of reasoning services [9]. When used to optimize reasoning services such as subsumption testing and MinA extraction, reachability-based modules have been criticized for only considering the subsumee of a subsumption entailment during the module extraction process [2], thus not sufficiently reducing the size of modules.

In this paper we address this shortcoming of reachability-based modules, with the aim of improving MinA extraction, as follows: We introduce a top-down heuristic which considers only the subsumer of an entailment and then combine it with standard reachability-based modules to form a bidirectional version of reachability. This new bidirectional version of the heuristic thus considers both the subsumee and subsumer in a subsumption entailment between concept names. For relatively sparse graphs this significantly reduces the size of modules extracted with almost no additional cost to that of extracting standard reachability-based modules.

Given a subsumption statement between single concept names, we show that every MinA is in fact a bidirectional reachability-based module in terms of itself. Using this property we implement very fast algorithms to extract all MinAs for acyclic $\mathcal{EL}$ TBoxes consisting of only primitive concept inclusions without performing a single subsumption test, thereby significantly reducing the runtime complexity of MinA extraction for these TBoxes.

In Section 2 we give a brief introduction to description logics and the notations as used in this paper. Section 3 introduces reachability-based modules [9], the new top-down reachability heuristic and finally defines bidirectional

reachability-based modules. Then in Section 4 we investigate the relationship between MinAs and the inexpressive Horn DL $\mathcal{HL}$ and extend the findings to $\mathcal{EL}$ TBoxes consisting of primitive concept inclusions. Lastly in Section 5 we provide empirical results of the various algorithms presented as tested on three generally large real world biomedical ontologies.

2 Preliminaries

In the standard set-theoretic semantics of concept descriptions, concepts are interpreted as subsets of a domain of interest, and roles as binary relations over this domain. An interpretation I consists of a non-empty set Δ^I (the *domain* of I) and a function $\cdot^I$ (the *interpretation function* of I) which maps each atomic concept A to a subset A^I of Δ^I, and each atomic role r to a subset r^I of $\Delta^I \times \Delta^I$. The interpretation function is extended to arbitrary concept and role descriptions, with the specifics depending on the particular description logic under consideration.

A DL knowledge base consists of a *TBox* which contains *terminological axioms* and an *ABox* which contains *assertions*; for the purposes of this paper we concern ourselves only with Tbox statements, or *general concept inclusions* (GCIs) of the form $C \sqsubseteq D$, where C and D are (possibly complex) concept descriptions. Here C is referred to as the *subsumee* and D as the *subsumer*. An interpretation I *satisfies* $C \sqsubseteq D$, written $I \Vdash C \sqsubseteq D$, iff $C^I \subseteq D^I$. In this paper, when the left hand side of a GCI consists of only a single concept name, the statement is referred to as a primitive concept inclusion.

An interpretation I satisfies a DL TBox $\mathcal{T}$ iff it satisfies every statement in $\mathcal{T}$. A TBox $\mathcal{T}$ *entails* a DL statement ϕ, written as $\mathcal{T} \models \phi$, iff every interpretation that satisfies $\mathcal{T}$ also satisfies ϕ.

Roughly speaking, DLs are defined by the constructors they provide. In this paper we consider the DLs $\mathcal{HL}$ and $\mathcal{EL}$. The constructors allowed for $\mathcal{EL}$ are conjunction ($\sqcap$) and existential restriction ($\exists$), with semantics defined as follows: $(C \sqcap D)^I = C^I \cap D^I$; $(\exists r.C)^I = \{x \in \Delta^I \mid \exists y \in \Delta^I : (x, y) \in r^I \wedge y \in C^I\}$. The only concept constructor allowed for $\mathcal{HL}$ is conjunction, with semantics as for $\mathcal{EL}$. Both $\mathcal{HL}$ and $\mathcal{EL}$ also have the distinguished top concept $\top$ with semantics $\top^I = \Delta^I$. Normalization for $\mathcal{HL}$ only allows GCIs of the form $A \sqsubseteq B$ and $A_1 \sqcap A_2 \sqsubseteq B$. For $\mathcal{EL}$, GCIs of the form $A \sqsubseteq \exists r.B$ and $\exists r.A \sqsubseteq B$ are also allowed. Given any concept description or subsumption statement α, Sig(α) is defined as the union of all concept and role names occurring in α.

Definition 1. (Module) *Let $\mathcal{L}$ be an arbitrary description language, $\mathcal{O}$ an $\mathcal{L}$ ontology, and σ a statement formulated in $\mathcal{L}$. Then, $\mathcal{O}' \subseteq \mathcal{O}$ is a* module for σ in $\mathcal{O}$ *(a σ-*module in $\mathcal{O}$*) whenever: $\mathcal{O} \models \sigma$ if and only if $\mathcal{O}' \models \sigma$. We say that $\mathcal{O}'$ is a module for a signature $\boldsymbol{S}$ in $\mathcal{O}$ (an* **S**-module in $\mathcal{O}$*) if, for every $\mathcal{L}$ statement σ with Sig(σ) $\subseteq \boldsymbol{S}$, $\mathcal{O}'$ is a σ-module in $\mathcal{O}$. Given the statement σ, if there is no $\mathcal{O}'' \subset \mathcal{O}'$ such that $\mathcal{O}'' \models \sigma$ then $\mathcal{O}'$ is a minimal σ-module.*

Given a subsumption statement $\sigma = A \sqsubseteq B$, a MinA is defined as a minimal set of axioms $\mathcal{O}'$ such that $\mathcal{O}' \models A \sqsubseteq B$. Though these are not usually referred to as modules in the literature, MinAs are by definition minimal modules for a specific statement of interest.

3 Bidirectional Reachability-based Modules for $\mathcal{EL}$

Extracting modules aims to preserve both subsumption and non-subsumption relationships in a subset of an ontology. This can be understood as the reachability problem in a directed graph [1], considering concept names as nodes and explicit subsumption relationships as edges in the graph, where each inclusion axiom $\alpha_L \sqsubseteq \alpha_R \in \mathcal{O}$ essentially specifies a collection of hyperedges from the connected node $\text{Sig}(\alpha_L)$ to each of the symbols in $\text{Sig}(\alpha_R)$.

Definition 2. (Bottom-up reachability-based modules [9])[1] *Let $\mathcal{O}$ be an $\mathcal{EL}$ ontology and $S \subseteq Sig(\mathcal{O})$ a signature. The set of S-reachable names in $\mathcal{O}$ is defined inductively as follows: (i) x is S-reachable in $\mathcal{O}$, for every $x \in S$; and (ii) for all inclusion axioms $\alpha_L \sqsubseteq \alpha_R$, if x is S-reachable in $\mathcal{O}$ for every $x \in Sig(\alpha_L)$, then y is S-reachable in $\mathcal{O}$ for every $y \in Sig(\alpha_R)$. We call an axiom $\alpha_L \sqsubseteq \alpha_R$ S-reachable in $\mathcal{O}$ if every element of $Sig(\alpha_L)$ is S-reachable in $\mathcal{O}$. The bottom-up reachability-based module for S in $\mathcal{O}$, denoted by $\mathcal{O}_S^{reach}$, consists of all S-reachable axioms in $\mathcal{O}$.*

When S is the single concept A, we write A-reachable and $\mathcal{O}_A^{reach}$. For $\mathcal{EL}$, axioms of the form $\top \sqsubseteq \alpha_R$ are such that $\text{Sig}(\top) = \emptyset$, thus they will form part of every reachability-based module extracted. Bottom-up reachability-based modules are in fact equivalent to $\bot$-locality based modules [4, 8].

A criticism that may be raised against these bottom-up reachability-based modules is that they contain many irrelevant axioms and in some cases do not reduce the size of the ontology at all [2]. This stems from the fact that $\mathcal{O}_A^{reach}$ considers only the subsumee A in $\mathcal{O} \models A \sqsubseteq B$; the subsumer B is never used to eliminate unwanted axioms. For example:

Example 1. Given the ontology $\mathcal{O} = \{A \sqsubseteq \exists r.D, \exists r.D \sqsubseteq B, E \sqsubseteq B, A \sqsubseteq F\}$, as well as the entailment $\mathcal{O} \models A \sqsubseteq B$, $\mathcal{O}_A^{reach}$ consists of axioms $\{A \sqsubseteq \exists r.D, \exists r.D \sqsubseteq B, A \sqsubseteq F\}$. $A \sqsubseteq F$ is irrelevant in terms of $\mathcal{O} \models A \sqsubseteq B$, yet it is included in $\mathcal{O}_A^{reach}$.

For large ontologies many such irrelevant axioms may be included in a bottom-up reachability-based module. We introduce modules based on the subsumer of an entailment namely top-down reachability-based modules. Formally:

[1] The original definition by Suntisrivaraporn does not have the qualifier 'bottom-up', but because we introduce 'top-down' reachability-based modules later on in Definition 3, the qualifier is used to avoid confusion.

Definition 3. (Top-down reachability-based module) *Let $\mathcal{O}$ be an $\mathcal{EL}$ ontology and $S \subseteq Sig(\mathcal{O})$ a signature. The set of $\overleftarrow{\text{S}}$-reachable names in $\mathcal{O}$ is defined inductively as follows: (i) x is $\overleftarrow{\text{S}}$-reachable in $\mathcal{O}$, for every $x \in S$; and (ii) for all inclusion axioms $\alpha_L \sqsubseteq \alpha_R$, if x is $\overleftarrow{\text{S}}$-reachable in $\mathcal{O}$ for some $x \in Sig(\alpha_R)$, then y is $\overleftarrow{\text{S}}$-reachable in $\mathcal{O}$ for every $y \in Sig(\alpha_L)$. We call an axiom $\alpha_L \sqsubseteq \alpha_R$ $\overleftarrow{\text{S}}$-reachable in $\mathcal{O}$ if some element of $Sig(\alpha_R)$ is $\overleftarrow{\text{S}}$-reachable. The top-down reachability-based module for S in $\mathcal{O}$, denoted by $\mathcal{O}^{reach}_{\overleftarrow{\text{S}}}$, consists of all $\overleftarrow{\text{S}}$-reachable axioms from $\mathcal{O}$.*

Algorithm 1 extracts a top-down reachability based module, given an $\mathcal{EL}$ TBox $\mathcal{O}$ and a signature S as input. `active-axioms`(x) are all those, and only those axioms $(\alpha_L \sqsubseteq \alpha_R) \in \mathcal{O}$ such that $x \in \text{Sig}(\alpha_R)$, thus every such axiom is also by definition top-down reachable. For a signature S we define `active-axioms` $(S) := \bigcup_{x \in S}$`active-axioms`(x).

Algorithm 1 (`Extract top-down reachability-based module`)

```
Procedure extract-top-down-module(O, S)
Input: O - EL ontology; S - signature
Output: O_S: top-down reachability-based module for S in O
1: O_S := ∅; queue := active-axioms(S)
3: while not empty(queue) do
4:    (α_L ⊑ α_R) := fetch(queue)
5:    O_S := O_S ∪ {α_L ⊑ α_R}
6:    queue := queue ∪ (active-axioms(Sig(α_L)) \O_S)
7: return O_S
```

Theorem 1. *[5] Let $\mathcal{O}$ be an $\mathcal{EL}$ ontology, n the number of axioms in $\mathcal{O}$, and $S \subseteq Sig(\mathcal{O})$ a signature. Algorithm 1 terminates after $O(n)$ steps and returns the top-down reachability-based module for S in $\mathcal{O}$.*

It is easy to show that top-down reachability-based modules are equivalent to a subset of $\top$-locality modules [4, 8]. These modules can be criticized in a similar manner to bottom-up reachability-based modules, in that they include many irrelevant axioms. Combining $\bot$-locality modules with $\top$-locality based modules allows us to extract so called nested locality modules denoted by $\top\bot$ or $\bot\top$ [8]. We introduce a slightly different form of module called *bidirectional reachability-based modules*, aimed towards finding small modules preserving subsumption relationships between single concept names.

Definition 4. (Bidirectional reachability-based module [6]) *The bi-directional reachability-based module, denoted $\mathcal{O}^{reach}_{A \leftrightarrow B}$, for the statement $A \sqsubseteq B$ in terms of $\mathcal{O}$, is defined as the set of all axioms $\alpha_L \sqsubseteq \alpha_R \in \mathcal{O}$ such that: for every $x_i \in Sig(\alpha_L)$, x_i is A-reachable in terms of $\mathcal{O}$, and α_R is $\overleftarrow{\text{B}}$-reachable in terms of $\mathcal{O}$. Any non-empty subset $\mathcal{O}' \subseteq \mathcal{O}^{reach}_{A \leftrightarrow B}$ such that $\mathcal{O}'^{reach}_{A \leftrightarrow B} = \mathcal{O}'$ is called a bidirectional reachability-based sub-module of $\mathcal{O}$ for the statement $A \sqsubseteq B$. $\mathcal{O}'^{reach}_{A \leftrightarrow B}$ is minimal if there exists no $\mathcal{O}'' \subset \mathcal{O}'$ such that $\mathcal{O}''^{reach}_{A \leftrightarrow B} = \mathcal{O}''$.*

These modules differ from nested locality modules as follows: Given a subsumption statement $A \sqsubseteq B$, and the $\bot\top$ module $\mathcal{O}'$, then $\mathcal{O}'$ will contain all axioms for the signature $S = \{A, B\}$, thus it will include all the axioms for the entailments $\mathcal{O}' \models A \sqsubseteq B$ and $\mathcal{O}' \models B \sqsubseteq A$. A bidirectional reachability-based module $\mathcal{O}''$, however, only contains axioms for the entailment $\mathcal{O}'' \models A \sqsubseteq B$. Using the notation that $\bot\{A, B\}$ represents the $\bot$ locality module for the signature $\{A, B\}$ the relationship between these modules can be illustrated as follows:

$$\mathcal{O}^{reach}_{A \leftrightarrow B} \subseteq \bot\{A\}\top\{B\} \subseteq \bot\{A\}\top\{B\} \cup \bot\{B\}\top\{A\} \subseteq \bot\top\{A, B\} \subseteq \bot\{A, B\}$$

The following example shows the relationship between a bidirectional reachability-based module, bidirectional reachability-based sub-modules and minimal bidirectional reachability-based modules.

Example 2. Given the ontology $\mathcal{O}$ consisting of the set of axioms: $\{\alpha_1 : A \sqsubseteq C_1$, $\alpha_2 : A \sqsubseteq D$, $\alpha_3 : D \sqsubseteq C_3$, $\alpha_4 : C_1 \sqsubseteq \exists R.C_2$, $\alpha_5 : C_2 \sqsubseteq C_3$, $\alpha_6 : C_3 \sqcap C_4 \sqsubseteq B$, $\alpha_7 : \exists r.C_3 \sqsubseteq C_4$, $\alpha_8 C_3 \sqsubseteq B$, $\alpha_9 : C_2 \sqsubseteq E$, $\alpha_{10} : E \sqsubseteq F\}$, as well as the statement $\mathcal{O} \models A \sqsubseteq B$, we have that:

- $\mathcal{O}^{reach}_{A} = \mathcal{O}$,
- $(\mathcal{O}^{reach}_{A})^{reach}_{\overline{\mathrm{B}}} = \mathcal{O}^{reach}_{A \leftrightarrow B}$ consist of axioms: $\{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7, \alpha_8\}$
- Given that the sets $\mathcal{O}_0$, $\mathcal{O}_1$, $\mathcal{O}_2$ and $\mathcal{O}_3$ are defined as follows: $\mathcal{O}_0 = \{\alpha_1, \alpha_4, \alpha_5, \alpha_6, \alpha_7\}$, $\mathcal{O}_1 = \{\alpha_2, \alpha_3, \alpha_8\}$, $\mathcal{O}_2 = \{\alpha_1, \alpha_4, \alpha_5, \alpha_8\}$, $\mathcal{O}_3 = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7\}$, then
  - $\mathcal{O}_0$, $\mathcal{O}_1$, $\mathcal{O}_2$ and $\mathcal{O}_3$ are bidirectional reachability-based sub-modules $\mathcal{O}_i \subseteq \mathcal{O}^{reach}_{A \leftrightarrow B}$, that is, $\mathcal{O}_i = \mathcal{O}_i{}^{reach}_{A \leftrightarrow B}$.
  - $\mathcal{O}_1$ and $\mathcal{O}_2$ are both minimal bidirectional reachability-based modules with $\mathcal{O}_1$ being the only one of these sets that is both a minimal bidirectional reachability-based module and a MinA for the statement $A \sqsubseteq B$ such that $\mathcal{O}_1 \models A \sqsubseteq B$.
  - $\mathcal{O}_3$ is a MinA for the statement $A \sqsubseteq B$ such that $\mathcal{O}_3 \models A \sqsubseteq B$ but $\mathcal{O}_3$ is not a minimal bidirectional reachability-based module.

The algorithms for both bottom-up and top-down reachability based modules extraction methods may now be applied in any order and in sequence to extract bidirectional reachability-based modules. Since $\mathcal{O}^{reach}_{\overline{\mathrm{B}}}$ is in general very large, we prefer to extract $(\mathcal{O}^{reach}_{A})^{reach}_{\overline{\mathrm{B}}}$.

An interesting property of bidirectional reachability-based modules for $\mathcal{EL}$ is that every MinA for a subsumption statement is a bidirectional reachability-based module in terms of itself. Formally:

Theorem 2. *[5] Given an $\mathcal{EL}$ TBox $\mathcal{T}$ and the statement $A \sqsubseteq B$ such that $\mathcal{T} \models A \sqsubseteq B$. Let $M_1 \subseteq \mathcal{T}$ be a MinA such that $M_1 \models A \sqsubseteq B$, then $M_1{}^{reach}_{A \leftrightarrow B} = M_1$.*

4 MinA extraction

By Theorem 2 every MinA is a bidirectional reachability-based module. In this section we show that for the DL $\mathcal{HL}$ every MinA is a minimal bidirectional

reachability-based module and that this property can be extended to acyclic $\mathcal{EL}$ TBoxes consisting of only primitive concept definitions. We also provide algorithms to compute and extract all MinAs for the given TBoxes.

Every MinA in $\mathcal{HL}$ is a *minimal* bidirectional reachability-based module in terms of itself. This is quite a subtle point, because MinAs are already minimal. Note, however, that MinAs are minimal with respect to the property of entailing a given statement of interest, whereas bidirectional reachability-based sub-modules are minimal with respect to the syntactic requirement for both bottom-up reachability and top-down reachability.

Theorem 3. *[5] Given a acyclic $\mathcal{HL}$ TBox $\mathcal{T}$ in normal form, the statement $A \sqsubseteq B$ and a MinA M_1 such that $M_1 \models A \sqsubseteq B$, then M_1 is a minimal bidirectional reachability-based module $M_1{}^{reach}_{A \leftrightarrow B}$ in terms of M_1.*

Next we show that every minimal bidirectional reachability-based module in $\mathcal{HL}$ for a statement $A \sqsubseteq B$ corresponds to a MinA.

Theorem 4. *[5] Given an acyclic $\mathcal{HL}$ TBox $\mathcal{T}$ in normal form, the statement $A \sqsubseteq B$ and a minimal bidirectional reachability-based module $M_1{}^{reach}_{A \leftrightarrow B}$, then $M_1 \models A \sqsubseteq B$.*

Theorems 3 and 4 allows us to conclude that there is a one-to-one correspondence between minimal bidirectional reachability-based modules and MinAs in $\mathcal{HL}$.

Corollary 1. *There is a one-to-one correspondence between MinAs and minimal bidirectional reachability-based modules in $\mathcal{HL}$.*

In order to extract all minimal bidirectional reachability-based modules we propose an algorithm originally inspired by the Earley [3] algorithm for parsing Context Free Grammars (CFG). Given a string to parse and a CFG the algorithm computes all possible parse trees in polynomial time. We employ a variation of the algorithm in order to compute a representation of all possible bidirectional reachability-based modules in $\mathcal{HL}$. A CFG consists of a set of CFG production rules formally defined as:

Definition 5. (CFG production rules) *Let X represent a single non-terminal, the symbol 'a' represents a single terminal and α and σ represent mixed strings of terminals and non-terminals, including the* null *string. CFG production rules have the form $X \rightarrow \alpha\sigma$ or $X \rightarrow a$.*

Any $\mathcal{HL}$ TBox can be transformed to an equivalent CFG by step by step transformation process [6, 5], with the reachability preserving CFG for an HL TBox is defined as:

Definition 6. ***Reachability preserving CFG for a $\mathcal{HL}$ TBox.***
Let $\mathcal{T}$ be an $\mathcal{HL}$ TBox in normal form and $A \sqsubseteq B$ a statement such that $\mathcal{T} \models A \sqsubseteq B$, then the reachability preserving CFG, denoted $\mathcal{CFG}_{\mathcal{T}}$, is a minimal set

of CFG production rules such that for each axiom $\alpha_L \sqsubseteq \alpha_R \in \mathcal{T}$: if $Sig(\alpha_L) = \emptyset$ the rule $x_i \rightarrow A \in \mathcal{CFG}_\mathcal{T}$ for each $x_i \in Sig(\alpha_R)$; for all other axioms the rule $x_i \rightarrow Sig(\alpha_L) \in \mathcal{CFG}_\mathcal{T}$; where the symbol A represents the only terminal symbol and the set $Sig(\mathcal{T})\backslash A$ represents the set of non-terminals.

The conversion process may be illustrated by the following example:

Example 3. Given the acyclic $\mathcal{HL}$ TBox $\mathcal{T}$ in normal form: $\mathcal{T} = \{A \sqsubseteq B_1, A \sqsubseteq B_2, B_1 \sqsubseteq C_1, B_1 \sqsubseteq D, B_2 \sqcap C_1 \sqsubseteq D, \top \sqsubseteq B_2\}$. Then $\text{CFG}_\mathcal{T}$ is given by: $\{ B_1 \rightarrow A, B_2 \rightarrow A, C_1 \rightarrow B_1, D \rightarrow B_1, D \rightarrow B_2 C_1\}$

Once the TBox has been converted to a CFG, we employ a parallel breadth-first algorithm, adapted from the Earley [3] algorithm, further optimized and improved from an algorithm earlier presented in [6]. The algorithm computes and indexes a representation of all bi-direction sub-modules in polynomial time.

Algorithm 2 (Sub-module computation) [5] *The algorithm consists of two sub-parts, the* predictor *and* completer*. For each state in CHART, the state $(X \rightarrow \alpha\beta)$ is evaluated and the appropriate sub-part executed:*
***Input**: Reachability preserving CFG for an $\mathcal{HL}$ TBox;*
***Output**: Reference table CHART capturing a representation of all $\mathcal{HL}$ sub-modules.*

1. ***Predictor:** Given the state $(X \rightarrow Y_1 \ldots Y_n)$, for all Y_i such that $(Y_i \rightarrow \sigma) \notin$ CHART, add all rules $(Y_i \rightarrow \sigma)$ to CHART.*
2. ***Completer:** If state $= (X \rightarrow Z_1 \ldots Z_m)$ with all Z_i terminals, then*
   - *add a pointer to this state in the completion table for X,and*
   - *if X is not a terminal symbol, then mark it a terminal symbol, and*
   - *if X is a new terminal symbol, then call the completer for each rule $(Y \rightarrow \ldots X \ldots) \in$CHART such that all symbols on the right hand side of the rule are terminal symbols.*

The algorithm executes all states iteratively in a top-down manner until no new states are available for processing. Given the statement $A \sqsubseteq B$, the production rule $S \rightarrow B$ is used to initialize CHART.

Theorem 5. *Given a acyclic $\mathcal{HL}$ TBox $\mathcal{T}$ in normal form and the statement of interest such that $\mathcal{T} \models A \sqsubseteq B$, with $CFG_\mathcal{T}$ the context free grammar associated with $\mathcal{T}$. If n is the number of production rules in $CFG_\mathcal{T}$, then Algorithm 2 computes a representation of all possible bidirectional reachability-based modules in $O(n^2)$ worst case running time.*

Once Algorithm 2 terminates, the chart returned contains a representation of all possible bidirectional reachability-based modules, and hence a representation of all MinAs. This set is essentially an indexed bidirectionally reachable module.

In order to obtain all individual MinAs from the CHART returned by Algorithm 2, we introduce an algorithm to extract all minimal bidirectional reachability-based modules from it. Due to the space limitations of this paper we do not give an implementation of the algorithm but refer the interested reader to [5].

Theorem 6. *Given the indexed bidirectional reachability-based $\mathcal{HL}$ CFG for the statement $A \sqsubseteq B$, $CFG_{\mathcal{O}}$ and the reference table CHART returned by Algorithm 2, the algorithm to extract all individual MinAs will extract all MinAs M_i such that $M_i \models A \sqsubseteq B$. Each M_i will be extracted in $O(m^2)$ worst case running time, where $m = |Sig(CFG_{\mathcal{O}})|$.*

The algorithm introduced may be extended to extract all MinAs for acyclic $\mathcal{EL}$ TBoxes consisting of only primitive concept definitions. However, we show that though all MinAs for these TBoxes are minimal bidirectional reachability-based modules, the converse does not hold.

Example 4. Let $\mathcal{T}$ be an acyclic $\mathcal{EL}$ TBox consisting of only primitive concept definitions. Further, let M_1 be a minimal bidirectional reachability-based module for the statement $A \sqsubseteq B$ consisting of the axioms $A \sqsubseteq \exists r.C$ and $C \sqsubseteq B$. Then ${M_1}^{reach}_{A \leftrightarrow B} = M_1$ and M_1 is minimal, but $M_1 \not\models A \sqsubseteq B$ unless $M_1 \models B \sqsubseteq \bot$. Hence M_1 is not a MinA for $\mathcal{T} \models A \sqsubseteq B$.

Theorem 7. *Let $\mathcal{T}$ be an $\mathcal{EL}$ TBox consisting of only primitive concept definitions in normal form, and let $A \sqsubseteq B$ be a statement such that $\mathcal{T} \models A \sqsubseteq B$. Then for every minimal bidirectional reachability-based module N_i such that $\alpha_L \sqsubseteq \exists r.C \in N_i$ we have that $N_i \not\models A \sqsubseteq B$. Further, for every minimal bidirectional reachability-based module M_i such that $\alpha_L \sqsubseteq \exists r.C \notin M_i$ we have that $M_i \models A \sqsubseteq B$.*

Consequently, the algorithms presented may be used in order to extract all minimal modules and thus MinAs for acyclic $\mathcal{EL}$ TBoxes consisting of only primitive concept definitions. When a minimal module includes axioms containing existential restrictions this module may simply be discarded as not being a MinA. The algorithm is complete in that it will extract all MinAs. However, since not all minimal modules extracted are MinAs, it is no longer sound. Soundness may however be obtained by simply making the test for the inclusion of existential restrictions part of the algorithm. When extending the problem of finding MinAs to general $\mathcal{EL}$ TBoxes, a staight forward extraction process is no longer possible and every possible matching between symbols needs to be calculated by the algorithm. Thus a simple iteration of all minimal bi-directional reachability based modules in order to find a single MinA results in an algorithm that runs in exponential worst case time.

Theorem 8. *Let $\mathcal{T}$ be an acyclic general $\mathcal{EL}$ TBox in normal form and $A \sqsubseteq B$ a statement of interest. Let CHART represent the resultant reference set returned by Algorithm 2. Let M_1 be a MinA such that $M_1 \models A \sqsubseteq B$ and P_1 represent the set of production rules for M_1. Now let m_i be the number of times a symbol C_i occurs on the right hand side of all production rules in P_1 and let k_i be the number of entries in CHART[C_i]. Then for the n possible symbols in P_1 there are a total of $\prod_{i=1}^{n} \sum_{j=1}^{j=m_i \leq k_i} C\binom{k_i}{j}$ bidirectional reachability-based modules.*

Though we believe that this theoretical worst case complexity will not pose a problem for real world $\mathcal{EL}$ medical ontologies, subsumption testing will be required once each minimal module have been extracted.

5 Empirical Results

In this section we test the algorithms presented in this paper and evaluate their performance in terms of three real world biomedical ontologies[2]: $\mathcal{O}_{Snomed}$ - The Systematized Nomenclature of Medicine, Clinical Terms; $\mathcal{O}_{Nci}$ - The Thesaurus of the US National Cancer Institute and $\mathcal{O}_{Go}$ - The Gene Ontology.

The algorithms presented were all implemented in Java as part of a plugin for the Protégé 4.1 (beta) ontology editor. All single threaded algorithms were tested on a Intel Quad Core based computer, with 6 Gig of RAM, running on Microsoft Windows 7 x64 and hosted in a 64 bit Java virtual machine. We did not implement nor utilise an optimized subsumption testing algorithm for inexpressive DLs. Subsumption testing were done by the standard HerMit[3] reasoner where neccesary.

Table 1 show the results of all bidirectional reachability-based modules extracted. The columns in the table are organised as follows: Ontology – the ontology for which the modules are being extracted; $|\ \mathcal{O}_A^{reach}\ |$ – the number of axioms in the reachability-based modules for all concepts $A \in$Sig($\mathcal{O}$); T($\mathcal{O}_A^{reach}$) – the average time, in seconds, required by the algorithm to extract all reachability-based modules; $|\ \mathcal{O}_{A\leftrightarrow B}^{reach}\ |$ – the average number of axioms for all bidirectional reachability-based modules; T($\mathcal{O}_{A\leftrightarrow B}^{reach}$) – the additional time, in seconds, required to extract the bidirectional reachability-based modules, i.e. Total time = T($\mathcal{O}_A^{reach}$) + T($\mathcal{O}_{A\leftrightarrow B}^{reach}$).

Average Values

Ontology	$\mathcal{O}_A^{reach}$	T($\mathcal{O}_A^{reach}$)	$\mathcal{O}_{A\leftrightarrow B}^{reach}$	T($\mathcal{O}_{A\leftrightarrow B}^{reach}$)
$\mathcal{O}_{Go}$	13.16	0.000032	4.48	0.000006
$\mathcal{O}_{Nci}$	25.68	0.000048	5.59	0.000006
$\mathcal{O}_{Snomed}$	27.70	0.040725	18.40	0.000175

Maximum Values

Ontology	$\mathcal{O}_A^{reach}$	T($\mathcal{O}_A^{reach}$)	$\mathcal{O}_{A\leftrightarrow B}^{reach}$	T($\mathcal{O}_{A\leftrightarrow B}^{reach}$)
$\mathcal{O}_{Go}$	68	0.000417	20.15	0.000666
$\mathcal{O}_{Nci}$	398	0.001916	55.00	0.000569
$\mathcal{O}_{Snomed}$	254	0.217781	222.06	0.004843

Median Values

Ontology	$\mathcal{O}_A^{reach}$	T($\mathcal{O}_A^{reach}$)	$\mathcal{O}_{A\leftrightarrow B}^{reach}$	T($\mathcal{O}_{A\leftrightarrow B}^{reach}$)
$\mathcal{O}_{Go}$	10	0.000026	3.86	0.000005
$\mathcal{O}_{Nci}$	11	0.000026	4.37	0.000005
$\mathcal{O}_{Snomed}$	16	0.001800	6.66	0.000008

Table 1. Bidirectional reachability-based module extraction

From the table we see that bidirectional reachability-based modules are between 30% and 80% smaller than standard reachability-based modules and may

[2] http://lat.inf.tu-dresden.de/systems/cel/
[3] http://hermit-reasoner.com/

be extracted at the additional cost of between 0.4% and 19.0% in the running time of the algorithm. The average runtime increases for the GO and NCI ontologies tested may seem excessively high. However, we note that the running times are measured in the low microsecond range. At these extremely small intervals the accuracy of our measuring tools is very low and the true runtime performance of the algorithms only becomes evident in relatively large ontologies. Therefore, the runtime performance of the algorithms for the SNOMED ontology gives a more accurate measure of the true performance of the algorithms. In terms of median values extracting bidirectional reachability-based modules results in very stable performance across all ontologies tested, with an approximate 59% decrease in the size of all modules extracted.

The MinA extraction algorithms were tested as follows: for every concept name $A \in$ in Sig($\mathcal{O}$) we extracted $\mathcal{O}_A^{reach}$, then Algorithm 2 was called for each concept name $B \in$Sig($\mathcal{O}_A^{reach}$) in order to extract $\mathcal{O}_{A\leftrightarrow B}^{reach}$. For each of these indexed bidirectional reachability-based modules we then extracted all possible minimal bidirectional reachability-based modules M_i. The standard HerMit reasoner was then called to test if $M_i \models A \sqsubseteq B$. This subsumption test is irrelevant and is only included for the sake of interest.

The columns in Table 2 are organised as follows: Ontology – the ontology for which the MinAs are being extracted; $\mid \mathcal{O}_{A\leftrightarrow B}^{reach} \mid$ – the average number of axioms for all bidirectional reachability-based modules; $\mid Min(\mathcal{O}_{A\leftrightarrow B}^{reach}) \mid$ – the average number of minimal bidirectional reachability-based modules; T($Min(\mathcal{O}_{A\leftrightarrow B}^{reach})$) – the additional time, in seconds, required to extract all minimal bidirectional reachability-based modules; %MinAs – the percentage of minimal bidirectional reachability-based modules that are MinAs; |MinA| – the average size of each MinA and T(MinA) – the additional time required to test subsumption for all minimal modules, i.e. to calculate the total time to extract all MinAs from the ontology = T($Min(\mathcal{O}_{A\leftrightarrow B}^{reach})$) + T(MinA).

Average Values

Ontology	$\mathcal{O}_{A\leftrightarrow B}^{reach}$	$\mid Min(\mathcal{O}_{A\leftrightarrow B}^{reach}) \mid$	T($Min(\mathcal{O}_{A\leftrightarrow B}^{reach})$)	%MinAs	|MinA|	T(MinA)
$\mathcal{O}_{Go}$	13	2.720188	0.000023	89.18%	3.298866	0.005472
$\mathcal{O}_{Nci}$	26	2.180851	0.000014	91.47%	3.721915	0.002842

Median Values

Ontology	$\mathcal{O}_{A\leftrightarrow B}^{reach}$	$\mid Min(\mathcal{O}_{A\leftrightarrow B}^{reach}) \mid$	T($Min(\mathcal{O}_{A\leftrightarrow B}^{reach})$)	%MinAs	|MinA|	T(MinA)
$\mathcal{O}_{Go}$	10	1.500000	0.000013	100.00%	3.000000	0.002327
$\mathcal{O}_{Nci}$	11	1.000000	0.000010	100.00%	3.500000	0.001452

Table 2. MinA extraction

On average there are between 2 and 3 minimal bidirectional reachability-based modules for each possible subsumption statement. From these, about 90% are MinAs, each of which contains between 3 and 4 axioms on average. The total additional time required to extract all minimal bidirectional reachability-based

modules, and thus MinAs, is in the low microsecond range. The most expensive costs incurred was subsumption testing, with a total running time for testing all minimal modules in the low to mid millisecond range. This testing however is unnecessary and is only included to illustrate the costs involved in testing all minimal modules for subsumption.

Once a bottom-up reachability-based module has been extracted, the additional runtime costs incurred to extract a bidirectional reachability module together with all minimal bidirectional reachability-based modules, and thus MinAs, is less that 1% of the cost of performing a subsumption test on a single MinA. This makes MinA extraction, for acyclic $\mathcal{EL}$ TBoxes consisting of only primitive concept definitions, negligible. The average reduction of 59% in the number of axioms for bidirectional reachability-based modules tested here, over that of standard reachability-based modules, indicates that for more expressive DLs in the $\mathcal{EL}$ family, bidirectional reachability-based modules may yield a significant improvement during MinA extraction to standard black-box algorithms.

Extension of the techniques presented here to more expressive DLs using hypergraph grammars, and relating it to the techniques and complexity results presented in [7], are topics of further research. We thank the anonymous reviewers for their comments on related and further work.

References

1. Ausiello, G., Franciosa, P.G., Frigioni, D.: Directed hypergraphs: Problems, algorithmic results, and a novel decremental approach. In: Proceedings of the Seventh Italian Conference on Theoretical Computer Science (ICTCS), LNCS, vol. 2202, pp. 312–327. Springer, London, UK (2001)
2. Du, J., Qi, G., Ji, Q.: Goal-directed module extraction for explaining OWL DL entailments. In: Bernstein, A., Karger, D.R., Heath, T., Feigenbaum, L., Maynard, D., Motta, E., Thirunarayan, K. (eds.) Proceedings ISWC'09, LNCS, vol. 5823, pp. 163–179. Springer, Berlin Heidelberg (2009)
3. Earley, J.: An efficient context-free parsing algorithm. Communications of the Association for Computing Machinery 13(2), 94–102 (1970)
4. Grau, B.C., Horrocks, I., Kazakov, Y., Sattler, U.: Modular reuse of ontologies: Theory and practice. Journal of Artificial Intelligence Research 31, 273–318 (2008)
5. Nortjé, R.: Module extraction for inexpressive description logics. Master's thesis, University of South Africa (2011), submitted.
6. Nortjé, R., Britz, K., Meyer, T.: Finding $\mathcal{EL}^+$ justifications using the Earley parsing algorithm. In: Meyer, T., Taylor, K. (eds.) Australasian Ontology Workshop 2009 (AOW 2009). CRPIT, vol. 112, pp. 27–35. ACS, Melbourne, Australia (2009)
7. Peñaloza, R., Sertkaya, B.: On the complexity of axiom pinpointing in the EL family of description logics. In: Lin, F., Sattler, U., Truszczynski, M. (eds.) Proceedings KR-10. AAAI Press, Toronto, Canada (2010)
8. Sattler, U., Schneider, T., Zakharyaschev, M.: Which kind of module should I extract? In: Grau, B.C., Horrocks, I., Motik, B., Sattler, U. (eds.) 22nd International Workshop on Description Logics (DL2009). CEUR-WS, Oxford, UK (2009)
9. Suntisrivaraporn, B.: Polynomial-Time Reasoning Support for Design and Maintenance of Large-Scale Biomedical Ontologies. Ph.D. thesis, Technical University of Dresden (2009)