Abox Satisfiability Reduced to Terminological Reasoning in Expressive Description Logics

Sergio Tessaris[1] and Ian Horrocks[2]

[1] Department of Computation
UMIST
Manchester, UK
sergio@co.umist.ac.uk
[2] Department of Computer Science
University of Manchester
Manchester, UK
horrocks@cs.man.ac.uk

Abstract. Description Logics knowledge bases are traditionally divided into a terminological part (Tbox), and an assertional part (Abox). However, most of recent results on practical algorithms are confined to terminological reasoning only. Due to the applications of Description Logics for databases and the so-called "Semantic Web", there is a growing interest for practical algorithms for Abox reasoning. In this paper we present an algorithm for deciding knowledge base satisfiability based on the idea of separating terminological and assertional reasoning. This modularity allows to build complete Description Logics systems, by reusing available terminological reasoners.

1 Introduction

Although Description Logics (DLs) have proved useful in a range of application domains, e.g., configuration [1] and reasoning about database schemas and queries [2], their original development was motivated by the desire to give a well defined and implementation independent semantics to knowledge representation systems based on semantic networks and frames [3], and to provide automated reasoning services for interesting inference problems, e.g., concept subsumption. Subsequent research has investigated the computational complexity of such inference problems for different DLs, and the design and implementation of (sound and complete) algorithms capable of solving these problems. Recently, the focus of this work has been on the development of algorithms and optimised implementations for increasingly expressive DLs, e.g., those supporting transitive roles (binary relations) and general inclusion axioms.[1]

The knowledge represented in a DL based system is often divided into two parts: a "terminological" part (called the Tbox) and an "assertional" part (called

[1] I.e., axioms asserting a subsumption relationship between arbitrarily complex concepts.

the Abox). The Tbox defines the structure of the domain, and consists of a set of axioms asserting, e.g., that one concept (class) subsumes (is a superclass of) another; the Abox describes a concrete example of the domain, and consists of a set of axioms asserting, e.g., that an individual is an instance of a concept or that one individual is related to another by a given role. Interesting inference problems for a DL based knowledge representation system include concept satisfiability and subsumption, realisation (what is the most specific concept an individual is an instance of) and retrieval (which individuals are instances of a given concept).

The use of DLs in knowledge representation has been highlighted by the recent explosion of interest in the so-called "Semantic Web" [4], where DLs are set to provide both the formal underpinnings and automated reasoning services for Semantic Web knowledge representation languages such as DAML+OIL [5]. The DL based design of these languages allows them to exploit both formal results (e.g., w.r.t. the decidability and complexity of key inference problems [6]) and implemented systems from DL research. Reasoning with Aboxes is likely to be of increasing importance, e.g., in Semantic Web applications, where it will be necessary not only to reason with concepts, but also with individuals (web resources) that instantiate them, and in particular to answer queries over sets of such individuals (e.g., see [7]).

Most of the current DL systems (for example RACER [8]) use optimised versions of the tableaux–based algorithm described in [9] for deciding the satisfiability of a Tbox or knowledge base.[2] In this paper we present an alternative algorithm based on the *precompletion* technique introduced in [10, 11]. We extended the original work to deal with transitive roles, functional roles and general inclusion axioms.

The main idea behind this technique is to eliminate Abox axioms specifying relationships between individuals by explicating the consequences of such relationships. Once these axioms have been eliminated, the assertions about a single individual can be independently verified using a standard Tbox reasoner. The algorithm is completely independent of the Tbox reasoner, so it can exploit existing systems with highly optimised Tbox reasoners, and could be used to add an Abox to such a system without re-implementing the Tbox reasoner [12].

The main purpose of this paper is to present the precompletion algorithm and to provide evidence of its correctness and completeness. However, in Section 5 we briefly comment on a prototypical implementation of this algorithm, and its empirical evaluation.[3]

The proof for correctness and completeness of the technique is in two parts: in Section 3 we present the method to enumerate the so called *precompletions* of a KB. We show that a KB is satisfiable iff one of these precompletions is satisfiable. In the second part (Section 4) we show that, for checking the satisfiability of a precompletion, role assertions can be ignored. In a precompletion the relevant elements are the concept assertions and the terminology. Each individual

[2] Many interesting inference problems, including both realisation and retrieval, can be reduced to knowledge base satisfiability.

[3] A more extended discussion can be found in [12] or [13].

is associated to its *individual concept*, which is the conjunction of the concepts appearing in Abox assertions about the individual itself. If all the individual concepts are satisfiable (with respect to the terminology), then their models can be combined in an interpretation satisfying the precompleted knowledge base.

Due to space restrictions, most of the proofs are only sketched or omitted; their full version can be found in Chapter 5 of [13].[4]

2 Preliminaries

2.1 $\mathcal{SH}f$ knowledge bases

The DL $\mathcal{SH}f$ is built over a signature of distinct sets of concept ($\mathcal{CN}$), role ($\mathcal{RN}$) and individual ($\mathcal{O}$) names. In addition, we distinguish two non-overlapping subsets of $\mathcal{RN}$ ($\mathcal{TRN}$ and $\mathcal{FRN}$) which denote the transitive and the functional roles. The set of all $\mathcal{SH}f$ concepts is the smallest set such that every concept name in $\mathcal{CN}$ and the symbols $\top$, $\bot$ are concepts, and if C and D are concepts and R a role name in $\mathcal{RN}$, then $\neg C$, $(C \sqcap D)$, $(C \sqcup D)$, $(\forall R.C)$, and $(\exists R.C)$ are concepts.

An *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consists of a nonempty domain $\Delta^{\mathcal{I}}$ and a interpretation function $\cdot^{\mathcal{I}}$. The interpretation function maps concepts into subsets of $\Delta^{\mathcal{I}}$, individual names into different elements of $\Delta^{\mathcal{I}}$,[5] and role names into binary relations over $\Delta^{\mathcal{I}}$ (i.e. subsets of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$). Complex concept expressions are interpreted according to the following equations (see [14])

$$\begin{aligned}
\top^{\mathcal{I}} &= \Delta^{\mathcal{I}} \qquad & (C \sqcap D)^{\mathcal{I}} &= C^{\mathcal{I}} \cap D^{\mathcal{I}} \\
\bot^{\mathcal{I}} &= \emptyset & (C \sqcup D)^{\mathcal{I}} &= C^{\mathcal{I}} \cup D^{\mathcal{I}} \\
& & \neg C^{\mathcal{I}} &= \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}} \\
(\forall R.C)^{\mathcal{I}} &= \left\{x \in \Delta^{\mathcal{I}} \mid \forall y (x,y) \in R^{\mathcal{I}} \Rightarrow y \in C^{\mathcal{I}}\right\} \\
(\exists R.C)^{\mathcal{I}} &= \left\{x \in \Delta^{\mathcal{I}} \mid \exists y (x,y) \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\right\}
\end{aligned}$$

In addition, the interpretation function must satisfy the transitive and functional restrictions on role names; i.e. for any $R \in \mathcal{TRN}$ if $(x,y) \in R^{\mathcal{I}}$ and $(y,z) \in R^{\mathcal{I}}$, then $(x,z) \in R^{\mathcal{I}}$, and for any $F \in \mathcal{FRN}$ if $(x,y) \in F^{\mathcal{I}}$ and $(x,z) \in F^{\mathcal{I}}$, then $y = z$.

A $\mathcal{SH}f$ knowledge base (KB) is a pair $\langle \mathcal{T}, \mathcal{A} \rangle$, where $\mathcal{T}$ is called *Tbox* and $\mathcal{A}$ is called *Abox*. The Tbox contains a finite set of axioms of the form $C \sqsubseteq D$ or $R \sqsubseteq S$; while $\mathcal{A}$ is the Abox and contains a finite set of assertions of the form $a{:}C$ or $\langle a,b \rangle{:}R$ (where C, D are $\mathcal{SH}f$ concepts, R, S role names, and a, b individual names). Intuitively, axioms in the Tbox describe intensional properties of all the elements of the domain, while assertions in the Abox assign properties of some named elements. We say that an interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ satisfies the axiom $C \sqsubseteq D$ ($R \sqsubseteq S$) iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ ($R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$), and the assertion $a{:}C$ ($\langle a,b \rangle{:}R$) iff $a^{\mathcal{I}} \in C^{\mathcal{I}}$ ($(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$).

[4] Its electronic version is available at `http://www.cs.man.ac.uk/~tessaris/papers/phd-thesis.ps.gz`

[5] This corresponds to the so-called Unique Name Assumption (UNA).

Role inclusion axioms contain only role names, and if there are cyclical definitions (e.g. $S \sqsubseteq R$ and $R \sqsubseteq S$), all the names involved in the cycle must correspond to the same binary relation in every interpretation (satisfying the axioms). For these reasons, we assume that role axioms are summarised by a partial order $\preceq$ defined over the set of role names.

We assumed that transitive and functional role names are distinct, and it is easy to realise that a subrole of a functional role is functional as well. Therefore, we impose the restriction that transitive roles cannot be sub-roles of any functional role.

2.2 Technical Definitions

Without loss of generality we assume that all the concept axioms in the Tbox are in the form $\top \sqsubseteq C$, where C is a concept expression. In DLs closed under negation, this assumption is not restrictive at all. Since, an arbitrary assertion $C_1 \sqsubseteq C_2$ can be transformed into the equivalent assertion $\top \sqsubseteq (\neg C_1 \sqcup C_2)$, which is in the required form.

A second assumption we adopt is that concept expressions are in *negation normal form*, where the $\neg\cdot$ constructor can appear only in front of concept names. Any concept expression can be transformed into an equivalent expression in normal form using the following rewriting rules.

$$\neg\neg C \equiv C \qquad \neg(C \sqcap D) \equiv \neg C \sqcup \neg D \qquad \neg\exists R.C \equiv \forall R.\neg C$$
$$\neg(C \sqcup D) \equiv \neg C \sqcap \neg D \qquad \neg\forall R.C \equiv \exists R.\neg C$$

Given a knowledge base $\Sigma = \langle \mathcal{T}, \mathcal{A} \rangle$, the *label* of an individual (written as $\mathcal{L}(\Sigma, o)$) is the set of concept expressions in the assertions on the individual itself. This is formally defined by $\{C \mid o{:}C \in \mathcal{A}\}$ if this set is not empty, or $\{\top\}$ otherwise. The *individual concept* expression $\bigsqcap \mathcal{L}(\Sigma, o)$ is defined as the conjunction of all the concept expressions in $\mathcal{L}(\Sigma, o)$: $C_1 \sqcap \ldots \sqcap C_n$ where $\{C_i \mid i = 1, \ldots, n\} = \mathcal{L}(\Sigma, o)$.

In order to simplify the formulae that define the precompletion algorithm, we define the role binary operators $\cdot \overset{o}{\approx}_{\mathcal{A}} \cdot$ (depending on the individual name o in $\mathcal{O}$, and an Abox $\mathcal{A}$). Intuitively, these operators take into account the possible interaction between the role hierarchy and the functional restrictions. An operator $\cdot \overset{o}{\approx}_{\mathcal{A}} \cdot$ holds between two role names R and S if they are functional, and the Abox assertions force the R and S successors of the individual name o to be the same element.

Definition 1. *Given two roles R and S, an individual o, and a KB $\Sigma = \langle \mathcal{T}, \mathcal{A} \rangle$, $R \overset{o}{\approx}_{\mathcal{A}} S$ holds iff:*

- *there is a role $R_0 \preceq R$ s.t. either $o{:}\exists R_0.C_0$ or $\langle o, o' \rangle{:}R_0$ is in $\mathcal{A}$; and*
- *there is a role $S_0 \preceq S$ s.t. either $o{:}\exists S_0.D_0$ or $\langle o, o'' \rangle{:}S_0$ is in $\mathcal{A}$; and*
- *there is a set of roles $\{R_1, \ldots R_{n-1}\}$, and a set of functional roles $\{F_1, \ldots, F_n\}$, s.t.*

- *either* $o{:}\exists R_i.C_i$ *or* $\langle o, o' \rangle{:}R_i$ *is in* $\mathcal{A}$, *for any* $i = 1, \ldots, n-1$ *and*
- $R \preceq F_1,\ S \preceq F_n,\ R_1 \preceq F_1,\ R_1 \preceq F_2,\ R_2 \preceq F_2,\ \ldots,\ R_{n-1} \preceq F_n.$

The $\cdot \overset{o}{\approx}_{\mathcal{A}} \cdot$ relation can be better understood by considering that it is describing a situation in which part of the role taxonomy looks like

$$
\begin{array}{ccccccccccccc}
 & F_1 & & F_2 & & & & F_{n-1} & & & F_n & & \\
\succeq & & \preceq\ \succeq & & \preceq & & \succeq & & \preceq & \succeq & & \preceq & \\
R & & R_1 & & & \cdots & & & R_{n-1} & & & & S \\
\curlyeqsucc & & & & & & & & & & & & \curlyeqsucc \\
R_0 & & & & & & & & & & & & S_0
\end{array}
$$

and for each of the role names $R,\ R_1, \ldots, R_{n-1},\ S$ the individual name o has a successor. In this case, the functional restrictions cause all these successors to be interpreted as the very same element. By the way it is defined, the relation $\cdot \overset{o}{\approx}_{\mathcal{A}} \cdot$ is symmetric (i.e. $R \overset{o}{\approx}_{\mathcal{A}} S \Rightarrow S \overset{o}{\approx}_{\mathcal{A}} R$). Next proposition shows that the definition captures the intuition behind the operators.

Proposition 1. *Let* $\Sigma = \langle \mathcal{T}, \mathcal{A} \rangle$ *be a KB, and* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ *be an interpretation satisfying* Σ. *For any individual name* o, *role names* R, S, *and elements* x, y *of* $\Delta^{\mathcal{I}}$, *if* $(o^{\mathcal{I}}, x) \in R^{\mathcal{I}}$, $(o^{\mathcal{I}}, y) \in S^{\mathcal{I}}$, *and* $R \overset{o}{\approx}_{\mathcal{A}} S$, *then* $x = y$.

Proof. All the roles $R, S, R_1, \ldots, R_{n-1}$ of Definition 1 are functional because they are included in functional roles $(F_1, \ldots, F_n)$; therefore $o^{\mathcal{I}}$ has at most one successor for any of these roles. We are going to show that all these successors are equal.

Note that for any R_i, the constraint $o{:}\exists R_i.C_i$ (or $\langle o, o' \rangle{:}R_i$) implies the existence of an element x_i in $\Delta^{\mathcal{I}}$ s.t. $(o^{\mathcal{I}}, x_i) \in {R_i}^{\mathcal{I}}$.

Let us consider $(o^{\mathcal{I}}, x_1) \in {R_1}^{\mathcal{I}}$, and $(o^{\mathcal{I}}, x) \in R^{\mathcal{I}}$. Since ${R_1}^{\mathcal{I}} \subseteq {F_1}^{\mathcal{I}}$ and $R^{\mathcal{I}} \subseteq {F_1}^{\mathcal{I}}$, then $\big\{(o^{\mathcal{I}}, x_1), (o^{\mathcal{I}}, x)\big\} \subseteq {F_1}^{\mathcal{I}}$. From the functionality of ${F_1}^{\mathcal{I}}$ we can conclude that $x_1 = x$.

The very same arguments can be applied to all pair of roles R_i, R_{i+1}, including the last R_{n-1}, S; therefore $x = x_1 = x_2 = \ldots = x_{n-1} = y$.

3 Precompletions of knowledge bases

Intuitively, a knowledge base is precompleted if all the information entailed by the presence of role assertions is exhibited in the form of concept assertions. The definition of a precompletion for a knowledge base $\langle \mathcal{T}, \mathcal{A} \rangle$ is given in a procedural way, as a new KB $\langle \mathcal{T}, \mathcal{A}_{pc} \rangle$ where the Abox is extended using the syntactic rules of Figure 1 as long as they are applicable. Due to the nondeterminism of the rules, several precompletions can be derived. A knowledge base is satisfiable if and only if a satisfiable precompletion can be derived.

The precompletion rules are designed in such a way that, whatever strategy of application is chosen, the process of completing a knowledge base always terminates (Proposition 2). The precompletion algorithm generates a finite, but possibly exponential, number of precompletions; which can be individually checked

$\mathcal{A} \rightarrow_{\sqsubseteq} \{o{:}C\} \cup \mathcal{A}$
if o is in $\mathcal{O}$, $\top \sqsubseteq C$ is in $\mathcal{T}$
and $o{:}C$ is not in $\mathcal{A}$.

$\mathcal{A} \rightarrow_{\sqcup} \{o{:}D\} \cup \mathcal{A}$
if $o{:}C_1 \sqcup C_2$ is in $\mathcal{A}$,
and $D = C_1$ or $D = C_2$
and neither $o{:}C_1$ nor $o{:}C_2$ is in $\mathcal{A}$.

$\mathcal{A} \rightarrow_{\exists^1} \{o'{:}C\} \cup \mathcal{A}$
if $o{:}\exists R.C$ and $\langle o, o' \rangle{:}S$ are in $\mathcal{A}$,
$R \stackrel{o}{\approx}_{\mathcal{A}} S$, and $o'{:}C$ is not in $\mathcal{A}$.

$\mathcal{A} \rightarrow_{\forall +} \{o'{:}\forall R.C\} \cup \mathcal{A}$
if $o{:}\forall T.C$ in $\mathcal{A}$, $\langle o, o' \rangle{:}S$ is in $\mathcal{A}$,
and there is $R \in \mathcal{TRN}$ such that $S \preceq R \preceq T$,
and $o'{:}\forall R.C$ is not in $\mathcal{A}$.

$\mathcal{A} \rightarrow_{\sqcap} \{o{:}C_1,\ o{:}C_2\} \cup \mathcal{A}$
if $o{:}C_1 \sqcap C_2$ is in $\mathcal{A}$,
and neither $o{:}C_1$ nor $o{:}C_2$ is in $\mathcal{A}$.

$\mathcal{A} \rightarrow_{\forall^1} \{o'{:}C\} \cup \mathcal{A}$
if $o{:}\forall R.C$ and $\langle o, o' \rangle{:}S$ are in $\mathcal{A}$,
there is $R' \preceq R$ s.t. $R' \stackrel{o}{\approx}_{\mathcal{A}} S$
and $o'{:}C$ is not in $\mathcal{A}$.

$\mathcal{A} \rightarrow_{\forall} \{o'{:}C\} \cup \mathcal{A}$
if $o{:}\forall R.C$ is in $\mathcal{A}$, and $\langle o, o' \rangle{:}S$ is in $\mathcal{A}$,
and $S \preceq R$, and $o'{:}C$ is not in $\mathcal{A}$.

Fig. 1. Precompletion rules for $\mathcal{SH}f$

for their consistency. The advantage over the original knowledge base is that they are simpler, enabling the use of techniques based on terminological reasoning.

Proposition 2. *The precompletion process always terminates, and any precompletion has a size which is polynomial w.r.t. the size of the knowledge base.*

Proof (Sketched). The number of new assertions introduced by the terminology via the $\rightarrow_{\sqsubseteq}$ rule is equal to the number of individual names in the KB multiplied by the number of concept axioms in the Tbox. The rules $\rightarrow_{\sqcap}$, $\rightarrow_{\sqcup}$, $\rightarrow_{\exists^1}$, $\rightarrow_{\forall^1}$, and $\rightarrow_{\forall}$ always introduce assertions smaller than the original ones. The only rule that introduces non decreasing assertions is $\rightarrow_{\forall +}$; however, its applicability is bounded by the number of role assertions, which is invariant.

For estimating the size of each precompletion we can use the argument that the number of different concept expressions that can be generated is polynomial w.r.t. the size of the KB.[6] Therefore the size of a precompletion cannot exceed the number of individual names multiplied by the number of concept expressions, and this number is still polynomial w.r.t. the size of the KB.

The satisfiability of a knowledge base and the satisfiability of its precompletions are strictly related. In fact, the knowledge base is satisfiable if and only if at least one of its precompletions is satisfiable (Proposition 3).

Proposition 3. *A knowledge base $\Sigma = \langle \mathcal{T}, \mathcal{A} \rangle$ is satisfiable if and only if it has a satisfiable precompletion.*

Proof. For the *if* direction, since Σ is included in all its precompletions, a model for a precompletion Σ_{pc} is a model for Σ as well.

[6] Again the only problem may come from the $\rightarrow_{\forall +}$ rule, but the number of formulae that it can generate is limited by the number of transitive role names.

For the *only if* direction, we show that given a model $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ for Σ, a satisfiable precompletion of Σ can be built. This precompletion $\Sigma_{pc} = \langle \mathcal{T}, \mathcal{A}_{pc} \rangle$ is built by extending $\mathcal{A}$ using a set of rules constrained by the model $\mathcal{I}$. The rules are the same as Figure 1 apart from the nondeterministic $\rightarrow_{\sqcup}$ rule, which is transformed into a deterministic one by using the model $\mathcal{I}$:

$\mathcal{A} \rightarrow_{\overline{\sqcup}} \{o{:}D\} \cup \mathcal{A}$
 if $o{:}C_1 \sqcup C_2$ in $\mathcal{A}$,
 and $D = C_1$ if $o^{\mathcal{I}} \in C_1^{\mathcal{I}}$ and $D = C_2$ otherwise
 and neither $o{:}C_1$ nor $o{:}C_2$ is in $\mathcal{A}$.

All the rules preserve satisfiability, in the sense that if $\mathcal{I}$ is a model for the Abox before the application of the rule, then it is a model for the extended Abox as well. Because of the preserved satisfiability, $\mathcal{I}$ must be a model for the precompleted knowledge base Σ_{pc} as well.

Here we show the proofs for the $\rightarrow_{\overline{\sqcup}}$ and $\rightarrow_{\forall^1}$ rules only, the rest of the rules follow a similar pattern.

$\rightarrow_{\overline{\sqcup}}$ If $\mathcal{I}$ is a model for $\langle \mathcal{T}, \mathcal{A} \rangle$, then $o^{\mathcal{I}} \in (C_1 \sqcup C_2)^{\mathcal{I}}$; therefore either $o^{\mathcal{I}} \in C_1^{\mathcal{I}}$ or $o^{\mathcal{I}} \in C_2^{\mathcal{I}}$. Suppose that $o^{\mathcal{I}} \in C_1^{\mathcal{I}}$, then $\mathcal{A}$ is extended by adding the assertion $o{:}C_1$ which is satisfied by $\mathcal{I}$, and analogously for the case in which $o^{\mathcal{I}} \in C_2^{\mathcal{I}}$.

$\rightarrow_{\forall^1}$ If $\mathcal{I}$ is a model for $\langle \mathcal{T}, \mathcal{A} \rangle$, then $(o^{\mathcal{I}}, o'^{\mathcal{I}}) \in S^{\mathcal{I}}$, and every element x s.t. $(o^{\mathcal{I}}, x) \in R^{\mathcal{I}}$ must be in $C^{\mathcal{I}}$.

Since $R' \overset{o}{\approx}_{\mathcal{A}} S$, there is a role $R_0 \preceq R'$ and an element x such that $(o^{\mathcal{I}}, x) \in {R_0}^{\mathcal{I}}$. In addition, $R' \preceq R$ therefore $(o^{\mathcal{I}}, x) \in R^{\mathcal{I}}$, which means that $x \in C^{\mathcal{I}}$. Finally, we can use Proposition 1 with R' and S for concluding that $x = o'^{\mathcal{I}}$, so $\mathcal{I}$ satisfies the assertion $o'{:}C$.

Since Proposition 2 ensures that the precompletions of a given knowledge base can be enumerated, the satisfiability checking can be performed on precompleted knowledge bases. Therefore the problem of checking the satisfiability of a $\mathcal{SH}f$ knowledge base can be reduced to the problem of verifying whether one of its precompletions is satisfiable.

4 Satisfiability of precompletions

In precompleted knowledge bases, the information carried by role assertions is made explicit; therefore all the relevant properties of an individual are in the form of concept assertions. The label of an individual completely characterises the properties of the individual, and it can be used to verify that these properties are not contradictory.

Since we are going to ignore the role assertions of a precompleted KB, first we must make sure that a precompleted KB does not contain any contradiction caused by role assertions. In $\mathcal{SH}f$ this case is restricted to assertions involving functional roles. We say that a precompletion $\Sigma_{pc} = \langle \mathcal{T}, \mathcal{A}_{pc} \rangle$ contains a *clash* iff there are two roles $R, R' \in \mathcal{RN}$, and individual names a, b, c with $b \neq c$, such that

$R \overset{a}{\approx}_{\mathcal{A}_{pc}} R'$, and $\{\langle a, b\rangle{:}R, \langle a, c\rangle{:}R'\} \subseteq \mathcal{A}_{pc}$. A precompletion containing a clash is trivially not satisfiable because it violates some the functional restrictions; in fact, by Proposition 1 the interpretations of b and c must coincide, which is in contradiction with the unique name assumption.

It is easy to see that if a precompleted KB is satisfiable, then each individual concept is satisfiable as well. More involved is the proof that the satisfiability of all the individual concepts guarantees the satisfiability of the whole KB. The rest of this section is devoted to show that this is the case.

If each individual concept $\bigsqcap \mathcal{L}(\Sigma_{pc}, o)$ is separately satisfiable with respect to the terminology, then for every individual name o there is an *individual model* $\mathcal{I}_o = (\Delta_o, \cdot^{\mathcal{I}_o})$ that witnesses the satisfiability. We are going to use these individual models to build a new interpretation which satisfies the precompleted KB.

Without loss of generality, we can assume that these individual models are tree-shaped,[7] and the root of the tree is in the interpretation of the individual concept. In fact, the enriched expressivity of $\mathcal{SH}f$ forces interpretations not being simple trees, but what we call *quasi transitive* trees (see [13]). These are trees where part of the branches can be transitively closed by the role transitivity. The effect is that one node can have more than a parent, but all the parents of any node belong to the same path. However, the main property we are interested in this context, is that the root node does not have entering edges. To simplify the notation, we assume that the individual model $\mathcal{I}_o$ contains the interpretation for the individual o (i.e. $o^{\mathcal{I}_o}$), which is the root of the quasi transitive tree. In addition, the domains of the models can be considered pairwise disjoint without loss of generality.

As anticipated, we are going to build an interpretation for the precompleted KB (called *union interpretation*) by combining the individual models. The domain of the union interpretation is the union of all the domains from each individual model. The interpretation function is defined in terms of the interpretation functions of each individual model. Each individual name is interpreted as the root of its corresponding model (i.e. $o^{\mathcal{I}_o}$), and concept names are interpreted as the union of their interpretations in the different models. Interpretation of roles involves something more than the union of individual interpretations. First, the role assertions in the Abox must be taken into account, as well as the effect of transitivity and role hierarchy. On top of that we must ensure that functional roles are still functional.

Consider for example a KB containing only the Abox assertions $\langle a, b\rangle{:}F$ and $a{:}\exists F.C$, where F is a functional role. The precompletion of this KB contains the Abox assertions $\langle a, b\rangle{:}F$, $a{:}\exists F.C$, and $b{:}C$. From the precompletion, two individual models can be derived: $\mathcal{I}_a$ for the concept $\exists F.C$ and $\mathcal{I}_b$ for C. If we try to merge these two models, together with the pair generated by the role assertion $\langle a, b\rangle{:}F$, the resulting interpretation would not satisfy the functional restriction on F. The solution to this problem relies on a more careful definition of the union interpretation, which takes into account the functional restrictions on role

[7] Roles represent the labelled edges connecting elements of the interpretation domain.

extensions. For this purpose we introduce the notion of *restricted interpretation* (written as $\underline{\mathcal{I}_o}$). The idea is to remove the links that will be added later on by means of role assertions in the Abox. We perform this operation only on functional roles, because they are the problematic ones.

Given a precompleted KB $\Sigma_{pc} = \langle \mathcal{T}, \mathcal{A}_{pc} \rangle$, and an individual o in Σ_{pc}, let $\mathcal{I}_o = (\Delta_o, \cdot^{\mathcal{I}_o})$ be the individual model for o. Its correspondent restricted individual interpretation function $\cdot^{\underline{\mathcal{I}_o}}$ is equal to the original interpretation function $\cdot^{\mathcal{I}_o}$ for concept and individual names; while for roles it is defined as

$$R^{\underline{\mathcal{I}_o}} = \begin{cases} R^{\mathcal{I}_o} \setminus \left\{ (o^{\mathcal{I}_o}, x) \mid (o^{\mathcal{I}_o}, x) \in R^{\mathcal{I}_o} \right\} & \text{if there are } R', R'' \text{ s.t. } R \preceq R', \\ & \langle o, o' \rangle{:}R'' \in \mathcal{A}_{pc}, \text{and } R' \overset{o}{\approx}_{\mathcal{A}_{pc}} R''; \\ R^{\mathcal{I}_o} & \text{otherwise.} \end{cases}$$

Note that the restricted interpretation depends on the knowledge base as well as on each individual model. It is used instead of the original interpretation function in Definition 2, where the extension of role names is defined in such a way that pairs removed by means of the restricted interpretations are substituted by pairs induced by Abox assertions. Transitive roles are not affected by the restricted interpretation because, by assumption, they cannot have functional super-roles.

Definition 2 (Union interpretation). *Given a precompleted knowledge base* $\Sigma_{pc} = \langle \mathcal{T}, \mathcal{A}_{pc} \rangle$ *and the individual models* $\mathcal{I}_o = (\Delta_o, \cdot^{\mathcal{I}_o})$ *for each individual* $o \in \mathcal{O}$*, then the* union interpretation $\overline{\mathcal{I}} = (\overline{\Delta}, \cdot^{\overline{\mathcal{I}}})$ *is defined as:*

$$\overline{\Delta} = \bigcup_{o \in \mathcal{O}} \Delta_o, \qquad A^{\overline{\mathcal{I}}} = \bigcup_{o \in \mathcal{O}} A^{\mathcal{I}_o}, \qquad o^{\overline{\mathcal{I}}} = o^{\mathcal{I}_o},$$

$$R^{\overline{\mathcal{I}}} = \begin{cases} \bigcup_{o \in \mathcal{O}} R^{\underline{\mathcal{I}_o}} \cup \left\{ (a^{\overline{\mathcal{I}}}, b^{\overline{\mathcal{I}}}) \mid \langle a, b \rangle{:}R' \in \mathcal{A}_{pc}, R \overset{a}{\approx}_{\mathcal{A}_{pc}} R' \right\} & \textit{if } R \notin \mathcal{TRN}; \\ \qquad \cup \left\{ (a^{\overline{\mathcal{I}}}, b^{\overline{\mathcal{I}}}) \mid \langle a, b \rangle{:}R' \in \mathcal{A}_{pc}, R' \preceq R \right\} & \\ \qquad \cup \bigcup_{S \preceq R, S \in \mathcal{TRN}} S^{\overline{\mathcal{I}}} & \\ (\bigcup_{o \in \mathcal{O}} R^{\underline{\mathcal{I}_o}} \cup \left\{ (a^{\overline{\mathcal{I}}}, b^{\overline{\mathcal{I}}}) \mid \langle a, b \rangle{:}R' \in \mathcal{A}_{pc}, R' \preceq R \right\})^{+} & \textit{if } R \in \mathcal{TRN}. \end{cases}$$

For each $o \in \mathcal{O}$*,* $A \in \mathcal{CN}$*,* $R \in \mathcal{RN}$*. The operator* $\cdot^{+}$ *builds the transitive closure of a relation.*[8]

Note that the definition is recursive because the union interpretation is used to build the interpretation for roles. However, the interpretation is well defined, because we assumed that the role hierarchy is acyclic (see Section 2.1).

The definition of union interpretation for concept and individual names is straightforward, while roles are more involved. The reason is that by considering precompleted Aboxes we ignored the role assertions. However, these must be added when we build the interpretation for the whole Abox. Role assertions require new pairs directly corresponding to the assertion themselves, represented

[8] Interpretation of complex concept expressions are defined as described in Section 2.1.

by the component $\left\{(a^{\overline{\mathcal{I}}}, b^{\overline{\mathcal{I}}}) \mid \langle a,b\rangle{:}R' \in \mathcal{A}_{pc}, R' \preceq R\right\}$. But this is not enough because of transitivity and functional restriction. Transitivity is guaranteed by the application of the $\cdot^+$ operator, and by the component $\bigcup_{S \preceq R, S \in \mathcal{TRN}} S^{\overline{\mathcal{I}}}$. While the $\left\{(a^{\overline{\mathcal{I}}}, b^{\overline{\mathcal{I}}}) \mid \langle a,b\rangle{:}R' \in \mathcal{A}_{pc}, R \overset{a}{\approx}_{\mathcal{A}_{pc}} R'\right\}$ component ensures the presence of pairs enforced by the interaction between functionality and role hierarchy.

4.1 Interpretation of roles

The crucial properties used are the disjointness of the domains, together with the quasi transitive model structure. These ensure that all the newly added pairs involve (or are "mediated" by) root nodes. Proofs in this section are omitted and they can be found in [13] (Section 5.2.1). Most of them consist of an induction on the role hierarchy, based on the definition of union interpretation.

No pair of elements, coming from the same individual domain, is in the union interpretation of a role unless it is in the individual interpretation of the role itself. Obviously, we must exclude the case of a pair induced by an Abox assertion. For example, the assertion $\langle a,a\rangle{:}R$ forces the pair $(a^{\mathcal{I}_a}, a^{\mathcal{I}_a})$ in $R^{\overline{\mathcal{I}}}$.

Proposition 4. *Given a precompleted knowledge base $\Sigma_{pc} = \langle \mathcal{T}, \mathcal{A}_{pc}\rangle$, and the union interpretation $\overline{\mathcal{I}} = (\overline{\Delta}, \cdot^{\overline{\mathcal{I}}})$ from the individual models $\mathcal{I}_o = (\Delta_o, \cdot^{\mathcal{I}_o})$ with $o \in \mathcal{O}$. For each role $R \in \mathcal{RN}$, individual name $o \in \mathcal{O}$, and elements $\{x,y\} \subseteq \Delta_o$:*

$$(x,y) \in R^{\overline{\mathcal{I}}} \text{ and } y \neq o^{\overline{\mathcal{I}}} \text{ implies } (x,y) \in R^{\mathcal{I}_o}$$

The following proposition shows that if two elements corresponding to individual names are related, then this is because of one or more Abox assertions.

Proposition 5. *Given a precompleted knowledge base $\Sigma_{pc} = \langle \mathcal{T}, \mathcal{A}_{pc}\rangle$, and the union interpretation $\overline{\mathcal{I}} = (\overline{\Delta}, \cdot^{\overline{\mathcal{I}}})$ from the individual models $\mathcal{I}_o = (\Delta_o, \cdot^{\mathcal{I}_o})$ with $o \in \mathcal{O}$. For any role $R \in \mathcal{RN}$ and elements x,y of $\overline{\Delta}$, if $(x,y) \in R^{\overline{\mathcal{I}}}$ and $y \in \mathcal{O}^{\overline{\mathcal{I}}}$ then:*

$$(x,y) \in \left\{(a^{\overline{\mathcal{I}}}, b^{\overline{\mathcal{I}}}) \mid \langle a,b\rangle{:}R' \in \mathcal{A}_{pc}, R' \preceq R\right\}, \ \textit{or}$$
$$(x,y) \in \left\{(a^{\overline{\mathcal{I}}}, b^{\overline{\mathcal{I}}}) \mid \langle a,b\rangle{:}R' \in \mathcal{A}_{pc}, R \overset{a}{\approx}_{\mathcal{A}_{pc}} R'\right\}, \ \textit{or}$$
$$(x,y) \in (\left\{(a^{\overline{\mathcal{I}}}, b^{\overline{\mathcal{I}}}) \mid \langle a,b\rangle{:}R' \in \mathcal{A}_{pc}, R' \preceq S\right\})^+ \textit{ for some transitive role } S \preceq R.$$

In the union interpretation there cannot be any connection between elements from different individual domains, without a path passing through a root node (Proposition 6). This property ensures that all the restrictions applying to a non-root element are never directly propagated from different individual domains, but induced through the root of its own individual domain.

Proposition 6. *Given a precompleted knowledge base $\Sigma_{pc} = \langle \mathcal{T}, \mathcal{A}_{pc}\rangle$, and the union interpretation $\overline{\mathcal{I}} = (\overline{\Delta}, \cdot^{\overline{\mathcal{I}}})$ from the individual models $\mathcal{I}_o = (\Delta_o, \cdot^{\mathcal{I}_o})$ with*

$o \in \mathcal{O}$. For any role $R \in \mathcal{RN}$, different individual names a, b, and elements $x \in \Delta_a$, $y \in \Delta_b$, if $(x, y) \in R^{\overline{\mathcal{I}}}$ then $x = a^{\overline{\mathcal{I}}}$, and $y = b^{\overline{\mathcal{I}}}$ or there is a role $S \in \mathcal{TRN}$ such that $S \preceq R$ and $\left\{(x, b^{\overline{\mathcal{I}}}), (b^{\overline{\mathcal{I}}}, y)\right\} \subseteq S^{\overline{\mathcal{I}}}$.

Functional restrictions are satisfied because we removed potential conflicts with Abox assertions by means of the restricted interpretation of roles.

Proposition 7. *Given a precompleted knowledge base $\Sigma_{pc} = \langle \mathcal{T}, \mathcal{A}_{pc} \rangle$, and the union interpretation $\overline{\mathcal{I}} = (\overline{\Delta}, \cdot^{\overline{\mathcal{I}}})$ from the individual models $\mathcal{I}_o = (\Delta_o, \cdot^{\mathcal{I}_o})$ with $o \in \mathcal{O}$. If a role R is included in a functional role F (i.e. $R \preceq F$), then $\sharp\left\{y | (x, y) \in R^{\overline{\mathcal{I}}}\right\} \leq 1$ for any element x of $\overline{\Delta}$.*

While the recursive definition of the union interpretation ensures that the hierarchy is satisfied (see Definition 2).

Proposition 8. *Given a precompleted knowledge base $\Sigma_{pc} = \langle \mathcal{T}, \mathcal{A}_{pc} \rangle$, and the union interpretation $\overline{\mathcal{I}} = (\overline{\Delta}, \cdot^{\overline{\mathcal{I}}})$ from the individual models $\mathcal{I}_o = (\Delta_o, \cdot^{\mathcal{I}_o})$ with $o \in \mathcal{O}$. For two arbitrary roles R and S, if $S \preceq R$ then $S^{\overline{\mathcal{I}}} \subseteq R^{\overline{\mathcal{I}}}$.*

4.2 Interpretation of concepts

The presence of new pairs in the interpretation of roles can modify the extension of concept forming constructors. In particular, this affects the universal constructor ($\forall R.C$); e.g. if one element was in the interpretation of $\forall R.C$, new elements related via R and not being in C may force the same element not being in $\forall R.C$ any more. This kind of "non-monotonicity" problem does not involve any element of the domain, but it is localised to root elements. Roughly speaking, the reason for this lies on the fact that the interpretation of roles, restricted to non–root elements, does not change (see Proposition 4).

Proposition 9. *For any individual name $a \in \mathcal{O}$ and concept expression D, $D^{\mathcal{I}_a} \setminus \left\{a^{\overline{\mathcal{I}}}\right\} \subseteq D^{\overline{\mathcal{I}}}$.*

The proof of this proposition consists of a structural induction on concept expressions, where the basic cases are the concept names. Intuitively, the property is true because no new successors are added to elements of $\Delta_a \setminus \left\{a^{\overline{\mathcal{I}}}\right\}$.

A restricted version of Proposition 9 is valid for elements corresponding to individual names (roots) w.r.t. concept expressions which appear as assertions in the precompleted KB (Proposition 10). This restricted property of the union interpretation (together with the more general one applying to non-roots) is sufficient to prove that all the axioms and assertions in the precompleted knowledge base are satisfied by the union interpretation (Proposition 11).

Proposition 10. *For any individual name $a \in \mathcal{O}$, $a{:}D \in \mathcal{A}_{pc}$ implies $a^{\overline{\mathcal{I}}} \in D^{\overline{\mathcal{I}}}$.*

Proof. The Proposition is proved by induction on the structure of the concept expression D. Here we show only the more interesting universal constructor case.

As induction hypothesis we assume that the proposition is valid for the concept C_1 and we show that it must be valid for $(\forall R.C_1)$ as well. Suppose that $(a^{\overline{\mathcal{I}}}, x) \in R^{\overline{\mathcal{I}}}$, then we distinguish three different cases according to the element x: x is one of the root elements (i.e. $x \in \mathcal{O}^{\overline{\mathcal{I}}}$), $x \in \Delta_a \setminus \left\{a^{\overline{\mathcal{I}}}\right\}$, or there is $b \neq a$ s.t. $x \in \Delta_b \setminus \left\{b^{\overline{\mathcal{I}}}\right\}$.

- If $x = o^{\overline{\mathcal{I}}}$ then the pair $(a^{\overline{\mathcal{I}}}, x)$ must be introduced by means of Abox assertions (see Proposition 5). Let us consider the three different cases.
  $(a^{\overline{\mathcal{I}}}, x) \in \left\{(a^{\overline{\mathcal{I}}}, b^{\overline{\mathcal{I}}}) \mid \langle a,b\rangle{:}R' \in \mathcal{A}_{pc}, R' \preceq R\right\}$ **:** there is a constraint $\langle a,b\rangle{:}R'$ with $R' \preceq R$ s.t. $x = b^{\overline{\mathcal{I}}}$. The constraint $b{:}C_1$ must be in $\mathcal{A}_{pc}$ because of the $\rightarrow_\forall$ rule.Therefore we can use the induction hypothesis for concluding that $x \in {C_1}^{\overline{\mathcal{I}}}$.
  $(a^{\overline{\mathcal{I}}}, x) \in \left\{(a^{\overline{\mathcal{I}}}, b^{\overline{\mathcal{I}}}) \mid \langle a,b\rangle{:}R' \in \mathcal{A}_{pc}, R \overset{a}{\approx}_{\mathcal{A}_{pc}} R'\right\}$ **:** there is a constraint $\langle a,b\rangle{:}R'$ with $R \overset{a}{\approx}_{\mathcal{A}_{pc}} R'$ s.t. $x = b^{\overline{\mathcal{I}}}$. The constraint $b{:}C_1$ must be in $\mathcal{A}_{pc}$ because of the $\rightarrow_{\forall^1}$ rule.Therefore we can use the induction hypothesis for concluding that $x \in {C_1}^{\overline{\mathcal{I}}}$.
  $(a^{\overline{\mathcal{I}}}, x) \in (\left\{(a^{\overline{\mathcal{I}}}, b^{\overline{\mathcal{I}}}) \mid \langle a,b\rangle{:}R' \in \mathcal{A}_{pc}, R' \preceq S\right\})^+$ **:** there is a set of constraints

  $$\{\langle a,o_1\rangle{:}S_1, \langle o_1,o_2\rangle{:}S_2, \ldots, \langle o_{n-1},o_n\rangle{:}S_n\} \subseteq \mathcal{A}_{pc}$$

  and a transitive role S s.t. $S_i \preceq S \preceq R$ for all $i = 1,\ldots,n$, and $x = {o_n}^{\overline{\mathcal{I}}}$.For each $i = 1,\ldots,n$, the constraints $o_i{:}(\forall S.C_1)$ and $o_i{:}C_1$ are in $\mathcal{A}_{pc}$ (because of the $\rightarrow_{\forall^+}$ rule and the $\rightarrow_\forall$). Therefore $o_n{:}C_1$ is in $\mathcal{A}_{pc}$ and $x \in {C_1}^{\overline{\mathcal{I}}}$ by the induction hypothesis.
- If $x \in \Delta_a \setminus \left\{a^{\overline{\mathcal{I}}}\right\}$ then $(a^{\overline{\mathcal{I}}}, x) \in R^{\mathcal{I}_a}$, because both $a^{\overline{\mathcal{I}}}$ and x are in the same individual domain (see Definition 2). In addition, $(\forall R.C_1) \in \mathcal{L}(\Sigma_{pc}, a)$ so $x \in {C_1}^{\mathcal{I}_a}$ because $\mathcal{I}_a$ is a model for the label $\mathcal{L}(\Sigma_{pc}, a)$. Using Proposition 9 we can conclude that $x \in {C_1}^{\overline{\mathcal{I}}}$.
- If there is $b \neq a$ s.t. $x \in \Delta_b \setminus \left\{b^{\overline{\mathcal{I}}}\right\}$, then by Proposition 6 there is a transitive role S s.t. $S \preceq R$ and both $(a^{\overline{\mathcal{I}}}, b^{\overline{\mathcal{I}}})$, $(b^{\overline{\mathcal{I}}}, x)$ are in $S^{\overline{\mathcal{I}}}$. By using the very same arguments as the first case we can show that the constraint $b{:}(\forall S.C_1)$ is in $\mathcal{A}_{pc}$; therefore we can conclude that $x \in {C_1}^{\overline{\mathcal{I}}}$ by using the same arguments as in the previous case.

It is clear that a model for the knowledge base is a model for every individual concept. Using the properties demonstrated in the previous pages we show that the union interpretation is a model for the precompleted knowledge base.

Proposition 11. *A clash-free $\mathcal{SH}f$ precompleted knowledge base $\Sigma_{pc} = \langle \mathcal{T}, \mathcal{A}_{pc}\rangle$ is satisfiable if and only if for each individual name $o \in \mathcal{O}$ the concept expression $\bigsqcap \mathcal{L}(\Sigma_{pc}, o)$ is satisfiable with respect the terminology $\mathcal{T}_{pc}$.*

Proof. The *only if* direction is easy because a model for Σ_{pc} is a model for $o{:}\mathcal{L}(\Sigma_{pc}, o)$ for any individual name o, so $\mathcal{L}(\Sigma_{pc}, o)$ must be satisfiable.

For the *if* direction we use the propositions defined in this section to show that every statement in the KB is satisfied by the union interpretation $\overline{\mathcal{I}} = (\overline{\Delta}, \cdot^{\overline{\mathcal{I}}})$.

$a{:}C$ **:** By Proposition 10, $a^{\overline{\mathcal{I}}} \in C^{\overline{\mathcal{I}}}$.
$\langle a, b\rangle{:}R$ **:** By construction of $R^{\overline{\mathcal{I}}}$ (see Definition 2) $(a^{\overline{\mathcal{I}}}, b^{\overline{\mathcal{I}}}) \in R^{\overline{\mathcal{I}}}$.
$\top \sqsubseteq C$ **:** Let x be an element of $\overline{\Delta}$, then there exists $a \in \mathcal{O}$ such that $x \in \Delta_a$, and $x \in C^{\mathcal{I}_a}$ because the model $\mathcal{I}_a$ satisfies $\mathcal{T}_{pc}$. There are two cases: $x = a^{\overline{\mathcal{I}}}$ or $x \in \Delta_a \setminus \left\{a^{\overline{\mathcal{I}}}\right\}$

- If $x = a^{\overline{\mathcal{I}}}$ then $x \in C^{\overline{\mathcal{I}}}$ because of Proposition 10 ($a{:}C$ is in the precompleted Abox $\mathcal{A}_{pc}$ because of the $\rightarrow_{\sqsubseteq}$ rule).
- If $x \neq a^{\overline{\mathcal{I}}}$, then $x \in C^{\overline{\mathcal{I}}}$ by Proposition 9.

$R \in \mathcal{TRN}$ **:** Transitivity of $R^{\overline{\mathcal{I}}}$ is guaranteed by construction (see Definition 2).
$R \in \mathcal{FRN}$ **:** Functionality is satisfied by Proposition 7.
$S \sqsubseteq R$ **:** If $S \preceq R$, then $S^{\overline{\mathcal{I}}} \subseteq R^{\overline{\mathcal{I}}}$ by Proposition 8.

5 Notes on implementation and evaluation

The algorithm described in this paper has been implemented in order to verify the feasibility of the approach in terms of performance. Although a description of the implemented system and the analysis of the results of the experiments is outside the scope of this paper, we mention this work for the sake of completeness. For more details the reader should refer to [12] or [13].

The experience with previous DL systems shows that the direct implementation of the tableaux–based satisfiability algorithms provides very poor performance. The development of other DL reasoners showed that the use of various optimisations and heuristics has a great impact on performance with application and synthetic knowledge bases. We adopted some of the well known optimisation techniques adopted by most of the state of the art DL reasoners (see [15, 16]), as well as techniques developed in conjunction with the precompletion algorithm. Moreover, the precompletion phase is completely separated from the terminological reasoning, so optimisation techniques implemented at the two levels do not interact adversely. For example, if a contradiction is found in an individual label (even in the modal part, by means of a call to the terminological reasoner) there is no reason to generate a full precompletion. For this reason, the terminological reasoner is called at different stages during the precompletion.

We used the Abox reasoner RACER (see [8]) to compare the results of our system. In Table 1 we provide a summary of the results obtained with the *synthetic Abox tests* of the DL benchmark suite (see [15, 17]). There are 9 classes of tests, each one containing different instances of problems of increasing difficulty. Each instance is automatically generated according to a schema related to the class, and it consists of a Tbox, an Abox and a set of instance checking queries.

	Optimised	Not Optimised	RACER	Max
k_branch_n	2	1	3	4
k_d4_n	2	1	2	4
k_dum_n	8	1	13	21
k_grz_n	7	2	10	10
k_lin_n	4	3	4	10
k_path_n	4	1	3	4
k_ph_n	6	6	5	7
k_poly_n	4	4	4	8
k_t4p_n	2	0	2	5

Table 1. Summary of experimental results

Results of the experiments are grouped by the class of tests, and each column shows the last instance of the test which has been solved within the given time-out of 500 seconds. As a reference, the two last columns show the results of the RACER system (version 1.2)[9] and the maximum number of test instances. The second and third columns show the results with and without optimisations respectively.

6 Conclusions

We have presented a precompletion style algorithm for deciding knowledge base satisfiability in the $\mathcal{SH}f$ description logic, and proved its soundness and completeness. Reasoning with a complete knowledge base, i.e., an Abox as well as a Tbox, will be of increasing importance in the application of Description Logics, e.g., in providing inference services for the Semantic Web.

The main advantages of the precompletion approach (w.r.t. tableaux algorithms that reason directly with the whole knowledge base) are its ability to exploit existing highly optimised Tbox reasoners, and the fact that it may be able to handle very large Aboxes by partitioning them into disconnected parts.

On the other hand, the logic described is a subset of that required by Semantic Web knowledge representation languages such as DAML+OIL, and it is not easy to see how the technique can be extended to deal with more expressive languages. There is, however, no known tableaux algorithm that is able to deal with the complete DAML+OIL language [18], and applications are already using reasoners that only deal with subsets of the language. Moreover, $\mathcal{SH}f$ is the largest language for which reasonable evidence exists as to the empirical tractability of highly optimised tableaux algorithms (for Tbox reasoning) [19]. It has yet to be shown that these results will extend to logics, such as $\mathcal{SHIQ}$, which are closer in expressive power to DAML+OIL.

[9] This was the version of the RACER system available at the time we performed the experiments. As pointed out by an anonymous reviewer, the latest version of RACER (1.6.7) improved the results in most of the classes.

References

1. McGuinness, D.L., Wright, J.R.: An industrial strength description logic-based configuration platform. IEEE Intelligent Systems (1998) 69–77
2. Calvanese, D., De Giacomo, G., Lenzerini, M.: On the decidability of query containment under constraints. In: Proc. of PODS-98. (1998) 149–158
3. Brachman, R.J., Schmolze, J.G.: An overview of the KL-ONE knowledge representation system. Cognitive Science **9** (1985) 171–216
4. Berners-Lee, T.: Weaving the Web. Harpur, San Francisco (1999)
5. Horrocks, I., Patel-Schneider, P.: The generation of DAML+OIL. In: Proc. of DL 2002, CEUR Workshop Proceedings (2001) 30–35
6. Donini, F.M., Lenzerini, M., Nardi, D., Nutt, W.: The complexity of concept languages. Information and Computation **134** (1997) 1–58
7. Broekstra, J., Kampman, A., van Harmelen, F.: Sesame: An architecture for storing and querying RDF data and schema information. In D. Fensel, J. Hendler, H.L., Wahlster, W., eds.: Semantics for the WWW. MIT Press (2001)
8. Haarslev, V., Möller, R.: RACER system description. In: Proc. of IJCAR-01. (2001)
9. Horrocks, I., Sattler, U., Tobies, S.: Reasoning with individuals for the description logic $\mathcal{SHIQ}$. In MacAllester, D., ed.: Proc. of CADE-2000. Number 1831 in LNCS, Springer-Verlag (2000) 482–496
10. Hollunder, B.: Algorithmic Foundations of Terminological Knowledge Representation Systems. PhD thesis, Universität des Saarlandes (1994)
11. Donini, F.M., Lenzerini, M., Nardi, D., Schaerf, A.: Deduction in concept languages: From subsumption to instance checking. J. of Logic and Computation **4** (1994) 423–452
12. Tessaris, S., Horrocks, I., Gough, G.: Evaluating a modular abox algorithm. In: Proc. of KR-02. (2002)
13. Tessaris, S.: Questions and answers: reasoning and querying in Description Logic. PhD thesis, University of Manchester (2001)
14. Schmidt-Schauß, M., Smolka, G.: Attributive concept descriptions with complements. Artificial Intelligence **48** (1991) 1–26
15. Haarslev, V., Möller, R.: An empirical evaluation of optimization strategies for abox reasoning in expressive description logics. In: Proc. of DL'99. (1999) 115–119
16. Horrocks, I., Patel-Schneider, P.F.: Optimising description logic subsumption. Journal of Logic and Computation **9** (1999) 267–293
17. Horrocks, I., Patel-Schneider, P.F.: DL system comparison. In: Proc. of DL'98. (1998)
18. Horrocks, I.: Daml+oil: a reason-able web ontology language. In: Proc. of EDBT 2002. (2002)
19. Horrocks, I.: Using an expressive description logic: FaCT or fiction? In: Proc. of KR-98. (1998) 636–647