Representing and sharing knowledge using SNOMED
Proceedings of the 3rd international conference on Knowledge Representation in Medicine (KR-MED 2008)
R. Cornet, K.A. Spackman (Eds)

Debugging SNOMED CT Using Axiom Pinpointing in the Description Logic $\mathcal{EL}^+$

Franz Baader and Boontawee Suntisrivaraporn
Institute for Theoretical Computer Science, TU Dresden, Germany

Abstract

SNOMED CT *is a large-scale medical ontology, which is developed using a variant of the inexpressive Description Logic $\mathcal{EL}$. Description Logic reasoning can not only be used to compute subsumption relationships between* SNOMED *concepts, but also to pinpoint the reason why a certain subsumption relationship holds by computing the axioms responsible for this relationship. This helps developers and users of* SNOMED CT *to understand why a given subsumption relationship follows from the ontology, which can be seen as a first step toward removing unwanted subsumption relationships.*
In this paper, we describe a new method for axiom pinpointing in the Description Logic $\mathcal{EL}^+$, which is based on the computation of so-called reachability-based modules. Our experiments on SNOMED CT *show that the sets of axioms explaining subsumption are usually quite small, and that our method is fast enough to compute such sets on demand.*

Introduction

Description Logics (DLs) [1] are a family of logic-based knowledge representation formalisms, which can be used to develop ontologies in a formally well-founded way. This is true both for expressive DLs, which are the logical basis of the Web Ontology Language OWL [2], and for inexpressive DLs of the $\mathcal{EL}$ family [3], which are used in the design of large-scale medical ontologies such as SNOMED CT[1] and the National Cancer Institute's ontology.[2]

One of the main advantages of employing a logic-based ontology language is that reasoning services can be used to derive implicit knowledge from the one explicitly represented. DL systems can, for example, classify a given ontology, i.e., compute all the subsumption (subconcept–superconcept) relationships between the concepts defined in the ontology. The advantage of using an inexpressive DL of the $\mathcal{EL}$ family is that classification is tractable, i.e., $\mathcal{EL}$ reasoners such as CEL [4] can compute the subsumption hierarchy of a given ontology in polynomial time.

Similar to writing large programs, building large-scale ontologies is an error-prone endeavor. Classification can help to alert the developer or user of an ontology to the existence of errors. For example, the subsumption relationship between "amputation of finger" and "amputation of upper limb" in SNOMED CT is clearly unintended [6, 7], and thus reveals a modeling error. However, given an unintended subsumption relationship in a large ontology like SNOMED CT with almost four hundred thousand axioms, it is not always easy to find the erroneous axioms responsible for it by hand. To overcome this problem, the DL community has recently invested quite some work on automating this process. Given a subsumption relationship or another questionable consequence, axiom pinpointing computes a minimal subset (all minimal subsets) of the ontology that have this consequence (called MinAs in the following). Most of the work on axiom pinpointing in DLs was concerned with rather expressive DLs (see, e.g., [8, 9, 10]). The only work that concentrated on pinpointing in the $\mathcal{EL}$ family of DLs was [11]. In addition to providing complexity results for pinpointing, [11] introduces a "pragmatic" algorithm for computing one MinA, which is based on a modified version of the classification algorithm used by the CEL reasoner [4]. Though this approach worked quite well for mid-size ontologies (see the experiments on a variant of the GALEN medical ontology described in [11]), it was not efficient enough to deal with large-scale ontologies like SNOMED CT.

[1] http://www.ihtsdo.org/our-standards/
[2] http://www.nci.nih.gov/cancerinfo/terminologyresources

In the present paper, we describe a new method for axiom pinpointing in the Description Logic $\mathcal{EL}^+$, which is based on the computation of so-called reachability-based modules [5]. Our experiments on SNOMED CT show that the sets of axioms explaining a given subsumption are usually quite small (78% of the MinAs we computed were of size ten or less), and that our method is fast enough (on average, it took one second to obtain a MinA) to compute these sets on demand, i.e., whenever the user asks for a MinA for a suspect subsumption relationship.

Axiom pinpointing in $\mathcal{EL}^+$

In this section, we first introduce the DL $\mathcal{EL}^+$, which is an extension of the DL $\mathcal{EL}$ used to define SNOMED CT. Then, we define minimal axiom sets (MinAs) for subsumption, and recall some of the known results about computing MinAs in $\mathcal{EL}^+$.

Syntax	Semantics
$\top$	$\Delta^{\mathcal{I}}$
$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
$\exists r.C$	$\{x \in \Delta^{\mathcal{I}} \mid \exists y \in \Delta^{\mathcal{I}} : (x,y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$
$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
$r_1 \circ \cdots \circ r_n \sqsubseteq s$	$r_1^{\mathcal{I}} \circ \cdots \circ r_n^{\mathcal{I}} \subseteq s^{\mathcal{I}}$

Table 1: Syntax and semantics of $\mathcal{EL}^+$.

Starting with a set of concept names CN and a set of role names RN, $\mathcal{EL}^+$ *concept descriptions* can be built using the constructors shown in the upper part of Table 1, i.e., every concept name $A \in \mathsf{CN}$ and the top concept $\top$ are $\mathcal{EL}^+$ concept descriptions, and if C, D are $\mathcal{EL}^+$ concept descriptions and $r \in \mathsf{RN}$ is a role name, then $C \sqcap D$ (conjunction) and $\exists r.C$ (existential restriction) are $\mathcal{EL}^+$ concept descriptions. Role chains of the form $r_1 \circ \cdots \circ r_n$ for $n \geq 0$ are called *role descriptions*. An $\mathcal{EL}^+$ *ontology* is a finite set of axioms of the form shown in the lower part of Table 1, where axioms of the form $C \sqsubseteq D$ are called general concept inclusions (GCIs) and of the form $r_1 \circ \cdots \circ r_n \sqsubseteq s$ role inclusions (RIs). An $\mathcal{EL}$ *ontology* is an $\mathcal{EL}^+$ ontology that does not contain RIs. We use $C \equiv D$ as an abbreviation for the two GCIs $C \sqsubseteq D, D \sqsubseteq C$.

The semantics of $\mathcal{EL}^+$ is defined in terms of *interpretations* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where the domain $\Delta^{\mathcal{I}}$ is a non-empty set of individuals, and the interpretation function $\cdot^{\mathcal{I}}$ maps each concept name $A \in \mathsf{CN}$ to a subset $A^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$ and each role name $r \in \mathsf{RN}$ to a binary relation $r^{\mathcal{I}}$ on $\Delta^{\mathcal{I}}$. The extension of $\cdot^{\mathcal{I}}$ to arbitrary concept descriptions is inductively defined, as shown in the semantics column of Table 1. An interpretation $\mathcal{I}$ is a *model* of an ontology $\mathcal{O}$ if, for each inclusion axiom in $\mathcal{O}$, the conditions given in the semantics column of Table 1 are satisfied.

The main reasoning problem in $\mathcal{EL}^+$ is the *subsumption problem*: given an $\mathcal{EL}^+$ ontology $\mathcal{O}$ and two $\mathcal{EL}^+$ concept descriptions C, D, check whether C is *subsumed* by D w.r.t. $\mathcal{O}$ (written $C \sqsubseteq_{\mathcal{O}} D$), i.e., whether $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ holds in all models of $\mathcal{O}$. The computation of all subsumption relationships between the concept names occurring in $\mathcal{O}$ is called *classification* of $\mathcal{O}$.

α_1	$\mathsf{AmpOfFinger}$	$\equiv$	$\mathsf{Amp} \sqcap \exists \mathsf{site}.\mathsf{Finger}_S$
α_2	$\mathsf{AmpOfHand}$	$\equiv$	$\mathsf{Amp} \sqcap \exists \mathsf{site}.\mathsf{Hand}_S$
α_3	$\mathsf{InjToFinger}$	$\equiv$	$\mathsf{Inj} \sqcap \exists \mathsf{site}.\mathsf{Finger}_S$
α_4	$\mathsf{InjToHand}$	$\equiv$	$\mathsf{Inj} \sqcap \exists \mathsf{site}.\mathsf{Hand}_S$
α_5	Finger_E	$\sqsubseteq$	Finger_S
α_6	Finger_P	$\sqsubseteq$	$\mathsf{Finger}_S \sqcap \exists \mathsf{part}.\mathsf{Finger}_E$
α_7	Hand_E	$\sqsubseteq$	Hand_S
α_8	Hand_P	$\sqsubseteq$	$\mathsf{Hand}_S \sqcap \exists \mathsf{part}.\mathsf{Hand}_E$
α_9	ULimb_E	$\sqsubseteq$	ULimb_S
α_{10}	ULimb_P	$\sqsubseteq$	$\mathsf{ULimb}_S \sqcap \exists \mathsf{part}.\mathsf{ULimb}_E$
α_{11}	Finger_S	$\sqsubseteq$	Hand_P
α_{12}	Hand_S	$\sqsubseteq$	ULimb_P

Figure 1: Ontology $\mathcal{O}_{\mathsf{Amp}}$ illustrating a faulty SEP-triplet encoding in SNOMED CT.

Figure 1 shows a small $\mathcal{EL}$ ontology defining concepts related to amputation/injury of hand and finger. It uses the so-called SEP-triplet encoding [12], in which anatomical concepts (like hand) are represented by three concepts: the structure concept (e.g, Hand_S, which stands for the hand and its proper parts), the part concept (e.g, Hand_P, which stands for the proper parts of the hand), and the entity concept (e.g, Hand_E, which stands for the entire hand). The axioms α_5–α_{10} constitute a completed SEP-triplet encoding for finger, hand, and upper limb. For example, axiom α_8 says that proper parts of the hand belong to the structure concept Hand_S, and they are parts of hand (i.e., linked with the role part to the entity concept Hand_E). Given this encoding, the fact that the finger is part of the hand can be expressed using axiom α_{11}. The main reason for using this encoding in SNOMED CT is that it can simulate transitivity reasoning for the role part, although transitivity of part cannot be expressed in $\mathcal{EL}$. For example, it is easy to see that the ontology $\mathcal{O}_{\mathsf{Amp}}$

implies that the finger is part of the upper limb, i.e., $\mathsf{Finger}_E \sqsubseteq_{\mathcal{O}_{\mathsf{Amp}}} \exists \mathsf{part}.\mathsf{ULimb}_E$. As a side-effect, the SEP-triplet encoding can also be used to simulate so-called right-identity rules [13], which allow to inherit properties along the part role. Consider the following subsumption relationships that hold in our example ontology:

$$\mathsf{AmpOfFinger} \sqsubseteq_{\mathcal{O}_{\mathsf{Amp}}} \mathsf{AmpOfHand}, \quad (1)$$

$$\mathsf{InjToFinger} \sqsubseteq_{\mathcal{O}_{\mathsf{Amp}}} \mathsf{InjToHand}. \quad (2)$$

While subsumption (2) actually makes sense (it is sensible to say that an injury to the finger is an injury to the hand), subsumption (1) is clearly undesirable. Subsumption (1) is an example of a false positive subsumption, which does indeed occur in SNOMED CT. It has been argued [6, 7] that this subsumption is due to a faulty SEP-triplet encoding. In fact, using the entity concepts instead of the structure concepts in the axioms α_1 and α_2 would have avoided this problem.

In $\mathcal{EL}^+$, one could actually dispense with the SEP-triplet encoding altogether since both transitivity and right-identity rules can be expressed using RIs. For example, $\mathsf{part} \circ \mathsf{part} \sqsubseteq \mathsf{part}$ expresses transitivity of the role part. An alternative and direct representation of anatomical concepts, as well as referring concepts like clinical findings and procedures, based on this additional expressive power of the DL $\mathcal{EL}^+$ is proposed in [6]. The new modeling is succinct and also avoids the above false positive subsumption (1).

For a small ontology like $\mathcal{O}_{\mathsf{Amp}}$, it is not hard to do the subsumption reasoning manually, and thus also to find the axioms responsible for a given subsumption relationship by hand. For a very large ontology like SNOMED CT, this manual approach to pinpointing the responsible axioms is very time-consuming, and thus should be automated. First, we give a formal definition of what automated pinpointing is actually supposed to compute.

Definition 1 (MinA). Let $\mathcal{O}$ be an $\mathcal{EL}^+$ ontology, and A, B concept names such that $A \sqsubseteq_{\mathcal{O}} B$. The set $\mathcal{S} \subseteq \mathcal{O}$ is a *minimal axiom set (MinA) for* $A \sqsubseteq_{\mathcal{O}} B$ if, and only if, $A \sqsubseteq_{\mathcal{S}} B$ and, for every $\mathcal{S}' \subset \mathcal{S}$, $A \not\sqsubseteq_{\mathcal{S}'} B$. $\diamond$

In our example, $\{\alpha_1, \alpha_2, \alpha_8, \alpha_{11}\}$ is the only MinA for subsumption (1), whereas $\{\alpha_3, \alpha_4, \alpha_8, \alpha_{11}\}$ is the only MinA for subsumption (2). As shown in [11], a given subsumption relationship w.r.t. an $\mathcal{EL}^+$ ontology may have exponentially many MinAs, and even deciding whether there is a MinA of cardinality $\leq k$ is an NP-complete problem. In contrast, one MinA can always be extracted in polynomial time. In [11], this was shown using the simple Algorithm 1, which requires linearly many (polynomial) subsumption tests. For a large ontology, however, this naive approach is not feasible. For example, for SNOMED CT it would require almost half a million subsumption tests for each MinA extraction.

Algorithm 1 Naive linear extraction of a MinA.

function lin-extract-mina$(A, B, \mathcal{O})$
$\mathcal{S} := \mathcal{O}$
for each axiom $\alpha \in \mathcal{O}$ **do**
  if $A \sqsubseteq_{\mathcal{S}\setminus\{\alpha\}} B$ **then**
    $\mathcal{S} := \mathcal{S} \setminus \{\alpha\}$
return $\mathcal{S}$

Algorithm 2 Logarithmic extraction of a MinA.

function log-extract-mina$(A, B, \mathcal{O})$
return log-extract-mina-r$(A, B, \emptyset, \mathcal{O})$
function log-extract-mina-r$(A, B, S, \mathcal{O})$
if $|\mathcal{O}| = 1$ **then**
  return $\mathcal{O}$
$S_1, S_2 :=$ halve$(\mathcal{O})$
if $A \sqsubseteq_{S \cup S_1} B$ **then**
  return log-extract-mina-r(A, B, S, S_1)
if $A \sqsubseteq_{S \cup S_2} B$ **then**
  return log-extract-mina-r(A, B, S, S_2)
$S_1' :=$ log-extract-mina-r$(A, B, S \cup S_2, S_1)$
$S_2' :=$ log-extract-mina-r$(A, B, S \cup S_1', S_2)$
return $S_1' \cup S_1'$

We can do much better by adopting the algorithm for computing prime implicates described in [14] to our problem. Basically, this algorithm applies binary search to find a MinA. Instead of taking out one axiom at a time, it partitions the ontology into two halves, and checks whether one of them entails the subsumption. If yes, it immediately recurses on that half, throwing away half of the axioms in one step. Otherwise, essential axioms are in both halves. In this case, the algorithm recurses on each half, using the other half as the "support set". Algorithm 2 describes this approach in more detail, where the function halve$(\mathcal{O})$ partitions $\mathcal{O}$ into $S_1 \cup S_2$ with $||S_1| - |S_2|| \leq 1$. It follows from the results in [14] that computing a MinA $\mathcal{S}$ for a given subsumption $A \sqsubseteq_{\mathcal{O}} B$ with Algorithm 2 requires $O\left((|\mathcal{S}| - 1) + |\mathcal{S}| log(|\mathcal{O}|/|\mathcal{S}|)\right)$ subsumption tests. This greatly improves on the naive algorithm. For instance, computing a MinA consisting of nine axioms for SNOMED CT requires about one hundred subsumption tests. Though this is much better than the almost half a million required by the naive algorithm, it is still not good enough to compute MinAs on demand (see below).

Modularization-based axiom pinpointing in $\mathcal{EL}^+$

Instead of applying Algorithm 1 or 2 directly to the whole ontology $\mathcal{O}$, one can first try to find a non-minimal (but hopefully small) subset $\mathcal{S} \subseteq \mathcal{O}$ with $A \sqsubseteq_{\mathcal{S}} B$ (called *nMinA* in the following), and then apply Algorithm 1 or 2 to this subset to obtain a MinA. In [11], we have sketched a modified version of the classification algorithm for $\mathcal{EL}^+$ [3, 4] that extracts such nMinAs. In the experiments on a version of GALEN described in [11], Algorithm 1 was then used to minimize these sets. Whereas the nMinA extraction was fast and produced quite small sets for GALEN, it crashed after a few hours because of space problems when applied to SNOMED CT.

To overcome this problem, we propose an algorithm for extracting nMinAs that is based on modularization. In the following, we introduce only those notions regarding modularization that are strictly necessary in the context of this paper. More details regarding the reachability-based modularization approach from which these notions are derived, as well as its connection to other work on modularization, can be found in [5].

Let $\mathcal{O}$ be an $\mathcal{EL}^+$ ontology, and A a concept name occurring in $\mathcal{O}$. We say that $\mathcal{S} \subseteq \mathcal{O}$ is a *subsumption module for A in $\mathcal{O}$* whenever $A \sqsubseteq_{\mathcal{O}} B$ if, and only if, $A \sqsubseteq_{\mathcal{S}} B$ holds for all concept names B occurring in $\mathcal{O}$. Obviously, if $\mathcal{S}$ is a subsumption module for A in $\mathcal{O}$ and $A \sqsubseteq_{\mathcal{O}} B$, then $\mathcal{S}$ is an nMinA for this subsumption, and Algorithm 1 or 2 can be used to compute a MinA $\mathcal{S}' \subseteq \mathcal{S}$ from $\mathcal{S}$. Thus, we know that a subsumption module for A contains a MinA for every valid subsumption relationship $A \sqsubseteq_{\mathcal{O}} B$. The reachability-based modules introduced below satisfy an even stronger property: they contain *all* MinAs for all valid subsumptions.

Definition 2. Let $\mathcal{O}$ be an $\mathcal{EL}^+$ ontology and A a concept name occurring in $\mathcal{O}$. The subsumption module $\mathcal{S}$ for A in $\mathcal{O}$ is called *strong* if the following holds for all concept names B occurring in $\mathcal{O}$: if $A \sqsubseteq_{\mathcal{O}} B$, then every MinA for $A \sqsubseteq_{\mathcal{O}} B$ is a subset of $\mathcal{S}$. ◇

Obviously, $\mathcal{O}$ itself is a strong subsumption module for every concept name A occurring in $\mathcal{O}$. The following definition (first introduced in [5]) yields strong subsumption modules that are usually much smaller than the whole ontology. For an $\mathcal{EL}^+$ entity X—i.e., either a (concept or role) description, a (concept or role) inclusion axiom, or an ontology—we write $\mathsf{Sig}(X)$ to denote the set of concept and role names occurring in the entity X.

Definition 3 (Reachability-based modules). Let $\mathcal{O}$ be an $\mathcal{EL}^+$ ontology and A a concept name occurring in $\mathcal{O}$. The *set of A-reachable names in $\mathcal{O}$* is the smallest set N of concept and role names such that

- A belongs to N;
- for all (concept/role) inclusion axioms $\alpha_L \sqsubseteq \alpha_R$ in $\mathcal{O}$, if $\mathsf{Sig}(\alpha_L) \subseteq \mathsf{N}$ then $\mathsf{Sig}(\alpha_R) \subseteq \mathsf{N}$.

We call an axiom $\alpha_L \sqsubseteq \alpha_R$ *A-reachable in $\mathcal{O}$* if every element of $\mathsf{Sig}(\alpha_L)$ is A-reachable in $\mathcal{O}$. The *reachability-based module for A in $\mathcal{O}$*, denoted by $\mathcal{O}_A^{\mathsf{reach}}$, consists of all A-reachable axioms from $\mathcal{O}$. ◇

In [5], it has been shown that $\mathcal{O}_A^{\mathsf{reach}}$ is indeed a subsumption module for A in $\mathcal{O}$. Here, we show the following stronger results.

Theorem 4. *Let $\mathcal{O}$ be an $\mathcal{EL}^+$ ontology and A a concept name. Then $\mathcal{O}_A^{\mathsf{reach}}$ is a* strong *subsumption module for A in $\mathcal{O}$.*

Proof. The fact that $\mathcal{O}_A^{\mathsf{reach}}$ is a subsumption module was already shown in [5]. To show that it is strong, assume that $A \sqsubseteq_{\mathcal{O}} B$ holds, but there is a MinA $\mathcal{S}$ for $A \sqsubseteq_{\mathcal{O}} B$ that is not contained in $\mathcal{O}_A^{\mathsf{reach}}$. Thus, there is an axiom $\alpha \in \mathcal{S} \setminus \mathcal{O}_A^{\mathsf{reach}}$. Let $\mathcal{S}_1$ be the subset of $\mathcal{S}$ that contains the A-reachable axioms. Note that $\mathcal{S}_1$ is a strict subset of $\mathcal{S}$ since $\alpha \notin \mathcal{S}_1$. We claim that $A \sqsubseteq_{\mathcal{S}} B$ implies $A \sqsubseteq_{\mathcal{S}_1} B$, which contradicts the assumption that $\mathcal{S}$ is a MinA for $A \sqsubseteq_{\mathcal{O}} B$.

To show the claim, we assume to the contrary that $A \not\sqsubseteq_{\mathcal{S}_1} B$, i.e., there is a model $\mathcal{I}_1$ of $\mathcal{S}_1$ such that $A^{\mathcal{I}_1} \not\subseteq B^{\mathcal{I}_1}$. We modify $\mathcal{I}_1$ to $\mathcal{I}$ by setting $y^{\mathcal{I}} := \emptyset$ for all (concept or role) names that are not A-reachable. It is easy to see that $A^{\mathcal{I}} \not\subseteq B^{\mathcal{I}}$. In fact, we have $A^{\mathcal{I}} = A^{\mathcal{I}_1}$ (since A is A-reachable), and $B^{\mathcal{I}} = B^{\mathcal{I}_1}$ or $B^{\mathcal{I}} = \emptyset$.

It remains to show that $\mathcal{I}$ is indeed a model of $\mathcal{S}$, i.e. satisfies all axioms $\beta_L \sqsubseteq \beta_R$ in $\mathcal{S}$. If β_L contains a name that is not A-reachable, then $(\beta_L)^{\mathcal{I}} = \emptyset$, and the axiom is trivially satisfied. Otherwise, this axiom belongs to $\mathcal{S}_1$, and the definition of A-reachability implies that all names in β_R are A-reachable as well. Consequently, $\mathcal{I}_1$ and $\mathcal{I}$ coincide on the names occurring in $\beta_L \sqsubseteq \beta_R$. Since $\mathcal{I}_1$ is a model of $\mathcal{S}_1$, we thus have $(\beta_L)^{\mathcal{I}} = (\beta_L)^{\mathcal{I}_1} \subseteq (\beta_R)^{\mathcal{I}_1} = (\beta_R)^{\mathcal{I}}$. ❑

As an immediate consequence of this theorem, instead of extracting a MinA for $A \sqsubseteq_{\mathcal{O}} B$ from $\mathcal{O}$, it is sufficient to extract a MinA for $A \sqsubseteq_{\mathcal{O}_A^{\mathsf{reach}}} B$ from $\mathcal{O}_A^{\mathsf{reach}}$. This is what the function extract-mina in

Algorithm 3 Modularization-based extraction of a MinA

function extract-mina$(A, B, \mathcal{O})$
1: $\mathcal{O}_A^{\mathsf{reach}} \leftarrow$ extract-module$(\mathcal{O}, A)$
2: **return** log-extract-mina$(A, B, \mathcal{O}_A^{\mathsf{reach}})$

function second-mina?$(A, B, \mathcal{O}_A^{\mathsf{reach}}, \mathcal{S}_1)$
1: **for** each axiom $\alpha \in \mathcal{S}_1$ **do**
2: $\quad\mathcal{O}' \leftarrow \mathcal{O}_A^{\mathsf{reach}} \backslash \{\alpha\}$
3: $\quad$**if** $A \sqsubseteq_{\mathcal{O}'} B$ **then**
4: $\quad\quad$**return** "second MinA exists"
5: **return** "MinA unique"

function extract-module$(\mathcal{O}, A)$
1: $\mathcal{O}_A \leftarrow \emptyset$
2: queue $\leftarrow$ active-axioms$(\{A\})$
3: **while not** empty(queue) **do**
4: $\quad(\alpha_L \sqsubseteq \alpha_R) \leftarrow$ fetch(queue)
5: $\quad$**if** $\mathsf{Sig}(\alpha_L) \subseteq \{A\} \cup \mathsf{Sig}(\mathcal{O}_A)$ **then**
6: $\quad\quad\mathcal{O}_A \leftarrow \mathcal{O}_A \cup \{\alpha_L \sqsubseteq \alpha_R\}$
7: $\quad\quad$queue $\leftarrow$ queue $\cup$ (active-axioms$(\mathsf{Sig}(\alpha_R)) \setminus \mathcal{O}_A$)
8: **return** $\mathcal{O}_A$

Algorithm 3 does. Note that, instead of the logarithmic extraction algorithm (Algorithm 2), we could also use the linear extraction algorithm (Algorithm 1). Since reachability-based modules are usually quite small, it is not a priori clear whether using the more complicated logarithmic algorithm really pays off (see the results of our experiments below). The function second-mina? in Algorithm 3 takes the extracted module and the first MinA as input, and checks if the subsumption in question still holds in the absence of one of the axioms in the MinA. In this case, this subsumption obviously must have more than one MinA. Note that, for this function to be correct, we really need to know that $\mathcal{O}_A^{\mathsf{reach}}$ is a *strong* subsumption module.

The function extract-module in Algorithm 3 realizes one way of computing reachability-based modules. The function call active-axioms used there yields, for a given set of names, all axioms that contain at least one of these names in their left-hand side. It is not hard to show that the call extract-module$(\mathcal{O}, A)$ indeed computes the reachability-based module for A in $\mathcal{O}$ (see [5] for more details). The experiments described in [5] show that extraction of reachability-based modules in SNOMED CT is usually quite fast, and the modules obtained this way are quite small. In the next section, we show that these positive results extend to the modularization-based extraction of MinAs.

Experimental Results

We have implemented the three algorithms described in this paper, using CEL [4] to compute subsumption. Our experiments use the January/2005 release of the DL version of SNOMED CT, which contains 379,691 concept names, 62 role names, and 379,704 axioms.[3] In the following, we call this ontology $\mathcal{O}^{\text{SNOMED}}$. The experiments were carried out on a PC with 2.40 GHz Pentium-4 processor and 1 GB of memory.

As stand-alone algorithms for computing a MinA, we applied Algorithm 1 and 2 only to the false positive subsumption $\mathsf{AmpOfFinger} \sqsubseteq_{\mathcal{O}^{\text{SNOMED}}} \mathsf{AmpOfHand}$. Algorithm 1 did not terminate on this input after 24 hours, whereas Algorithm 2 required 26:05 minutes (1,565 seconds) to compute a MinA of cardinality 6. (Note that the actual modelling of "amputation of finger" and "amputation of hand" in SNOMED CT differs from the one given in Fig. 1 due to the use of role groups and of two different roles to express location in SNOMED CT. Thus, the computed MinA also differs from the one given above. However, it also shows that the reason for the unintended subsumption is the incorrect use of the SEP-triplet encoding.)

Algorithm 3 performs much better for the amputation example. The reachability-based module $\mathcal{O}^{\text{SNOMED}}_{\mathsf{AmpOfFinger}}$ contains 57 axioms, and was computed in 0.04 seconds. Extracting a MinA for $\mathsf{AmpOfFinger} \sqsubseteq_{\mathcal{O}^{\text{SNOMED}}_{\mathsf{AmpOfFinger}}} \mathsf{AmpOfHand}$ from $\mathcal{O}^{\text{SNOMED}}_{\mathsf{AmpOfFinger}}$ using the logarithmic minimization algorithm then took only half a second. An application of second-mina? then showed that the extracted MinA is the only one for this subsumption.

We have also applied Algorithm 3 to a large number of subsumption relationships that follow from $\mathcal{O}^{\text{SNOMED}}$. Since there are more than five million such subsumptions, testing the algorithm on all of them was not feasible: assuming an average extraction time of 1 second, this would have required 58 days. For this reason, we sampled 0.5% of all concepts in each top-level category C in SNOMED CT. Let us denote the set of samples for category C by c-samples(C). For each sampled concept A, all positive subsumptions $A \sqsubseteq_{\mathcal{O}^{\text{SNOMED}}} B$ with A as subsumee were considered.

The first column of Table 2 shows the top-level categories and the second the number of sampled subsumption relationships with the subsumee in this category. The next four columns give the time needed to compute and the size of the corresponding modules and MinAs. The values in square brackets give the time required by the

[3] The *DL version* is also known in the SNOMED lingo as the 'stated form,' while *axioms* here boil down to (primitive) concept definitions.

SNOMED category C $A \in$ c-samples(C)	♯Subs. samples	Time to extract module $\mathcal{O}_A^{\text{SNOMED}}$ (avg/max)	Size of $\mathcal{O}_A^{\text{SNOMED}}$ (avg/max)	Time to extract MinA for $A \sqsubseteq_{\mathcal{O}_A^{\text{SNOMED}}} B$ (avg/max)	Size of MinA for $A \sqsubseteq_{\mathcal{O}_A^{\text{SNOMED}}} B$ (avg/max)	%Subs. with one MinA
Attribute	25	< 0.01 / < 0.01	5.12/8	0.05 / 0.09 [0.15 / 0.18]	3.16/7	100
Body structure	4 738	< 0.01 / 0.01	40.76 / 76	0.41 / 4.19 [0.63 / 2.24]	5.54 / 18	64.16
Clinical Finding	11 112	0.03 / 3.97	71.50 / 143	1.66 / 9.58 [1.15 / 5.04]	9.00 / 34	63.00
Context-dependent categories	208	0.01 / 0.03	0.14 / 108	0.37 / 1.43 [0.63 / 1.77]	4.10 / 13	95.67
Environments & geographical locations	51	< 0.01 / < 0.01	7.65 / 9	0.07 / 0.12 [0.17 / 0.19]	3.82 / 8	100
Events	28	< 0.01 / < 0.01	4.64 / 6	0.04 / 0.08 [0.13 / 0.16]	2.32 / 5	100
Observable entity	253	< 0.01 / < 0.01	8.26 / 12	0.08 / 0.18 [0.18 / 0.24]	3.68 / 8	90.12
Organism	1 429	< 0.01 / 0.01	13.03 / 21	0.09 / 0.20 [0.25 / 0.36]	4.72 / 13	65.01
Pharmaceutical/biologic product	1 233	< 0.01 / 0.01	31.41 / 60	0.16 / 0.47 [0.50 / 0.91]	3.68 / 10	81.51
Physical force	6	< 0.01 / < 0.01	7.00 / 7	0.38 / 0.58 [0.16 / 0.17]	2.83 / 5	50.00
Physical object	166	< 0.01 / < 0.01	9.35 / 12	0.09 / 0.19 [0.19 / 0.24]	4.18 / 11	93.98
Procedure	5 183	0.02 / 0.05	71.89 / 146	0.62 / 5.21 [0.15 / 4.38]	8.65 / 36	66.29
Qualifier value	216	< 0.01 / 0.01	6.68 / 11	0.06 / 0.13 [0.16 / 0.22]	2.67 / 7	87.96
Social context	204	< 0.01 / 0.01	10.11 / 15	0.18 / 0.47 [0.21 / 0.28]	3.59 / 9	77.45
Special concept	1 272	< 0.01 / 0.01	5.00 / 5	0.05 / 0.09 [0.14 / 0.14]	2.5 / 4	100
Specimen	38	0.01 / 0.02	67.74 / 127	0.19 / 0.59 [1.06 / 2.11]	4.55 / 13	81.58
Staging and scales	20	< 0.01 / < 0.01	5.60 / 8	0.04 / 0.08 [0.14 / 0.18]	2.8 / 7	100
Substance	1 295	< 0.01 / 0.01	14.76 / 32	0.16 / 0.41 [0.19 / 0.51]	4.17 / 12	72.82
Overall in SNOMED CT	27 477	0.02 / 3.97	53.21 / 146	1.03 / 9.58 [0.67 / 5.04]	7.11 / 36	68.26

Table 2: Empirical results of the modularization-based axiom pinpointing on SNOMED CT (time in seconds; size in number of axioms).

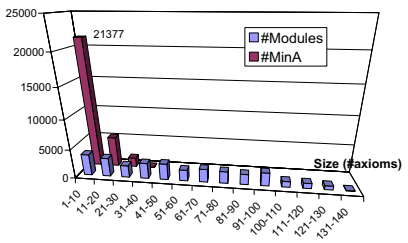

Figure 2: Module and MinA size distribution.

modularization-based pinpointing algorithm, but with the naive linear minimization algorithm instead of the logarithmic one. In all four columns, we give both average and maximum values. The last column shows the percentage of subsumptions that have only one MinA. Interestingly, more than two thirds of all subsumptions have only one MinA. The overall empirical results for the 27,477 sampled subsumptions (about 0.5% of all subsumptions) are given in the last row of the table. These results show that, on average, a MinA can be computed within one second, and its size is smaller than 10. Thus, MinAs can indeed be computed on demand, and their size is small enough such that they can then be inspected by hand.
Surprisingly, the linear minimization algorithm performed better in our experiments than the logarithmic one. An explanation for this is probably that, unlike the experiments of Algorithm 1 and 2 on the whole ontology, the modules are already quite small, and thus the overhead required by the logarithmic algorithm does not pay off. Figure 2 depicts the size distribution of our sampled modules and MinAs. As easily visible from the chart, the modules are quite small, but the MinAs are even smaller. In fact, the majority of all subsumptions (78%) have a MinA of size ten or less.

Conclusions

We have introduced a new method for axiom pinpointing in the DL $\mathcal{EL}^+$ that is based on the computation of reachability-based modules. The experiments carried out on SNOMED CT show that this method is fast enough to extract a minimal axiom set (MinA) for a given subsumption on demand. In addition, the extracted MinAs are usually quite small and can therefore be inspected by users and designers of SNOMED CT by hand. In the future, we will extend the approach such that it can (i) extract all MinAs, (ii) provide natural language explanations for subsumption, and (iii) give suggestions for how to revise the ontology to get rid of an unwanted subsumption.

Acknowledgements

The first author was partially supported by NICTA, Canberra Research Lab, and the second by the German Research Foundation (DFG) under grant BA 1122/11-1.

Address for Correspondence

Franz Baader and Boontawee Suntisrivaraporn
TU Dresden, Theoretical Computer Science,
01062 Dresden, Germany
{baader,meng}@tcs.inf.tu-dresden.de

References

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

[2] I. Horrocks, P. F. P.-Schneider, and F. van Harmelen. From SHIQ and RDF to OWL: The making of a web ontology language. *Journal of Web Semantics*, 1(1):7–26, 2003.

[3] F. Baader, S. Brandt, and C. Lutz. Pushing the $\mathcal{EL}$ envelope. In *Proc. IJCAI 2005*.

[4] F. Baader, C. Lutz, and B. Suntisrivaraporn. CEL—a polynomial-time reasoner for life science ontologies. In *Proc. IJCAR 2006*, Springer LNAI 4130, 2006.

[5] B. Suntisrivaraporn. Module extraction and incremental classification: A pragmatic approach for $\mathcal{EL}^+$ ontologies. In *Proc. ESWC 2008*, Springer LNCS, 2008. To appear.

[6] B. Suntisrivaraporn, F. Baader, S. Schulz, and K. Spackman. Replacing SEP-triplets in SNOMED CT using tractable description logic operators. In *Proc. AIME 2007*, Springer LNCS 4594, 2007.

[7] U. Hahn S. Schulz, K. Mark. Spatial location and its relevance for terminological inferences in bio-ontologies. *BMC Bioinformatics*, 2007.

[8] S. Schlobach and R. Cornet. Non-standard reasoning services for the debugging of description logic terminologies. In *Proc. IJCAI 2003*.

[9] B. Parsia, E. Sirin, and A. Kalyanpur. Debugging OWL ontologies. In *Proc. WWW 2005*.

[10] T. Meyer, K. Lee, R. Booth, and J. Z. Pan. Finding maximally satisfiable terminologies for the description logic $\mathcal{ALC}$. In *Proc. (AAAI 2006)*. AAAI Press/The MIT Press, 2006.

[11] F. Baader, R. Peñaloza, and B. Suntisrivaraporn. Pinpointing in the description logic $\mathcal{EL}^+$. In *Proc. KI 2007*, Springer LNAI 4667, 2007.

[12] S. Schulz, M. Romacker, and U. Hahn. Part-whole reasoning in medical ontologies revisited: Introducing SEP triplets into classification-based description logics. *JAMIA*, 1998.

[13] K.A. Spackman. Managing clinical terminology hierarchies using algorithmic calculation of subsumption: Experience with SNOMED-RT. *JAMIA*, 2000. Fall Symposium Special Issue.

[14] A. R. Bradley and Z. Manna. Checking safety by inductive generalization of counterexamples to induction. In *Proc. FMCAD 2007*.