The revival of structural subsumption in tableau-based description logic reasoners

V. Haarslev[1], R. Möller[2], S. Wandelt[2]

[1] Concordia University, Montreal, Canada
[2] Hamburg University of Technology, Hamburg, Germany*

Abstract. The paper summarizes our experiences with optimization techniques for well-known tableau-based description logic reasoning systems, and analyzes the performance of very simple techniques to cope with Tboxes whose bulk axioms just use a less expressive language such as $\mathcal{ELH}$, whereas some small parts of the Tbox use a language as expressive as $\mathcal{SHIQ}$. The techniques analyzed in this paper have been tested with RacerPro, but they can be embedded into other tableau-based reasoners such as, e.g., Fact++ or Pellet in a seamless way.

1 Introduction

In practical applications involving ontologies, it should be the goal to formulate important tasks as reasoning problems, and existing description logic inference systems should be used to actually solve these problems. Due to our experiences, for specific tasks of this kind, very often expressive description logics such as $\mathcal{ALC}$ or $\mathcal{SHIQ}$ are used. For large parts of the application, however, a description logic such as $\mathcal{EL}$ or maybe $\mathcal{ELH}$ might be sufficient. For practical applications it is a realistic assumption that description logic reasoning systems should be tailored towards Tboxes consisting of very large sets of (acyclic) concept definitions with concept descriptions from, for instance, the $\mathcal{ELH}$ language on the one hand, and some smaller set of axioms involving $\mathcal{ALC}$ or $\mathcal{SHIQ}$ concept descriptions on the other. Due to our experiences, supporting this kind of Tboxes is of utmost importance for the acceptance of description logic reasoners in industrial contexts.

It has been argued in the literature [1–3] that less expressive languages have their merits for applications using very large Tboxes because in these less expressive languages, subsumption is a polynomial inference problem. Also, recently, non-obvious results have been achieved that identify subsumption as a polynomial problem even in the presence of generalized concept inclusions (GCIs) for the description logic $\mathcal{EL}^{++}$ [4–6].

Subsumption is an important inference problem in many application contexts, and it is the predominant inference problem used at development time when a Tbox is classified. Classification of large Tboxes is a well-investigated, but still not an easy inference problem from a practical point of view. Exploiting the results in [6], for this task very promising results have been reported on algorithms implemented in the CEL description logic system [7]. In this work, among

* This paper has been partially funded by the TONES Project, FP6-7603, 6th EU Framework Programme.

other knowledge bases, the authors consider a huge Tbox, which is a variant of SNOMED-CT [8] with approximately 380,000 concept names for which, to a large extent, corresponding concept definitions exist. If an application is to be built that uses a Tbox of this size, it might be the case that additional concept definitions using concept descriptions from $\mathcal{ALC}$ or even $\mathcal{SHIQ}$ should be added in order to solve certain application tasks using description logic reasoning. However, then, description logic inference systems tailored to the $\mathcal{EL}$ language (and its extension $\mathcal{EL}^{++}$) can no longer be used for classification purposes at the current state of the art. The problem is that one often cannot have the cake (description logics systems ensuring fast classification) and eat it too (i.e., use the expressive power of description logics for solving application problems).

Due to our experiences, for instance, value restrictions are indeed used in applications, and are supported by W3C syntaxes for description logic languages (such as OWL Lite or OWL DL). Thus, following our line of argumentation, in these kinds of applications, $\mathcal{SHIQ}$ reasoners such as FaCT++, Pellet, or RacerPro seem to be appropriate. Experiments have indicated that Tboxes of the SNOMED family are hard for these reasoners [5].

In this paper we summarize our experiences with optimization techniques for well-known tableau-based reasoning systems, and analyze the performance of very simple techniques to cope with Tboxes whose bulk axioms just use a less expressive language such as $\mathcal{ELH}$, whereas some small parts of the Tbox use a language as expressive or more expressive as $\mathcal{ALC}$. The techniques analyzed in this paper have been tested with RacerPro, but they can be embedded into other tableau-based reasoners such as, e.g., Fact++ or Pellet in a seamless way.

The paper is structured as follows. In Section 3 we first give a deeper introduction to the problems of tableau-based reasoning techniques in the context of classifying $\mathcal{ELH}$. Then, in Section 4, a technique is specified that fits well into tableau-based reasoning systems, and, due to our experiments in Section 5, can solve the classification problem for important example Tboxes of the class considered in this paper. Section 6 concludes the paper and analyses the pros and cons of the approach.

2 Syntax and semantics of $\mathcal{ALC}$ and $\mathcal{ELH}$

For a given application problem one chooses a set of elementary descriptions for *concepts* and *roles* representing unary and binary predicates, respectively. Elementary descriptions are also called *atomic descriptions*, or just *names* for brevity.

In the following, we use letters A and R for atomic concept and role descriptions, respectively. In $\mathcal{ALC}$ (Attributive Language with full Complement), *descriptions for complex concepts* C or D can be inductively built using the following grammar:

We introduce the concept descriptions $\top$ and $\bot$ as abbreviations for $A \sqcup \neg A$ and $A \sqcap \neg A$, respectively. Concept descriptions may be written in parentheses in order to avoid scoping ambiguities. A concept description C which is contained in a concept description D is called a *subconcept* (of D).

For defining the semantics of concept and role descriptions we consider *interpretations* $\mathcal{I}$ that consist of a non-empty set $\Delta^{\mathcal{I}}$, the domain, and an inter-

$$
\begin{array}{rll}
C, D \longrightarrow & A & \mid \text{atomic concept description} \\
& C \sqcap D & \mid \text{conjunction} \\
& C \sqcup D & \mid \text{disjunction} \\
& \neg C & \mid \text{negation} \\
& \exists R.C & \mid \text{existential restriction} \\
& \forall R.C & \mid \text{value restriction}
\end{array}
$$

pretation function $\cdot^{\mathcal{I}}$, which assigns to every atomic concept description A a set $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ and to every (atomic) role R a set $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. For complex concept descriptions the interpretation function is extended as follows:

$$
\begin{aligned}
(C \sqcap D)^{\mathcal{I}} &= C^{\mathcal{I}} \cap D^{\mathcal{I}} \\
(C \sqcup D)^{\mathcal{I}} &= C^{\mathcal{I}} \cup D^{\mathcal{I}} \\
(\neg C)^{\mathcal{I}} &= \Delta^{\mathcal{I}} \backslash C^{\mathcal{I}} \\
(\exists R.C)^{\mathcal{I}} &= \{x \mid \exists y.(x, y) \in R^{\mathcal{I}} \text{ and } y \in C^{\mathcal{I}}\} \\
(\forall R.C)^{\mathcal{I}} &= \{x \mid \forall y. \text{ if } (x, y) \in R^{\mathcal{I}} \text{ then } y \in C^{\mathcal{I}}\}
\end{aligned}
$$

The semantics of description logics is based on the notion of satisfiability. An interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ *satisfies* a concept description C if $C^{\mathcal{I}} \neq \emptyset$. In this case, $\mathcal{I}$ is called a *model* for C.

A *Tbox* is a set of so-called *generalized concept inclusions* $C \sqsubseteq D$. A concept definition $A \doteq C$ in a Tbox $\mathcal{T}$ is an abbreviation of $A \sqsubseteq C$ and $C \sqsubseteq A$, given there is no other GCI $A \sqsubseteq D$ in $\mathcal{T}$ and A is not referred to in C, either directly or indirectly via other GCIs.

An interpretation $\mathcal{I}$ *satisfies* a GCI $C \sqsubseteq D$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$. An interpretation is a *model* of a Tbox if it satisfies all GCIs in the TBox. A concept description C *is subsumed by* a concept description D w.r.t. a Tbox if the GCI $C \sqsubseteq D$ is satisfied in all models of the Tbox. In this case, we also say that D *subsumes* C, or D is a *subsumer of* C.

The set of *parents* of an atomic concept is the set of concept names which are most-specific subsumers w.r.t. a Tbox $\mathcal{T}$. The set of *children* is defined analogously. The *classification* problem for $\mathcal{T}$ is the problem of computing the parents for every concept name occurring in $\mathcal{T}$. The resulting lattice is called *taxonomy*.

2.1 Decision problems and their reductions

The definitions given in the previous section can be paraphrased as decision problems. The *concept satisfiability* problem is to check whether a model for a concept description exists. The Tbox satisfiability problem is to check whether there exists a model for the Tbox. The *concept subsumption* problem is to check whether $C \sqsubseteq D$ holds in all models of the Tbox. The *classification* problem is to compute the taxonomy, and is reduced to a quadratic number of concept subsumption problems in the worst-case.

2.2 The description logics $\mathcal{EL}$, $\mathcal{ELH}$, and others

Given the definition of $\mathcal{ALC}$ in the previous section, we introduce description logic $\mathcal{EL}$ by imposing restrictions on the set of possible complex concept descriptions. Only conjunction and existential restrictions are allowed in $\mathcal{EL}$. In $\mathcal{ELH}$ it is possible to additionally specify a set of role inclusions of the form $R \sqsubseteq S$ where R and S are atomic role descriptions. The set of superroles of a role w.r.t. a set of role inclusions is defined in the obvious way.

A role inclusion is satisfied if there exists an interpretation $\mathcal{I}$ such that $R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$. In addition to GCIs, role inclusions can appear in Tboxes in $\mathcal{ELH}$. Since $\mathcal{EL}$ (and $\mathcal{ELH}$) do not support negation, concept descriptions are always satisfiable (also with respect to Tboxes).

For a definition of the description logic $\mathcal{SHIQ}$ see [9]. $\mathcal{EL}^{++}$ is defined in [4].

3 The problem, an observation and a hypothesis

The problem with contemporary $\mathcal{SHIQ}$ classification techniques is that, in the worst case, subsumption tests for the form $C \sqsubseteq D$ with respect to a Tbox $\mathcal{T}$ are translated to a test of the form $\neg SAT_{\mathcal{T}}(C \sqcap \neg D)$, i.e., subsumption is reduced to non-satisfiability. The introduction of negation might introduce disjunctions. Since concept descriptions can be deeply nested and usually introduce many conjuncts, due to the negation, many disjuncts are the result. In tableau-based reasoners, a technique called lazy evaluation [10] is used, with the effect that even more disjunction appear implicitly when concept names are "replaced" with their definition (see [10] for details).

During Tbox classification very many subsumption tests have to be made, and concerning practical Tboxes it is well-known that most of these tests return false, i.e, $C \sqcap \neg D$ is indeed satisfiable. With the underlying tableau-based prover technology, the search for a model in the space of disjunctions is slow, however. The subsumption problem formulated in this way is solved with an exponential algorithm due to the introduction of negation. It is not the case that in the experiments discussed below for a specific subsumption test a combinatorial explosion occurs, causing the test to run hours. But, although a single subsumption test does not matter much for the whole classification process, at the current state of the art the tableau prover machinery just seems to be too heavy-weight, and the enormous amount of subsumption tests provides for a situation in which classifications times are not tolerable.

An initial experiment with RacerPro (see also [5]) indicated that the above-mentioned brute-force approach needed days to (correctly) classify a variant of SNOMED-CT that uses the language $\mathcal{ELH}$ [11]. We call this version SNOMED-$\mathcal{ELH}$. Note that RacerPro implements standard optimization techniques for classification as described in [12, 13] as well as some others [14]. In addition, as analyzed by Tsarkov et al. [15], in Tboxes of this kind for quite a substantial number of concept names the axioms directly indicate the correct place in the taxonomy (i.e., the parents and children can be identified using static analysis). The technique is known as the identification of "completely defined concepts" (CD optimization, [15]). Our experiments with an implementation of this technique in RacerPro also indicate the effectiveness of this method (see below).

However, still we were not satisfied with classification runtimes (a couple of hours for SNOMED-$\mathcal{ELH}$).

A backbone optimization technique for classifying Tboxes using tableau-based reasoners is called "pseudo model merging" [13]. Every concept name CN satisfiable w.r.t. a Tbox $\mathcal{T}$ is associated with a so-called pseudo model data structure $M = (L, \neg L, S^{\exists}, S^{\forall})$ where L is a set of positive literals (concept names), $\neg L$ is a set of negative literals, $S^{\exists}$ is a set of concept descriptions of the form $\exists R.C$, and $S^{\forall}$ is a set of concept descriptions of the form $\forall R.C$. The data structure is computed in the obvious way from the label of the root node in a tableau test to implement $SAT_{\mathcal{T}}$ for a concept description C. The idea of the algorithm is to show that C and $\neg D$ do neither "interact" in terms of positive and negative literals nor "interact" w.r.t. existential restrictions and value restrictions (see [13] and Section 4 for details). If there is no interaction, the conjunction $C \sqcap \neg D$ is satisfiable, and C is not subsumed by D.

Although pseudo model merging can be seen as a cheap trick, it is surprisingly effective for a variety of expressive description logics [16]. However, our tests indicate that this backbone optimization technique for classifying Tboxes is not effective at all for SNOMED-$\mathcal{ELH}$. Indeed, the technique just introduces overhead if used in the standard way. Since subsumption tests introduce disjunctions (see the arguments above), in many cases, the "wrong" pseudo models are tried in the merging process (for details of the merging process see [13, 16]), and the computational efforts do not lead to any results. On the contrary, the heavy-weight tableau machinery is still needed.

The observation that realistic knowledge bases in use today use inexpressive languages for most parts of the Tboxes led us to the insight that one might use the pseudo models computed anyhow in tableau-based reasoners in a different way. Instead of "merging" the pseudo models computed for C and $\neg D$ (in the sense of model merging) one could try to "embed" the pseudo model of D into the pseudo model of C. This technique has been used for many years in description logic systems and is known as structural subsumption. Deep pseudo models (with pseudo models computed also for the qualifications in existential restrictions) as introduced in [16] can be seen as so-called description graphs [3]. An algorithm that checks whether the description graphs can be embedded can be used to decide subsumption in case certain conditions are met (for details see below).

There exists an algorithm for this problem that runs in polynomial time (and it works bottom-up). A simple graph-embedding algorithm, which is the basis of previous structural subsumption algorithms (back to KL-ONE), works top-down. It takes two deep pseudo models (or description graphs) as input and checks for graph embeddings with respect to existential restrictions. In the worst-case it is exponential, however.

Static analysis has to reveal in which cases the structural subsumption algorithm can be applied, and in which cases the tableau-based techniques are really required to be sound and complete. It was our research hypothesis that for Tboxes such as SNOMED-$\mathcal{ELH}$ augmented with few $\mathcal{SHIQ}$ axioms an "acceptable" performance for classification can be achieved.

This paper describes our experiments and the results we obtained w.r.t. whether the hypothesis can be verified or not. To our surprise, structural subsumption, i.e., a prestidigitation, implemented with an algorithm that is expo-

nential in the worst case, can dramatically speed up classification times while retaining soundness and completeness. SNOMED-$\mathcal{ELH}$ with 380,000 concept names and augmented with some $\mathcal{SHIQ}$ axioms can be classified in 12 minutes on a standard laptop computer.

4 Structural subsumption as model embedding

Model merging is a standard optimization technique for description logic classifiers, and we define the procedure here for the sake of completeness by considering the case for the description logic $\mathcal{ALC}$. The approach can easily be extended to more expressive description logics such as $\mathcal{SHIQ}$.

The idea of merging pseudo models (pmodels) for a subsumption test $C \sqsubseteq D$ is to evaluate $pmodels\_mergable(pmodel(C), pmodel(\neg D))$ whose definition is given in Algorithm 1.

Algorithm 1 $pmodels\_mergable?((L_1, \neg L_1, S_1^{\exists}, S_1^{\forall}), (L_2, \neg L_2, S_2^{\exists}, S_2^{\forall}))$

return $(L_1 \cap \neg L_2 = \emptyset) \wedge (\neg L_1 \cap L_2 = \emptyset)$
$\wedge$ **not exists** $(\exists R.C) \in S_1^{\exists}$ **such that**
there exists $(\forall S.D) \in S_2^{\forall}$ **with** $S \in ancestors(R)$
$\wedge$ **not exists** $(\exists R.C) \in S_2^{\exists}$ **such that**
there exists $(\forall S.D) \in S_1^{\forall}$ **with** $S \in ancestors(R)$

The function *ancestors* applied to a role R returns the set of superroles of R including R.

Since the pseudo models are computed anyway during classification in a standard tableau-based classification algorithm, the idea is to exploit these data structures also in a different way, namely in a structural subsumption test for the language $\mathcal{ELH}$.

Every $\mathcal{ELH}$ concept C is associated with a pseudo model data structure $M = (L, \{\}, S^{\exists}, \{\})$, i.e., there are no negated literals and no value restrictions. We assume that the data structure $pmodel(C)$ being associated with C is computed on demand (with respect to a given Tbox). As has been mentioned before, an $\mathcal{ELH}$ concept is always satisfiable.

If now during classification it is to be checked whether A_1 subsumes A_2 this can be done with a call to Algorithm 2 in the form of

$$pmodel\_embeddable?(pmodel(A_1), pmodel(A_2))$$

Algorithm 2 $pmodel\_embeddable?((L_1, \{\}, S_1^{\exists}, \{\}), (L_2, \{\}, S_2^{\exists}, \{\}))$

return $filter(L_1) \subseteq L_2$
$\wedge$ **for all** $(\exists R.C) \in S_1^{\exists}$
there exists $(\exists S.D) \in S_2^{\exists}$ **such that**
$R \in ancestors(S)$
$\wedge\ pmodel\_embeddable?(pmodel(C), pmodel(D))$

Note that we do not claim that the idea behind Algorithm 2 is new. The algorithm is presented here in order to discuss details in the context of pseudo models.

The function $filter$ is used for the following purpose. Given a Tbox

$$\{A \doteq \exists R.C, \quad A_1 \sqsubseteq B \sqcap \exists R.C, \quad A_2 \doteq B \sqcap A\}$$

it is obvious that A_1 is subsumed by A_2, and it holds that[3]

$$pmodel(A_1) = (\{A_1, B\}, \{\}, \{\exists R.C\}, \{\})$$

$$pmodel(A_2) = (\{A_2, B, A\}, \{\}, \{\exists R.C\}, \{\}).$$

However, the pseudo model of A_2 cannot be embedded into the pseudo model of A_1 due to the defined concepts A_2 and A in the literal list. The function $filter$ eliminates the concept names for which there exists a concept definition (and keeps those for which there exists no definitions or only GCIs with the name on the left-hand side).

The information encoded in the pseudo model is usually exploited for the model merging process. For the implementation of model embedding, however, defined concepts such as A_1 should be transparent. The function $filter$ used in Algorithm 2 eliminates all concept names from L_1 for which there exists a concept definition and keeps the names for which there exists a GCI with the name on the left-hand side.

It can be easily seen that Algorithm 2 can exhibit exponential behavior in the worst case. Our conjecture was that for realistic knowledge bases, the combinatorial behavior just does not occur because, usually, there are only very few existential restrictions for a particular role R used in a concept definition. In order to verify this hypothesis we wanted to conduct several experiments with SNOMED-$\mathcal{ELH}$. The goal was however, not to build a dedicated $\mathcal{ELH}$ prover but to integrate this technique into a standard $\mathcal{SHIQ}$ prover such as RacerPro. The above-mentioned model embedding technique should hardly introduce any overhead since pseudo models are computed anyway for the standard model merging optimization techniques (see above). Nevertheless, a static analysis is required in order to identify when the model embedding technique is actually applicable for two concepts A_1, A_2 to be checked for subsumption.

The applicability conditions of *pmodel_embeddable*? are defined as specified in Algorithm 3.

Algorithm 3 *pmodel_embedding_applicable*?(A_1, A_2, T)

return $\neg$*meta_constraints*?(T)
$\wedge$ *elh_concept*?$(A_1, T) \wedge$ *elh_concept*?(A_2, T)
$\wedge \neg$*self_referencing*?$(A_1, T) \wedge \neg$*self_referencing*?(A_2, T)

The function *meta_constraint*? checks if GCI absorption was successful, i.e., the function tests whether there are GCIS left after GCI absorption. The function

[3] We assume that taxonomic encoding [13] is not used.

elh_concept? exploits results from a static analysis of the axioms in the Tbox. For a concept name A it is determined whether all subconcepts "reachable" via concept definitions or GCIs are $\mathcal{ELH}$ concepts. The function *self_referencing*? determines whether a concept name A is cyclic with respect to the Tbox axioms.

In Algorithm 4 the subsumption test is defined. The function *shiq_subsumes*? applies the standard subsumption test with optimization techniques as described in [12–14].

Algorithm 4 *subsumes*?(A_1, A_2, T)

if *pmodel_embedding_applicable*?(A_1, A_2, T) **then**
  return *pmodel_embeddable*?$(pmodel(A_1), pmodel(A_2))$
else
  return *shiq_subsumes*?(A_1, A_2, T)

5 Evaluation

We conducted several experiments to verify the hypothesis that Algorithm 2, which is worst-case exponential, dramatically speeds up classification times for SNOMED-$\mathcal{ELH}$ even in the case where there are some GCIs that use $\mathcal{ALC}$ concept descriptions.

Table 1 shows the results that we obtained for classifying SNOMED-$\mathcal{ELH}$ with approximately 380,000 concept names using RacerPro 1.9.2 beta (Intel 2.4GHz, Core 2 Duo, Mac OS X, Leopard, 64bit).

CD optimization	pseudo model embedding	*ELH*?	runtime
disabled	disabled	yes	several days
enabled	disabled	yes	several hours
disabled	enabled	yes	minutes (15 min)
enabled	enabled	yes	minutes (12 min)

Table 1. Effects of optimization techniques.

The results shown in Table 1 reveal that the CD optimization technique (see Section 3) has a dramatic effect (line 2) but model embedding has an even larger effect (line 3). Using both techniques in combination yields a slightly better performance (line 4).

We have also run CEL on SNOMED-$\mathcal{ELH}$ (Intel Pentium IV 2.8 GHz, Linux, 32bit). RacerPro and CEL need approximately the same time (half an hour on the Intel P4 machine under Linux).

RacerPro can still be used if we "spoil" the Tbox with an $\mathcal{ALC}$ axiom of the form $A_1 \sqsubseteq \forall R.A_2$ for arbitrary SNOMED-$\mathcal{ELH}$ concept names A_1 and A_2 and role names R. Further experiments have been carried out to investigate in which situations the performance will degrade. Figure 1 indicates that the effect is

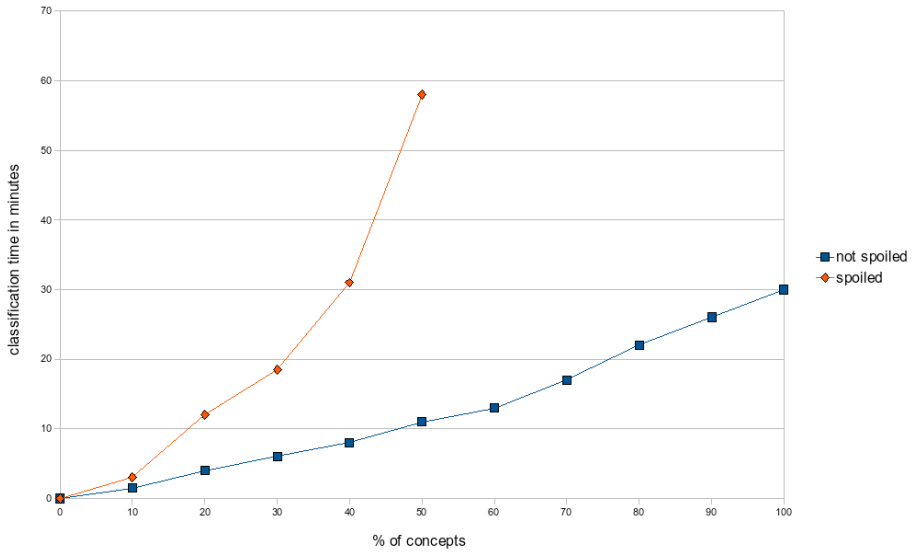

Fig. 1. A SNOMED classification test (Intel Pentium IV 2.8 GHz, Linux, 32bit) with value restrictions added to the definition of 50 randomly selected parents of $\bot$. An increasing fragment of the Tbox is used (10%, 20%, ..., 100%). The line with boxes indicates the runtimes (in minutes) without value restrictions, the line with diamonds shows the runtimes with "spoiling" value restrictions.

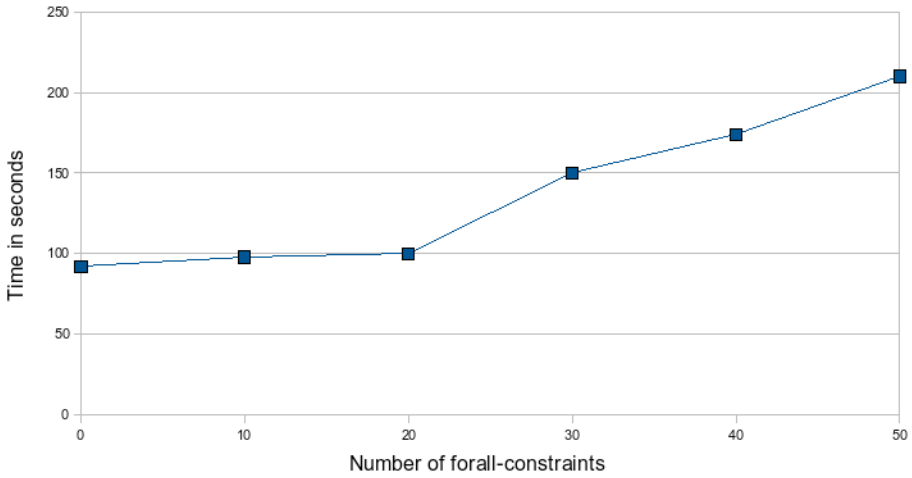

Fig. 2. A SNOMED classification test (Intel Pentium IV 2.8 GHz, Linux, 32bit) with an increasing number of value restrictions added to the definition of randomly selected parents of $\bot$. Only the first 10 % of SNOMED are used for this test (37900 concept names).

dramatic. A few value restrictions can cause long classification times. The effect is less dramatic if there are "few" concept names in the Tbox (see Figure 2).

Note that we are aware of the fact that RacerPro cannot even be used in cases where CEL could be expected to still be quite fast, i.e., if the extensions that $\mathcal{EL}^{++}$ includes were used in the Tbox. In addition, the optimization techniques presented in this paper cannot be used in the presence of cycles. The cases in which the techniques can indeed be used are automatically detected, however. CEL cannot be run in the spoiled case.

6 Conclusion

The techniques investigated in this paper are attractive for tableau-based description logic systems because they are very easy to implement. Data structures being computed anyway (pseudo models) are now also used for model embedding as a kind of structural subsumption test. The proposed technique requires a static analysis for implementing the applicability check defined in Algorithm 3. We conjecture that the information needed in Algorithm 3 is already computed in all optimized tableau-based reasoners existing today.

The evaluation with SNOMED-$\mathcal{ELH}$ shows encouraging results. Nevertheless, the tests reveal that even a relatively small number of value restrictions can cause problems with large Tboxes. Nevertheless, we can have the bulk $\mathcal{ELH}$ Tboxes that meet sporadic $\mathcal{SHIQ}$ requirements for small parts.

Furthermore, we found it surprising that an exponential algorithm, namely a (naive) structural subsumption test, can be used to classify Tboxes such as SNOMED-$\mathcal{ELH}$. With *pmodel_embeddable*? the "right structures" are made available. The potential problem that there are many non-effective recursive calls of *pmodel_embeddable*? does not occur in practical Tboxes, and SNOMED-$\mathcal{ELH}$ seems to be a good example for this effect. An implementation of a polynomial algorithm for *pmodel_embeddable*? and its evaluation in the context of SNOMED-$\mathcal{ELH}$ is left for further studies.

The experiments shed some light on the real problem with tableau-based subsumption tests. The problem is not the exponential worst-case behavior per se. The worst-case-exponential algorithm for *pmodel_embeddable*? works fine for SNOMED-$\mathcal{ELH}$. In our opinion, the problem is that in standard tableau reasoners the pseudo models are retained, but the tableau structures used for deciding the problem $subsumes?(D, C) = \neg SAT(C \sqcap \neg D)$ are always discarded after the test is completed. This could be avoided if also tableau structures for C and $\neg D$ were retained and somehow manipulated in an effective way to get the answer. We conjecture that it would be possible to also deal with, for instance, value restrictions effectively in a tableau-based algorithm if the "right structures" were kept between multiple calls to *subsumes*. This is subject to further research as well.

References

1. Brachman, R.J., Levesque, H.J.: The tractability of subsumption in frame-based description languages. In: Proc. of the 4th Nat. Conf. on Artificial Intelligence (AAAI'84). (1984) 34–37

2. Borgida, A., Patel-Schneider, P.F.: A semantics and complete algorithm for subsumption in the CLASSIC description logic. J. of Artificial Intelligence Research **1** (1994) 277–308
3. Baader, F.: Terminological cycles in a description logic with existential restrictions. In Gottlob, G., Walsh, T., eds.: Proceedings of the 18th International Joint Conference on Artificial Intelligence, Morgan Kaufmann (2003) 325–330
4. Brandt, S.: Polynomial time reasoning in a description logic with existential restrictions, GCI axioms, and—what else? In de Mantáras, R.L., Saitta, L., eds.: Proceedings of the 16th European Conference on Artificial Intelligence (ECAI-2004), IOS Press (2004) 298–302
5. Baader, F., Lutz, C., Suntisrivaraporn, B.: Is tractable reasoning in extensions of the description logic $\mathcal{EL}$ useful in practice? In: Proceedings of the Methods for Modalities Workshop (M4M-05), Berlin, Germany (2005)
6. Baader, F., Brandt, S., Lutz, C.: Pushing the $\mathcal{EL}$ envelope. In: Proc. of the 19th Int. Joint Conf. on Artificial Intelligence (IJCAI 2005). (2005) 364–369
7. Baader, F., Lutz, C., Suntisrivaraporn, B.: CEL—a polynomial-time reasoner for life science ontologies. In Furbach, U., Shankar, N., eds.: Proceedings of the 3rd International Joint Conference on Automated Reasoning (IJCAR'06). Volume 4130 of Lecture Notes in Artificial Intelligence., Springer-Verlag (2006) 287–291
8. Spackman, K.A., Campbell, K.E., Cote, R.A.: SNOMED RT: A reference terminology for health care. J. of the American Medical Informatics Association (1997) 640–644 Fall Symposium Supplement.
9. Horrocks, I., Sattler, U., Tobies, S.: Reasoning with individuals for the description logic $\mathcal{SHIQ}$. In McAllester, D., ed.: Proc. of the 17th Int. Conf. on Automated Deduction (CADE 2000). Volume 1831 of Lecture Notes in Computer Science., Springer (2000) 482–496
10. Horrocks, I., Tobies, S.: Reasoning with axioms: Theory and practice. In: Proc. of the 7th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR 2000). (2000) 285–296
11. Spackman, K.A.: Managing clinical terminology hierarchies using algorithmic calculation of subsumption: Experience with snomed-rt. J. of the American Medical Informatics Association (2000) Fall Symposium Special Issue.
12. Baader, F., Franconi, E., Hollunder, B., Nebel, B., Profitlich, H.J.: An empirical analysis of optimization techniques for terminological representation systems or: Making KRIS get a move on. Applied Artificial Intelligence. Special Issue on Knowledge Base Management **4** (1994) 109–132
13. Horrocks, I.: Optimisation techniques for expressive description logics. Technical Report UMCS-97-2-1, University of Manchester, Department of Computer Science (1997)
14. Haarslev, V., Möller, R.: High performance reasoning with very large knowledge bases: A practical case study. In: Proc. of the 17th Int. Joint Conf. on Artificial Intelligence (IJCAI 2001). (2001) 161–168
15. Tsarkov, D., Horrocks, I., Patel-Schneider, P.F.: Optimizing terminological reasoning for expressive description logics. J. of Automated Reasoning **39** (2007) 277–316
16. Haarslev, V., Möller, R., Turhan, A.Y.: Exploiting pseudo models for TBox and ABox reasoning in expressive description logics. In: Proc. of the Int. Joint Conf. on Automated Reasoning (IJCAR 2001). Volume 2083 of Lecture Notes in Artificial Intelligence., Springer (2001)