A Proof-Theoretic Subsumption Reasoner for Hybrid $\mathcal{EL}$-TBoxes

Franz Baader, Novak Novakovic, and Boontawee Suntisrivaraporn

Theoretical Computer Science, TU Dresden, Germany

Abstract. Hybrid $\mathcal{EL}$-TBoxes combine general concept inclusions (GCIs), which are interpreted with descriptive semantics, with cyclic concept definitions, which are interpreted with greatest fixpoint (gfp) semantics. We introduce a proof-theoretic approach that yields a polynomial-time decision procedure for subsumption in $\mathcal{EL}$ w.r.t. hybrid TBoxes, and present preliminary experimental results regarding the performance of the reasoner Hyb that implements this decision procedure.

1 Introduction

The $\mathcal{EL}$-family of description logics (DLs) is a family of inexpressive DLs whose main distinguishing feature is that they provide their users with existential restrictions rather than value restrictions as the main concept constructor involving roles. The core language of this family is $\mathcal{EL}$, which has the top concept ($\top$), conjunction ($\sqcap$), and existential restrictions ($\exists r.C$) as concept constructors. This family has recently drawn considerable attention since, one the one hand, the subsumption problem stays tractable (i.e., decidable in polynomial time) in situations where the corresponding DL with value restrictions becomes intractable. In particular, subsumption in $\mathcal{EL}$ is tractable both w.r.t. cyclic TBoxes interpreted with gfp or descriptive semantics [3] and w.r.t. general TBoxes (i.e., finite sets of GCIs) interpreted with descriptive semantics [8, 4]. On the other hand, although of limited expressive power, $\mathcal{EL}$ is nevertheless used in applications, e.g., to define biomedical ontologies. For example, both the large medical ontology Snomed ct [15] and the Gene Ontology [1] can be expressed in $\mathcal{EL}$, and the same is true for large parts of the medical ontology Galen [14]. To support such applications, several $\mathcal{EL}$ reasoners have been developed. Implementations of the polynomial-time decision procedures for subsumption w.r.t. cyclic $\mathcal{EL}$-TBoxes introduced in [3] were described in [16]. An optimised version of the algorithm dealing with the case of general TBoxes [8, 4] was implemented in the CEL system [6, 7], which is able to classify very large ontologies such as Snomed.

In some cases, it would be advantageous to have both GCIs interpreted with descriptive semantics and cyclic concept definitions interpreted with gfp-semantics available in one TBox. One motivation for such hybrid TBoxes comes from the area of non-standard inferences in DLs. For example, if one wants to support the so-called bottom-up construction of DL knowledge bases, then one needs to compute least common subsumers (lcs) and most specific concepts (msc) [5]. In [2], it was shown that the lcs and the msc in $\mathcal{EL}$ always exist and can be computed in polynomial time if cyclic definitions that are interpreted with gfp-semantics are available. In contrast, if cyclic definitions or GCIs are interpreted with descriptive semantics, neither the lcs nor the msc need to exist.

Hybrid $\mathcal{EL}$ TBoxes have first been introduced in [10]. Basically, such a TBox consists of two parts $\mathcal{T}$ and $\mathcal{F}$, where $\mathcal{T}$ is a cyclic TBox whose primitive concepts occur in the GCIs of the general TBox $\mathcal{F}$. However, defined concepts of $\mathcal{T}$ must not occur in $\mathcal{F}$. It was shown in [10] that subsumption w.r.t. such hybrid TBoxes can still be decided in polynomial time. The algorithm uses reasoning w.r.t. the general TBox $\mathcal{F}$ to extend the cyclic TBox $\mathcal{T}$ to a cyclic TBox $\widehat{\mathcal{T}}$ such that subsumption can then be decided considering only $\widehat{\mathcal{T}}$. An implementation of this approach is described in [12]. It uses the reasoner for cyclic $\mathcal{EL}$-TBoxes with gfp semantics described in [16] to classify the extended TBox $\widehat{\mathcal{T}}$. The reasoning w.r.t. the general TBox, which is required to compute the extension $\widehat{\mathcal{T}}$, employs a preliminary implementation of the algorithm in [8], which is also described in [16].[1] In [9] it was shown that, w.r.t. hybrid $\mathcal{EL}$-TBoxes, the lcs and msc always exits and can be computed in polynomial time.

An approach for deciding subsumption in $\mathcal{EL}$ that significantly differs from the ones described in [3, 8, 4] was introduced in [11]. It is based on sound and complete Gentzen-style proof calculi for subsumption w.r.t. cyclic TBoxes interpreted with gfp semantics and for subsumption w.r.t. general TBoxes interpreted with descriptive semantics. These calculi yield polynomial-time decision procedures since they satisfy an appropriate sub-description property.

In this paper, we show that we can obtain a polynomial-time decision procedure for subsumption w.r.t. hybrid $\mathcal{EL}$-TBoxes by combining the two calculi introduced in [11]. We also report on first experimental results regarding the performance of our implementation of this decision procedure in the system Hyb.[2] Notice that both general $\mathcal{EL}$-TBoxes interpreted by descriptive semantics and cyclic $\mathcal{EL}$-TBoxes interpreted with gfp semantics are special cases of the hybrid $\mathcal{EL}$-TBoxes. Thus, our decision procedure can also classify these kinds of TBoxes.

2 Hybrid $\mathcal{EL}$-TBoxes

Starting with a set N_{con} of concept names and a set N_{role} of role names, $\mathcal{EL}$-*concept descriptions* are built using the concept constructors top concept ($\top$), conjunction ($\sqcap$), and existential restrictions ($\exists r.C$). The semantics of $\mathcal{EL}$ is defined in the usual way, using the notion of an interpretation $\mathcal{I} = (\mathcal{D}_\mathcal{I}, \cdot^\mathcal{I})$, which consists of a nonempty domain $\mathcal{D}_\mathcal{I}$ and an interpretation function $\cdot^\mathcal{I}$ that assigns binary relations on $\mathcal{D}_\mathcal{I}$ to role names and subsets of $\mathcal{D}_\mathcal{I}$ to concept descriptions, as shown in the semantics column of Table 2.

A *concept definition* is an expression of the form $A \equiv C$ where A is a concept name and C is a concept description, and a *general concept inclusion* (GCI)

[1] This implementation is not as efficient as the one later developed for the CEL system.

[2] see `http://lat.inf.tu-dresden.de/systems/Hyb/`.

Name	Syntax	Semantics
concept name	A	$A^{\mathcal{I}} \subseteq \mathcal{D}_{\mathcal{I}}$
role name	r	$r^{\mathcal{I}} \subseteq \mathcal{D}_{\mathcal{I}} \times \mathcal{D}_{\mathcal{I}}$
top-concept	$\top$	$\top^{\mathcal{I}} = \mathcal{D}_{\mathcal{I}}$
conjunction	$C \sqcap D$	$(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$
exist. restriction	$\exists r.C$	$(\exists r.C)^{\mathcal{I}} = \{x \mid \exists y : (x,y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$
concept definition	$A \equiv C$	$A^{\mathcal{I}} = C^{\mathcal{I}}$
subsumption	$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$

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

is an expression of the form $C \sqsubseteq D$, where C, D are concept descriptions. An interpretation $\mathcal{I}$ is a *model* of a concept definition or GCI if it satisfies the respective condition given in the semantics column of Table 2. This semantics for GCIs and concept definitions is usually called *descriptive semantics.* A *TBox* is a finite set $\mathcal{T}$ of concept definitions that does not contain multiple definitions, i.e., $\{A \equiv C, A \equiv D\} \subseteq \mathcal{T}$ implies $C = D$. Note that we do *not* require TBoxes to be *acyclic*, i.e., there may be cyclic dependencies among the concept definitions. A *general TBox* is a finite set of GCIs. The interpretation $\mathcal{I}$ is a *model* of the TBox $\mathcal{T}$ (the general TBox $\mathcal{F}$) iff it is a model of all concept definitions (GCIs) in $\mathcal{T}$ (in $\mathcal{F}$). The name *general* TBox is justified by the fact that concept definitions $A \equiv C$ can of course be expressed by GCIs $A \sqsubseteq C, C \sqsubseteq A$. However, in our hybrid TBoxes we will interpret concept definitions by greatest fixpoint semantics rather than by descriptive semantics.

We assume in the following that the set of concept names N_{con} is partitioned into the set of *primitive concepts* N_{prim} and the set of *defined concepts* N_{def}. In a hybrid TBox, concept names occurring on the left-hand side of a concept definition are required to come from the set N_{def}, whereas GCIs may not contain concept names from N_{def}.

Definition 1 (Hybrid $\mathcal{EL}$-TBoxes). *A* hybrid $\mathcal{EL}$-TBox *is a pair $(\mathcal{F}, \mathcal{T})$, where $\mathcal{F}$ is a general $\mathcal{EL}$-TBox containing only concept names from N_{prim}, and $\mathcal{T}$ is an $\mathcal{EL}$-TBox such that $A \equiv C \in \mathcal{T}$ implies $A \in N_{def}$.*

An example of a hybrid $\mathcal{EL}$-Tbox, taken form [10], is given in Fig. 1. It defines the concepts 'disease of the connective tissue,' 'bacterial infection,' and 'bacterial pericarditis' using the cyclic definitions in $\mathcal{T}$. The general TBox $\mathcal{F}$ states some properties that the primitive concepts and roles occurring in $\mathcal{T}$ must satisfy, such as the fact that a disease located on connective tissue also acts on the connective tissue.

In general, the idea underlying the definition of hybrid TBoxes is the following: $\mathcal{F}$ can be used to constrain the interpretation of the primitive concepts and roles, whereas $\mathcal{T}$ tells us how to interpret the defined concepts occurring in it, once the interpretation of the primitive concepts and roles is fixed.

$$\begin{array}{lrl}
\mathcal{T}: & \mathsf{ConnTissDisease} \equiv & \mathsf{Disease} \sqcap \exists \mathsf{acts\_on}.\mathsf{ConnTissue} \\
 & \mathsf{BactInfection} \equiv & \mathsf{Infection} \sqcap \exists \mathsf{causes}.\mathsf{BactPericarditis} \\
 & \mathsf{BactPericarditis} \equiv & \mathsf{Inflammation} \sqcap \exists \mathsf{has\_loc}.\mathsf{Pericardium} \\
 & & \sqcap\, \exists \mathsf{caused\_by}.\mathsf{BactInfection} \\
 & & \\
\mathcal{F}: & \mathsf{Disease} \sqcap \exists \mathsf{has\_loc}.\mathsf{ConnTissue} \sqsubseteq & \exists \mathsf{acts\_on}.\mathsf{ConnTissue} \\
 & \mathsf{Inflammation} \sqsubseteq & \mathsf{Disease} \\
 & \mathsf{Pericardium} \sqsubseteq & \mathsf{ConnTissue}
\end{array}$$

Fig. 1. A small hybrid $\mathcal{EL}$-TBox.

A *primitive interpretation* $\mathcal{J}$ is defined like an interpretation, with the only difference that it does not provide an interpretation for defined concepts. A primitive interpretation can thus interpret concept descriptions built over N_{prim} and N_{role}, but it cannot interpret concept descriptions containing elements of N_{def}. Given a primitive interpretation $\mathcal{J}$, we say that the (full) interpretation $\mathcal{I}$ is *based on* $\mathcal{J}$ if it has the same domain as $\mathcal{J}$ and its interpretation function coincides with $\mathcal{J}$ on N_{prim} and N_{role}.

Given two interpretations $\mathcal{I}_1$ and $\mathcal{I}_2$ based on the same primitive interpretation $\mathcal{J}$, we define

$$\mathcal{I}_1 \preceq_{\mathcal{J}} \mathcal{I}_2 \quad \text{iff} \quad A^{\mathcal{I}_1} \subseteq A^{\mathcal{I}_2} \text{ for all } A \in N_{def}.$$

It is easy to see that the relation $\preceq_{\mathcal{J}}$ is a partial order on the set of interpretations based on $\mathcal{J}$. In [3] the following was shown: given an $\mathcal{EL}$-TBox $\mathcal{T}$ and a primitive interpretation $\mathcal{J}$, there exists a unique model $\mathcal{I}$ of $\mathcal{T}$ such that

- $\mathcal{I}$ is based on $\mathcal{J}$;
- $\mathcal{I}' \preceq_{\mathcal{J}} \mathcal{I}$ for all models $\mathcal{I}'$ of $\mathcal{T}$ that are based on $\mathcal{J}$.

We call such a model $\mathcal{I}$ a *gfp-model* of $\mathcal{T}$.

Definition 2 (Semantics of hybrid $\mathcal{EL}$-TBoxes). *The interpretation $\mathcal{I}$ is a* hybrid model *of the hybrid $\mathcal{EL}$-TBox $(\mathcal{F}, \mathcal{T})$, iff $\mathcal{I}$ is a gfp-model of $\mathcal{T}$ and the primitive interpretation $\mathcal{J}$ it is based on is a model of $\mathcal{F}$.*

It is well-known that gfp-semantics coincides with descriptive semantics for acyclic TBoxes. Thus, if $\mathcal{T}$ is actually acyclic, then $\mathcal{I}$ is a hybrid model of $(\mathcal{F}, \mathcal{T})$ according to the semantics introduced in Definition 2 iff it is a model of $\mathcal{T} \cup \mathcal{F}$ w.r.t. descriptive semantics, i.e., iff $\mathcal{I}$ is a model of every GCI in $\mathcal{F}$ and of every concept definition in $\mathcal{T}$.

3 Subsumption w.r.t. Hybrid $\mathcal{EL}$-TBoxes

Based on the semantics for hybrid TBoxes introduced above, we can now define the main inference problem that we want to solve in this paper.

$$\begin{array}{llllll}
C \sqsubseteq_n C & \text{(Refl)} & C \sqsubseteq_n \top & \text{(Top)} & C \sqsubseteq_0 D & \text{(Start)} \\[2ex]
\dfrac{C \sqsubseteq_n E}{C \sqcap D \sqsubseteq_n E} & \text{(AndL1)} & \dfrac{D \sqsubseteq_n E}{C \sqcap D \sqsubseteq_n E} & \text{(AndL2)} & \dfrac{C \sqsubseteq_n D \quad C \sqsubseteq_n E}{C \sqsubseteq_n D \sqcap E} & \text{(AndR)} \\[2ex]
 & & \dfrac{C \sqsubseteq_n D}{\exists r.C \sqsubseteq_n \exists r.D} & \text{(Ex)} & & \\[2ex]
\dfrac{C \sqsubseteq_n D}{A \sqsubseteq_n D} & \text{(DefL)} & \dfrac{D \sqsubseteq_n C}{D \sqsubseteq_{n+1} A} & \text{(DefR)} & \dfrac{C \sqsubseteq_n E \quad F \sqsubseteq_n D}{C \sqsubseteq_n D} & \text{(GCI)} \\
\text{for } A \equiv C \in \mathcal{T} & & \text{for } A \equiv C \in \mathcal{T} & & \text{for } E \sqsubseteq F \in \mathcal{F} &
\end{array}$$

Fig. 2. The rule system HC

Definition 3 (Subsumption w.r.t. hybrid $\mathcal{EL}$-TBoxes). *Let $(\mathcal{F}, \mathcal{T})$ be a hybrid $\mathcal{EL}$-TBox, and A, B defined concepts occurring on the left-hand side of a definition in $\mathcal{T}$. Then A* is subsumed by *B* w.r.t. *$(\mathcal{F}, \mathcal{T})$ (written $A \sqsubseteq_{gfp,\mathcal{F},\mathcal{T}} B$) iff $A^{\mathcal{I}} \subseteq B^{\mathcal{I}}$ holds for all hybrid models $\mathcal{I}$ of $(\mathcal{F}, \mathcal{T})$.*

Defining (and computing) subsumption only for concept names A, B defined in $\mathcal{T}$ rather than for arbitrary concept descriptions C, D is not a real restriction since one can always add definitions with the right-hand sides C, D to $\mathcal{T}$.

Assume that the hybrid $\mathcal{EL}$-TBox $(\mathcal{F}, \mathcal{T})$ is given, and that we want to decide whether, for given defined concepts A, B, the subsumption relationship $A \sqsubseteq_{\mathrm{gfp},\mathcal{F},\mathcal{T}} B$ holds or not. Following the ideas in [11], we introduce a sound and complete Gentzen-style calculus for subsumption. The reason why this calculus yields a decision procedure is basically that it has the sub-description property, i.e., application of rules can be restricted to sub-descriptions of concept descriptions occurring in $\mathcal{F}$ or $\mathcal{T}$.

A *sequent for* $(\mathcal{F}, \mathcal{T})$ is of the form $C \sqsubseteq_n D$, where C, D are sub-descriptions of concept descriptions occurring in $\mathcal{F}$ or $\mathcal{T}$, and $n \geq 0$. The rules of the **H**ybrid $\mathcal{EL}$-TBox **C**alculus HC depicted in Fig. 2 can be used to derive new sequents from sequents that have already been derived. For example, the sequents in the first row of the figure can always be derived without any prerequisites, using the rules Refl, Top, and Start, respectively. Using the rule AndR, the sequent $C \sqsubseteq_n D \sqcap E$ can be derived in case both $C \sqsubseteq_n D$ and $C \sqsubseteq_n E$ have already been derived. Note that the rule Start applies only for $n = 0$. Also note that, in the rule DefR, the index is incremented when going from the prerequisite to the consequent.

Fig. 3 shows a derivation in HC w.r.t. the hybrid $\mathcal{EL}$-TBox from Fig. 1, where obvious abbreviations of concept and role names have been made. This derivation tree demonstrates that the sequent $\mathsf{BactPericarditis} \sqsubseteq_{n+1} \mathsf{ConnTissDisease}$ can

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\mathsf{Infl} \sqsubseteq_n \mathsf{Infl} \quad \mathsf{D} \sqsubseteq_n \mathsf{D}}{\mathsf{Infl} \sqsubseteq_n \mathsf{D}}}{\mathsf{Infl} \sqcap \exists\mathsf{hl}.\mathsf{P} \sqsubseteq_n \mathsf{D}} \quad \dfrac{\dfrac{\dfrac{\dfrac{\mathsf{Infl} \sqsubseteq_n \mathsf{Infl} \quad \mathsf{D} \sqsubseteq_n \mathsf{D}}{\mathsf{Infl} \sqsubseteq_n \mathsf{D}}}{\mathsf{Infl} \sqcap \exists\mathsf{hl}.\mathsf{P} \sqsubseteq_n \mathsf{D}} \quad \dfrac{\dfrac{\dfrac{\mathsf{P} \sqsubseteq_n \mathsf{P} \quad \mathsf{CT} \sqsubseteq_n \mathsf{CT}}{\mathsf{P} \sqsubseteq_n \mathsf{CT}}}{\exists\mathsf{hl}.\mathsf{P} \sqsubseteq_n \exists\mathsf{hl}.\mathsf{CT}}}{\mathsf{Infl} \sqcap \exists\mathsf{hl}.\mathsf{P} \sqsubseteq_n \exists\mathsf{hl}.\mathsf{CT}}}{\mathsf{Infl} \sqcap \exists\mathsf{hl}.\mathsf{P} \sqsubseteq_n \mathsf{D} \sqcap \exists\mathsf{hl}.\mathsf{CT}} \quad \exists\mathsf{ao}.\mathsf{CT} \sqsubseteq_n \exists\mathsf{ao}.\mathsf{CT}}{\mathsf{Infl} \sqcap \exists\mathsf{hl}.\mathsf{P} \sqsubseteq_n \exists\mathsf{ao}.\mathsf{CT}}}{\mathsf{Infl} \sqcap \exists\mathsf{hl}.\mathsf{P} \sqsubseteq_n \mathsf{D} \sqcap \exists\mathsf{ao}.\mathsf{CT}}}{\mathsf{Infl} \sqcap \exists\mathsf{hl}.\mathsf{P} \sqcap \exists\mathsf{cb}.\mathsf{Bl} \sqsubseteq_n \mathsf{D} \sqcap \exists\mathsf{ao}.\mathsf{CT}}}{\mathsf{BP} \sqsubseteq_n \mathsf{D} \sqcap \exists\mathsf{ao}.\mathsf{CT}}}{\mathsf{BP} \sqsubseteq_{n+1} \mathsf{CTD}}$$

Fig. 3. An example of a derivation in HC.

be derived for every $n \geq 0$. Note that we can also derive BactPericarditis $\sqsubseteq_0$ ConnTissDisease using the rule Start.

The calculus HC defines binary relations $\sqsubseteq_n$ for $n \in \{0, 1, \ldots\} \cup \{\infty\}$ on the set of sub-descriptions of concept descriptions occurring in $\mathcal{F}$ or $\mathcal{T}$:

Definition 4. *Let C, D be sub-descriptions of the concept descriptions occurring in $\mathcal{F}$ or $\mathcal{T}$. Then $C \sqsubseteq_n D$ holds iff the sequent $C \sqsubseteq_n D$ can be derived using the rules of HC. In addition, $C \sqsubseteq_\infty D$ holds iff $C \sqsubseteq_n D$ holds for all $n \geq 0$.*

The calculus HC is sound and complete for subsumption w.r.t. hybrid $\mathcal{EL}$-TBoxes in the following sense.

Theorem 1 (Soundness and Completeness of HC). *Let $(\mathcal{F}, \mathcal{T})$ be a hybrid $\mathcal{EL}$-TBox, and A, B defined concepts occurring on the left-hand side of a definition in $\mathcal{T}$. Then $A \sqsubseteq_{gfp,\mathcal{F},\mathcal{T}} B$ iff $A \sqsubseteq_\infty B$ holds.*

A detailed proof of this theorem is given in [13]. Though the rules of HC are taken from the sound and complete subsumption calculi introduced in [11] for subsumption w.r.t. cyclic $\mathcal{EL}$-TBoxes interpreted with gfp-semantics and for subsumption w.r.t. general $\mathcal{EL}$-TBoxes interpreted with descriptive semantics, respectively, the proof that their combination is sound and complete for the case of hybrid $\mathcal{EL}$-TBoxes requires non-trivial modifications of the proofs given in [11]. Nevertheless, we think that these proofs are simpler and easier to comprehend than the ones given in [10, 12] for the correctness of the reduction-based subsumption algorithm for hybrid $\mathcal{EL}$-TBoxes introduced there.

In our example, we have BactPericarditis $\sqsubseteq_\infty$ ConnTissDisease, and thus soundness of HC implies that the subsumption relationship BactPericarditis $\sqsubseteq_{\mathrm{gfp},\mathcal{F},\mathcal{T}}$ ConnTissDisease holds.

It is not hard to show that $\sqsubseteq_0$ is the universal relation on sub-descriptions of the concept descriptions occurring in $\mathcal{F}$ or $\mathcal{T}$, and that $\sqsubseteq_{n+1} \subseteq \sqsubseteq_n$ holds for all $n \geq 0$ (see [13] for a proof). Thus, to compute $\sqsubseteq_\infty$ we can start with the universal relation $\sqsubseteq_0$, and then compute $\sqsubseteq_1, \sqsubseteq_2, \ldots$, until for some m we have $\sqsubseteq_m = \sqsubseteq_{m+1}$, and thus $\sqsubseteq_m = \sqsubseteq_\infty$. Since the set of sub-descriptions is finite, the computation of each relation $\sqsubseteq_n$ can be done in finite time, and we can be sure that there always exists an m such that $\sqsubseteq_m = \sqsubseteq_{m+1}$. This shows that the calculus HC indeed yields a subsumption algorithm.

4 The Algorithm and its Implementation

In this section, we describe how to develop a practical algorithm based on the decision procedure for subsumption w.r.t. hybrid $\mathcal{EL}$-TBoxes introduced in the previous section. As mentioned above, the algorithm starts with the universal relation $\sqsubseteq_0$, and then iteratively compute the relations $\sqsubseteq_n$ for increasing n until a fixpoint is reached, i.e., an m such that $\sqsubseteq_m = \sqsubseteq_{m+1}$. There are two possibilities for computing $\sqsubseteq_n$ given $\sqsubseteq_{n-1}$: the bottom-up approach and the top-down approach, where bottom-up and top-down is meant w.r.t. the proof trees of the calculus HC (like the one depicted in Fig. 3).

The *bottom-up* search starts with a target sequent $C \sqsubseteq_n D$ as the root and then non-deterministically applies one of the HC rules backwards, i.e., it extends the proof tree with the prerequisites needed to apply the rule in a way that derives the target sequent. Clearly, the initial sequent $C \sqsubseteq_n D$ is provable iff all the spawned prerequisites are provable. Thus, one continues in the same way with the prerequisites as target sequents. If eventually all the prerequisites are instances of the rules (Refl), (Top), or (Start), or have been proved in the previous iteration (i.e., when computing $\sqsubseteq_{n-1}$), then the proof tree is complete, implying that the initial sequent can be derived. Due to the presence of GCIs, termination of this search procedure is, however, not guaranteed. In order to regain termination we need a blocking mechanism to ensure that the same sequent is not tried to be proved twice along the same path. One can also optimise the search procedure by caching already proved sequents and by reusing them in other branches of the proof tree whenever appropriate. Though the termination problem could in principle be solved by blocking, this approach was nevertheless not chosen for our implementation of the Hyb reasoner. On the one hand, the bottom-up search is non-deterministic, with backtracking necessary to try out different HC rules at various levels of the proof tree. On the other hand, the fact that the bottom-up approach is goal oriented is not really useful here since we need to compute the whole relation $\sqsubseteq_n$ anyway. In fact, otherwise we would not be able to detect that a fixpoint has been reached.

In the remainder of this section, we focus on the top-down approach, which is the one realized in the Hyb reasoner.

The algorithm implemented in Hyb

The *top-down* approach does not start with a single target sequent, i.e., the root of a proof tree. Instead, it starts with all potential leaf sequents, i.e., all instances of (Refl) and (Top). Instances of (Start) are considered only in the computation of $\sqsubseteq_0$. The algorithm maintains two tables of derivable subsumptions:

$$\mathsf{ds} : \mathsf{Subs}_{(\mathcal{F},\mathcal{T})} \times \mathsf{Subs}_{(\mathcal{F},\mathcal{T})} \to \{0,1\} \text{ and } \mathsf{ds}^- : \mathsf{Subs}_{(\mathcal{F},\mathcal{T})} \times \mathsf{Subs}_{(\mathcal{F},\mathcal{T})} \to \{0,1\},$$

where $\mathsf{Subs}_{(\mathcal{F},\mathcal{T})}$ denotes the set of sub-descriptions of the concept descriptions occurring in $\mathcal{F}$ or $\mathcal{T}$. Intuitively, when computing $\sqsubseteq_n$, then ds assigns 1 to (C, D) if we have already computed that $C \sqsubseteq_n D$ holds, whereas ds^- has full information about $\sqsubseteq_{n-1}$, i.e., $\mathsf{ds}^-(C, D) = 1$ iff $C \sqsubseteq_{n-1} D$ holds.

For the case $n = 0$, the rule (Start) tells us that we must initialise $\mathsf{ds}[C, D]$ to 1 for all sub-descriptions $C, D \in \mathsf{Subs}_{(\mathcal{F},\mathcal{T})}$. We divide the rules of HC into two groups, namely, (DefR) and the others. In each iteration, the algorithm first exhaustively applies all the rules other than (DefR). When this is done, and an instance of (DefR) is applicable, then it *advances* to the next generation (i.e., from n to $n + 1$) by performing the following steps:

1. set $\mathsf{ds}^-[C, D] := \mathsf{ds}[C, D]$ for all sub-descriptions C, D;
2. set $\mathsf{ds}[C, D]$ to 1 if $C = D$ or $D = \top$, and 0 otherwise; and
3. set $\mathsf{ds}[D, A]$ to 1 if $\mathsf{ds}^-[D, C] = 1$ and $A \equiv C \in \mathcal{T}$.

Obviously, Step 2 corresponds to the application of the rule (Refl) and (Top), while Step 3 realizes the rule (DefR). After this initialisation, all other rules are again exhaustively applied to compute $\sqsubseteq_{n+1}$.

To guide the rule application within one iteration, we use a candidate queue that stores potentially applicable rule instances. These queues are initialised according to Steps 2 and 3 introduced above. A rule application changes some value $\mathsf{ds}[C, D]$ from 0 to 1, which may in turn trigger another rule application. This is taken care of by appropriately augmenting the candidate queue.

The algorithm stops when no more rule applies in the current iteration, and the two tables of derivable subsumptions coincide, i.e., $\mathsf{ds}[C, D] = \mathsf{ds}^-[C, D]$ for all $C, D \in \mathsf{Subs}_{(\mathcal{F},\mathcal{T})}$. This means that the fixpoint has been reached, and the table ds can be used to correctly answer subsumption queries for all sub-descriptions occurring in the hybrid TBox $(\mathcal{F}, \mathcal{T})$.

The algorithm terminates in time polynomial in the size of the input since

1. iterations can only remove derivable subsumptions due to the fact that $\sqsubseteq_{n+1} \subseteq \sqsubseteq_n$, and there are at most a quadratic number of entries in the tables ds;
2. there are polynomially many applicable rule instances in each iteration, and
3. each rule instance is applied at most once, and its application takes only polynomial time.

Optimisations

Despite running in polynomial-time, an unoptimised implementation of the algorithm would not behave well in practice. Here we describes some of the optimisation techniques that we have employed in the implementation of the Hyb reasoner.

Monotonicity. As mentioned before, the mapping $n \mapsto \sqsubseteq_n$ is monotone, i.e., if $C \sqsubseteq_n D$ does not hold, then neither can $C \sqsubseteq_{n+1} D$. Thus, if we have $\mathsf{ds}^-[C, D] = 0$, then it is clear that we can never get $\mathsf{ds}[C, D] = 1$. This fact can be used to avoid attempting certain rule applications for which it is clear that they can never be successful.

Invariant derivations. For $n \geq 1$ we can distinguish derivable sequents $C \sqsubseteq_n D$ that require the rule (DefR) from those that do not. In fact, the latter set of sequents is the same for all $n \geq 1$. Thus, it is enough to compute them once, when computing $\sqsubseteq_1$. Afterwards, they are simply transferred to the current relation $\sqsubseteq_n$ without the need to recompute them.

Single iteration. If the hybrid TBox contains only GCIs and no concept definitions, i.e., $\mathcal{T}$ is empty, then a single iteration of the algorithm is sufficient, i.e., it is enough to compute $\sqsubseteq_1$. Thus, we can avoid the initialisation of ds for the case $n = 0$, which corresponds to disabling the rule (Start). Also, it is enough to create and maintain just one table ds for the case $n = 1$. This way, the space requirement of the algorithm is reduced by half.

The algorithm together with the optimisation techniques sketched above has been implemented in our Hyb reasoner, using Common LISP. More details can be found on the system's Web page `http://lat.inf.tu-dresden.de/systems/hyb/`.

5 Experimental Results

This section describes our experiments on several life science ontologies. To evaluate the Hyb reasoner, we have compared its performance with that of the reasoner for cyclic $\mathcal{EL}$-TBox with gfp-semantics from [16], the reasoner for hybrid $\mathcal{EL}$-TBox from [12], and CEL [7]. All the experiments were performed on a PC with a 1.7 GHz Pentium-4 CPU and 512 MB of physical memory, running Linux version 2.6.14. All participating reasoners were written in Common LISP, and we used Allegro CL version 8.1 as the runtime environment.

Our benchmarks comprise the toy ontology from Figure 1, the Gene Ontology (Go), the National Cancer Institute's thesaurus (Nci), and several subsets of the Galen Medical Knowledge Base (Galen).[3] Since none of the reasoners, apart from CEL, supports role axioms, we have removed role axioms from all the ontologies. These benchmark ontologies are referred to as $\mathcal{O}^{\mathsf{toy}}$, $\mathcal{O}^{\mathsf{Go}}$, $\mathcal{O}^{\mathsf{Nci}}$, and $\mathcal{O}^{\mathsf{Galen}}$ in the following. Some information on the size of these ontologies is given in the upper part of Table 2. There, $\mathcal{O}^{\mathsf{Galen}}$ with a subscript refers to its various subsets used in the experiments. Note that $\mathcal{O}^{\mathsf{toy}}$ is the only hybrid TBox, while $\mathcal{O}^{\mathsf{Go}}$ and $\mathcal{O}^{\mathsf{Nci}}$ are in principle acyclic TBoxes,[4] and $\mathcal{O}^{\mathsf{Galen}}$ is a general TBox. The classification times given in the lower part of the table are in second, where 'unatt.' means that the reasoner did not terminate on the input after five hours.

[3] More information regarding these ontologies is available at `http://lat.inf.tu-dresden.de/~meng/toyont.html`.

[4] All GCIs in $\mathcal{O}^{\mathsf{Go}}$ and $\mathcal{O}^{\mathsf{Nci}}$ are of the form $A \sqsubseteq C$ with A a concept name, and can be absorbed into the concept definitions $A \equiv P \sqcap C$ with P a fresh concept name.

	$\mathcal{O}^{\mathsf{Go}}$	$\mathcal{O}^{\mathsf{Nci}}$	$\mathcal{O}^{\mathsf{Galen}}$	$\mathcal{O}_1^{\mathsf{Galen}}$	$\mathcal{O}_2^{\mathsf{Galen}}$	$\mathcal{O}_3^{\mathsf{Galen}}$	$\mathcal{O}^{\mathsf{toy}}$
♯Definitions	0	0	699	24	689	699	3
♯GCIs	16 803	46 807	3 252	2 713	339	0	3
♯Concept names	16 806	27 652	2 049	1 798	642	420	6
♯Role names	1	50	159	109	133	128	4
Hyb	3 482	16 252	10 457	290	745	158	0.01
Reasoner from [16]	592	unatt.	n/a	n/a	n/a	12 310	n/a
Reasoner from [12]	unatt.	unatt.	unatt.	unatt.	unatt.	unatt.	0.02
CEL	2.24	11.98	10	1.35	0.94	0.77	n/a

Table 2. Benchmarks and classification times (in seconds).

We write 'n/a' to express that the reasoner does not support this type of TBox. In particular, the gfp-reasoner from[16] cannot handle GCIs, while CEL does not support the gfp-semantics used for hybrid TBoxes.

We can observe that Hyb can classify all the benchmark ontologies, and that it outperforms both the gfp-reasoner from [16] (except for the case of $\mathcal{O}^{\mathsf{Go}}$) and the one for hybrid TBoxes from [12]. It should be noted, however, that the gfp-reasoner from [16] was a prototypical implementation without many optimisations. The reasoners for hybrid TBoxes from [12] uses this gfp-reasoner from [16] and it also employes a quite preliminary reasoner for general $\mathcal{EL}$-TBoxes, also described in [16].

The significantly better performance of CEL did not come as a surprise. On the one hand, Hyb can deal with a more expressive TBox formalism. On the other hand, we have spent quite some work on optimising CEL, whereas the current implementation of Hyb is still rather preliminary. Given this, we find the fact that Hyb can (in contrast to the reasoners from [16] and [12]) classify all the benchmark ontologies very promising.

6 Conclusion

In this paper, we have described a Gentzen-style calculus for subsumption w.r.t. hybrid $\mathcal{EL}$-TBoxes, which is an extension to the case of hybrid TBoxes of the calculi for general TBoxes and for cyclic TBoxes with gfp-semantics that have been introduced in [11]. Based on this calculus, we have developed a polynomial-time decision procedure for subsumption w.r.t. hybrid $\mathcal{EL}$-TBoxes, and have sketched its implementation in the Hyb reasoner. Our experiments on life-science ontologies have shown that an acceptable performance of reasoning w.r.t. hybrid semantics can be achieved in practice. Since the main motivation for considering hybrid TBoxes was that, w.r.t. them, the lcs and msc always exist, the natural next step is to develop a proof-theoretic approach to computing the lcs and the msc. For the lcs, such an approach is already described in [13]. We are currently working on the case of the msc. Other future work in this direction is to further optimise the algorithm described in this paper, and to extend it to more expressive DLs from the $\mathcal{EL}$ family.

References

1. M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock. Gene ontology: tool for the unification of biology. *Nat Genet*, 25(1):25–29, May 2000.
2. F. Baader. Least common subsumers and most specific concepts in a description logic with existential restrictions and terminological cycles. In *Proc. IJCAI'03*, Morgan Kaufmann, 2003.
3. F. Baader. Terminological cycles in a description logic with existential restrictions. In *Proc. IJCAI'03*, Morgan Kaufmann, 2003.
4. F. Baader, S. Brandt, and C. Lutz. Pushing the $\mathcal{EL}$ envelope. In *Proc. IJCAI'05*, Morgan Kaufmann, 2005.
5. F. Baader, R. Küsters, and R. Molitor. Computing least common subsumers in description logics with existential restrictions. In *Proc. IJCAI'99*, Morgan Kaufmann, 2003.
6. F. Baader, C. Lutz, and B. Suntisrivaraporn. Is tractable reasoning in extensions of the description logic $\mathcal{EL}$ useful in practice? In *Proc. M4M-05*, 2005.
7. F. Baader, C. Lutz, and B. Suntisrivaraporn. CEL—a polynomial-time reasoner for life science ontologies. In *Proc. IJCAR'06*, Springer LNAI 4130, 2006.
8. S. Brandt. Polynomial time reasoning in a description logic with existential restrictions, GCI axioms, and—what else? In *Proc. ECAI'04*, IOS Press, 2004.
9. S. Brandt. *Standard and Non-standard reasoning in Description Logics*. Ph.D. dissertation, Institute for Theoretical Computer Science, TU Dresden, Germany, 2006.
10. S. Brandt and J. Model. Subsumption in w.r.t. hybrid TBoxes. In *Proc. KI'05*, Springer LNAI 3698, 2005.
11. M. Hofmann. Proof-theoretic approach to description-logic. In *Proc. LICS'05*, IEEE Press, 2005.
12. J. Model. Subsumtion in $\mathcal{EL}$ bezüglich hybrider TBoxen. Diploma thesis, Institute for Theoretical Computer Science, TU Dresden, Germany, 2005.
13. N. Novakovic. Proof-theoretic approach to subsumption and least common subsumer in $\mathcal{EL}$ w.r.t. hybrid TBoxes. Master thesis, Institute for Theoretical Computer Science, TU Dresden, Germany, 2007.
14. A. Rector and I. Horrocks. Experience building a large, re-usable medical ontology using a description logic with transitivity and concept inclusions. In *Proc. AAAI'97*, AAAI Press, 1997.
15. K. Spackman. Managing clinical terminology hierarchies using algorithmic calculation. *Journal of the American Medical Informatics Association*, Fall Symposium Special Issue, 2000.
16. B. Suntisrivaraporn. Optimization and implementation of subsumption algorithms for the description logic $\mathcal{EL}$ with cyclic TBoxes and general concept inclusion axioms. Master thesis, Institute for Theoretical Computer Science, TU Dresden, Germany, 2005.