Modular Combination of Reasoners for Ontology Classification

Ana Armas Romero, Bernardo Cuenca Grau, Ian Horrocks

Department of Computer Science. University of Oxford

Abstract. Classification is a fundamental reasoning task in ontology design, and there is currently a wide range of reasoners highly optimised for classification of $\mathcal{SROIQ}$ ontologies. Existing reasoners, however, do not exploit the fact that most of the axioms in many realistic $\mathcal{SROIQ}$ ontologies are expressed in some lightweight DL, such as $\mathcal{EL}^{++}$. In this paper, we propose a novel reasoning technique that allows us to completely classify a large subset of the signature of a $\mathcal{SROIQ}$ ontology by relying only on a reasoner for a given lightweight DL. We also show how this information can then be exploited by the fully-fledged $\mathcal{SROIQ}$ reasoner HermiT to complete the classification of the ontology.

1 Introduction

Classification —the problem of identifying the subsumption relationships between all pairs of atomic concepts occurring in the input ontology— is a fundamental reasoning task in ontology design. The decision problems associated to classification, however, have a very high worst-case complexity for expressive DLs; in particular, subsumption w.r.t. an ontology is 2NEXPTIME-complete for $\mathcal{SROIQ}$ [14] —the DL underlying the standard ontology language OWL 2 [5].

Despite these discouraging complexity results, considerable effort has been devoted to making classification feasible in practice. As a result, many reasoning algorithms and optimisation techniques have been developed, and there is currently a wide range of highly-optimised reasoners, such as Pellet [19], FaCT++ [20], RacerPro [9] and HermiT [6], that support classification of ontologies written in expressive description logics.

Since individual subsumption tests performed during classification can be computationally very expensive, most DL reasoners implement variants of the well-known Enhanced Traversal Algorithm [2], which reduces the number of required subsumption tests. Sophisticated optimisation techniques are also implemented on top of these algorithms to further reduce the number of potentially expensive subsumption tests [10].

A widely implemented technique is the *told subsumptions* optimisation [10], which provides an inexpensive way of computing subsumption relationships that hold in the input ontology. In typical ontologies, however, most candidate subsumption relationships between atomic concepts will not hold; hence, efficiently identifying and exploiting such *non-subsumption* relationships becomes critical

in practice, and several optimisation techniques have been developed with this goal in mind. In particular, the *completely defined concepts* optimisation [21] identifies a fragment of the ontology for which told subsumption provides complete information; furthermore *model-merging* and other related techniques exploit the computations performed during individual concept satisfiability tests to detect non-subsumptions [10, 8, 6]. However, although these techniques have proved effective in practice, the classification of very large ontologies can still require a large number of expensive subsumption tests.

In recent years, there has been a growing interest in so-called *lightweight DL*s. The description logic $\mathcal{EL}^{++}$ [1], for example, can capture several prominent ontologies, and allows classification to be performed in polynomial time. Reasoners specifically designed for $\mathcal{EL}^{++}$, such as CEL [3] and ELK [15], can classify ontologies as large as SNOMED CT in a few seconds.

Unfortunately, many ontologies fall outside the $\mathcal{EL}^{++}$ fragment, and so cannot be classified using $\mathcal{EL}^{++}$ reasoners. In many cases, however, such ontologies contain only a relatively small number of non $\mathcal{EL}^{++}$ axioms. For example, out of the 219,224 axioms in the latest version of NCI, only 65 are non $\mathcal{EL}^{++}$. Being able to use an $\mathcal{EL}^{++}$ reasoner to efficiently compute most of the subsumptions and non-subsumptions required to classify these ontologies could lead to significant improvements in both performance and scalability.

In this paper, we propose a technique where a reasoner for some DL $\mathcal{L}$ is used as "black box" by a reasoner for a more expressive logic $\mathcal{L}'$. We focus on the case where $\mathcal{L}'$ is $\mathcal{SROIQ}$, and we present a classification algorithm that, given a $\mathcal{SROIQ}$ ontology $\mathcal{O}$, proceeds as follows:

1. It computes a signature $\Sigma^{\mathcal{L}} \subseteq \mathsf{Sig}(\mathcal{O})$ and a fragment $\mathcal{M}^{\mathcal{L}} \subseteq \mathcal{O}$ written in $\mathcal{L}$ such that the concepts in $\Sigma^{\mathcal{L}}$ can be completely classified using only the axioms in $\mathcal{M}^{\mathcal{L}}$; more precisely, $\Sigma^{\mathcal{L}}$ and $\mathcal{M}^{\mathcal{L}}$ will be such that, for each atomic concept $A \in \Sigma^{\mathcal{L}}$ and each $B \in \mathsf{Sig}(\mathcal{O}) \cup \{\top, \bot\}$, we have $\mathcal{O} \models A \sqsubseteq B$ iff $\mathcal{M}^{\mathcal{L}} \models A \sqsubseteq B$.
2. It classifies $\mathcal{M}^{\mathcal{L}}$ using an $\mathcal{L}$-reasoner and feeds (in a compact way) the obtained (non-)subsumptions to a $\mathcal{SROIQ}$-reasoner, such as HermiT, that can effectively exploit this information [6].

Step 1 involves two important technical challenges. First, $\Sigma^{\mathcal{L}}$ should be as large as possible; in particular, for ontologies with only a few non-$\mathcal{L}$ axioms, it is reasonable to expect $\Sigma^{\mathcal{L}}$ to contain most of the ontology's signature. Second, $\mathcal{M}^{\mathcal{L}}$ must be *complete* for $\Sigma^{\mathcal{L}}$. Although techniques such as the completely defined concepts optimisation can be used to identify a complete fragment, these techniques are very restricted; thus, we exploit module extraction techniques [4, 7], which, in addition to giving completeness guarantees, are more generally applicable, more flexible, and more robust.

We believe that our results are interesting from both a theoretical and a practical point of view. We show that given a $\mathcal{SROIQ}$ ontology $\mathcal{O}$ that is not captured by any known polynomial fragment of $\mathcal{SROIQ}$, it is often possible to identify a large subset Σ of $\mathsf{Sig}(\mathcal{O})$ such that all subsumers of concepts in Σ w.r.t. $\mathcal{O}$ can be computed using a polynomial time classification algorithm. From

a practical point of view, our first experiments with a prototype implementation suggest the potential of this approach for optimising classification.

This paper is supplemented by an Appendix containing additional technical details.

2 Preliminaries

We adopt standard DL notation, as well as standard notions of signature, interpretations, entailment, satisfiability and subsumption. We also assume basic familiarity with the description logics $\mathcal{SROIQ}$ [11] and $\mathcal{EL}^{++}$ [1]. When talking about *ontologies* and *axioms* we will implicitly refer to $\mathcal{SROIQ}$-ontologies and $\mathcal{SROIQ}$-axioms, respectively.

We denote with $\mathsf{Sig}(\mathcal{O})$ (respectively, $\mathsf{Sig}(\alpha)$) the signature of an ontology $\mathcal{O}$ (respectively, of an axiom α). Furthermore, given an ontology $\mathcal{O}$ and a DL $\mathcal{L} \subseteq \mathcal{SROIQ}$, we denote with $\mathcal{O}_{\mathcal{L}}$ the subset of $\mathcal{L}$-axioms in $\mathcal{O}$.

2.1 Module Extraction

Intuitively, a module $\mathcal{M}$ for an ontology $\mathcal{O}$ w.r.t. a signature Σ is an ontology $\mathcal{M} \subseteq \mathcal{O}$ such that $\mathcal{M}$ entails the same axioms over Σ as $\mathcal{O}$.

This intuition is typically formalised using different notions of a *conservative extension* [16, 4]. In this paper, we define modules in terms of a model-theoretic notion of conservative extension.

Definition 1 (Model Conservative Extension). *Let $\mathcal{O}$ be an ontology and let $\Sigma \subseteq \mathsf{Sig}(\mathcal{O})$. We say that $\mathcal{O}$ is a* model conservative extension *of $\mathcal{M} \subseteq \mathcal{O}$ w.r.t. Σ if, for every model $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ of $\mathcal{M}$, there exists a model $\mathcal{J} = (\Delta^{\mathcal{J}}, \cdot^{\mathcal{J}})$ of $\mathcal{O}$ such that $\Delta^{\mathcal{I}} = \Delta^{\mathcal{J}}$ and $X^{\mathcal{I}} = X^{\mathcal{J}}$ for every symbol $X \in \Sigma$.*

Definition 2 (Module). *Let $\mathcal{O}$ be an ontology and let Σ be a signature. We say that $\mathcal{M} \subseteq \mathcal{O}$ is a module in $\mathcal{O}$ w.r.t. Σ if $\mathcal{O}$ is a model conservative extension of $\mathcal{M}$ w.r.t. Σ.*

In particular, if $\mathcal{M}$ is a module in $\mathcal{O}$ w.r.t. Σ, then the following condition holds: for each axiom α with $\mathsf{Sig}(\alpha) \subseteq \Sigma$, we have $\mathcal{M} \models \alpha$ iff $\mathcal{O} \models \alpha$.

The problem of checking whether $\mathcal{M}$ is a module in $\mathcal{O}$ w.r.t. Σ, however, is already undecidable for $\mathcal{EL}^{++}$ [17], so approximations are typically needed in practice. The following sufficient condition for model conservativity is known to work well in practice [4].

Definition 3 ($\emptyset$-locality). *Let Σ be a signature and let $\mathcal{O}$ be an ontology. An interpretation $\mathcal{I}$ is $\emptyset$-local for Σ if for every atomic concept $A \notin \Sigma$ and every atomic role $R \notin \Sigma$, we have $A^{\mathcal{I}} = R^{\mathcal{I}} = \emptyset$. An axiom α is $\emptyset$-local for Σ if $\mathcal{I} \models \alpha$ for each $\mathcal{I}$ that is $\emptyset$-local for Σ. An ontology $\mathcal{O}$ is $\emptyset$-local for Σ if every axiom in $\mathcal{O}$ is $\emptyset$-local for Σ.*

Checking $\emptyset$-locality for $\mathcal{SROIQ}$ axioms is, however, a PSPACE-complete problem [4]. Since our goal is to optimise classification, checking $\emptyset$-locality might still be too costly. Instead, we will use $\bot$-locality — a well-known sufficient syntactic condition for $\emptyset$-locality which has been successfully used for both ontology reuse and reasoning problems [4, 12, 18, 7].

The precise grammar defining $\bot$-locality for $\mathcal{SROIQ}$ is given for reference in the Appendix, and can also be found in the literature [7, 4]. It suffices to consider that, for each $\mathcal{O}$ and Σ, $\bot$-locality implies $\emptyset$-locality and it can be checked in polynomial time. Furthermore, the following property holds [7, 4]:

Proposition 1. *If an axiom α is $\bot$-local w.r.t. a signature Σ, then α is $\bot$-local w.r.t. Σ' for any $\Sigma' \subseteq \Sigma$.*

We can use $\bot$-locality to define the notion of a $\bot$-module. The aforementioned properties of $\bot$-locality ensure that, if $\mathcal{M}$ is a $\bot$-module w.r.t. Σ in $\mathcal{O}$ as defined next, then it is also a module w.r.t. Σ in $\mathcal{O}$.

Definition 4 ($\bot$-module). *An ontology $\mathcal{M} \subseteq \mathcal{O}$ is a $\bot$-module in $\mathcal{O}$ w.r.t. Σ if $\mathcal{O} \setminus \mathcal{M}$ is $\bot$-local for $\Sigma \cup \mathsf{Sig}(\mathcal{M})$.*

Clearly, there is a unique smallest $\bot$-module for a given $\mathcal{O}$ and Σ (the smallest subset $\mathcal{M} \subseteq \mathcal{O}$ s.t. $\mathcal{O} \setminus \mathcal{M}$ is $\bot$-local for $\Sigma \cup \mathsf{Sig}(\mathcal{M})$). In what follows, we refer to such smallest module as *the* $\bot$-module in $\mathcal{O}$ w.r.t. Σ and we denote it $\mathcal{M}_{[\mathcal{O},\Sigma]}$.

In addition to being modules as in Definition 2, $\bot$-modules also enjoy an additional property that makes them especially well-suited for optimising ontology classification [7].

Proposition 2. *Let $\mathcal{O}$ be an ontology, let A, B be concepts in $\mathsf{Sig}(\mathcal{O}) \cup \{\top, \bot\}$, let $\Sigma \subseteq \mathsf{Sig}(\mathcal{O})$ with $A \in \Sigma$, and let $\mathcal{M} \subseteq \mathcal{O}$ be a $\bot$-module in $\mathcal{O}$ w.r.t. Σ. Then $\mathcal{O} \models A \sqsubseteq B$ iff $\mathcal{M} \models A \sqsubseteq B$.*

2.2 Ontology Classification in HermiT

The reasoner HermiT implements a classification algorithm [6] that differs significantly from the standard Enhanced Traversal Algorithm [2] implemented in most other DL reasoners. The key feature of HermiT's classification algorithm that makes it especially well-suited for our purposes is that it exploits sets $\mathbf{K}$ and $\mathbf{P}$ of pairs $\langle A, B\rangle$ of atomic concepts representing *known subsumptions* and *possible subsumptions*, respectively. These sets are used to reduce the number of required tests during classification. Information about non-subsumptions is implicitly stored in these sets (as it would be too costly to store it explicitly), i.e., if $\mathbf{A} = \{\langle A, B\rangle \mid A, B$ are atomic concept names in $\mathsf{Sig}(\mathcal{O})\}$, then $\mathbf{A} \setminus (\mathbf{K} \cup \mathbf{P})$ is the set of known non-subsumptions.

The algorithm works in two clearly distinct phases. In the *initialisation phase*, sets $\mathbf{K}$ and $\mathbf{P}$ are given initial values using information obtained from satisfiability tests performed on atomic concepts. In the *classification phase*, $\mathbf{K}$ is

augmented with pairs from $\mathbf{P}$ until $\mathbf{K}$ contains all the entailed subsumptions and $\mathbf{P}$ is empty.

Additional technical details about HermiT's classification algorithm are provided in the Appendix.

3 Modular Classification of Ontologies

Given a $\mathcal{SROIQ}$ ontology $\mathcal{O}$ and a description logic $\mathcal{L} \subseteq \mathcal{SROIQ}$, our first goal is to identify a signature $\Sigma^{\mathcal{L}} \subseteq \mathsf{Sig}(\mathcal{O})$ such that $\mathcal{M}_{[\mathcal{O},\Sigma^{\mathcal{L}}]} \subseteq \mathcal{O}_{\mathcal{L}}$. We call any such subset of $\mathsf{Sig}(\mathcal{O})$ an *$\mathcal{L}$-signature for $\mathcal{O}$*. Section 3.1 addresses the problem of identifying as large an $\mathcal{L}$-signature as possible.

We can then use an $\mathcal{L}$-reasoner to compute from $\mathcal{M}_{[\mathcal{O},\Sigma^{\mathcal{L}}]}$ complete classification information about the atomic concepts in $\Sigma^{\mathcal{L}}$ —by Proposition 2, given any $A \in \Sigma^{\mathcal{L}}$ and $B \in \mathsf{Sig}(\mathcal{O}) \cup \{\top, \bot\}$ we have $\mathcal{O} \models A \sqsubseteq B$ iff $\mathcal{M}_{[\mathcal{O},\Sigma^{\mathcal{L}}]} \models A \sqsubseteq B$.

HermiT's classification algorithm needs to be slightly modified in order to exploit the information computed by the $\mathcal{L}$-reasoner. In section 3.2 we show how to adapt the initialisation phase to efficiently encode this information into $\mathbf{K}$ and $\mathbf{P}$. Additional technical information about our modification of HermiT's algorithm (including a proof of correctness) is given in the Appendix.

3.1 Computing an $\mathcal{L}$-signature

The definition of $\bot$-module immediately suggests a simple "guess and check" algorithm for computing a (maximal) $\mathcal{L}$-signature for $\mathcal{O}$: consider all subsets $\Sigma \subseteq \mathsf{Sig}(\mathcal{O})$ in decreasing size order and, for each of them, check whether $\mathcal{M}_{[\mathcal{O},\Sigma]}$ is an $\mathcal{L}$-ontology.

Our goal in practice, however, is to optimise classification; hence, we propose a more practical algorithm. Although our algorithm is not guaranteed to compute a maximal $\mathcal{L}$-signature, it can be implemented very efficiently and, as shown in the evaluation section, it typically computes large $\mathcal{L}$-signatures, provided that $\mathcal{O}_{\mathcal{L}}$ is a large enough fragment of $\mathcal{O}$.

We will exploit the fact that every $\mathcal{L}$-signature $\Sigma^{\mathcal{L}}$ *must* satisfy the following property $(\star)$. If $(\star)$ does not hold, then $\mathcal{M}_{[\mathcal{O},\Sigma^{\mathcal{L}}]}$ will contain some non $\mathcal{L}$-axiom.

$$\text{Property } (\star): \quad \mathcal{O} \setminus \mathcal{O}_{\mathcal{L}} \text{ is } \bot\text{-local w.r.t. } \Sigma^{\mathcal{L}}$$

Example 1. Consider $\mathcal{L} = \mathcal{EL}$ and the following ontology

$$\mathcal{O}^{\mathsf{ex}} = \{A \sqsubseteq B, \exists R.C \sqsubseteq D, E \sqsubseteq \forall S.A, \exists R.D \sqsubseteq \neg B\}$$

Note that the set of $\mathcal{L}$-axioms in $\mathcal{O}^{\mathsf{ex}}$ is $\mathcal{O}^{\mathsf{ex}}_{\mathcal{L}} = \{A \sqsubseteq B, \exists R.C \sqsubseteq D\}$. Furthermore, the signature of $\mathcal{O}^{\mathsf{ex}}_{\mathcal{L}}$, namely $\Sigma_1 = \{A, B, C, D, R\}$, is not an $\mathcal{L}$-signature for $\mathcal{O}^{\mathsf{ex}}$; indeed, the non $\mathcal{L}$-axiom $\exists R.D \sqsubseteq \neg B$ is not $\bot$-local w.r.t Σ_1.

In contrast, we have that $\mathcal{O}^{\mathsf{ex}} \setminus \mathcal{O}^{\mathsf{ex}}_{\mathcal{L}} = \{E \sqsubseteq \forall S.A, \exists R.D \sqsubseteq \neg B\}$ is $\bot$-local w.r.t. $\Sigma_2 = (\mathsf{Sig}(\mathcal{O}^{\mathsf{ex}}) \setminus \mathsf{Sig}(\mathcal{O}^{\mathsf{ex}}_{\mathcal{L}})) = \{C\}$. Furthermore, $\mathcal{M}_{[\mathcal{O}^{\mathsf{ex}},\Sigma_2]} = \emptyset$; hence, Σ_2 is an $\mathcal{L}$-signature for $\mathcal{O}^{\mathsf{ex}}$, and we can ensure that $\mathcal{O}^{\mathsf{ex}} \not\models C \sqsubseteq X$ for each atomic concept $X \in \mathsf{Sig}(\mathcal{O}^{\mathsf{ex}})$ different from C. $\diamondsuit$

Although Example 1 might suggest that property ($\star$) is also a *sufficient* condition for $\Sigma^{\mathcal{L}}$ to be an $\mathcal{L}$-signature in $\mathcal{O}$, this is unfortunately not the case.

Example 2. Consider $\Sigma_3 = \{A, C, D, R, S\}$; clearly, $\mathcal{O}^{\mathsf{ex}} \setminus \mathcal{O}^{\mathsf{ex}}_{\mathcal{L}}$ is $\bot$-local w.r.t Σ_3 and hence ($\star$) holds for Σ_3. However, Σ_3 is not an $\mathcal{L}$-signature for $\mathcal{O}^{\mathsf{ex}}$.

By Definition 4, each axiom in $\mathcal{O}^{\mathsf{ex}} \setminus \mathcal{M}_{[\mathcal{O}^{\mathsf{ex}}, \Sigma_3]}$ must be $\bot$-local w.r.t. signature $\Sigma_3 \cup \mathsf{Sig}(\mathcal{M}_{[\mathcal{O}^{\mathsf{ex}}, \Sigma_3]})$ (and not just w.r.t Σ_3). Axiom $\alpha = A \sqsubseteq B$ is not $\bot$-local w.r.t. Σ_3, so we have $\alpha \in \mathcal{M}_{[\mathcal{O}^{\mathsf{ex}}, \Sigma_3]}$. But then, we have $B \in \mathsf{Sig}(\mathcal{M}_{[\mathcal{O}^{\mathsf{ex}}, \Sigma_3]})$ and hence the non $\mathcal{L}$-axiom $\beta = \exists R.D \sqsubseteq \neg B$ is not $\bot$-local w.r.t. $\Sigma_3 \cup \mathsf{Sig}(\mathcal{M}_{[\mathcal{O}^{\mathsf{ex}}, \Sigma_3]})$.

We can address this problem by reducing Σ_3 to $\Sigma_4 = \Sigma_3 \setminus \{A\}$. The corresponding $\bot$-module for Σ_4 then becomes $\mathcal{M}_{[\mathcal{O}^{\mathsf{ex}}, \Sigma_4]} = \{\exists R.C \sqsubseteq D\}$, which is an $\mathcal{L}$-ontology; thus, Σ_4 is an $\mathcal{L}$-signature for $\mathcal{O}^{\mathsf{ex}}$. $\diamondsuit$

Example 2 suggests an algorithm for computing an $\mathcal{L}$-signature for $\mathcal{O}$, which can be intuitively described as follows.

1. Reduce $\Sigma_0 = \mathsf{Sig}(\mathcal{O})$ to a subset Σ_1 of Σ_0 such that $\mathcal{S}_0 = \mathcal{O} \setminus \mathcal{O}_{\mathcal{L}}$ is $\bot$-local w.r.t. Σ_1 (thus satisfying ($\star$)).
2. Compute the axioms $\mathcal{S}_1$ in $\mathcal{M}_{[\mathcal{O}, \Sigma_1]}$ containing symbols not in Σ_1.
3. Reduce Σ_1 to a subset Σ_2 of Σ_1 such that $\mathcal{S}_1$ is $\bot$-local w.r.t. Σ_2.
4. Repeat Steps [2-4] until the set of axioms computed in Step 2 is empty.

Note that there can be many ways to perform the signature reduction required in Steps 1 and 4. For instance, Σ_2 and Σ_3 in Examples 1 and 2 are both possible reductions of $\mathsf{Sig}(\mathcal{O}^{\mathsf{ex}})$ in Step 1. These acceptable reductions can be characterised using a function

$$\mathsf{localise} : \mathcal{P}(\mathsf{Sig}(\mathcal{O})) \times \mathcal{P}(\mathcal{O}) \to \mathcal{P}(\mathsf{Sig}(\mathcal{O}))$$

such that, given $\Sigma \in \mathcal{P}(\mathsf{Sig}(\mathcal{O}))$ and $\mathcal{S} \in \mathcal{P}(\mathcal{O})$ not $\bot$-local w.r.t. Σ, $\mathsf{localise}(\Sigma, \mathcal{S})$ returns

- Σ if $\mathcal{S} = \emptyset$.
- a subset $\Sigma' \subset \Sigma$ such that every axiom in $\mathcal{S}$ is $\bot$-local w.r.t. Σ' if $\mathcal{S} \neq \emptyset$ and Σ' exists.
- $\emptyset$ otherwise.

Given a particular $\mathsf{localise}$ function, Algorithm 1 accepts a $\mathcal{SROIQ}$ ontology $\mathcal{O}$ and returns either the pair $\langle \mathbf{false}, \emptyset \rangle$ or a pair $\langle \mathbf{true}, \Sigma^{\mathcal{L}} \rangle$ with $\Sigma^{\mathcal{L}} \subseteq \mathsf{Sig}(\mathcal{O})$ an $\mathcal{L}$-signature for $\mathcal{O}^{\mathsf{ex}}$. Termination and correctness are granted by Theorem 1.

Theorem 1. *Let $\mathcal{S}_i$, Σ_i ($i \geq 0$) be defined by the following construction:*

$$\begin{array}{lll} (i = 0): & \Sigma_0 = \mathsf{Sig}(\mathcal{O}) & \mathcal{S}_0 = \mathcal{O} \setminus \mathcal{O}_{\mathcal{L}} \\ (i \geq 1): & \Sigma_i = \mathsf{localise}(\Sigma_{i-1}, \mathcal{S}_{i-1}) & \mathcal{S}_i = \{\alpha \in \mathcal{M}_{[\mathcal{O}, \Sigma_i]} \mid \mathsf{Sig}(\alpha) \not\subseteq \Sigma_i\} \end{array}$$

Let $\Sigma^{\mathcal{L}} := \bigcap_{i \geq 0} \Sigma_i$. Then, the following properties hold:

1. *There exists $k < |\mathsf{Sig}(\mathcal{O})|$ such that either $\Sigma_k = \emptyset$ or $\mathcal{S}_k = \emptyset$.*
2. *Either $\Sigma^{\mathcal{L}} = \emptyset$ or $\mathcal{M}_{[\mathcal{O}, \Sigma^{\mathcal{L}}]} \subseteq \mathcal{O}_{\mathcal{L}}$.*

Algorithm 1 $\mathcal{L}$-signature($\mathcal{O}$)

Input: a $\mathcal{SROIQ}$ ontology $\mathcal{O}$

```
Σ := Sig(O)
S := O \ O_L
canLocalise = true
while S ≠ ∅ and canLocalise do
    Σ := localise(Σ, S)
    if Σ = ∅ then
        canLocalise := false
    else
        S := {α ∈ M_[O,Σ] | Sig(α) ⊈ Σ}
return ⟨canLocalise, Σ⟩
```

Proof. We first show Claim 1. Suppose $\Sigma_i \neq \emptyset$ for each $i \geq 0$. A straightforward inductive argument would show that $\Sigma_j \subseteq \Sigma_i$ for each $j > i \geq 0$. Furthermore, $\Sigma_0 = \mathsf{Sig}(\mathcal{O})$, so it cannot be the case that $\Sigma_j \subset \Sigma_i$ for each $0 \leq i < j \leq |\mathsf{Sig}(\mathcal{O})|$. Therefore, there must be some $k < |\mathsf{Sig}(\mathcal{O})|$ such that $\Sigma_{k+1} = \Sigma_k$; by the definition of $\mathsf{localise}$, this implies that $\mathcal{S}_k = \emptyset$.

We finally show Claim 2. Suppose $\Sigma^{\mathcal{L}} \neq \emptyset$. It is enough to prove that each $\alpha \in \mathcal{O} \setminus \mathcal{O}_{\mathcal{L}}$ is $\bot$-local w.r.t. $\Sigma^{\mathcal{L}} \cup \mathsf{Sig}(\mathcal{M}_{[\mathcal{O},\Sigma^{\mathcal{L}}]})$.

First, we are going to see that $\mathsf{Sig}(\mathcal{M}_{[\mathcal{O},\Sigma^{\mathcal{L}}]}) \subseteq \Sigma^{\mathcal{L}}$. According to Claim 1, there exists $k < |\mathsf{Sig}(\mathcal{O})|$ such that $\mathcal{S}_k = \emptyset$. This implies that, for each axiom $\alpha \in \mathcal{M}_{[\mathcal{O},\Sigma_k]}$, we have $\mathsf{Sig}(\alpha) \subseteq \Sigma_k$. It is easy to see that $\mathcal{S}_k = \emptyset$ also implies that $\Sigma_j = \Sigma_k$ for each $j > k$. Together with the fact that $\Sigma_j \subseteq \Sigma_i$ for each $j > i \geq 0$, this implies $\Sigma^{\mathcal{L}} = \bigcap_{i \geq 0} \Sigma_i = \Sigma_k$. But then for each $\alpha \in \mathcal{M}_{[\mathcal{O},\Sigma^{\mathcal{L}}]} = \mathcal{M}_{[\mathcal{O},\Sigma_k]}$ we have $\mathsf{Sig}(\alpha) \subseteq \Sigma_k = \Sigma^{\mathcal{L}}$, and so $\mathsf{Sig}(\mathcal{M}_{[\mathcal{O},\Sigma^{\mathcal{L}}]}) \subseteq \Sigma^{\mathcal{L}}$.

Now we can just prove that each $\alpha \in \mathcal{O} \setminus \mathcal{O}_{\mathcal{L}}$ is $\bot$-local w.r.t. $\Sigma^{\mathcal{L}}$. Because $\Sigma^{\mathcal{L}} = \bigcap_{i \geq 0} \Sigma_i \neq \emptyset$, in particular it must be the case that $\Sigma_0 \neq \emptyset$. By definition of $\mathsf{localise}$, either $\mathcal{O} \setminus \mathcal{O}_{\mathcal{L}} = \emptyset$ —in which case it is immediate that $\mathcal{M}_{[\mathcal{O},\Sigma^{\mathcal{L}}]} \subseteq \mathcal{O}_{\mathcal{L}}$— or every axiom in $\mathcal{S}_0 = \mathcal{O} \setminus \mathcal{O}_{\mathcal{L}}$ is $\bot$-local w.r.t. $\Sigma_1 = \mathsf{localise}(\Sigma_0, \mathcal{S}_0)$. Then, by Proposition 1, each $\alpha \in \mathcal{O} \setminus \mathcal{O}_{\mathcal{L}}$ is $\bot$-local w.r.t. $\Sigma^{\mathcal{L}} \subseteq \Sigma_1$. □

In practice, it is more convenient to use the $\mathcal{L}$-reasoner to classify $\mathcal{O}_{\mathcal{L}}$, instead of $\mathcal{M}_{[\mathcal{O},\Sigma^{\mathcal{L}}]}$. Once $\Sigma^{\mathcal{L}}$ has been computed, the following proposition shows that $\mathcal{O}_{\mathcal{L}}$ provides as much information as $\mathcal{M}_{[\mathcal{O},\Sigma^{\mathcal{L}}]}$ about the classification of $\mathcal{O}$. Furthermore, in general $\mathcal{M}_{[\mathcal{O},\Sigma^{\mathcal{L}}]} \subset \mathcal{O}_{\mathcal{L}}$ so additional subsumption relationships might be obtained by classifying $\mathcal{O}_{\mathcal{L}}$.

Proposition 3. *Let $\Sigma^{\mathcal{L}}$ be an $\mathcal{L}$-signature for an ontology $\mathcal{O}$. Then for each atomic concept $A \in \Sigma^{\mathcal{L}}$ and each $B \in \mathsf{Sig}(\mathcal{O}) \cup \{\top, \bot\}$ we have*

$$\mathcal{O} \models A \sqsubseteq B \text{ iff } \mathcal{O}_{\mathcal{L}} \models A \sqsubseteq B$$

Proof. Consider an atomic concept $A \in \Sigma^{\mathcal{L}}$ and $B \in \mathsf{Sig}(\mathcal{O}) \cup \{\top, \bot\}$. By monotonicity, because $\mathcal{O}_{\mathcal{L}} \subseteq \mathcal{O}$, we know that

$$\mathcal{O} \not\models A \sqsubseteq B \text{ implies } \mathcal{O}_{\mathcal{L}} \not\models A \sqsubseteq B$$

Algorithm 2 $\mathcal{L}$-ModularClassification($\mathcal{O}$)

Input: a $\mathcal{SROIQ}$ ontology $\mathcal{O}$

1: $\mathcal{O}_\mathcal{L} := \{\alpha \in \mathcal{O} \mid \alpha \text{ is an } \mathcal{L}\text{-axiom}\}$
2: $\Sigma^\mathcal{L} := \mathcal{L}\text{-signature}(\mathcal{O})$ ▷ See Algorithm 1
3: $H_{\mathcal{O}_\mathcal{L}} := \mathcal{L}\text{-classification}(\mathcal{O}_\mathcal{L})$
4: $H :=$ HermiTclassification$(\mathcal{O}_\mathcal{L}, H_{\mathcal{O}_\mathcal{L}}, \Sigma^\mathcal{L})$ ▷ See Section 3.2 and Appendix
5: **return** H

By monotonicity, because $\mathcal{M}_{[\mathcal{O},\Sigma^\mathcal{L}]} \subseteq \mathcal{O}_\mathcal{L}$ (by Theorem 1), it is the case that $\mathcal{M}_{[\mathcal{O},\Sigma^\mathcal{L}]} \models A \sqsubseteq B$ implies $\mathcal{O}_\mathcal{L} \models A \sqsubseteq B$. Now $\mathcal{M}_{[\mathcal{O},\Sigma^\mathcal{L}]}$ is a $\bot$-module in $\mathcal{O}$ w.r.t. $\Sigma^\mathcal{L}$, so by Proposition 2, $\mathcal{O} \models A \sqsubseteq B$ implies $\mathcal{M}_{[\mathcal{O},\Sigma^\mathcal{L}]} \models A \sqsubseteq B$, and

$$\mathcal{O} \models A \sqsubseteq B \text{ implies } \mathcal{O}_\mathcal{L} \models A \sqsubseteq B$$

Therefore, for each atomic concept $A \in \Sigma^\mathcal{L}$ and $B \in \mathsf{Sig}(\mathcal{O}) \cup \{\top, \bot\}$ we have $\mathcal{O} \models A \sqsubseteq B$ if and only if $\mathcal{O}_\mathcal{L} \models A \sqsubseteq B$, as required. □

3.2 Adapting HermiT's Initialisation Phase

As mentioned in Section 2.2, HermiT's classification algorithm works with (disjoint) sets $\mathbf{K}$ and $\mathbf{P}$ of known and possible subsumptions, respectively. We next discuss how we can use the information extracted from $\mathcal{O}_\mathcal{L}$ by the $\mathcal{L}$-reasoner in the initialisation of $\mathbf{K}$ and $\mathbf{P}$.

Let $\mathbf{K}' = \{\langle A, B\rangle \in \mathsf{Sig}(\mathcal{O}) \times (\mathsf{Sig}(\mathcal{O}) \cup \{\top, \bot\}) \mid \mathcal{O}_\mathcal{L} \models A \sqsubseteq B\}$ be the positive subsumptions extracted from $\mathcal{O}_\mathcal{L}$ by the $\mathcal{L}$-reasoner. We can clearly complement the initialisation of $\mathbf{K}$ by simply adding $\mathbf{K}'$ to $\mathbf{K}$.

To improve the initialisation of $\mathbf{P}$, we can simply make sure that no pair $\langle A, B\rangle \in \Sigma^\mathcal{L} \times \mathsf{Sig}(\mathcal{O})$ is ever added to $\mathbf{P}$. Indeed, by Proposition 3, if $\mathcal{O} \models A \sqsubseteq B$ then $\langle A, B\rangle$ must already be in $\mathbf{K}'$; otherwise, we must have $\mathcal{O} \not\models A \sqsubseteq B$ and there is no need to consider the pair $\langle A, B\rangle$ as a possible subsumption.

We include in the Appendix a slightly modified version of the intialisation algorithm in HermiT that is capable of exploiting the information extracted from $\mathcal{O}_\mathcal{L}$ by the $\mathcal{L}$-reasoner in the way just explained.

Algorithm 2 describes, at an abstract level, how the entire classification process can be performed with our modular technique for a particular $\mathcal{L} \subseteq \mathcal{SROIQ}$ and a particular function localise.

4 Implementation and Experiments

We have implemented our algorithms in Java using the OWL API.[1] Our implementation of the localise function is based on the locality module extractor described in [12], which is publicly available.[2]

[1] `http://owlapi.sourceforge.net/`
[2] `http://www.cs.ox.ac.uk/isg/tools/ModuleExtractor`

Table 1. Test ontologies.

	Number of axioms		Signature	
Ontology	Total	$\mathcal{EL}^{++}$	Size	Concepts
SNOMED$^{\sqcup}$	582,364	582,362	291,207	291,145
NCI	219,224	219,159	91,497	91,225
FMA-SNOMED	385,146	385,142	159,415	159,328

Table 2. $\mathcal{L}$-signature and classification times for $\mathcal{L} = \mathcal{EL}^{++}$.

	$\Sigma^{\mathcal{L}}$			Classification time(s)	
Ontology	Size	Concepts	Time (s)	HermiT	Modular
SNOMED$^{\sqcup}$	280,985 (96%)	280,923	15.3	2,016.5	189.9
NCI	85,411 (93%)	85,139	7.6	74.9	32.0
FMA-SNOMED	33,124 (21%)	33,046	14.3	876.5	790.6

In the implementation of localise, symbols required to make a set of axioms $\bot$-local are selected greedily axiom by axiom. When selecting symbols, we rely on heuristics that try to keep as many roles as possible within $\Sigma^{\mathcal{L}}$. This is because ontologies contain many more concepts than roles, and each role typically occurs in a large number of axioms; thus, having a role outside $\Sigma^{\mathcal{L}}$ is likely to cause many other symbols to be left outside $\Sigma^{\mathcal{L}}$.

In our experiments, we have used the ontologies given in Table 1:

- SNOMED$^{\sqcup}$ is a modification of the well-known SNOMED ontology (v. January 2010), where two axioms containing disjunction have been added (using feedback obtained from SNOMED's developers).
- NCI is the latest version of the National Cancer Institute Thesaurus. This ontology contains 65 non $\mathcal{EL}^{++}$ axioms.
- FMA-SNOMED is the ontology obtained from the integration of (a fragment of) the Foundational Model of Anatomy (FMA) and (a fragment of) SNOMED using ontology mappings [13]. In this case, all the non $\mathcal{EL}^{++}$ axioms come from FMA.

Our results are summarised in Table 2. The first two columns in the table provide the total size and number of concepts in the $\mathcal{EL}^{++}$-signature. The third column indicates the time required to compute the $\mathcal{EL}^{++}$-signature using the algorithm described in Section 3.1. Finally, the last two columns provide the total classification time using (the latest version of) HermiT, and the classification time required to complete the classification of $\mathcal{O}_{\mathcal{L}}$ as described in Section 3.2. For convenience of implementation, we have also classified $\mathcal{O}_{\mathcal{L}}$ using HermiT (and this time has not been included in the table); however, the reasoner ELK can classify $\mathcal{O}_{\mathcal{L}}$ in all cases in just a few seconds (e.g., ELK can classify SNOMED using concurrent classification techniques in about 5 seconds [15]).

We can observe that 96% of the symbols in SNOMED$^{\sqcup}$ (and 93% of the symbols in NCI) are included in the $\mathcal{EL}^{++}$-signature; thus, all subsumers of concepts

in this signature can be completely determined using an $\mathcal{EL}^{++}$-reasoner. Note, however, that the size of the $\mathcal{EL}^{++}$-signature for FMA-SNOMED is comparatively much smaller. This is due to the structure of FMA, which contains several non $\mathcal{EL}^{++}$ axioms about roles that are widely used in the ontology. For example, the domain of the role hasMass is defined as a disjunction of very general concepts, such as MaterialThing; since role hasMass is outside the $\mathcal{EL}^{++}$-signature, so will be MaterialThing (and, as a consequence, also the many concepts subsumed by MaterialThing).

Finally, concerning classification times, our results suggest the potential of our techniques. Improvements are especially substantial for both SNOMED$^{\sqcup}$ and NCI, where the $\mathcal{EL}^{++}$-signature is very large.

5 Conclusion and Future Work

In this paper, we have proposed a technique for classifying a $\mathcal{SROIQ}$ ontology $\mathcal{O}$ by exploiting a reasoner for a fragment $\mathcal{L}$ of $\mathcal{SROIQ}$. Our technique allows us to show that the subsumers of many concepts in $\mathcal{O}$ can be completely determined using only the $\mathcal{L}$-reasoner. Although our implementation is still at a very prototypical stage, our preliminary experiments show the potential of our approach in practice.

Our work is only very preliminary, and there are many interesting possibilities for future work.

- Our heuristics for computing an $\mathcal{L}$-signature $\Sigma^{\mathcal{L}}$ are rather naive and there is plenty of room for improvement. For example, it might be possible to explore modular decomposition techniques to compute larger $\mathcal{L}$-signatures [22].
- HermiT's initialisation phase could be further improved to make better use of the information obtained from the $\mathcal{L}$-reasoner.
- We are using $\bot$-modules, which provide very strong preservation guarantees (they preserve even *models*). It would be interesting to devise novel techniques for extracting modules that are more "permissive", in the sense that they only provide preservation guarantees for atomic subsumptions.
- Our technique could also be applied to a different notion of locality, as long as it satisfied a result analogous to Proposition 2.
- It would be interesting to explore ontology rewriting techniques that complement module extraction. For example, we could rewrite $\mathcal{O}$ into an $\mathcal{L}$-ontology $\mathcal{O}'$ such that $\mathcal{O}' \models \mathcal{O}$, in which case the classification of $\mathcal{O}'$ would provide an "upper bound" to the classification of $\mathcal{O}$.

Acknowledgements. This work was supported by the Royal Society, the EU FP7 project SEALS and the EPSRC projects ConDOR, ExODA, and LogMap.

References

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

2. F. Baader, E. Franconi, B. Hollunder, B. Nebel, and H.J. Profitlich. An empirical analysis of optimization techniques for terminological representation systems. *Applied Intelligence*, 4(2):109–132, 1994.
3. F. Baader, C. Lutz, and B. Suntisrivaraporn. CEL - a polynomial-time reasoner for life science ontologies. In *Proc. of IJCAR*, pages 287–291, 2006.
4. B. Cuenca Grau, I. Horrocks, Y. Kazakov, and U. Sattler. Modular reuse of ontologies: Theory and practice. *JAIR*, 31:273–318, 2008.
5. B. Cuenca Grau, I. Horrocks, B. Motik, B. Parsia, P. F. Patel-Schneider, and U. Sattler. OWL 2: The next step for OWL. *J. Web Semantics (JWS)*, 6(4):309–322, 2008.
6. B. Glimm, I. Horrocks, B. Motik, R. Shearer, and G. Stoilos. A novel approach to ontology classification. *J. of Web Semantics*, 10(1), 2011.
7. B. Cuenca Grau, C. Halaschek-Wiener, Y. Kazakov, and B. Suntisrivaraporn. Incremental classification of description logics ontologies. *JAR*, 44(4):337–369, 2010.
8. V. Haarslev and R. Möller. High performance reasoning with very large knowledge bases: A practical case study. In *Proc. IJCAI*, pages 161–168, 2001.
9. V. Haarslev and R. Möller. Racer system description. In *Proc. of IJCAR*, pages 701–705, 2001.
10. I. Horrocks. Implementation and optimisation techniques. In *The Description Logic Handbook: Theory, Implementation, and Applications*, chapter 9, pages 306–346. 2003.
11. I. Horrocks, O. Kutz, and U. Sattler. The even more irresistible $\mathcal{SROIQ}$. In *Proc. of KR*, pages 57–67, 2006.
12. E. Jimenez-Ruiz, B. Cuenca Grau, T. Schneider, U. Sattler, and R. Berlanga. Safe and economic re-use of ontologies: a logic-based methodology and tool support. In *Proc. of ESWC*, 2008.
13. E. Jiménez-Ruiz and B. Cuenca Grau. Logmap: Logic-based and scalable ontology matching. In *Proc. of ISWC*, 2011.
14. Y. Kazakov. $\mathcal{RIQ}$ and $\mathcal{SROIQ}$ are harder than $\mathcal{SHOIQ}$. In *Proc. of KR*, pages 274–284, 2008.
15. Y. Kazakov, M. Krötzsch, and F. Simančík. Concurrent classification of $\mathcal{EL}$ ontologies. In *Proc. of ISWC*, volume 7032, 2011.
16. C. Lutz, D. Walther, and F. Wolter. Conservative extensions in expressive description logics. In *Proc. of IJCAI*, pages 453–458, 2007.
17. C. Lutz and F. Wolter. Conservative extensions in the lightweight description logic $\mathcal{EL}$. In *Proc. of CADE-21*, volume 4603, 2007.
18. U. Sattler, T. Schneider, and M. Zakharyaschev. Which kind of module should I extract? In *Proc. of DL*, 2009.
19. E. Sirin, B. Parsia, B. Cuenca Grau, A. Kalyanpur, and Y. Katz. Pellet: A practical OWL DL reasoner. *J. of Web Semantics*, 5(2):51–53, 2007.
20. D. Tsarkov and I. Horrocks. FaCT++ description logic reasoner: System description. In *Proc. of IJCAR*, volume 4130, pages 292–297, 2006.
21. D. Tsarkov, I. Horrocks, and P.F. Patel-Schneider. Optimizing terminological reasoning for expressive description logics. *JAR*, 39(3):277–316, 2007.
22. C. Del Vescovo, B. Parsia, U. Sattler, and T. Schneider. The modular structure of an ontology: Atomic decomposition. In *Proc. of IJCAI*, pages 2232–2237, 2011.

A Definition of $\bot$-locality for $\mathcal{SROIQ}$

Definition 5 ($\bot$-locality for $\mathcal{SROIQ}$). *Let Σ be a signature and consider the sets $\boldsymbol{Bot}(\Sigma)$ and $\boldsymbol{Top}(\Sigma)$ syntactically defined as follows:*

$$\begin{aligned}\boldsymbol{Bot}(\Sigma) ::= {} & A^{\bot} \mid \neg C^{\top} \mid C^{\bot} \sqcap C \mid C \sqcap C^{\bot} \mid \exists R^{\bot}.C \mid \exists R.C^{\bot} \mid \\ & \geq nS^{\bot}.C \mid \geq nS.C^{\bot} \mid \exists S^{\bot}.\mathsf{Self}\end{aligned}$$

$$\boldsymbol{Top}(\Sigma) ::= \top \mid \neg C^{\bot} \mid C_1^{\top} \sqcap C_2^{\top} \mid \geq 0R.C$$

With $n \geq 1$, $A^{\bot}$ an atomic concept outside Σ, $R^{\bot}$ an atomic role outisde Σ or the inverse of such an atomic role, $S^{\bot}$ a simple atomic role outside Σ or the inverse of such an atomic role, $C^{\bot} \in \boldsymbol{Bot}(\Sigma)$, $C^{\top}, C_1^{\top}, C_2^{\top} \in \boldsymbol{Top}(\Sigma)$, $C^{\bot} \in \boldsymbol{Bot}(\Sigma)$, and C, R, S any concept, role and simple role, respectively.

An axiom α is $\bot$-local w.r.t. Σ if it has one of the following forms: $C^{\bot} \sqsubseteq C$, $C \sqsubseteq C^{\top}$, $\mathsf{Dis}(S, S^{\bot})$, $\mathsf{Dis}(S^{\bot}, S)$ or $w^{\bot} \sqsubseteq R$, with $w^{\bot}$ being a chain $R_1 \ldots R_n$ of roles such that some R_i is an atomic role not in Σ or the inverse of such a role.

B Modified Initialisation Algorithm for K and P

We have added some modifications to the initialisation algorithm presented in [6] to encode the subsumption information computed from $\mathcal{O}_{\mathcal{L}}$ by the L-reasoner into $\mathbf{K}$ and $\mathbf{P}$. The result is Algorithm 3. There are two additional arguments to the function that the algorithm computes, namely an $\mathcal{L}$-signature, $\Sigma^{\mathcal{L}}$, and the set $\mathbf{K}'$ of all pairs $\langle A, B\rangle \in \mathsf{Sig}(\mathcal{O}) \times (\mathsf{Sig}(\mathcal{O}) \cup \{\top, \bot\})$ such that $\mathcal{O}_{\mathcal{L}} \models A \sqsubseteq B$. There are also modifications in lines 4, 12 and 16. We refer the reader to [6] for definitions of the auxiliary functions used in the algorithm.

Proposition 4 grants correctness of the general classification algorithm given in [6] if Algorithm 3 is used to initialise $\mathbf{K}$ and $\mathbf{P}$. Let $\rightsquigarrow_{\mathbf{K}}$ be the reflexive and transitive closure of the relation induced by the set of pairs $\mathbf{K}$.

Proposition 4. *When applied to a satisfiable ontology $\mathcal{O}$, a set of atomic concepts $S \subseteq \mathsf{Sig}(\mathcal{O})$, an $\mathcal{L}$-signature $\Sigma^{\mathcal{L}}$ and the set*

$$\mathbf{K}' = \{\langle A, B\rangle \in \mathsf{Sig}(\mathcal{O}) \times (\mathsf{Sig}(\mathcal{O}) \cup \{\top, \bot\}) \mid \mathcal{O}_{\mathcal{L}} \models A \sqsubseteq B\}$$

Algorithm 3 terminates. Let $\mathbf{K}$ and $\mathbf{P}$ be the relations produced by the algorithm; then, for all atomic concepts $A, B \in S$, the following properties hold:

1. *$A \rightsquigarrow_{\mathbf{K}} B$ implies $\mathcal{O} \models A \sqsubseteq B$*
2. *If A is unsatisfiable then $A \rightsquigarrow_{\mathbf{K}} \bot$.*
3. *If A is satisfiable and $\mathcal{O} \models A \sqsubseteq B$ then either $A \rightsquigarrow_{\mathbf{K}} B$ or a class A' exists such that $A \rightsquigarrow_{\mathbf{K}} A'$, $\langle A', B\rangle \in \mathbf{P}$ and $\mathcal{O} \models A' \sqsubseteq B$.*

Proof. This proof follows closely and refers to the proof for the analogous Proposition in [6].

Algorithm 3 $\mathsf{InitialiseRelations}(\mathcal{O}, S, \underline{\Sigma^{\mathcal{L}}}, \mathbf{K}')$

Input: an ontology $\mathcal{O}$, a set S of classes to be classified, <u>an $\mathcal{L}$-signature $\Sigma^{\mathcal{L}}$ and the set $\mathbf{K}'$ of all subsumption relationships entailed by $\mathcal{O}_{\mathcal{L}}$</u>

1: $\mathbf{K} := \mathbf{K}' \cup \mathsf{explicit}(\mathcal{O})$
2: $(V, H, \rho) := \mathsf{hierarchy}(S, \mathbf{K}, \bot, \top)$
3: $\mathbf{P} := \emptyset$
4: $\mathsf{ToTest} := \{C \mid \langle \bot, C\rangle \in H\}\underline{\setminus \Sigma^{\mathcal{L}}}$
5: **while** $\mathsf{ToTest} \neq \emptyset$ **do**
6: choose and remove C from ToTest
7: **if** $\mathbf{P}|_C = \emptyset$ and $\langle C, \bot\rangle \notin K$
8: $\mathcal{A} := \mathsf{buildPreModel}(C(s_0), \emptyset, \mathcal{O})$ // s_0 is fresh
9: **if** $\mathsf{unsatisfiable} \in \mathcal{A}$ **then** // C is unsatisfiable
10: **for all** D such that $D \rightsquigarrow_H C$ and $\langle D, \bot\rangle \notin \mathbf{K}$ **do**
11: add $\langle D, \bot\rangle$ to $\mathbf{K}$ and remove D from ToTest
12: **for all** $\langle D, E\rangle \in H$ with $\langle E, \bot\rangle \notin K$ <u>and $E \notin \Sigma^{\mathcal{L}}$</u> **do**
13: add E to ToTest
14: **else**
15: add $\langle C, D\rangle$ to $\mathbf{K}$ for each $D \in \mathsf{knownSubsumers}(S, s_0, \mathcal{A}\}$
16: **for all** s in $\mathcal{A}$, all $D \in \mathcal{L}_{\mathcal{A}}(s)$, and all $E\underline{\notin \Sigma^{\mathcal{L}}}$ such that $D \rightsquigarrow_H E$ **do**
17: **if** $\mathbf{P}|_E = \emptyset$ **then**
18: $\mathbf{P} := \mathbf{P} \cup \{\langle E, F\rangle \mid F \in \mathsf{possibleSubsumers}(S, \mathcal{O}, E, s, \mathcal{A}, \mathbf{K})\}$
19: **else**
20: $\mathbf{P} := \mathbf{P} \setminus \{\langle E, F\rangle \mid F \notin \mathsf{possibleSubsumers}(S, \mathcal{O}, E, s, \mathcal{A}, \mathbf{K})\}$
21: remove each $\langle E_1, E_2\rangle$ from $\mathbf{P}$ such that $E_1 \rightsquigarrow_{\mathbf{K}} E_2$
22: **return**$(\mathbf{K}, \mathbf{P})$

The argument for termination in the original proof still holds. The same happens for Claim 1.

To prove Claims 2 and 3 we use the same auxiliary result ♦ used in [6] and a variation of ♣: let H be the hierarchy constructed in line 2 and A an arbitrary concept name occurring in H; then the following properties hold at the beginning of each iteration of the while-loop.

- (♦) If A is satisfiable and $\mathbf{P}|_A \neq \emptyset$, then, for each concept $B \in S$ such that $\mathcal{O} \models A \sqsubseteq B$, we have $B \in \mathbf{P}|_A$.
- (♣) If $\mathbf{P}|_A = \emptyset$, $A \not\rightsquigarrow_{\mathbf{K}} \bot$ and $A \notin \Sigma^{\mathcal{L}}$, then there exists some atomic concept $F \in \mathsf{ToTest}$ such that $F \not\rightsquigarrow_{\mathbf{K}} \bot$, $F \rightsquigarrow_H A$ and $\mathbf{P}|_G = \emptyset$ for each atomic concept G such that $F \rightsquigarrow_H G$ and $G \rightsquigarrow_H A$.

with $\mathbf{P}|_A = \{B \mid \langle A, B\rangle \in \mathbf{P}\}$ for each $A \in S$.

The argument used to prove (♦) in [6] works here as well.

We are going to apply induction on the iterations of the while-loop to show that any atomic concept A occurring in H satisfies (♣).

Base Case: At the beginning of the first iteration, ToTest contains all atomic concepts of H 'above' $\bot$; hence, for an arbitrary atomic concept $F \in \mathsf{ToTest}$, we have $F \not\rightsquigarrow_H \bot$. Therefore, if $\mathbf{P}|_A = \emptyset$, $A \not\rightsquigarrow_H \bot$ and $A \notin \Sigma^{\mathcal{L}}$, then property (♣) is satisfied for $F = A$.

Induction Step: Assume that property (♣) holds for A at the beginning of iteration i. We show that (♣) also holds for A at the end of iteration $i+1$. The claim is nontrivial only if $\mathbf{P}|_A = \emptyset$, $A \not\rightsquigarrow_{\mathbf{K}} \bot$ and $A \notin \Sigma^{\mathcal{L}}$. Since A satisfies the induction hypothesis, there is some atomic concept $F \in \mathsf{ToTest}$ such that $F \not\rightsquigarrow_{\mathbf{K}} \bot$, $F \rightsquigarrow_H A$ and $\mathbf{P}|_G = \emptyset$ for each atomic concept G such that $F \rightsquigarrow_H G$ and $G \rightsquigarrow_H A$. Let C be an arbitrary concept chosen in line 6. If C does not satisfy the condition in line 7, then $\mathbf{P}|_C \neq \emptyset$ or $C \rightsquigarrow_{\mathbf{K}} \bot$, so $C \neq F$ and thus F satisfies property (♣) for A at the end of the iteration. If C satisfies the condition in line 7, we have two possibilities.

First, assume that C is satisfiable, and let $\mathcal{A}$ be the premodel obtained in line 8. For an arbitrary atomic concept D, if $\mathbf{P}|_D = \emptyset$ at the beginning but not at the end of the loop, then by the condition in line 16 we have $\mathbf{P}_E \neq \emptyset$ at the end of the loop for each concept E such that $D \rightsquigarrow_H E$ and $E \notin \Sigma^{\mathcal{L}}$. Consequently, if $\mathbf{P}|_G \neq \emptyset$ at the end of the loop for some atomic concept G such that $F \rightsquigarrow_H G$ and $G \rightsquigarrow_H A$, then $\mathbf{P}|_A \neq \emptyset$ at the end of the loop as well (since $A \notin \Sigma^{\mathcal{L}}$), so property (♣) is satisfied for A at the end of the iteration. Otherwise, since lines 14-20 never add a pair of the form $\langle F, \bot\rangle$ to $\mathbf{K}$, concept F satisfies satisfies property (♣) for A at the end of the iteration.

Second, assume that C is unsatisfiable; then, property (♣) can be affected only if $F \rightsquigarrow_H C$. To summarise, we have $F \rightsquigarrow_H C$ and $F \rightsquigarrow_H A$, where H is a directed acyclic relation; but then a 'highest' atomic concept D in H exists that occurs on the path from F to C and on the path from F to A. More formally, there exists D such that

- $F \rightsquigarrow_H D$,
- $D \rightsquigarrow_H A$,
- $D \rightsquigarrow_H C$, and
- for each concept D' different from D such that $D \rightsquigarrow_H D'$ and $D' \rightsquigarrow_H A$, we have $D' \not\rightsquigarrow_H C$.

Furthermore, since $F \not\rightsquigarrow_H \bot$, we also have $D \not\rightsquigarrow_{\mathbf{K}} \bot$. Class D will clearly eventually be considered in line 10. If $D = A$, then $\langle A, \bot\rangle$ is added to $\mathbf{K}$ in line 11, so A trivially satisfies property (♣) at the end of the iteration. If $D \neq A$, a class E exists such that $\langle D, E\rangle \in H$ and $E \rightsquigarrow_H A$; since $A \not\rightsquigarrow_{\mathbf{K}} \bot$, for each such E we have $E \not\rightsquigarrow_{\mathbf{K}} \bot$; furthermore, by property (♣), we have $\mathbf{P}|_G = \emptyset$ for each class G such that $E \rightsquigarrow_H G$ and $G \rightsquigarrow_H A$. By definition of the $\mathsf{hierarchy}$ and $\mathsf{explicit}$ functions (see [6]), it is immediate that $E \rightsquigarrow_H A$ implies $\mathcal{O} \models E \sqsubseteq A$. Now suppose $E \in \Sigma_{\mathcal{L}}$, then we would have $\mathcal{M}_{[\mathcal{O},\Sigma^{\mathcal{L}}]} \models E \sqsubseteq A$ and thus necessarily $A \in \mathsf{Sig}(\mathcal{M}_{[\mathcal{O},\Sigma^{\mathcal{L}}]}) \subseteq \Sigma^{\mathcal{L}}$ (Theorem 1); but $A \notin \Sigma^{\mathcal{L}}$ by hypothesis, so it must be $E \notin \Sigma^{\mathcal{L}}$. At least one such E is added to ToTest in line 13, so E satisfies property (♣) for A at the end of the iteration.

This completes the proof of property (♣) and we next prove Claims 2 and 3.

(Claim 2) Consider an arbitrary unsatisfiable atomic concept $A \in S$. By definition of the $\mathsf{hierarchy}$ function, an atomic concept A' occurring in H exists such that $A \in \rho(A')$. If A occurs in $\mathbf{K}$, then we clearly have $A \rightsquigarrow_{\mathbf{K}} A'$ and $A' \rightsquigarrow_{\mathbf{K}} A$. If A does not occur in $\mathbf{K}$, then, since A is unsatisfiable and $\top$ and $\bot$

are the only two elements that can occur in H but not in $\mathbf{K}$, we have $A = \bot$; but then $A' = A$, and so we have $A \rightsquigarrow_{\mathbf{K}} A'$ and $A' \rightsquigarrow_{\mathbf{K}} A$ in this case too (each atomic concept is reachable from itself). Class A' satisfies property (♣); furthermore, we have $\mathsf{ToTest} = \emptyset$ upon termination, so by the contrapositive of property (♣) either $A' \rightsquigarrow_{\mathbf{K}} \bot$, or $\mathbf{P}|_{A'} \neq \emptyset$ or $A \in \Sigma^{\mathcal{L}}$. Note, however, that unsatisfiable atomic concepts never appear in premodels, so the algorithm never adds a pair of the form $\langle A', C\rangle$ to $\mathbf{P}$. Thus either $A' \rightsquigarrow_{\mathbf{K}} \bot$ or $A \in \Sigma^{\mathcal{L}}$. If $A' \rightsquigarrow_{\mathbf{K}} \bot$, then consequently $A \rightsquigarrow_{\mathbf{K}} \bot$ as well. If $A \in \Sigma^{\mathcal{L}}$ then, by Proposition 2, if $\mathcal{O} \models A \sqsubseteq \bot$ also $\mathcal{O}_{\mathcal{L}} \models A \sqsubseteq \bot$ and we must already have $\langle A, \bot\rangle \in \mathbf{K}' \subseteq \mathbf{K}$, so $A \rightsquigarrow_{\mathbf{K}} \bot$.

(Claim 3) Consider an arbitrary satisfiable atomic concept $A \in S$ and an arbitrary atomic concept $B \in S$ such that $\mathcal{O} \models A \sqsubseteq B$. As before, whether A occurs in $\mathbf{K}$ or not, there is some atomic concept A' occurring in H such that $A \rightsquigarrow_{\mathbf{K}} A'$, $A' \rightsquigarrow_{\mathbf{K}} A$ and either $\mathbf{P}|'_A \neq \emptyset$, or $A' \rightsquigarrow_{\mathbf{K}} \bot$ or $A \in \Sigma^{\mathcal{L}}$. By Claim 1 and the fact that A is satisfiable, it cannot be the case that $A' \rightsquigarrow_{\mathbf{K}} \bot$. Therefore either $\mathbf{P}|_{A'} \neq \emptyset$ or $A \in \Sigma^{\mathcal{L}}$. If $\mathbf{P}|_{A'} \neq \emptyset$ then, by property (♦) we have $B \in \mathbf{P}|_{A'}$ at the end of the while-loop and Claim 3 holds at this point; pair $\langle A', B\rangle$ can be removed from $\mathbf{P}$ in line 21, but then $A' \rightsquigarrow_{\mathbf{K}} B$, and so we have $A \rightsquigarrow_{\mathbf{K}} B$ and Claim 3 holds after line 21 too. If $A \in \Sigma^{\mathcal{L}}$, then, by Proposition 2, if $\mathcal{O} \models A \sqsubseteq \bot$ also $\mathcal{O}_{\mathcal{L}} \models A \sqsubseteq B$ and we must already have $\langle A, B\rangle \in \mathbf{K}' \subseteq \mathbf{K}$, so $A \rightsquigarrow_{\mathbf{K}} B$.