TECHNISCHE UNIVERSITÄT DRESDEN

Dresden University of Technology
Institute for Theoretical Computer Science
Chair for Automata Theory

LTCS–Report

Module Extraction and Incremental Classi cation: A Pragmatic Approach for $\mathcal{EL}^+$ Ontologies

Boontawee Suntisrivaraporn

LTCS-Report 07-03

Lehrstuhl fur Automatentheorie
Institut fur Theoretische Informatik
TU Dresden
http://lat.inf.tu-dresden.de

Nothnitzer Str. 46
01187 Dresden
Germany

Module Extraction and Incremental Classi cation: A Pragmatic Approach for $\mathcal{EL}^+$ Ontologies

Boontawee Suntisrivaraporn

Theoretical Computer Science, TU Dresden, Germany
meng@tcs.inf.tu-dresden.de

14 December 2007

Abstract

The description logic $\mathcal{EL}^+$ has recently proved practically useful in the life science domain with presence of several large-scale biomedical ontologies such as Snomed ct. To deal with ontologies of this scale, standard reasoning of classi cation is essential but not su cient. The ability to extract relevant fragments from a large ontology and to incrementally classify it has become more crucial to support ontology design, maintenance and re-use. In this paper, we propose a pragmatic approach to module extraction and incremental classi cation for $\mathcal{EL}^+$ ontologies and report on empirical evaluations of our algorithms which have been implemented as an extension of the CEL reasoner.

Contents

1 Introduction

In the past few years, the $\mathcal{EL}$ family of description logics (DLs) has received an increasing interest and been intensively studied (see, e.g., [2, 3, 4, 9]). The attractiveness of the $\mathcal{EL}$ family is twofold: on the one hand, it is computationally tractable, i.e. subsumption is decidable in polytime; on the other hand, it is expressive enough to formulate many life science ontologies. Examples include the Gene Ontology, the thesaurus of the US National Cancer Institute (Nci), the Systematized Nomenclature of Medicine, Clinical Terms (Snomed ct), and large part (more than 95%) of the Galen Medical Knowledge Base (Galen). We lay emphasis on Snomed ct which comprises 0.5 million axioms and is now a standardized clinical terminology adopted by health care sectors in several countries [1].

Being a standard ontology, Snomed has been designed to comprehensively cover a whole range of concepts in the medical and clinical domains. For this reason, it is often the case that only a small part is actually needed in a speci c application. The ability to automate extraction of meaningful sub-ontologies that cover all relevant information is becoming important to support re-use of typically comprehensive standardized ontologies. Several techniques for syntactic module extraction have been proposed [10, 12, 7], since semantic extraction is highly complex [7]. Though (deductive) conservative extension could be used as a su cient condition for extracting a module, it is unfortunately too expensive (ExpTime-complete already in $\mathcal{EL}$ with GCIs [9]). In Section 3 of the present paper, we de ne a new kind of module, called *reachability-based modules*, which is motivated by a once-employed optimization technique in the CEL system. Also, we propose an algorithm for extracting modules of this kind and show some interesting properties.

Despite being classi able by modern DL reasoners, design and maintenance of large-scale ontologies like Snomed ct requires additional reasoning support. This is due to the fact that an ontology under development evolves continuously, and the developer often has to undergo the long process of *full classi cation* after addition of a few new axioms. Though classi cation of Snomed requires less than half an hour (see [3] or Table 1 in the present paper), the ontology developer is not likely willing to wait that long for a single change. In the worst case, she may end up not using automated reasoning support which could have helped identify potential modeling errors at an early stage. In Section 4, we propose a *goal-directed* variant of the $\mathcal{EL}^+$ classi cation algorithm developed in [4] which can be used for testing subsumption queries prior to full classi cation. Section 5 presents an extension of the algorithm in [4] to cater for two ontologies: the permanent ontology $\mathcal{O}_p$ which has been carefully modeled, and axioms of which are not supposed to be modi ed; and, the temporary ontology $\mathcal{O}_t$ that contains new axioms currently being authored. The extended algorithm reuses information from the previous classi cation of $\mathcal{O}_p$ and thus dispense with the need of the full classi cation of $\mathcal{O}_p \cup \mathcal{O}_t$. We call reasoning in this setting *restricted incremental classi cation.*

All algorithms proposed in this paper have been implemented in the CEL reasoner [3] and various experiments on realistic ontologies have been performed. The experiments and some promising results are discussed in Section 6.

2 Preliminaries

The present paper focuses on the sub-Boolean DL $\mathcal{EL}^+$ [4], which is the underlying logical formalism of the CEL reasoner [3]. Similar to other DLs, an $\mathcal{EL}^+$ signature is the disjoint union $\mathsf{S} = \mathsf{CN} \cup \mathsf{RN}$ of the sets of concept names and role names. $\mathcal{EL}^+$ *concept descriptions (or complex concepts)* can be de ned inductively as follows: each 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 concept conjunction $C \sqcap D$ and existential restriction $\exists r.C$ are $\mathcal{EL}^+$ concept descriptions. An $\mathcal{EL}^+$ *ontology* $\mathcal{O}$ is a nite set of *general concept inclusion (GCI)* axioms $C \sqsubseteq D$ and *role inclusion (RI)* axioms r_1 $r_n \sqsubseteq s$ with C, D $\mathcal{EL}^+$ concept descriptions and r_i, s role names. Concept equivalences and (primitive) concept de nitions are expressible using GCIs, whereas RIs can be used to express various role axioms, such as re exivity ($\sqsubseteq r$), transitivity (r $r \sqsubseteq r$), right-identity (r $s \sqsubseteq r$), and role hierarchy ($r \sqsubseteq s$) axioms. Figure 1 illustrates an example in the medical domain. For convenience, we write $\mathsf{Sig}(\mathcal{O})$ (resp., $\mathsf{Sig}(\ )$, $\mathsf{Sig}(C)$) to denote the signature of the ontology $\mathcal{O}$ (resp., the axiom , the concept C), i.e. concept and role names occurring in it.

The main inference problem for concepts is *subsumption query*: given an ontology $\mathcal{O}$ and two concept descriptions C, D, check if C is subsumed by (i.e. more speci c than) D w.r.t. $\mathcal{O}$, written $C \sqsubseteq_{\mathcal{O}} D$. From our example ontology, it is not di cult to draw that $\mathsf{Pericarditis} \sqsubseteq_{\mathcal{O}_{\mathsf{ex}}} \exists \mathsf{has\text{-}state.NeedsTreatment}$. The identi cation of subsumption relationships between *all* pairs of concept names occurring in $\mathcal{O}$ is known as *ontology classi cation.*

The semantics of $\mathcal{EL}^+$ ontologies, as well as of subsumption, is de ned by means of interpretations in the standard way, and we refer the reader to [4, 2].

1	Pericardium	$\sqsubseteq$	Tissue $\sqcap$ $\exists$contained-in.Heart
2	Endocardium	$\sqsubseteq$	Tissue $\sqcap$ $\exists$part-of.HeartValve
3	Pericarditis	$\sqsubseteq$	In ammation $\sqcap$ $\exists$has-location.Pericardium
4	Endocarditis	$\sqsubseteq$	In ammation $\sqcap$ $\exists$has-location.Endocardium
5	In ammation	$\sqsubseteq$	Disease $\sqcap$ $\exists$acts-on.Tissue
6	Disease $\sqcap$ $\exists$has-location.Heart	$\sqsubseteq$	HeartDisease
7	HeartDisease	$\sqsubseteq$	$\exists$has-state.NeedsTreatment
8	part-of part-of	$\sqsubseteq$	part-of
9	has-location contained-in	$\sqsubseteq$	has-location

Figure 1: An example $\mathcal{EL}^+$ ontology $\mathcal{O}_{\mathsf{ex}}$.

3 Modules Based on Connected Reachability

In this section, we introduce a new kind of module based on *connected reachability*, and propose an algorithm for extracting the modules of this kind. We also show that, in the DL $\mathcal{EL}^+$, our modules indeed correspond to modules based on syntactic locality rst introduced in [7]. We start by giving the general de nition of module:

De nition 1 (Modules for an axiom and a signature). Let $\mathcal{O}$ be an $\mathcal{EL}^+$ ontology, and $\mathcal{O}'$ a (possibly empty) set of axioms from $\mathcal{O}$. We say that $\mathcal{O}'$ is a *module for an axiom in* $\mathcal{O}$ (for short, *-module in* $\mathcal{O}$) if: $\mathcal{O}' \models$ i $\mathcal{O} \models$.

We say that $\mathcal{O}'$ is a *module for a signature* $\mathbf{S}$ if for every axiom with $\mathsf{Sig}(\ )$ $\mathbf{S}$, we have that $\mathcal{O}'$ is an -module in $\mathcal{O}$. $\Diamond$

Intuitively, a module of an ontology $\mathcal{O}$ is a subset $\mathcal{O}'$ $\mathcal{O}$ that preserves an axiom of interest or the axioms over a signature of interest. Observe that this is a very generic de nition, in the sense that the whole ontology is itself a module. In the following, we are interested in certain su cient conditions that not only extract a module according to De nition 1 but also guarantee relevancy of extracted axioms. Note that if $\mathcal{O} \models$, a justi cation (minimal axiom set that has the consequence) is a minimal -module in $\mathcal{O}$. A justi cation covers one axiom, not the axioms over a signature, thus it is normally expensive to obtain and involve standard inference reasoning, such as subsumption. For this reason, various syntactic approaches to extracting ontology fragments have been proposed in the literature [10, 12, 7]. In [7], Cuenca Grau et al. introduced a kind of module based on so-called syntactic locality. Here, we recap the notion of syntactic locality modulo the DL $\mathcal{EL}^+$.

De nition 2 (Locality-based modules). Let $\mathcal{O}$ be an $\mathcal{EL}^+$ ontology and $\mathbf{S}$ a signature. The following grammar recursively de nes the set of concepts $\mathbf{Con}^\perp(\mathbf{S})$:

$$\mathbf{Con}^\perp(\mathbf{S}) ::= A^\perp \mid (C^\perp \sqcap C) \mid (C \sqcap C^\perp) \mid (\exists r.C^\perp) \mid (\exists r^\perp.C)$$

with r is a role name, C a concept description, $A^\perp, r^\perp \notin \mathbf{S}$, and $C^\perp \in \mathbf{Con}^\perp(\mathbf{S})$.

An $\mathcal{EL}^+$ axiom is *syntactically local* w.r.t. $\mathbf{S}$ if it is one of the following forms: *(1)* $R^\perp \sqsubseteq s$ where s is a role name and $R^\perp$ is a role name $r^\perp \notin \mathbf{S}$ or a role composition r_1 r_n with $r_i \notin \mathbf{S}$ for some i n, or *(2)* $C^\perp \sqsubseteq C$ where $C^\perp \in \mathbf{Con}^\perp(\mathbf{S})$ and C is a concept description. We write $\mathsf{local}(\mathbf{S})$ to denote the collection of all $\mathcal{EL}^+$ axioms that are syntactically local w.r.t. $\mathbf{S}$.

If $\mathcal{O}$ can be partitioned into $\mathcal{O}'$ and $\mathcal{O}''$ s.t. every axiom in $\mathcal{O}''$ is syntactically local w.r.t. $\mathbf{S} \cup \mathsf{Sig}(\mathcal{O}')$, then $\mathcal{O}'$ is a *locality-based module* for $\mathbf{S}$ in $\mathcal{O}$.

$\Diamond$

Now we consider the optimization techniques of "reachability" that are used to heuristically determine obvious subsumption and non-subsumption relationships. The reachability heuristic for non-subsumptions can easily be exploited in module extraction for $\mathcal{EL}^+$ ontologies. To obtain a more satisfactory module size, however, we introduce a more appropriate (stronger) reachability notion and develop an algorithm for extracting modules based on this notion.

De nition 3 (Strong/weak reachability). Let $\mathcal{O}$ be an $\mathcal{EL}^+$ ontology, A, B concept names in $\mathcal{O}$. The strong (weak) reachability graph $\mathcal{G}_s(\mathcal{O})$ ($\mathcal{G}_w(\mathcal{O})$) for $\mathcal{O}$ is a tuple (V_s, E_s) ((V_w, E_w)) with $V_s = \mathsf{CN}(\mathcal{O})$ ($V_w = \mathsf{CN}(\mathcal{O})$) and E_s (E_w) the smallest set containing an edge (A, B) if $A \sqsubseteq D \in \mathcal{O}$ s.t. B is a conjunct in D (if $C \sqsubseteq D \in \mathcal{O}$ s.t. A occurs in C and B occurs in D).

We say that B is *strongly reachable* (*weakly reachable*) from A in $\mathcal{O}$ if there is a path from A to B in $\mathcal{G}_s(\mathcal{O})$ ($\mathcal{G}_w(\mathcal{O})$). $\Diamond$

Observe that B is strongly reachable from A in $\mathcal{O}$ implies $A \sqsubseteq_{\mathcal{O}} B$, while $A \sqsubseteq_{\mathcal{O}} B$ implies that B is weakly reachable from A in $\mathcal{O}$.

The weak reachability graph $\mathcal{G}_w(\mathcal{O})$ for $\mathcal{O}$ can be extended in a straightforward way to cover all the symbols in $\mathcal{O}$, i.e. also role names. Precisely, we de ne the extension as $\mathcal{G}'_w(\mathcal{O}) := (\mathsf{Sig}(\mathcal{O}), E'_w)$ with $(x, y) \in E'_w$ i there is an axiom $_L \sqsubseteq$ $_R \in \mathcal{O}$ s.t. $x \in \mathsf{Sig}(\ _L)$ and $y \in \mathsf{Sig}(\ _R)$. A module for $\mathbf{S} = \{A\}$ in an ontology $\mathcal{O}$ based on extended weak reachability can be extracted as follows: construct $\mathcal{G}'_w(\mathcal{O})$, extract all the paths from A in $\mathcal{G}_w(\mathcal{O})$, and nally, accumulate axioms responsible for the edges in those paths. However, this kind of module is relatively large, and many axioms are often irrelevant. For example, any GCIs with $\mathsf{Disease}$ appearing on the left-hand side, such as $\mathsf{Disease} \sqcap \exists\mathsf{has\text{-}location.Brain} \sqsubseteq \mathsf{BrainDisease}$, would be extracted as part of the module for $\mathbf{S} = \{\mathsf{Pericarditis}\}$. This axiom is irrelevant since $\mathsf{Pericarditis}$ does not refer to Brain and thus $\mathsf{BrainDisease}$. Such a module would end up comprising de nitions of all disease concepts. To rule out this kind of axioms, we make the notion of reachability graph stronger as follows: All symbols appearing on the left-hand side (e.g., $\mathsf{Disease}$, $\mathsf{has\text{-}location}$ and Brain) are viewed as a connected node in the graph, which has an edge to each symbol (e.g., $\mathsf{BrainDisease}$) on the right-hand side of the axiom. The connected node is reachable from x i all symbols participating in it are reachable from x. In our example, since both $\mathsf{has\text{-}location}$ and Brain are not reachable from $\mathsf{Pericarditis}$, neither is $\mathsf{BrainDisease}$. Therefore, the axiom is not extracted as part of the re ned module.

De nition 4 (Connected reachability and modules). Let $\mathcal{O}$ be an $\mathcal{EL}^+$ ontology, $\mathbf{S}$ $\quad \mathsf{Sig}(\mathcal{O})$ a signature, and $x, y \in \mathsf{Sig}(\mathcal{O})$ concept or role names. We say that x is *connectedly reachable* from $\mathbf{S}$ w.r.t. $\mathcal{O}$ (for short, *reachable* from $\mathbf{S}$ or $\mathbf{S}$-*reachable*) i $x \in \mathbf{S}$ or there is an axiom (either GCI or RI) $\ _L \sqsubseteq \ _R \in \mathcal{O}$ s.t. $x \in \mathsf{Sig}(\ _R)$ and, for all $y \in \mathsf{Sig}(\ _L)$, y is reachable from $\mathbf{S}$.

We say that an axiom $\ _L \sqsubseteq \ _R$ is *connected reachable* from $\mathbf{S}$ w.r.t. $\mathcal{O}$ (for short, $\mathbf{S}$-*reachable*) if, for all $x \in \mathsf{Sig}(\ _L)$, x is $\mathbf{S}$-reachable. *The reachability-based module* for $\mathbf{S}$ in $\mathcal{O}$, denoted by $\mathcal{O}^{\mathsf{reach}}_{\mathbf{S}}$, is the smallest set of all $\mathbf{S}$-reachable axioms, i.e. $\mathcal{O}^{\mathsf{reach}}_{\mathbf{S}} = \{ \ \in \mathcal{O} | \ $ is $\mathbf{S}$-reachable w.r.t. $\mathcal{O}\}$. $\Diamond$

Intuitively, x is reachable from y w.r.t. $\mathcal{O}$ means that y syntactically refers to x, either directly or indirectly via axioms in $\mathcal{O}$. If x, y are concept names, then the reachability suggests a potential subsumption relationship $y \sqsubseteq_{\mathcal{O}} x$. Note, in particular, that axioms of the forms $\top \sqsubseteq D$ and $\ \sqsubseteq r$ in $\mathcal{O}$ are reachable from any symbol in $\mathsf{Sig}(\mathcal{O})$ because $\mathsf{Sig}(\top) = \mathsf{Sig}(\ ) = \emptyset$, and therefore occur in every module. In our example, $\mathcal{O}^{\mathsf{reach}}_{\{\mathsf{Pericarditis}\}}$ contains axioms $\ _1$, $\ _3$, $\ _5$– $\ _7$ and $\ _9$. We now show some properties of connected reachability and reachability-based modules that are essential for establishing the subsequent lemmas:

Proposition 5 (Properties of reachability and $\mathcal{O}^{\mathsf{reach}}_{\mathbf{S}}$). *Let $\mathcal{O}$ be an $\mathcal{EL}^+$ ontology, $\mathbf{S}, \mathbf{S}_1, \mathbf{S}_2$ $\quad \mathsf{Sig}(\mathcal{O})$ signatures, x, y, z symbols in $\mathsf{Sig}(\mathcal{O})$, and A, B concept names in $\mathsf{CN}(\mathcal{O})$. Then, the following properties hold:*

1. *If $\mathbf{S}_1$ $\quad \mathbf{S}_2$, then $\mathcal{O}^{\mathsf{reach}}_{\mathbf{S}_1}$ $\quad \mathcal{O}^{\mathsf{reach}}_{\mathbf{S}_2}$.*

2. *If x is y-reachable and y is z-reachable, then x is z-reachable.*

3. *If x is reachable from y w.r.t. $\mathcal{O}$, then $\mathcal{O}^{\mathsf{reach}}_{\{x\}}$ $\quad \mathcal{O}^{\mathsf{reach}}_{\{y\}}$*

4. *$x \in \mathbf{S} \cup \mathsf{Sig}(\mathcal{O}^{\mathsf{reach}}_{\mathbf{S}})$ if, and only if, x is reachable from $\mathbf{S}$ w.r.t. $\mathcal{O}$.*

5. *If B is* not *connected reachable from A, then $A \not\sqsubseteq_{\mathcal{O}} B$.*

Proof.

To show Point 1, it is enough to show, for each axiom $= ({}_L \sqsubseteq {}_R) \in \mathcal{O}$, that $\in \mathcal{O}^{\mathsf{reach}}_{\mathbf{S}_1}$ implies $\in \mathcal{O}^{\mathsf{reach}}_{\mathbf{S}_2}$. By de nition, it follows from $\in \mathcal{O}^{\mathsf{reach}}_{\mathbf{S}_1}$ that x is $\mathbf{S}_1$-reachable for all $x \in \mathsf{Sig}({}_L)$. Since $\mathbf{S}_1 \quad \mathbf{S}_2$, x is also $\mathbf{S}_2$-reachable. Again, by de nition, we have $\in \mathcal{O}^{\mathsf{reach}}_{\mathbf{S}_2}$.

We can prove Point 2 by induction on the connected reachability of y to x. Induction Start: $y = x$. Then, x is z-reachable. Induction Step: there exists an axiom ${}_L \sqsubseteq {}_R \in \mathcal{O}$ such that $x \in \mathsf{Sig}({}_R)$ and, for all $x' \in \mathsf{Sig}({}_L)$, x' is y-reachable. By I.H., x' is z-reachable, implying by de nition that x is z-reachable.

Point 2 can now be used to prove Point 3. It su ces to show that $\in \mathcal{O}^{\mathsf{reach}}_{\{x\}}$ implies $\in \mathcal{O}^{\mathsf{reach}}_{\{y\}}$, for each $= ({}_L \sqsubseteq {}_R) \in \mathcal{O}$. By de nition, $\in \mathcal{O}^{\mathsf{reach}}_{\{x\}}$ implies that, for all $z \in \mathsf{Sig}({}_L)$, z is x-reachable. Since x is y-reachable, Point 3 implies that z is y-reachable. This means that is y-reachable, thus $\in \mathcal{O}^{\mathsf{reach}}_{\{y\}}$.

"Only if" direction of Point 4: Trivial if $x \in \mathbf{S}$. If $x \in \mathsf{Sig}(\mathcal{O}^{\mathsf{reach}}_{\mathbf{S}})$, then there is an $= ({}_L \sqsubseteq {}_R) \in \mathcal{O}^{\mathsf{reach}}_{\mathbf{S}}$ s.t. $x \in \mathsf{Sig}( )$. Since such an is $\mathbf{S}$-reachable, all $x' \in \mathsf{Sig}({}_L)$ must be $\mathbf{S}$-reachable. By de nition, every $x' \in \mathsf{Sig}({}_R)$ is also reachable. "If" direction: Assume that x is $\mathbf{S}$-reachable. By de nition, if x is $\mathbf{S}$-reachable, then $x \in \mathbf{S}$, or there is an $= ({}_L \sqsubseteq {}_R) \in \mathcal{O}$ s.t. $x \in \mathsf{Sig}({}_R)$ and, for all $y \in \mathsf{Sig}({}_L)$, y is reachable from $\mathbf{S}$. It is trivial that $x \in \mathbf{S} \cup \mathsf{Sig}(\mathcal{O}^{\mathsf{reach}}_{\mathbf{S}})$ in the rst case. In the latter case, we have that is $\mathbf{S}$-reachable, implying by de nition that $\in \mathcal{O}^{\mathsf{reach}}_{\mathbf{S}}$. Thus, $x \in \mathsf{Sig}( ) \quad \mathbf{S} \cup \mathsf{Sig}(\mathcal{O}^{\mathsf{reach}}_{\mathbf{S}})$.

To prove Point 5, we assume that B is not connectedly reachable from A. De ne an interpretation $\mathcal{I} = ({}^{\mathcal{I}}, {}^{\mathcal{I}})$ by setting ${}^{\mathcal{I}} := \{a\}$, $A'^{\mathcal{I}} := \{a\}$ for all A-reachable concept names A', $r^{\mathcal{I}} := \{(a, a)\}$ for all A-reachable role names r, and $x^{\mathcal{I}} := \emptyset$ for all concept and role names x unreachable from A. It is easy to see that $A^{\mathcal{I}} \;/\; B^{\mathcal{I}}$ with a the witness. It remains to show that $\mathcal{I}$ is a model of $\mathcal{O}$. With A-reachability, the ontology $\mathcal{O}$ can be partitioned into $\mathcal{O}' \cup \mathcal{O}''$ with $\mathcal{O}' := \{ \in \mathcal{O} \mid$ is A-reachable$\}$, and $\mathcal{O}'' := \mathcal{O} \backslash \mathcal{O}'$. For each $= ({}_L \sqsubseteq {}_R) \in \mathcal{O}'$, we have that all symbols in $\mathsf{Sig}( )$ are reachable from A and thus are interpreted as $\{a\}$ and $\{(a, a)\}$, respectively. It follows that ${}_L{}^{\mathcal{I}} = {}_R{}^{\mathcal{I}} = \{a\}$ if is a GCI, and ${}_L{}^{\mathcal{I}} = {}_R{}^{\mathcal{I}} = \{(a, a)\}$ otherwise. In both cases, we have that $\mathcal{I} \models$. For each $= ({}_L \sqsubseteq {}_R) \in \mathcal{O}''$, there is a symbol $x \in \mathsf{Sig}({}_L)$ unreachable from A. By construction of $\mathcal{I}$, we have $x^{\mathcal{I}} = \emptyset$, implying ${}_L{}^{\mathcal{I}} = \emptyset$. Thus, $\mathcal{I} \models$ as required.

❑

The converse of Point 5 is not true in general, for instance, Pericarditis involves Tissue, but the corresponding subsumption does not follow from the ontology. This suggests that we could use connected reachability as a heuristic for answering negative subsumption, in a similar but ner way as in weak reachability.

We outline our algorithm for extracting the reachability-based module given a signature $\mathbf{S}$ and an ontology $\mathcal{O}$ in Algorithm 1. Similar to the technique developed in [4], we view the input ontology $\mathcal{O}$ as a mapping $\mathsf{active\text{-}axioms} : \mathsf{Sig}(\mathcal{O}) \to \mathcal{O}$

Algorithm 1 extract-module

Input: $\mathcal{O}$: $\mathcal{EL}^+$ ontology; $\mathbf{S}$: signature
Output: $\mathcal{O}_\mathbf{S}$: reachability-based module for $\mathbf{S}$ in $\mathcal{O}$

```
1: $\mathcal{O}_\mathbf{S}$     $\emptyset$
2: queue     active-axioms($\mathbf{S}$)
3: **while not** empty(queue) **do**
4:     ( $_L \sqsubseteq$   $_R$)     fetch(queue)
5:     **if** Sig(  $_L$)     $\mathbf{S} \cup$ Sig($\mathcal{O}_\mathbf{S}$) **then**
6:         $\mathcal{O}_\mathbf{S}$     $\mathcal{O}_\mathbf{S} \cup \{$  $_L \sqsubseteq$   $_R\}$
7:         queue     queue $\cup$ (active-axioms(Sig(  $_R$)) $\setminus \mathcal{O}_\mathbf{S}$)
8: **return** $\mathcal{O}_\mathbf{S}$
```

with active-axioms(x) comprising all and only axioms $_L \sqsubseteq$ $_R \in \mathcal{O}$ such that x occurs in $_L$. The main di erences, compared to $\widehat{\mathcal{O}}$ mapping in Section 4 are that active-axioms does not assume the input ontology to be in normal form, and that it is de ned for both concept and role names. The intuition is that every axiom $\in$ active-axioms(x) is "active" for x, in the sense that y could be connectedly reachable via   from x for some $y \in$ Sig($\mathcal{O}$). For convenience, we de ne active-axioms($\mathbf{S}$) := $\bigcup_{x \in \mathbf{S}}$ active-axioms(x) for a signature $\mathbf{S}$ Sig($\mathcal{O}$).

It is easy to see that each axiom Algorithm 1 extracts to $\mathcal{O}_\mathbf{S}$ is $\mathbf{S}$-reachable. The fact that all $\mathbf{S}$-reachable axioms are extracted to $\mathcal{O}_\mathbf{S}$ can be proved by induction on connected reachability.

Proposition 6 (Algorithm 1 produces $\mathcal{O}_\mathbf{S}^{\mathsf{reach}}$). *Let $\mathcal{O}$ be an $\mathcal{EL}^+$ ontology and* $\mathbf{S}$ Sig($\mathcal{O}$) *a signature. Then, Algorithm 1 returns the reachability-based module for* $\mathbf{S}$ *in* $\mathcal{O}$.

In fact, connected reachability can be reduced to propositional Horn clause implication. The idea is to translate each $\mathcal{EL}^+$ axiom $_L \sqsubseteq$ $_R$ into the Horn clause $l_1 \wedge$ $\wedge\, l_m \rightarrow r_1 \wedge$ $\wedge\, r_n$ where $l_i \in$ Sig($_L$) and $r_i \in$ Sig($_R$). Given a signature $\mathbf{S}$ and a symbol x, x is $\mathbf{S}$-reachable i x is implied by $\bigwedge_{y \in \mathbf{S}} y$ w.r.t. the Horn clauses. The Dowling-Gallier algorithm [5] can check this in linear time.

Lemma 7 ($\mathcal{O}_A^{\mathsf{reach}}$ preserves $A \sqsubseteq_\mathcal{O} B$). *Let $\mathcal{O}$ be an $\mathcal{EL}^+$ ontology, $A \in$* CN($\mathcal{O}$)*, and $\mathcal{O}_A^{\mathsf{reach}}$ the reachability-based module for $\mathbf{S} = \{A\}$ in $\mathcal{O}$. Then, for any $= A \sqsubseteq B$ with $B \in$* CN($\mathcal{O}$)*, $\mathcal{O} \models$ i $\mathcal{O}_A^{\mathsf{reach}} \models$.*

Proof. "If" direction immediately follows from monotonicity of $\mathcal{EL}^+$. For "only if", we show that if $\mathcal{O}_A^{\mathsf{reach}} \not\models$, then $\mathcal{O} \not\models$. Assume that $\mathcal{O}_A^{\mathsf{reach}} \not\models$ $= (A \sqsubseteq B)$ and that B is connectedly reachable from A in $\mathcal{O}$, for otherwise, Point 5 of Proposition 5 implies that $\mathcal{O} \not\models$, and we are done. Since $\mathcal{O}_A^{\mathsf{reach}} \not\models$, there is a model $\mathcal{I}_A$ of $\mathcal{O}_A^{\mathsf{reach}}$ such that $A^{\mathcal{I}_A}$ / $B^{\mathcal{I}_A}$. Extend $\mathcal{I}_A$ to $\mathcal{I}$ by setting $x^I := \emptyset$ for all $x \in$ Sig($\mathcal{O}$)$\setminus$Sig($\mathcal{O}_A^{\mathsf{reach}}$). Since $\mathcal{I}$ is an extension of $\mathcal{I}_A$ and $A, B \in$ Sig($\mathcal{O}_A^{\mathsf{reach}}$), $\mathcal{I}$

is a model of $\mathcal{O}_A^{\mathsf{reach}}$ and $A^{\mathcal{I}}$ / $B^{\mathcal{I}}$. For each axiom $= ({}_L \sqsubseteq {}_R) \in \mathcal{O} \backslash \mathcal{O}_A^{\mathsf{reach}}$, we have that $\mathsf{Sig}({}_L)$ / $\mathsf{Sig}(\mathcal{O}_A^{\mathsf{reach}})$, since is not A-reachable. It follows that there is an $x \in \mathsf{Sig}({}_L)$ s.t. $x^{\mathcal{I}} = \emptyset$, implying by the semantics of $\mathcal{EL}^+$ that ${}_L{}^{\mathcal{I}} = \emptyset$. Thus, $\mathcal{I} \models$. ❑

This property suggests that, to query subsumption, it is enough to extract and maintain only linearly many modules, i.e. one for each concept name. Precisely, the module $\mathcal{O}_A$ can be used to correctly answer subsumption $A \sqsubseteq_{\mathcal{O}}^{?} B$ for any concept name $B \in \mathsf{Sig}(\mathcal{O})$. In the following, we show a tight relationship between our reachability-based modules and locality-based modules. Since locality-based modules also enjoy the property stated by Lemma 7, it is indeed an immediate corollary of the following result:

Lemma 8 ($\mathcal{O}_{\mathbf{S}}^{\mathsf{reach}}$ is the minimal locality-based module). *Let $\mathcal{O}$ be an $\mathcal{EL}^+$ ontology,* $\mathbf{S}$ $\mathsf{Sig}(\mathcal{O})$ *a signature. Then, $\mathcal{O}_{\mathbf{S}}^{\mathsf{reach}}$ is the minimal locality-based module for* $\mathbf{S}$ *in* $\mathcal{O}$.

Proof. First, we show that $\mathcal{O}_{\mathbf{S}}^{\mathsf{reach}}$ is a locality-based module. To prove this, it su ces to show that, for each axiom $= ({}_L \sqsubseteq {}_R) \in \mathcal{O} \backslash \mathcal{O}_{\mathbf{S}}^{\mathsf{reach}}$, is syntactically local w.r.t. $\mathbf{S} \cup \mathsf{Sig}(\mathcal{O}_{\mathbf{S}}^{\mathsf{reach}})$. Since $\mathcal{O}_{\mathbf{S}}^{\mathsf{reach}}$ contains only $\mathbf{S}$-reachable axioms, ${}_L$ is not $\mathbf{S}$-reachable, i.e. there exists an $x \in \mathsf{Sig}({}_L)$ such that x is not $\mathbf{S}$-reachable. By Point 4 of Proposition 5, $x \notin \mathbf{S} \cup \mathsf{Sig}(\mathcal{O}_{\mathbf{S}}^{\mathsf{reach}})$. Since x occurs in ${}_L$, by De nition 2, is syntactically local w.r.t. $\mathbf{S} \cup \mathsf{Sig}(\mathcal{O}_{\mathbf{S}}^{\mathsf{reach}})$, as required.

It remains to show that $\mathcal{O}_{\mathbf{S}}^{\mathsf{reach}}$ is minimal. Assume to the contrary that a smaller set $\mathcal{O}_{\mathbf{S}}^{\mathsf{reach}} \backslash \{ \ \}$ is a locality-based module, for some axiom $= ({}_L \sqsubseteq {}_R) \in \mathcal{O}_{\mathbf{S}}^{\mathsf{reach}}$. By de nition, each axiom $\in \mathcal{O} \backslash (\mathcal{O}_{\mathbf{S}}^{\mathsf{reach}} \backslash \{ \ \})$ is syntactically local w.r.t. $\mathbf{S}' = \mathbf{S} \cup \mathsf{Sig}(\mathcal{O}_{\mathbf{S}}^{\mathsf{reach}} \backslash \{ \ \})$. In particular, is syntactically local w.r.t. $\mathbf{S}'$. Our claim is that is not reachable from $\mathbf{S}$ w.r.t. $\mathcal{O}$. This contradicts the fact that $\in \mathcal{O}_{\mathbf{S}}^{\mathsf{reach}}$. ↯

Claim: Let $\mathbf{S}' = \mathbf{S} \cup \mathsf{Sig}(\mathcal{O}_{\mathbf{S}}^{\mathsf{reach}} \backslash \{ \ \})$ with syntactically local w.r.t. $\mathbf{S}'$. Then, is not reachable from $\mathbf{S}$ w.r.t. $\mathcal{O}$.

Since $= ({}_L \sqsubseteq {}_R)$ is syntactically local w.r.t. $\mathbf{S}'$, there exists an $x \in \mathsf{Sig}({}_L)$ s.t. $x \notin \mathbf{S}'$. There are two mutually disjoint cases: $x \notin \mathbf{S} \cup \mathsf{Sig}(\mathcal{O}_{\mathbf{S}}^{\mathsf{reach}})$ or $x \in \mathsf{Sig}(\ ) \backslash (\mathbf{S} \cup \mathsf{Sig}(\mathcal{O}_{\mathbf{S}}^{\mathsf{reach}} \backslash \{ \ \}))$. In the former case, x (thus,) is not $\mathbf{S}$-reachable by Point 4 of Proposition 5. In the latter case, x does not occur in any other axioms from $\mathcal{O}_{\mathbf{S}}^{\mathsf{reach}}$ apart from . In order for x to be $\mathbf{S}$-reachable, x must occur on the right-hand side of some axiom. Since x occurs only in , it means that $x \in \mathsf{Sig}({}_R)$. But, since x occurs on the left-hand side of as well, x cannot be $\mathbf{S}$-reachable.

❑

So, Algorithm 1 can be used to extract a locality-based module in an $\mathcal{EL}^+$ ontology. The main di erence, in contrast to the algorithm used in [7, 6], is that our algorithm considers only "active" axioms for ${}_R$ when a new axiom ${}_L \sqsubseteq {}_R$

is extracted. Also, testing whether an $\mathcal{EL}^+$ axiom $\ = (\ _L \sqsubseteq \ _R)$ is non-local w.r.t. a signature $\mathbf{S} \cup \mathsf{Sig}(\mathcal{O}_\mathbf{S})$ boils down to testing $\mathbf{S}$-reachability of , which is a simpler operation of testing set inclusion $\mathsf{Sig}(\ _L) \ ^? \ \mathbf{S} \cup \mathsf{Sig}(\mathcal{O}_\mathbf{S})$. This is due to the fact that any concept description and role composition $\ _L$, with $x \in \mathsf{Sig}(\ _L)$ interpreted as the empty set, is itself interpreted as the empty set. This observation could be used to optimize module extraction for ontologies in expressive description logics.

4 Goal-Directed Subsumption Algorithm

In general, the techniques developed for module extraction have a number of potential applications, including optimization of standard reasoning, incremental classi cation and ontology re-use. An obvious way to exploit module extraction to speed up standard reasoning, such as subsumption $\ \sqsubseteq^?_\mathcal{O}\ $, is to rst extract the module $\mathcal{O}^{\mathsf{reach}}$ for $\{\ \}$ in $\mathcal{O}$, and then query the subsumption $\ \sqsubseteq^?_{\mathcal{O}^{\mathsf{reach}}}\ $, i.e. against the module instead of the original ontology. Based on the assumption that modules are relatively much smaller than the ontology, this optimization should be highly e ective. In this section, however, we argue that module extraction actually does not help speed up standard reasoning in $\mathcal{EL}^+$. This stems from the deterministic nature of the reasoning algorithm for deciding subsumption in $\mathcal{EL}^+$, which is in contrast to non-deterministic tableau-based algorithms for expressive logics, such as $\mathcal{SHOIQ}$.

In fact, with small modi cations to the $\mathcal{EL}^+$ classi cation algorithm (rst introduced in [2] for $\mathcal{EL}^{++}$ and later re ned for implementation in [4]), we obtain a subsumption testing algorithm. The modi ed algorithm does not actually have to perform steps irrelevant to the subsumption in question – *the goal*. We call this variant the *goal-directed subsumption algorithm*.

Algorithm 2 outlines the modi ed core procedure goal-directed-process to replace process of Figure 4 in [4]. The procedure process-new-edge, as well as essential data structures, i.e. $\widehat{\mathcal{O}}$, queue, R, S, remains intact. In particular, we view the (normalized) input ontology $\mathcal{O}$ as a mapping $\widehat{\mathcal{O}}$ from concepts (appearing on the left-hand side of some GCI) to sets of queue entries. Here, $\mathbf{B}$ denotes the set of all concept names appearing in the conjunction $B_1 \sqcap \quad \sqcap B_n$.

The main di erence is the initialization of S, thus of queue. Since we are interested in the particular subsumption $\ \sqsubseteq\ $, we "activate" only by initializing $S(\ )$ with $\{\ , \top\}$ and queue$(\ )$ with $\widehat{\mathcal{O}}(\ ) \cup \widehat{\mathcal{O}}(\top)$. We activate a concept name B *only* when it becomes the second component of a tuple added to some $R(r)$ and has not been activated previously (see lines 8-9 in goal-directed-process of Algorithm 2). Thereby, $S(B)$ and queue(B) are initialized accordingly. Queues are processed in the same fashion as before except that and are now being monitored (Line 6), so that immediately after is added to $S(\ )$, the algorithm terminates with the positive answer (Line 7). Otherwise, goal-directed-process terminates normally, and the next queue entry will be fetched (Line 3 in subsumes? of Algorithm 2) and processed (Line 4). Unless 'positive' is returned, queues processing is continued until they are all empty. In this case, the algorithm returns 'negative'.

It is important to note that the goal-directed algorithm activates only concept names relevant to the target subsumption $\sqsubseteq$, i.e. reachable via $R(\ )$ from . The subsumer sets of concept names that do not become activated are not populated. Moreover, axioms that are involved in rule applications during the computation of subsumes?($\sqsubseteq$) are those from the reachability-based module $\mathcal{O}^{\mathsf{reach}}$ in $\mathcal{O}$. The following proposition states this correlation:

Proposition 9 (subsumes?($\sqsubseteq$) only requires axioms in $\mathcal{O}^{\mathsf{reach}}$). *Let $\mathcal{O}$ be an ontology in $\mathcal{EL}^+$ normal form, and $\mathcal{O}^{\mathsf{reach}}$ the reachability-based module for* $\{\ \}$ *in $\mathcal{O}$. Then,* subsumes?($\sqsubseteq$) *only requires axioms in $\mathcal{O}^{\mathsf{reach}}$* $\mathcal{O}$.

Proof. Assume that Algorithm 2 requires , for some axiom $\in \mathcal{O}$, i.e. is used in a rule application and thus causes addition to either $S(\ )$ or $R(\ )$. Before we can prove the proposition, we need the following invariants:

Inv1: If a concept name A is activated, then A is -reachable w.r.t. $\mathcal{O}$.

Inv2: If $B \in S(A)$ for some concept name A, then B is -reachable w.r.t. $\mathcal{O}$.

Inv3: If $(A, B) \in R(r)$ for some role name r, then r is -reachable w.r.t. $\mathcal{O}$.

Inv4: If $(\mathbf{B} \to B) \in$ queue(A) and, for all $B' \in \mathbf{B}$, B' is -reachable, then B is -reachable (a special case, if $(\emptyset \to B) \in$ queue(A), then B is -reachable); and, if $\exists r.B \in$ queue(A), then r and B are -reachable.

Inv5: If r is processed by process-new-edge, then r is -reachable w.r.t. $\mathcal{O}$.

Preservation of these invariants can be proved by induction on on execution of the algorithm. Induction start: is activated (Line 1 of subsumes?). By de nition, is -reachable w.r.t. $\mathcal{O}$. Recall that only activated concepts A can be processed by goal-directed-process. Induction step: we show the following four cases. Other cases can be easily shown in a similar fashion.

At Line 2 of goal-directed-process, B is added to $S(A)$ if $\mathbf{B}$ $S(A)$. By Inv2, every $B' \in \mathbf{B}$ is reachable from A. Since A is activated, Inv1 together with Point 2 of Proposition 5 implies that B' is -reachable. By the rst part of Inv4, B is reachable from , thus preserving Inv2.

At Line 3, elements from $\widehat{\mathcal{O}}(B)$ are added to queue(A). There are three potential kinds of axioms involved, i.e. $B \sqsubseteq B'$, $B \sqsubseteq \exists r.B'$, and $\mathbf{B} \sqsubseteq B'$ s.t.

$B \in \mathbf{B}$. In the rst two cases, B is -reachable due to Inv2 and $B \in S(A)$. By de nition, B' (r, B', resp.) is -reachable, thus preserving Inv4. In the last case, the rst part of Inv4 follows immediately from the de nition of connected reachability with $\mathsf{Sig}(\ _L) = \mathbf{B}$ and $\mathsf{Sig}(\ _R) = \{B'\}$.

At Line 9, B is activated. Since $\exists r.B$ occurred in queue(A), r, B are -reachable by the second part of Inv4. Thus, Inv1 is preserved.

At Line 10, r is -reachable by the same argument above, preserving Inv5. Procedure process-new-edge calls itself recursively potentially with a di erent role name v. Given that r is -reachable, it is trivial to see that s in Line 1 is also -reachable. Invocation at Line 5 and 7 can be shown in a parallel manner, and we only treat the former. By Inv3, $(A', A) \in R(u)$ implies that u is -reachable. Since $u \ \ s \sqsubseteq v \in \mathcal{O}$ and both u, s are reachable from , v is also -reachable, preserving Inv5.

Now we show that is indeed -reachable w.r.t. $\mathcal{O}$, thus in $\mathcal{O}^{\mathsf{reach}}$. We do case distinction w.r.t. the normal form of .

$X \sqsubseteq Y$ is required when $\widehat{\mathcal{O}}(X)$ augments some queue (Line 3). X is -reachable by Inv2 and the fact that it has been added to some $S(A)$. Obviously, is -reachable by de nition.

$\mathbf{X} \sqsubseteq Y$ is required when $\widehat{\mathcal{O}}(X)$ augments some queue (Line 3), for an $X \in \mathbf{X}$, and $\mathbf{X} \ \ \ S(A)$ for some concept name A. By Inv2, all $X' \in \mathbf{X}$ are -reachable. Obviously, is -reachable by de nition.

$X \sqsubseteq \exists r.Y$ (analogy to the rst case).

$\exists r.X \sqsubseteq Y$ is required $\widehat{\mathcal{O}}(\exists r.X)$ augments some queue (Line 5 in goal-directed-process and Line 3 in process-new-edge). Since $R(r)$ is not empty, Inv3 implies that r is -reachable. Also, X is -reachable since X occurs in some $S(A)$. By de nition, is -reachable.

$r \sqsubseteq s$ is required when it participates in the outer **for**-loop. Since r is -reachable by Inv5, s is also -reachable.

$u \ \ s \sqsubseteq v$ is required when the conditions at Line 4 (resp, Line 6) are satis ed. Obviously, is -reachable since both u and s are.

❏

Intuitively, the proposition suggests that our goal-directed subsumption algorithm inherently takes into account the notion of reachability-based module, i.e. it applies rules only to relevant axioms in the module. In fact, the preprocessing overhead of extracting relevant modules makes the overall computation time for a single subsumption query longer. This has been empirically con rmed in our experiments (see the last paragraph of Section 6).

Despite what has been said, module extraction is still useful for, e.g., ontology re-use, explanation, and full- edged incremental reasoning [6].

5 Duo-Ontology Classi cation

Unlike tableaux-based algorithms, the polynomial subsumption algorithm [2, 4] inherently classi es the input ontology by making all subsumptions between *concept names* explicit. This algorithm can be used to query subsumption between concept names occurring in the ontology, but complex subsumptions, such as

$$\mathsf{In\ \ ammation} \sqcap \exists \mathsf{has\text{-}location.Heart} \sqsubseteq^{?}_{\mathcal{O}_{\mathsf{ex}}} \mathsf{HeartDisease} \sqcap \exists \mathsf{has\text{-}state.NeedsTreatment}$$

cannot be answered directly. First, the ontology $\mathcal{O}_{\mathsf{ex}}$ from Figure 1 has to be augmented to $\mathcal{O}'_{\mathsf{ex}} := \mathcal{O}_{\mathsf{ex}} \cup \{A \sqsubseteq \mathsf{In\ \ ammation} \sqcap \exists \mathsf{has\text{-}location.Heart}, \mathsf{HeartDisease} \sqcap \exists \mathsf{has\text{-}state.NeedsTreatment} \sqsubseteq B\}$ with A, B new concept names, and then the subsumption test $A \sqsubseteq^{?}_{\mathcal{O}'_{\mathsf{ex}}} B$ can be carried out to decide the original complex subsumption. Since A, B are new names not occurring in $\mathcal{O}_{\mathsf{ex}}$, our complex subsumption holds i $A \sqsubseteq_{\mathcal{O}'_{\mathsf{ex}}} B$. This approach is e ective but ine cient unless only one such complex subsumption is queried for each ontology. Constructing and normalizing the augmented ontology every time each subsumption is tested is not likely to be acceptable in practice, especially when the background ontology is large. For instance, normalization of SNOMED CT takes more than one minute.

In this section, we propose an extension to the re ned algorithm (henceforth referred to as *the original algorithm*) developed in [4] to cater for a *duo-ontology* $\mathcal{O} = (\mathcal{O}_p \cup \mathcal{O}_t)$ with $\mathcal{O}_p$ a *permanent* $\mathcal{EL}^+$ ontology and $\mathcal{O}_t$ a set of *temporary* GCIs. Intuitively, $\mathcal{O}_p$ is the input ontology of which axioms have been read in and processed before, while $\mathcal{O}_t$ contains temporary GCIs that are asserted later. The main purpose is to reuse the information made available by preprocessing and classifying $\mathcal{O}_p$. Once $\mathcal{O}_p$ has been classi ed, the classi cation of $\mathcal{O}_p \cup \mathcal{O}_t$ should not start from scratch, but rather use the existing classi cation information together with the new GCIs from $\mathcal{O}_t$ to do incremental classi cation.

In our extension, we use two sets of the core data structures $\widehat{\mathcal{O}}(\ ), R(\ ), S(\ )$, but retain a single set of queues $\mathsf{queue}(\ )$. The mappings $\widehat{\mathcal{O}}_p, R_p, S_p$ are initialized and populated exactly as in the original algorithm, i.e. $\widehat{\mathcal{O}}_p$ encodes axioms in $\mathcal{O}_p$, and R_p, S_p store subsumption relationships inferred from $\mathcal{O}_p$. Similarly, the mapping $\widehat{\mathcal{O}}_t$ encodes axioms in $\mathcal{O}_t$, but R_t, S_t represent additional inferred subsumptions drawn from $\mathcal{O}_p \cup \mathcal{O}_t$ that are not already present in R_p, S_p, respectively. The extended algorithm is based on the tenet that description logics are monotonic, i.e. $\mathcal{O}_p \models$ implies $\mathcal{O}_p \cup \mathcal{O}_t \models$. There may be an additional consequence such that $\mathcal{O}_p \not\models$ but $\mathcal{O}_p \cup \mathcal{O}_t \models$. The extended algorithm stores such a consequence in a separate set of data structures, viz. R_p, S_p. Analogously to the original algorithm, queue entries are repeatedly fetched and processed until all queues are empty. Instead of the procedures process and process-new-edge, we use the extended versions for duo-ontology classi cation as outlined in Algorithm 3.

The extended algorithm's behavior is identical to that of the original one [4] if $\mathcal{O}_p$ has not been classi ed. In particular, $\widehat{\mathcal{O}}_p(\ ) \cup \widehat{\mathcal{O}}_t(\ )$ here is equivalent to $\widehat{\mathcal{O}}(\ )$ in [4] given that $\mathcal{O} = (\mathcal{O}_p \cup \mathcal{O}_t)$. Since no classi cation has taken place, $S_p(A) = R_p(r) = \emptyset$ for all concept name A and role name r. Initialization and processing of queues are done in the same manner with the only di erence that inferred consequences are now put in R_t and S_t.

If $\mathcal{O}_p$ has been classi ed (thus, S_p, R_p have been populated), then proper initialization has to be done w.r.t. previously inferred consequences (i.e. S_p, R_p) and new GCIs (i.e. $\widehat{\mathcal{O}}_t$). To this end, we initialize the data structures by setting:

for each role name $r \in \mathsf{RN}(\mathcal{O})$, $R_t(r) := \emptyset$;

for each *old* concept name $A \in \mathsf{CN}(\mathcal{O}_p)$, $S_t(A) := \emptyset$ and
$\mathsf{queue}(A) := \bigcup_{X \in S_p(A)} \widehat{\mathcal{O}}_t(X) \ \cup \ \bigcup_{\{(A,B) \in R_p(r), X \in S_p(B)\}} \widehat{\mathcal{O}}_t(\exists r.X)$;

for each *new* concept name $A \in \mathsf{CN}(\mathcal{O}_t) \backslash \mathsf{CN}(\mathcal{O}_p)$, $S_t(A) := \{A, \top\}$
$\mathsf{queue}(A) := \widehat{\mathcal{O}}_t(A) \cup \widehat{\mathcal{O}}_t(\top)$.

After initialization, queue processing is carried out by Algorithm 3 until all queues are empty. Observe the structural analogy between these procedures and the original ones in [4]. Observe also the key di erence: information is always retrieved from both sets of data structures, e.g., $S_p(A) \cup S_t(A)$ in Line 1, while modi cations are only made to the temporary set of data structures, e.g., $S_t(A) := S_t(A) \cup \{B\}$ in Line 2. The correctness of this algorithm can be shown following the proof's structures in the appendix of [4] w.r.t. additional subsumption consequences obtained during incremental classi cation.

Lemma 10 (Correctness of Algorithm 3). *Let $\mathcal{O} = (\mathcal{O}_p \cup \mathcal{O}_t)$ be a duo-ontology, and S_p, R_p be the results after the original algorithm terminates on $\mathcal{O}_p$. Then, the extended algorithm (Algorithm 3), applied to $\mathcal{O}_t$, incrementally classi es $\mathcal{O}_t$ against $\mathcal{O}_p$ (i.e. classi es $\mathcal{O}$) in time polynomial in the size of $\mathcal{O}$. That is, $B \in S_p(A) \cup S_t(A)$ i $A \sqsubseteq_{\mathcal{O}} B$ for all $A, B \in \mathsf{CN}(\mathcal{O})$.*

In our example, we set $\mathcal{O}_p$ to $\mathcal{O}_{\mathsf{ex}}$ and $\mathcal{O}_t$ to the set of the two new GCIs. We can run the extended algorithm on $\mathcal{O}_p \cup \mathcal{O}_t$ and reuse existing information in S_p and R_p, if any. After termination, our complex subsumption boils down to the set membership test $B \in^? S_p(A) \cup S_t(A) = S_t(A)$. To decide next subsumptions, only $\mathcal{O}_t, R_t, S_t$ and queue need to be initialized, leaving the background ontology $\mathcal{O}_p$ and possibly its classi cation information R_t, S_t intact.

Interestingly, this algorithm can be used e ectively in certain scenarios of incremental classi cation. Consider $\mathcal{O}_p$ as a well-developed, permanent ontology, and $\mathcal{O}_t$ as a small set of temporary axioms currently being authored. Obviously, if the permanent ontology is large, it would be impractical to reclassify from scratch every time some new axioms are to be added. Algorithm 3 incrementally classi es $\mathcal{O}_t$ against $\mathcal{O}_p$ and its classi cation information. If the inferred consequences are satisfactory, the temporary axioms can be committed to the permanent ontology by merging the two sets of data structures. Otherwise, axioms in $\mathcal{O}_t$ and their inferred consequences could be easily retracted, since these are segregated from $\mathcal{O}_p$ and its consequences. To be precise, we simply dump the values of $\mathcal{O}_t(\ ), R_t(\ )$ and $S_t(\ )$, when the temporary axioms are retracted.

6 Experiments and Empirical Results

This section describes the experiments and results of the three algorithms we proposed in the present paper: module extraction, goal-directed subsumption query, and duo-ontology classi cation. We have implemented the three algorithms and integrated them as new features in the CEL reasoner [3] version 1.0b.[1] All the experiments have been carried out on a standard PC: 2.40 GHz Pentium-4 processor and 1 GB of physical memory. In order to show interesting properties of reachability-based modules and scalability of subsumption and incremental classi cation in $\mathcal{EL}^+$, we have selected a few large ontologies from the medical domain. Our test suite comprises Snomed ct, NCI, and the $\mathcal{EL}^+$ fragments[2] of Galen and NotGalen, denoted respectively by $\mathcal{O}^{\text{Snomed}}$, $\mathcal{O}^{\text{NCI}}$, $\mathcal{O}^{\text{Galen}}$, and $\mathcal{O}^{\text{NotGalen}}$.[3] The Galen ontology shall not be confused with the original version of Galen, the latter of which is almost 10 times smaller and commonly used in DL benchmarking. The sizes of our test suite ontologies are shown in the second and third columns of Table 1. The last but one column shows the time CEL needs to classify each ontology, while the last presents in percentage the ratio of positive subsumption relationships between concept names. Observe that all ontologies

[1]Available at `http://lat.inf.tu-dresden.de/systems/cel/`

[2]The full Galen medical ontology is precisely based on $\mathcal{SHIF}$ dispensed with disjunction and value restriction. The description logic $\mathcal{EL}^+$ can indeed express most of the axioms, namely 95.75%, and we obtained this fragment for experimental purposes by dropping role inverse and functionality axioms.

[3]Obtainable at `http://lat.inf.tu-dresden.de/ meng/toyont.html`

Ontologies	♯Concepts/roles	♯Concept/role axioms	C. time (sec)	Pos. subs. (%)
$\mathcal{O}^{\text{NotGalen}}$	2 748 / 413	3 937 / 442	7.36	0.6013
$\mathcal{O}^{\text{Galen}}$	23 136 / 950	35 531 / 1 016	512.72	0.1648
$\mathcal{O}^{\text{NCI}}$	27 652 / 70	46 800 / 140	7.01	0.0441
$\mathcal{O}^{\text{Snomed}}$	379 691 / 62	379 691 / 13	1 671.23	0.0074

Table 1: $\mathcal{EL}^+$ ontology test suite

have a very low ratio of positive subsumption (less than 1%); in particular, less than a ten-thousandth of potential subsumptions *actually* hold in $\mathcal{O}^{\text{Snomed}}$.

Modularization: For each ontology $\mathcal{O}$ in the test suite and each concept name $A \in \mathsf{CN}(\mathcal{O})$, we extracted the reachability-based module $\mathcal{O}_A^{\mathsf{reach}}$. Statistical data concerning the sizes of modules and times required to extract them are presented in Table 2. Observe that it took a tiny amount of time to extract a single module based on connected reachability, with the maximum time less than four seconds. However, extracting large number of modules (i.e. one for each concept name) required considerably more time and even longer than classi cation. This was nevertheless the rst implementation that was not highly optimized. Several optimization techniques could be employed in module extraction, especially recursive extraction as suggested by Point 3 of Proposition 5 and the counting techniques from [5]. To empirically support Lemma 8, we have compared our modularization algorithm to that from [6, 7]. As expected, the results of both algorithms coincide w.r.t. $\mathcal{O}^{\text{NotGalen}}$ and $\mathcal{O}^{\text{NCI}}$, while we were unable to obtain locality-based modularization results w.r.t. the other two ontologies.[4]

Interestingly, module extraction reveals important structural dependencies that re ect complexity of the ontology. Though very large, concepts in $\mathcal{O}^{\text{NCI}}$ and $\mathcal{O}^{\text{Snomed}}$ are loosely connected w.r.t. reachability which makes it relatively easy to classify. In contrast, $\mathcal{O}^{\text{Galen}}$ contains more complex dependencies[5], thus is hard to classify.

To realize the pattern of module sizes in these ontologies, we also present them in a distribution chart in Figure 2. We used the whole module size data w.r.t. $\mathcal{O}^{\text{NotGalen}}$, $\mathcal{O}^{\text{NCI}}$, and $\mathcal{O}^{\text{Snomed}}$. For comparison purposes, however, we only used the module sizes of the smaller group in $\mathcal{O}^{\text{Galen}}$. For each ontology, the X-axis ranges over the sizes of modules in ten of axioms, whereas the Y-axis shows in percentage the number of modules that have the respective size. As obviously

[4]By setting the Java heap space to 0.8 GB on our benchmarking machine, it took 2.89 and 53.07 seconds to extract all modules in $\mathcal{O}^{\text{NotGalen}}$ and $\mathcal{O}^{\text{NCI}}$, respectively, whereas it failed due to memory exhaustion on $\mathcal{O}^{\text{Galen}}$ and $\mathcal{O}^{\text{Snomed}}$.

[5]Based on the statistical data analysis, there are two clearly distinct groups of concepts in $\mathcal{O}^{\text{Galen}}$: the rst with module sizes between 0 and 523 (med. 39; avg. 59.29) and the second between 14 791 and 15 545 (med. 14 792; avg. 14 829). Surprisingly, there is no module of size between those of these two groups.

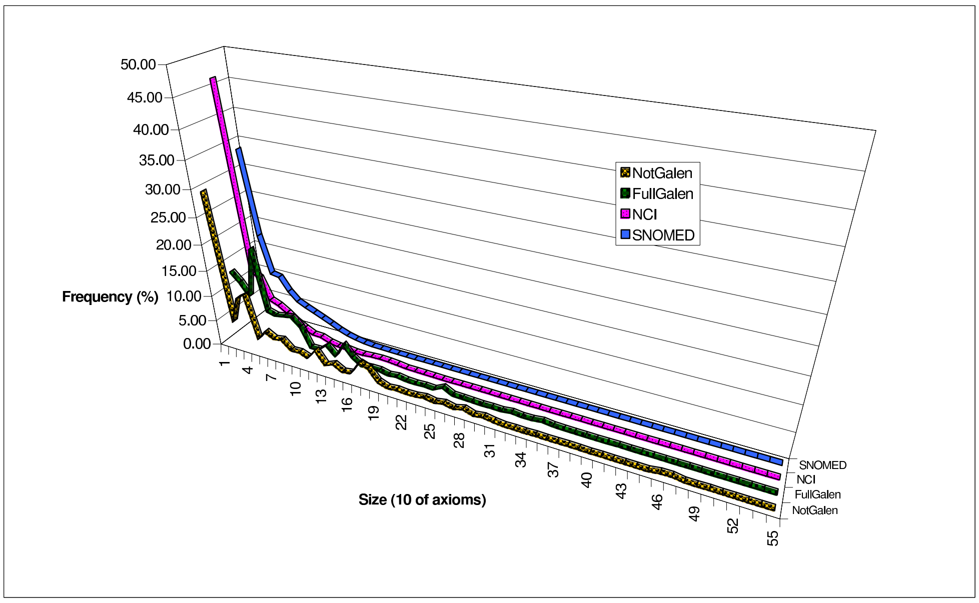

Figure 2: Distribution chart for sizes of the reachability-based modules.

depicted by the chart, the reachability-based modules are very small, in particular, in the case of $\mathcal{O}^{\text{NCI}}$ and $\mathcal{O}^{\text{Snomed}}$. In fact, more than 90% of modules in these two ontologies have less than 90 axioms.

Duo-ontology classi cation: As mentioned before, there are at least two applications of Algorithm 3, viz. complex subsumption query and (restricted) incremental classi cation. For complex subsumption query, we have adopted the "activation" idea from Algorithm 2 to quickly answer the query. To perform meaningful experiments, it is inevitable to involve a domain expert to obtain sensible test data. Though we have done so w.r.t. $\mathcal{O}^{\text{Snomed}}$, the numbers of complex subsumption queries and additional axioms are very small compared to the ontology size.[6] For this reason, we have developed our test strategy as follows: for each ontology $\mathcal{O}$ and various numbers n, we have (i) partitioned $\mathcal{O}$ into $\mathcal{O}_p$ and $\mathcal{O}_t$ such that $\mathcal{O}_t$ contains n% of GCIs from $\mathcal{O}$; (ii) classi ed $\mathcal{O}_p$ normally; nally, (iii) incrementally classi ed $\mathcal{O}_t$ against $\mathcal{O}_p$. The average computation times for several runs of (ii) and (iii) are shown in the left and right columns of each ontology in Table 3, respectively. It requires only 4% (resp., 15%, 35%, and 38%) of the total classi cation time for $\mathcal{O}^{\text{Snomed}}$ (resp., for $\mathcal{O}^{\text{Galen}}$, $\mathcal{O}^{\text{NCI}}$, and $\mathcal{O}^{\text{NotGalen}}$) to incrementally classify up to 1% of all axioms, i.e. about four-thousand axioms in the case of $\mathcal{O}^{\text{Snomed}}$.

Subsumption: To evaluate our goal-directed algorithm, we have run sub-

[6]On average, a typical complex subsumption query against $\mathcal{O}^{\text{Snomed}}$ took 0.00153 *milli*seconds, while incremental classi cation of one axiom needed 48.74 seconds.

sumption tests between *random* pairs of concept names without any heuristics.[7] Average/maximum querying times (in second) are 0.09/1.51 for $\mathcal{O}^{\text{NotGalen}}$, 124.01/254.31 for $\mathcal{O}^{\text{Galen}}$, 0.0034/0.44 for $\mathcal{O}^{\text{NCI}}$, and 0.0183/3.32 for $\mathcal{O}^{\text{Snomed}}$. Notice that subsumption requires a negligible amount of time and not much more than extracting a module in the case of $\mathcal{O}^{\text{NCI}}$ and $\mathcal{O}^{\text{Snomed}}$. Interestingly, subsumption querying times are roughly proportional to module sizes, which re ects the nature of the goal-directed algorithm as stated in Proposition 9.

7 Related Work

Recently, various techniques for extracting fragments of ontologies have been proposed in the literature. An example is the algorithm proposed in [12] which was developed speci cally for Galen. The algorithm traverses in de nitional order and into existential restrictions but does not take into account other dependencies, e.g., role hierarchy and GCIs. If applied to our example ontology $\mathcal{O}_{\mathsf{ex}}$, the algorithm extracts only $_1$, $_3$ and $_5$ as its segmentation output for Pericarditis. This is obviously not a module because we lose the subsumption Pericarditis $\sqsubseteq_{\mathcal{O}_{\mathsf{ex}}}$ HeartDisease. Another example is the Prompt-Factor tool [10] which implements an algorithm that, given an ontology $\mathcal{O}$ and a signature $\mathbf{S}$, retrieves a subset $\mathcal{O}_1$ $\mathcal{O}$ by retrieving to $\mathcal{O}_1$ axioms that contain symbols in $\mathbf{S}$ and extending $\mathbf{S}$ with $\mathsf{Sig}(\mathcal{O}_1)$ until a xpoint is reached. This is similar to our modules based on *weak* reachability, but it does not distinguish symbols occurring on lhs and rhs of axioms. In our example, the tool will return the whole ontology as output for $\mathbf{S} = \{\mathsf{Pericarditis}\}$, even though several axioms are irrelevant. As we have shown, modules based on syntactic locality [7] are equivalent to our reachability-based modules relative to $\mathcal{EL}^+$ ontologies. Since reachability is much simpler to check, our algorithm has proved more e cient.

Incremental classi cation and reasoning have received much attention in the recent years. In [8, 11], the so-called model-caching techniques have been investigated for application scenarios that only ABox is modi ed. A technique for incremental schema reasoning has recently been proposed in [6]: it utilizes modules to localize rami cations of changes and performs additional reasoning only on a ected modules. Since module extraction is somewhat expensive and has to be redone once the ontology is modi ed, it remains to be shown empirically whether this approach scales. All above-mentioned works focus on expressive languages. Here, however, we developed a very speci c approach to (restricted) incremental classi cation in $\mathcal{EL}^+$. Since the technique exploits the facts that the original $\mathcal{EL}^+$

[7]Since there are about *144 billion pairs* of concept names in the case of $\mathcal{O}^{\text{Snomed}}$ and some subsumption queries against $\mathcal{O}^{\text{Galen}}$ took a few minutes, performing subsumption queries between *all* pairs would not be feasible. Therefore, one thousand random pairs of subsumption were tested against $\mathcal{O}^{\text{Galen}}$, and one million random pairs against each of the other ontologies.

algorithm maintains completed subsumer sets, it is not immediately obvious how this may bene t tableau-based algorithms for expressive DLs.

8 Conclusion

In this paper, we have introduced a new kind of module (based on connected reachability) and proposed an algorithm to extract them from $\mathcal{EL}^+$ ontologies. We have shown that these are equivalent to locality-based modules w.r.t. $\mathcal{EL}^+$ ontologies and empirically demonstrated that modules can be extracted in reasonable time and are reasonably small. Also, we have proposed a goal-directed variant of the algorithm in [4] for testing subsumption prior to classi cation and have extended this algorithm to cater for a duo-ontology which can be utilized to answer complex subsumption queries and to do (restricted) incremental classi cation. Our empirical results have evidently con rmed that the proposed algorithms are practically feasible in large-scale ontology applications.

Despite not being directly useful to speed up standard reasoning in $\mathcal{EL}^+$, modularization obviously bene ts ontology re-use and explanation. As future work, we shall study the e ectiveness of using modules to optimize axiom pinpointing, which is the cornerstone of explanation support.

Acknowledgement: The author would like to acknowledge Franz Baader and Carsten Lutz for their valuable suggestions and Christian H.-Wiener for his willingness in comparing the two modularization approaches. This work has been supported by the EU project Thinking ONtologiES (aka, TONES) and the DFG project under grant BA 1122/11-1.

References

[1] The systematized nomenclature of medicine, clinical terms (Snomed ct). The International Health Terminology Standards Development Organisation, 2007. `http://www.ihtsdo.org/our-standards/`.

[2] F. Baader, S. Brandt, and C. Lutz. Pushing the $\mathcal{EL}$ envelope. In *Proc. of the 19th Int. Joint Conf. on Arti cial Intelligence (IJCAI-05)*, Edinburgh, UK, 2005. Morgan-Kaufmann Publishers.

[3] F. Baader, C. Lutz, and B. Suntisrivaraporn. CEL—a polynomial-time reasoner for life science ontologies. In U. Furbach and N. Shankar, editors, *Proc. of the 3rd Int. Joint Conf. on Automated Reasoning (IJCAR-06)*, volume 4130 of *Lecture Notes in Arti cial Intelligence*, pages 287–291. Springer-Verlag, 2006.

[4] F. Baader, C. Lutz, and B. Suntisrivaraporn. Is tractable reasoning in extensions of the description logic $\mathcal{EL}$ useful in practice? *Journal of Logic, Language and Information, Special Issue on Method for Modality (M4M)*, 2007. To appear.

[5] W. F. Dowling and J. Gallier. Linear-time algorithms for testing the satis ability of propositional horn formulae. *Journal of Logic Programming*, 1(3):267–284, 1984.

[6] B. Cuenca Grau, C. Halaschek-Wiener, and Y. Kazakov. History matters: Incremental ontology reasoning using modules. In *Proceedings of ISWC*, Busan, South Korea, 2007. Springer.

[7] B. Cuenca Grau, I. Horrocks, Y. Kazakov, and U. Sattler. Just the right amount: Extracting modules from ontologies. In *Proceedings of WWW*, pages 717–726, Ban , Canada, 2007. ACM.

[8] V. Haarslev and R. Moller. Incremental query answering for implementing document retrieval services. In *Proc. of the Int. Workshop on Description Logics (DL-03)*, pages 85–94, 2003.

[9] C. Lutz and F. Wolter. Conservative extensions in the lightweight description logic $\mathcal{EL}$. In *Proc. of the 21st Conf. on Automated Deduction*. Springer, 2007.

[10] N. Noy and M. Musen. The PROMPT suite: Interactive tools for ontology mapping and merging. *Int. J. of Human-Computer Studies*, 2003.

[11] B. Parsia, C. Halaschek-Wiener, and E. Sirin. Towards incremental reasoning through updates in OWL-DL. In *Proc. of Reasoning on the Web Workshop*, 2006.

[12] J. Seidenberg and A. Rector. Web ontology segmentation: Analysis, classi -cation and use. In *Proc. of WWW*. ACM, 2006.

Algorithm 2 Goal-directed subsumption algorithm

```
Procedure subsumes( $\sqsubseteq$ )
Input: ( $\sqsubseteq$ ): target subsumption
Output: 'positive' or 'negative' answer to the subsumption
 1: activate( )
 2: while not empty(queue($A$)) for some $A \in \mathsf{CN}(\mathcal{O})$ do
 3:    $X$    fetch(queue($A$))
 4:    if goal-directed-process($A, X,$  $\sqsubseteq$ ) then
 5:       return 'positive'
 6: return 'negative'

Procedure goal-directed-process($A, X,$  $\sqsubseteq$ )
Input: $A$: concept name; $X$: queue entry; ( $\sqsubseteq$ ): target subsumption
Output: 'positive' or 'unknown' answer to the subsumption
 1: if $X = \mathbf{B} \to B$, $\mathbf{B}$  $S(A)$ and $B \notin S(A)$ then
 2:    $S(A) := S(A) \cup \{B\}$
 3:    queue($A$) := queue($A$) $\cup\ \widehat{\mathcal{O}}(B)$
 4:    for all concept names $A'$ and role names $r$ with $(A', A) \in R(r)$ do
 5:       queue($A'$) := queue($A'$) $\cup\ \widehat{\mathcal{O}}(\exists r.B)$
 6:    if $A =$   and $B =$   then
 7:       return 'positive'
 8: if $X = \exists r.B$ and $(A, B) \notin R(r)$ then
 9:    activate($B$)
10:    process-new-edge($A, r, B$)
11: return 'unknown'

Procedure process-new-edge($A, r, B$)
Input: $A, B$: concept names; $r$: role name;
 1: for all role names $s$ with $r \sqsubseteq_{\mathcal{O}} s$ do
 2:    $R(s) := R(s) \cup \{(A, B)\}$
 3:    queue($A$) := queue($A$) $\cup \bigcup_{\{B' \mid B' \in S(B)\}} \widehat{\mathcal{O}}(\exists s.B')$
 4:    for all concept name $A'$ and role names $u, v$ with $u$   $s \sqsubseteq v \in \mathcal{O}$ and
       $(A', A) \in R(u)$ and $(A', B) \notin R(v)$ do
 5:       process-new-edge($A', v, B$)
 6:    for all concept name $B'$ and role names $u, v$ with $s$   $u \sqsubseteq v \in \mathcal{O}$ and
       $(B, B') \in R(u)$ and $(A, B') \notin R(v)$ do
 7:       process-new-edge($A, v, B'$)
```

Algorithm 3 Processing queue entries in duo-ontology classi cation

```
Procedure process-duo($A, X$)
Input: $A$: concept name; $X$: queue entry;
1: if $X = \mathbf{B} \rightarrow B$, $\mathbf{B}$   $S_p(A) \cup S_t(A)$ and $B \notin S_p(A) \cup S_t(A)$ then
2:     $S_t(A) := S_t(A) \cup \{B\}$
3:     queue($A$) := queue($A$) $\cup \widehat{\mathcal{O}}_p(B) \cup \widehat{\mathcal{O}}_t(B)$
4:     for all $A'$ and $r$ with $(A', A) \in R_p(r) \cup R_t(r)$ do
5:         queue($A'$) := queue($A'$) $\cup \widehat{\mathcal{O}}_p(\exists r.B) \cup \widehat{\mathcal{O}}_t(\exists r.B)$
6: if $X = \exists r.B$ and $(A, B) \notin R_p(r) \cup R_t(r)$ then
7:     process-new-edge($A, r, B$)

Procedure process-new-edge-duo($A, r, B$)
Input: $A, B$: concept names; $r$: role name;
1: for all role names $s$ with $r \sqsubseteq_{\mathcal{O}_p} s$ do
2:     $R_t(s) := R_t(s) \cup \{(A, B)\}$
3:     queue($A$) := queue($A$) $\cup \bigcup_{\{B' \mid B' \in S_p(B) \cup S_t(B)\}} (\widehat{\mathcal{O}}_p(\exists s.B') \cup \widehat{\mathcal{O}}_t(\exists s.B'))$
4:     for all concept name $A'$ and role names $u, v$ with $u$   $s \sqsubseteq v \in \mathcal{O}_p$ and
       $(A', A) \in R_p(u) \cup R_t(u)$ and $(A', B) \notin R_p(v) \cup R_t(v)$ do
5:         process-new-edge-duo($A', v, B$)
6:     for all concept name $B'$ and role names $u, v$ with $s$   $u \sqsubseteq v \in \mathcal{O}_p$ and
       $(B, B') \in R_p(u) \cup R_t(u)$ and $(A, B') \notin R_p(v) \cup R_t(v)$ do
7:         process-new-edge-duo($A, v, B'$)
```

Ontologies	Extraction time				Module size (%)		
	median	average	maximum	total	median	average	maximum
$\mathcal{O}^{\textsc{NotGalen}}$	< 0.01	0.00	0.01	2.38	35 (1.27)	68.64 (2.50)	495 (18.00)
$\mathcal{O}^{\textsc{Galen}}$	0.01	0.04	0.85	960	178 (0.77)	7092 (30.65)	15 545 (67.18)
$\mathcal{O}^{\text{NCI}}$	< 0.01	0.00	0.17	3.43	12 (0.026)	28.97 (0.062)	436 (0.929)
$\mathcal{O}^{\textsc{Snomed}}$	< 0.01	0.01	3.83	3 744	18 (0.005)	30.31 (0.008)	262 (0.069)

Table 2: Module extraction (time in second; size in number of axioms)

♯Temp. axioms	$\mathcal{O}^{\textsc{NotGalen}}$		$\mathcal{O}^{\textsc{Galen}}$		$\mathcal{O}^{\textsc{NCI}}$		$\mathcal{O}^{\textsc{Snomed}}$	
$(\lvert\mathcal{O}_t\rvert)$	C. time	IC. time	C. time	IC. time	C. time	IC. time	C. time	IC. time
0.2%	6.53	1.75	486.19	56.94	5.10	2.00	1 666.43	55.86
0.4%	6.50	1.88	484.89	59.37	4.81	2.15	1 663.51	57.97
0.6%	6.48	2.45	482.13	62.34	4.78	2.37	1 661.49	68.58
0.8%	6.43	2.88	466.97	80.52	4.70	2.54	1 652.84	83.27
1.0%	6.38	4.46	450.61	109.81	4.59	3.19	1 640.11	93.89

Table 3: Incremental classi cation (in second)