ISSN 1361 - 6161

THE UNIVERSITY of MANCHESTER

Computer Science
University of Manchester

Hybrid Reasoning for Description Logics with Transitive Roles and Axioms

Sergio Tessaris

Department of Computer Science
University of Manchester
Technical Report Series
UMCS-99-7-1

Hybrid Reasoning for Description Logics with Transitive Roles and Axioms*

Sergio Tessaris

Department of Computer Science
University of Manchester
Oxford Road, Manchester, UK.
`tessaris@cs.man.ac.uk`

June 8, 1999

*Refereed by: Enrico Franconi

Abstract

This report presents an original algorithm for correct and complete reasoning with Description Logics representation systems involving both terminological and assertional parts.

Description Logics knowledge representation systems are based of a family of formal languages for describing complex structured classes. These languages comprise boolean operators and quantification over class attributes, as well as the declaration of elements of the classes and their properties. DL knowledge bases are partitioned in a terminological part which describes the general organisation of the classes, and an assertional part for describing the properties of each individual. In spite of its inherent intractability, practical algorithm have been recently developed for terminological reasoning; however on the hybrid reasoning side there is still a lack of results. The aim of this research is to investigate new techniques for hybrid reasoning, in order to provide a complete and efficient DL knowledge representation system.

1 Introduction

Description Logics (DLs) constitute a family of formalisms for the structured representation of information. They developed from research on semantic networks and frames, with the characteristic of having a formally defined semantics which enables sound and complete reasoning. DLs represent knowledge in terms of objects (individuals) grouped into classes (concepts). In addition they provide structuring mechanisms for characterising the relevant properties of classes as relations (roles) between individuals, as well as inclusion relation between concepts (is–a).

Transitive roles play an important part in the representation of much "real-world" information; they allow objects to be described by referring to their parts without specifying the particular structure of the decomposition (see Artale et al. [1996]). Several previous papers (see Horrocks and Gough [1997], Horrocks and Sattler [1998]) have also shown the efficacy of using a logic that includes a set of transitive roles, without incurring the cost entailed by the transitive closure operator. This report extends the above work by presenting an algorithm for reasoning within a knowledge representation which also includes ABox assertions.

The approach used is based on an extension of the so–called *precompletion* technique Hollunder [1994], Donini et al. [1994] for reducing the KB satisfiability problem to concept satisfiability. The precompletion approach has been successfully used for both providing correct and complete algorithms, and analysing the complexity of the KB satisfiability problem. Previous works focused on DL knowledge bases with empty terminology, and languages not including transitive roles. With the work presented in this report we tackle the problem of generalising the precompletion technique for KBs with general inclusion axioms and transitive roles.

The main idea behind this technique consists of a correctness–preserving process which eliminates specific information regarding dependencies between individuals, while maintaining the consequences of such information. Once these dependencies are eliminated, the assertions about a single individual can be independently verified ignoring the fact that an individual is involved. The precompletion, or elimination of the dependencies, is performed by adding new assertions using a set of nondeterministic syntactic rules. Because of the nondeterminism of the rules, many different precompletions can be derived from a single knowledge base, which is satisfiable if and only if at least one of these precompletions is satisfiable.

The main advantage of this technique lies in the fact that the concept satisfiability tests can be performed by using any available terminological reasoner (providing a sufficiently expressive DL language); in particular, impressive results have been recently obtained by optimised tableau–based systems like FaCT (see Franconi et al. [1998]). Indeed, one of the primary motivations for reexamining the precompletion approach is the availability of such optimised systems. A first implementation of the algorithm is under development; this implementation will also be used as a test-bed for further optimisations which are currently being investigated.

2 $\mathcal{ALCH}_{\mathcal{R}^+}$ knowledge bases

$\mathcal{ALCH}_{\mathcal{R}^+}$ knowledge bases are based on a description logic comprising $\mathcal{ALC}$ concept expressions with a special set of transitive role names. Knowledge bases contain assertions about individuals, as well as a terminology with inclusion axioms between concept expressions and role names.

2.1 DL expressions

Concept expressions are built using a set of concept names ($\mathcal{CN}$) and role names ($\mathcal{RN}$); the subset $\mathcal{RN}_+ \subseteq \mathcal{RN}$ is the set of transitive roles. Valid concept expressions are defined by the following syntax:

$$C \quad ::= \quad A \mid \top \mid \bot \mid \neg A \mid C_1 \sqcap C_2 \mid C_1 \sqcup C_2 \mid \forall R.C \mid \exists R.C$$

where $A \in \mathcal{CN}$ and $R \in \mathcal{RN}$. The meaning of such expressions is given by a Tarski style model theoretic semantics using *interpretations*. An interpretation $\mathcal{I}$ is a pair $(\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where $\Delta^{\mathcal{I}}$ is the domain and $\cdot^{\mathcal{I}}$ an interpretation function. The function $\cdot^{\mathcal{I}}$ maps each concept name in $\mathcal{CN}$ to a subset of $\Delta^{\mathcal{I}}$ and each role name in $\mathcal{RN}$ to a binary relation over $\Delta^{\mathcal{I}}$ (a subset of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$). Transitive roles (those in $\mathcal{RN}_+$) must satisfy the condition $R^{\mathcal{I}} = (R^{\mathcal{I}})^+$, where $\cdot^+$ indicates transitive closure. The following equations provide the semantics for concept expressions:

$$\begin{aligned}
\top^{\mathcal{I}} &= \Delta^{\mathcal{I}} \\
\bot^{\mathcal{I}} &= \emptyset \\
(\neg A)^{\mathcal{I}} &= \Delta^{\mathcal{I}} \setminus A^{\mathcal{I}} \\
(C_1 \sqcap C_2)^{\mathcal{I}} &= C^{\mathcal{I}}{}_1 \cap C^{\mathcal{I}}{}_2 \\
(C_1 \sqcup C_2)^{\mathcal{I}} &= C^{\mathcal{I}}{}_1 \cup C^{\mathcal{I}}{}_2 \\
(\forall R.C)^{\mathcal{I}} &= \{i \in \Delta^{\mathcal{I}} \mid \forall j.\, (i,j) \in R^{\mathcal{I}} \Rightarrow j \in C^{\mathcal{I}}\} \\
(\exists R.C)^{\mathcal{I}} &= \{i \in \Delta^{\mathcal{I}} \mid \exists j.\, (i,j) \in R^{\mathcal{I}} \wedge j \in C^{\mathcal{I}}\}
\end{aligned}$$

2.2 DL knowledge bases

A DL knowledge base is a pair $\Sigma = \langle \mathcal{T}, \mathcal{A} \rangle$, where $\mathcal{T}$ is a *TBox* and $\mathcal{A}$ an *ABox*. The TBox, or terminology, contains assertions about the concept and role names; the semantics of those assertions is given by means of an interpretation $\mathcal{I}$:

Syntax	Semantics
$\top \sqsubseteq C$	$\top^{\mathcal{I}} \subseteq C^{\mathcal{I}}$
$S \sqsubseteq R$	$S^{\mathcal{I}} \subseteq R^{\mathcal{I}}$

where C is a concept expression, and R,S are role names. Restricting the syntax of the axioms to forms having the concept $\top$ on the left hand side does not limit the expressivity of the language. In fact a general inclusion axiom $C_1 \sqsubseteq C_2$ is equivalent to the axiom $\top \sqsubseteq C_2 \sqcup \neg C_1$.

The role inclusion axioms induce a partial order on the set of role names, indicated by the symbol $\preceq$. For all role names $R \preceq R$, and $S \preceq R$ if the assertion $S \sqsubseteq R$ is in the terminology, or there is a concept name R' such that $S \sqsubseteq R'$ is in the terminology and $R' \preceq R$.

The ABox, or assertional part, contains assertions about a set of individual names $\mathcal{O}$. Each one of those names is interpreted by a different element of the interpretation domain through the interpretation function:

Syntax	Semantics
$a : C$	$a^{\mathcal{I}} \in C^{\mathcal{I}}$
$\langle a, b \rangle : R$	$(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$

where a,b are individual names, C a concept expression, and R a role name.

A knowledge base Σ is said to be satisfiable if and only if there exists at least one interpretation which satisfies all the assertions in the knowledge base (is a model for the knowledge base). A concept expression is said to be satisfiable with respect to a terminology, if there exists an interpretation, satisfying all the assertions in the terminology, in which the extension of the concept expression is not empty.

3 Precompletions of knowledge bases

The proof for correctness and completeness of the technique is divided in two parts: in this section we present a method for deriving a set of simpler knowledge bases called *precompletions*, and show that this set characterises the satisfiability of the original knowledge base. The second part, in Section 4 gives a method for checking the satisfiability of such a precompletion using a terminological reasoner.

The purpose of the precompletion is to generate a simpler (not smaller) knowledge base where the role assertions about individuals can be ignored because they are superfluous. Precompletions of knowledge bases are built using a set of nondeterministic syntactic rules which extend the ABox of the original knowledge base. It will be proved that a knowledge base is satisfiable if and only if a satisfiable precompletion can be derived.

In Section 4 we show that for checking the satisfiability of a precompletion all the individual names and role assertions can be ignored. In a precompletion the elements that do matter are the concept assertions together with the terminology; each individual can be associated to a concept, namely the conjunction of the concept assertions for that individual. If all those individual concepts are satisfiable with respect to the terminology, then their models can be combined in a model for the precompleted knowledge base.

Intuitively, a knowledge base is precompleted if all the information entailed by the presence of role assertions is exhibited in form of concept assertions. The purpose of the precompletion is generating a simpler (not smaller) knowledge base where the role assertions can be ignored because they are superfluous. The definition of a precompletion for a knowledge base $\Sigma = (\mathcal{T}, \mathcal{A})$ is given in a procedural way as a new KB where the ABox is obtained by extending $\mathcal{A}$ using the nondeterministic syntactic rules in Figure 1 as long as they are applicable.

Definition 1 *A* precompletion *of a knowledge base $\Sigma = (\mathcal{T}, \mathcal{A})$ is defined as a knowledge base $\Sigma_{pc} = (\mathcal{T}, \mathcal{A}_{pc})$, where $\mathcal{A}_{pc}$ is built extending $\mathcal{A}$ by applying the syntactic rules in Figure 1 as long as they are applicable.*

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 with the same set of precompletions (Proposition 1).

Proposition 1 *The precompletion process always terminates.*

PROOF. (Sketched)
The number of new assertions introduced using the terminology by the $\rightarrow_{\sqsubseteq}$ rule is finitely bounded, and the rules $\rightarrow_{\sqcap}$, $\rightarrow_{\sqcup}$ and $\rightarrow_{\forall}$ always introduce assertions smaller than the original ones. The only rule that introduce non decreasing assertions is the $\rightarrow_{\forall+}$, but its applicability is bounded by the number of role assertions which is invariant.

□

The precompletion algorithm generates a finite, but possibly exponential, number of precompletions; which can be individually checked for their consistency. The advantage over the original knowledge base is that they are simpler, enabling the use of techniques based on terminological reasoning (see Section 4.2).

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 there is at least one of its precompletions which is satisfiable (Proposition 2). Since Proposition 1 ensures that all the precompletions of a given knowledge base can be enumerated, the satisfiability checking can be performed on precompleted knowledge bases.

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

$$\mathcal{A} \rightarrow_{\sqcap} \{o : C_1,\ o : C_2\} \cup \mathcal{A}$$
if $o : C_1 \sqcap C_2$ in $\mathcal{A}$,
and both $o : C_1$ and $o : C_2$ are not in $\mathcal{A}$.

$$\mathcal{A} \rightarrow_{\sqcup} \{o : D\} \cup \mathcal{A}$$
if $o : C_1 \sqcup C_2$ in $\mathcal{A}$,
and $D = C_1$ or $D = C_2$
and $o : D$ is not in $\mathcal{A}$.

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

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

Figure 1: Precompletion rules for $\mathcal{ALCH}_{\mathcal{R}+}$

Proposition 2 *A knowledge base $\Sigma = (\mathcal{T}, \mathcal{A})$ is satisfiable if and only if it has a satisfiable precompletion.*

PROOF.

$\Leftarrow$ Since Σ is included in all its precompletions, a model for a precompletion Σ_{pc} is a model for Σ as well.

$\Rightarrow$ Given a model $\mathcal{I}$ for Σ, a satisfiable precompletion of Σ can be built. This precompletion $\Sigma_{pc} = (\mathcal{T}, \mathcal{A}_{pc})$ is built by extending $\mathcal{A}$ using 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 $o : D$ is not 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:

$\rightarrow_{\sqsubseteq}$ If $\mathcal{I}$ is a model for $(\mathcal{T}, \mathcal{A})$, then each element of the domain is in $C^{\mathcal{I}}$ (including $o^{\mathcal{I}}$ for each $o \in \mathcal{O}$). Therefore the assertion $o : C$ is satisfied.

$\rightarrow_{\sqcap}$ If $\mathcal{I}$ is a model for $(\mathcal{T}, \mathcal{A})$, then $o^{\mathcal{I}} \in (C_1 \sqcap C_2)^{\mathcal{I}}$. Therefore $o^{\mathcal{I}} \in C_1^{\mathcal{I}}$ and $o^{\mathcal{I}} \in C_2^{\mathcal{I}}$.

$\rightarrow_{\overline{\sqcup}}$ If $\mathcal{I}$ is a model for $(\mathcal{T}, \mathcal{A})$, then $o^{\mathcal{I}} \in (C_1 \sqcup C_2)^{\mathcal{I}}$; therefore $o^{\mathcal{I}} \in C_1^{\mathcal{I}}$ 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 $o^{\mathcal{I}} \in C_2^{\mathcal{I}}$.

$\rightarrow_{\forall}$ If $\mathcal{I}$ is a model for $(\mathcal{T}, \mathcal{A})$, then $o^{\mathcal{I}} \in (\forall R.C)^{\mathcal{I}}$, $(o^{\mathcal{I}}, o'^{\mathcal{I}}) \in S^{\mathcal{I}}$ and $S^{\mathcal{I}} \subseteq R^{\mathcal{I}}$. So $(o^{\mathcal{I}}, o'^{\mathcal{I}}) \in R^{\mathcal{I}}$ as well, therefore $o'^{\mathcal{I}} \in C^{\mathcal{I}}$.

$\rightarrow_{\forall+}$ If $\mathcal{I}$ is a model for $(\mathcal{T}, \mathcal{A})$, then $o^{\mathcal{I}} \in (\forall T.C)^{\mathcal{I}}$, $(o^{\mathcal{I}}, o'^{\mathcal{I}}) \in S^{\mathcal{I}} \subseteq R^{\mathcal{I}} \subseteq T^{\mathcal{I}}$ and $R^{\mathcal{I}} = (R^{\mathcal{I}})^{+}$. If there is any $x \in \Delta$ such that $(o'^{\mathcal{I}}, x) \in R^{\mathcal{I}}$, then $(o^{\mathcal{I}}, x) \in R^{\mathcal{I}}$ as well because R is transitive. The element x should be in $C^{\mathcal{I}}$ because $o^{\mathcal{I}} \in (\forall T.C)^{\mathcal{I}}$ and $R^{\mathcal{I}} \subseteq T^{\mathcal{I}}$; therefore $o'^{\mathcal{I}} \in (\forall R.C)^{\mathcal{I}}$.

Because of the preserved satisfiability, $\mathcal{I}$ is a model for the precompleted knowledge base Σ_{pc} as well.

□

The following Proposition 3 makes explicit the logical properties of precompletions hidden in the rules of Figure 1. In such a way, wherever a precompletion is considered, would not be necessary to refer to the algorithm for its generation, but the properties will be directly used.

Proposition 3 *A precompleted $\mathcal{ALCH}_{\mathcal{R}^+}$ knowledge base $\Sigma_{pc} = (\mathcal{T}_{pc}, \mathcal{A}_{pc})$, containing assertions about the set of individual names $\mathcal{O}$, satisfies all the following conditions:*

1. *if $o \in \mathcal{O}$ and $\top \sqsubseteq C \in \mathcal{T}_{pc}$ then $o : C \in \mathcal{A}_{pc}$;*
2. *if $a : C_1 \sqcap C_2 \in \mathcal{A}_{pc}$ then $a : C_1 \in \mathcal{A}_{pc}$ and $a : C_2 \in \mathcal{A}_{pc}$;*
3. *if $a : C_1 \sqcup C_2 \in \mathcal{A}_{pc}$ then $a : C_1 \in \mathcal{A}_{pc}$ or $a : C_2 \in \mathcal{A}_{pc}$;*
4. *if $\{a : \forall R.C, \langle a, b\rangle : S\} \subseteq \mathcal{A}_{pc}$ and $S \preceq R$ then $b : C \in \mathcal{A}_{pc}$;*
5. *if $\{a : \forall T.C, \langle a, b\rangle : S\} \subseteq \mathcal{A}_{pc}$ and there is an $R \in \mathcal{RN}_{+}$ such that $S \preceq R \preceq T$, then $b : \forall R.C \in \mathcal{A}_{pc}$.*

Proof.
Each property corresponds to the pre–conditions of one of the rules of Figure 1. So if the property is not verified, the corresponding rule is still applicable and Σ_{pc} is not yet a precompletion.

□

Although the number of precompletions of a knowledge base could be exponential with respect to the KB size, the size of each precompletion is polynomial.

4 Satisfiability of precompletions

In this section we show how the satisfiability of a precompletion can be reduced to the concept satisfiability problem. This reduction enables us to close the circle and presenting the main result for a general KB in Section 4.3. Before considering the precompletion satisfiability, in Section 4.1 we recall some well known properties of the concept satisfiability problem. These will be essential for the understanding of the following sections.

4.1 Models of Concepts

Analogously to the knowledge base satisfiability definition of Section 2.2, the satisfiability for a concept expression C with respect to a terminology $\mathcal{T}$ is defined by mean of interpretations. A concept expression is said to be satisfiable with respect to a terminology if and only if there exists at least one interpretation which satisfies every axiom in the terminology, providing a nonempty extension for the concept expression itself ($C^{\mathcal{I}} \neq \emptyset$). These interpretations are said to be models for the expression with respect the given terminology.

An interpretation can be seen as a graph, where the nodes are elements of the domain and the interpretation of roles gives the (labelled) edges. For a large class of DLs, such models enjoy the so called *tree model property*, which is defined to be the fact that if a concept expression is satisfiable, then there exists a (possibly infinite) tree structured model. In other words, the graph is connected, each node has no more than one parent, and there are no cycles. Moreover, the individual which corresponds to the root of the tree is in the extension of the concept expression.

Proposition 4 *An $\mathcal{ALCH}_{\mathcal{R}^+}$ concept C is satisfiable with respect an $\mathcal{ALCH}_{\mathcal{R}^+}$ terminology $\mathcal{T}$ if and only if there exists a tree structured interpretation $\mathcal{I} = (\Delta, \cdot^{\mathcal{I}})$ which is a model for both C and $\mathcal{T}$, and whose root element is in the extension of C ($C^{\mathcal{I}}$).*

PROOF.
The proposition follows from the existence of a correct and complete tableaux–based satisfiability algorithm (see Horrocks [1998]). Such an algorithm builds a tree model starting from an individual which is in the extension of C.

$\square$

The root of a tree model $\mathcal{I}$ will be indicated by $\widehat{\mathcal{I}}$. It is worth noticing that the models can be infinite, and in this case there can be unbounded paths.

4.2 Satisfiability of precompleted KBs

As seen in Proposition 2, providing a knowledge base satisfiability checking algorithm for precompleted knowledge bases is general enough. The advantage in concentrating on such KBs is that the consistency can be verified using a terminological reasoner (Theorem 1). Given a knowledge base, the *label* of an individual is the set of concept expressions associated appearing in the concept assertions involving the individual itself.

Definition 2 *Given a knowledge base $\Sigma = (\mathcal{T}, \mathcal{A})$ containing assertions about named individuals $\mathcal{O}$, the* label *of an individual $o \in \mathcal{O}$ with respect to Σ is defined to be the collection of all concept expressions in the assertions about the individual name o:*

$$\mathcal{L}(\Sigma, o) = \begin{cases} \{C \mid o : C \in \mathcal{A}\} & \textit{if there is a } C \textit{ such that } o : C \in \mathcal{A} \\ \{\top\} & \textit{otherwise} \end{cases}$$

By using the label of an individual, the concept expression $\bigsqcap \mathcal{L}(\Sigma, o)$ is defined as the conjunction of all the concept expressions in $\mathcal{L}(\Sigma, o)$:

$$\bigsqcap \mathcal{L}(\Sigma, o) = C_1 \sqcap \ldots \sqcap C_n$$

where $\{C_i \mid i = 1, \ldots, n\} = \mathcal{L}(\Sigma, o)$.

Since in precompleted knowledge bases the information carried by role assertions is made explicit, all the properties of an individual are in the form of concept assertions. The label of an individual completely characterises its properties, and can be used for verifying that those properties are consistent.

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* $\mathcal{I}_o = (\Delta_o, \cdot^{\mathcal{I}_o})$, which witnesses the satisfiability. As seen in Section 4.1, it can be assumed that each individual model has a tree structure.

The domains of the models can be considered pairwise disjoint without loss of generality. Let be $\mathcal{I}_u$ and $\mathcal{I}_v$ two models of $\bigsqcap \mathcal{L}(\Sigma, u)$ and $\bigsqcap \mathcal{L}(\Sigma, v)$ respectively, whose domains Δ_u and Δ_v overlap. A new domain Δ'_u can be chosen having the same cardinality as Δ_u, and being disjoint from Δ_v. An arbitrary bijection can be defined between the sets Δ_u and Δ'_u. Moreover the defined bijection can be used to build a new interpretation in which every element of Δ_u is substituted by the mapped element in Δ'_u. It is easy to show that that new interpretation $\mathcal{I}'_u = (\Delta'_u, \cdot^{\mathcal{I}_u})$ is a model for the corresponding concept $\bigsqcap \mathcal{L}(\Sigma, u)$ with respect to the given terminology.

Starting from the individual models $\mathcal{I}_o$, an interpretation for the overall knowledge base can be build by combining them. 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, which is defined because to each individual name corresponds a label (see Definition 2).

- Concept names are interpreted as the union of their interpretations in the different models.
- Interpretation of role names is slightly more involved because the union of the single interpretations is not enough. In addition there must be present the pairs corresponding to the role assertions in the ABox, including those induced by both role hierarchy, and transitive roles.

Definition 3 *Given a precompleted knowledge base* $\Sigma_{pc} = (\mathcal{T}, \mathcal{A}_{pc})$ *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:*

$$\overline{\Delta} = \bigcup_{o\in\mathcal{O}} \Delta_o$$

$$\cdot^{\overline{\mathcal{I}}} = \begin{cases} o^{\overline{\mathcal{I}}} & \mapsto \widehat{\mathcal{I}_o} \\ A^{\overline{\mathcal{I}}} & \mapsto \bigcup_{o\in\mathcal{O}} A^{\mathcal{I}_o} \\ R^{\overline{\mathcal{I}}} & \mapsto \bigcup_{o\in\mathcal{O}} R^{\mathcal{I}_o} \cup \{(a^{\overline{\mathcal{I}}}, b^{\overline{\mathcal{I}}}) \mid \langle a,b\rangle : R \in \mathcal{A}_{pc}\} \cup \bigcup_{S\preceq R} S^{\overline{\mathcal{I}}} \\ R_+^{\overline{\mathcal{I}}} & \mapsto (\bigcup_{o\in\mathcal{O}} R_+^{\mathcal{I}_o} \cup \{(a^{\overline{\mathcal{I}}}, b^{\overline{\mathcal{I}}}) \mid \langle a,b\rangle : R_+ \in \mathcal{A}_{pc}\} \cup \bigcup_{S\preceq R_+} S^{\overline{\mathcal{I}}})^+ \end{cases}$$

For each $o \in \mathcal{O}$, $A \in \mathcal{CN}$, $R \in \mathcal{RN} \setminus \mathcal{RN}_+$ *and* $R_+ \in \mathcal{RN}_+$.

The fact that the union interpretation is a model for the precompleted KB is intuitive for the propositional part (concept conjunction, disjunction and negation), but not so for the modal part. In particular, trouble can arise from the interaction between the universal quantification and the newly added pairs. However, the domain disjointness and tree structure assumptions guarantee that the properties of role interpretations overcome this problem. This point is developed in the following section.

4.2.1 Interpretation of roles

The disjointness of the domains guarantees that the individual model trees do not overlap, so the role interpretations $R^{\mathcal{I}_o}$ are pairwise disjoint. This property is crucial for the structure of the role extension in the union interpretation, since it ensures that new links are added only to root nodes (those associated to individual names):

Proposition 5 *Given a precompleted knowledge base* $\Sigma_{pc} = (\mathcal{T}, \mathcal{A}_{pc})$, *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 pairs* $(x,y) \in R^{\overline{\mathcal{I}}}$:

$$x \in (\Delta_o \setminus \{o^{\overline{\mathcal{I}}}\}) \Rightarrow (x,y) \in R^{\mathcal{I}_o}$$

PROOF.
This proposition can easily be proved by induction using the partial order $\preceq$ over the set of roles $\mathcal{RN}$. In fact for the basic cases the interpretation become $\bigcup_{o\in\mathcal{O}} R^{\mathcal{I}_o}$ for non transitive roles, and $(\bigcup_{o\in\mathcal{O}} R^{\mathcal{I}_o} \cup \{(a^{\overline{\mathcal{I}}}, b^{\overline{\mathcal{I}}}) \mid \langle a,b\rangle : R \in \mathcal{A}_{pc}\})^+$ for the transitive roles, which add only pairs involving roots because $R^{\mathcal{I}_o}$ is already transitive.

The induction step for the non–transitive case is trivial because of the fact that the three components of the definition $\bigcup_{o\in\mathcal{O}} R^{\mathcal{I}_o}$, $\{(a^{\overline{\mathcal{I}}}, b^{\overline{\mathcal{I}}}) \mid \langle a,b\rangle : R \in \mathcal{A}_{pc}\}$, and $\bigcup_{S\preceq R_+} S^{\overline{\mathcal{I}}}$ satisfy the condition (see Definition 3). The transitive case is proved by showing that, given the role extension $\overline{R} = \bigcup_{o\in\mathcal{O}} R^{\mathcal{I}_o} \cup \{(a^{\overline{\mathcal{I}}}, b^{\overline{\mathcal{I}}}) \mid \langle a,b\rangle : R \in \mathcal{A}_{pc}\} \cup \bigcup_{S\preceq R} S^{\overline{\mathcal{I}}}$, for each natural n the role extension $\overline{R}^n$ satisfies the condition.[1] Clearly is true for $n = 1$ because is the same as for the non–transitive case. Let the condition holding for $i = 1 \dots n$ (i.e. for each $(x,y) \in \overline{R}^i$ $x \in (\Delta_o \setminus \{o^{\overline{\mathcal{I}}}\}) \Rightarrow (x,y) \in R^{\mathcal{I}_o}$), and the pair (x,y) being in the extension of $\overline{R}^{n+1}$. Then there is a z such that $(x,z) \in \overline{R}$ and $(z,y) \in \overline{R}^n$, in addition $\{(x,z),(z,y)\} \subseteq R^{\mathcal{I}_o}$ for the induction hypothesis; therefore $(x,y) \in R^{\mathcal{I}_o}$ because of $R^{\mathcal{I}_o}$ is transitive.

[1] The extension of $\overline{R}^n$ is defined by $\overline{R}^1 = \overline{R}$, and $\overline{R}^{n+1} = \overline{R} \circ \overline{R}^n$

□

One of the consequences of the last proposition is that if a pair of individuals from different individual domains is in the extension of a role, then the first element of the pair must be a root. This can easily seen because from Proposition 5 and the individual domain disjointness, there cannot be a non–root individual connected outside its individual domain.

The second important property of role extensions in union interpretation is that there cannot be a 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 an individual are via the root of its own individual domain, instead of by being directly propagated from different individual domains.

Proposition 6 *Given a precompleted knowledge base* $\Sigma_{pc} = (\mathcal{T}, \mathcal{A}_{pc})$*, 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* $R \in \mathcal{RN}$ *and pair* $(u^{\overline{\mathcal{I}}}, x) \in R^{\overline{\mathcal{I}}}$*, if* $x \in \Delta_v$ *where* $u \neq v$*, then either*

1. $x = v^{\overline{\mathcal{I}}}$ *or*
2. *there is a role* $S \in \mathcal{RN}_+$ *such that* $S \preceq R$ *and both* $(u^{\overline{\mathcal{I}}}, v^{\overline{\mathcal{I}}})$*,* $(v^{\overline{\mathcal{I}}}, x)$ *are in* $S^{\overline{\mathcal{I}}}$*.*

PROOF.
The proof follows the same structure of the previous Proposition 5. Basic cases are trivial because for non–transitive roles just pairs involving roots are added, while for transitive roles the new added paths always pass through roots.

Induction step for the non–transitive roles is verified because of the property is verified by all the components. Transitive roles are handled by showing that for any natural n, if $(u^{\overline{\mathcal{I}}}, x)$ is in $\overline{R}^n$ then $x = v^{\overline{\mathcal{I}}}$ or there exists a role $S \in \mathcal{RN}_+$ such that $S \preceq R$ and both $(u^{\overline{\mathcal{I}}}, v^{\overline{\mathcal{I}}})$, $(v^{\overline{\mathcal{I}}}, x)$ are in $S^{\overline{\mathcal{I}}}$. With $n = 1$ the very same considerations as non–transitive roles apply. Let the condition holding for n, and the pair $(u^{\overline{\mathcal{I}}}, x)$ being in the extension of $\overline{R}^{n+1}$; then there is a z such that $(u^{\overline{\mathcal{I}}}, z) \in \overline{R}$ and $(z, x) \in \overline{R}^n$. There can be three cases: $z = v^{\overline{\mathcal{I}}}$, $z \in \Delta_v$ with $z \neq v^{\overline{\mathcal{I}}}$, and $z = o^{\overline{\mathcal{I}}}$ with $o \neq v$.

1. If $z = v^{\overline{\mathcal{I}}}$, then the condition is satisfied because $\overline{R}^i \subseteq R$ for any i and R is transitive.
2. If $z \in \Delta_v$ with $z \neq v^{\overline{\mathcal{I}}}$, then $(z, x) \in R^{\mathcal{I}_o}$ by Proposition 5 so $(z, x) \in R^{\overline{\mathcal{I}}}$. On the other hand there is a transitive role S such that $S \preceq R$ and both $(u^{\overline{\mathcal{I}}}, v^{\overline{\mathcal{I}}})$, $(v^{\overline{\mathcal{I}}}, z)$ are in $S^{\overline{\mathcal{I}}}$ (is the basic case for $n = 1$); therefore $\{(u^{\overline{\mathcal{I}}}, v^{\overline{\mathcal{I}}}), (v^{\overline{\mathcal{I}}}, z), (z, x)\} \subseteq R^{\overline{\mathcal{I}}}$ and since R is transitive the condition is verified.
3. If $z = o^{\overline{\mathcal{I}}}$ with $o \neq v$, then by using the induction hypothesis $\{(o^{\overline{\mathcal{I}}}, v^{\overline{\mathcal{I}}}), (v^{\overline{\mathcal{I}}}, x)\} \subseteq R^{\overline{\mathcal{I}}}$ and since $(u^{\overline{\mathcal{I}}}, z) \in R^{\overline{\mathcal{I}}}$ as well, and R is transitive the condition is verified.

□

4.2.2 Interpretation of concepts

It is easy to see that interpretation of roles satisfies the role assertions in the KB, because of their simple form (see Section 2); however concept assertions are more problematic. In fact, although the interpretation of a concept name is simply the union of the interpretations of individual models, the extension of some concept expressions can violate the semantics of concept forming operators because of the presence of new pairs in the interpretation of roles. For example the simple knowledge base $\{\langle a, a\rangle : R\}$ has one single individual model which can be composed by the domain $\Delta_a = \{0\}$ with the interpretation function $a^{\mathcal{I}_a} = 0$: the union interpretation maps R to $(0, 0)$ and a to 0. It is easy to verify that $a^{\mathcal{I}_a} \in (\forall R.C)^{\mathcal{I}_a}$, but in the interpretation function it is no longer true.

This problem is localised to root individuals, because the interpretation of roles restricted to non–root individuals does not change. In fact, the interpretation of arbitrary concept expressions, restricted to non–root individuals is a monotonic extension of the one in individual models:

Proposition 7 *For any individual name $a \in \mathcal{O}$ and concept expression D, $D^{\mathcal{I}_a} \setminus \{a^{\overline{\mathcal{I}}}\} \subseteq D^{\overline{\mathcal{I}}}$.*

PROOF.
The proposition is proved by induction on the structure of the concept expression D. The basic case of a concept name is true by construction of $\overline{\mathcal{I}}$ (see Definition 3). If it is true for the concept expressions C_1 and C_2, then for each more complex expression:

$\neg A$ Let be $x \in \Delta_a \setminus \{a^{\overline{\mathcal{I}}}\}$ such that $x \notin A^{\overline{\mathcal{I}_a}}$, then $x \notin A^{\overline{\mathcal{I}}}$ by construction (the individual domains are disjoint), so $x \in (\neg A)^{\overline{\mathcal{I}}}$.

$(C_1 \sqcap C_2)$ By definition $(C_1 \sqcap C_2)^{\overline{\mathcal{I}}} = C_1^{\overline{\mathcal{I}}} \cap C_2^{\overline{\mathcal{I}}}$ and $C_1^{\mathcal{I}_a} \subseteq C_1^{\overline{\mathcal{I}}}$, $C_2^{\mathcal{I}_a} \subseteq C_2^{\overline{\mathcal{I}}}$ for the induction hypothesis; therefore $(C_1 \sqcap C_2)^{\mathcal{I}_a} \subseteq (C_1 \sqcap C_2)^{\overline{\mathcal{I}}}$.

$(C_1 \sqcup C_2)$ It is analogous to the $\sqcap$ case.

$(\exists R.C_1)$ If $x \in (\exists R.C_1)^{\mathcal{I}_a} \setminus \{a^{\overline{\mathcal{I}}}\}$, then there exists $y \in \Delta_a$ such that $(x, y) \in R^{\mathcal{I}_a}$ and $y \in C_1^{\mathcal{I}_a}$. For the construction of $\overline{\mathcal{I}}$ the pair $(x, y) \in R^{\overline{\mathcal{I}}}$ (see Definition 3), and $y \in C_1^{\overline{\mathcal{I}}}$ for the induction hypothesis; therefore $x \in (\exists R.C_1)^{\overline{\mathcal{I}}}$.

$(\forall R.C_1)$ Suppose that $x \in (\forall R.C_1)^{\mathcal{I}_a} \setminus \{a^{\overline{\mathcal{I}}}\}$, and an arbitrary $(x, y) \in R^{\overline{\mathcal{I}}}$, then for Proposition 5 (x, y) is in $R^{\mathcal{I}_a}$ so $y \in C_1^{\mathcal{I}_a}$. In addition, y cannot be equal to $a^{\overline{\mathcal{I}}}$ because of the tree model property of $\mathcal{I}_a$. Therefore $y \in C_1^{\overline{\mathcal{I}}}$ for the induction hypothesis.

□

For root individuals this monotonicity property does not hold for arbitrary concept expressions; however, it is still valid for concept expressions which appear as assertions in the precompleted KB (Proposition 8). This restricted property (together with the more general 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 9).

Proposition 8 *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.

- The basic case is a concept name: If $a : A$ is in $\mathcal{A}_{pc}$ then $A \in \mathcal{L}(\Sigma_{pc}, a)$ so $a \in A^{\mathcal{I}_a}$; therefore $a^{\overline{\mathcal{I}}} \in A^{\overline{\mathcal{I}}}$ because $A^{\mathcal{I}_a} \subseteq A^{\overline{\mathcal{I}}}$.
- If it is true for the concept expressions C_1 and C_2, then for each more complex expression:

  $\neg A$ If $a : \neg A \in \mathcal{A}_{pc}$, then $\neg A \in \mathcal{L}(\Sigma_{pc}, a)$ so $a^{\overline{\mathcal{I}}} \notin A^{\mathcal{I}_a}$. For the disjointness of the individual domains $a^{\overline{\mathcal{I}}} \notin A^{\overline{\mathcal{I}}}$ (see Definition 3), therefore $a^{\overline{\mathcal{I}}} \in (\neg A)^{\overline{\mathcal{I}}}$

  $(C_1 \sqcap C_2)$ If $a : (C_1 \sqcap C_2) \in \mathcal{A}_{pc}$, then both $a : C_1$ and $a : C_2$ are in $\mathcal{A}_{pc}$ (Proposition 3). For the induction hypothesis $a^{\overline{\mathcal{I}}}$ is in both $C_1^{\overline{\mathcal{I}}}$ and $C_2^{\overline{\mathcal{I}}}$, so it is in $(C_1 \sqcap C_2)^{\overline{\mathcal{I}}}$ as well.

  $(C_1 \sqcup C_2)$ If $a : (C_1 \sqcap C_2) \in \mathcal{A}_{pc}$, then either $a : C_1$ or $a : C_2$ is in $\mathcal{A}_{pc}$ (Proposition 3). If $a : C_1$ is in $\mathcal{A}_{pc}$, then for induction hypothesis $a^{\overline{\mathcal{I}}} \in C_1^{\overline{\mathcal{I}}}$; therefore $a^{\overline{\mathcal{I}}} \in (C_1 \sqcup C_2)^{\overline{\mathcal{I}}}$ as well. The same if $a : C_2$ is in $\mathcal{A}_{pc}$.

  $(\exists R.C_1)$ If $a : (\exists R.C_1)$ is in $\mathcal{A}_{pc}$ then $(\exists R.C_1) \in \mathcal{L}(\Sigma_{pc}, a)$; so it exists $x \in \Delta_a$ such that $(a^{\overline{\mathcal{I}}}, x) \in R^{\mathcal{I}_a}$ and $x \in C_1^{\mathcal{I}_a}$. The individual x is different from $a^{\overline{\mathcal{I}}}$ because the model $\mathcal{I}_a$ is a tree (see Definition 3), so $x \in C_1^{\overline{\mathcal{I}}}$ (Proposition 7). In addition, $R^{\mathcal{I}_a} \subseteq R^{\overline{\mathcal{I}}}$ (see Definition 3), therefore $a^{\overline{\mathcal{I}}} \in (\exists R.C_1)^{\overline{\mathcal{I}}}$.

($\forall R.C_1$) Suppose that $(a^{\overline{\mathcal{I}}}, x) \in R^{\overline{\mathcal{I}}}$, then either $x \in \Delta_a \setminus \{a^{\overline{\mathcal{I}}}\}$, $x = a^{\overline{\mathcal{I}}}$, it exists a $b \neq a$ such that $x \in \Delta_b \setminus \{b^{\overline{\mathcal{I}}}\}$, or $x = b^{\overline{\mathcal{I}}}$. Each one of the four cases is considered:

- If $x \in \Delta_a \setminus \{a^{\overline{\mathcal{I}}}\}$, then $(a^{\overline{\mathcal{I}}}, x) \in R^{\mathcal{I}_a}$ for Proposition 5. In addition, the concept $(\forall R.C_1)$ is in $\mathcal{L}(\Sigma_{pc}, a)$ (see Definition 3) so $x \in C_1^{\mathcal{I}_a}$. By Proposition 7 $C_1^{\mathcal{I}_a} \setminus \{a^{\overline{\mathcal{I}}}\} \subseteq C_1^{\overline{\mathcal{I}}}$, so $x \in C_1^{\overline{\mathcal{I}}}$.
- Since $(a^{\overline{\mathcal{I}}}, a^{\overline{\mathcal{I}}}) \notin R^{\mathcal{I}_a}$ because of the tree structure of the model $\mathcal{I}_a$, this case is the same as $x = b^{\overline{\mathcal{I}}}$.
- If $x = b^{\overline{\mathcal{I}}}$ then either $\langle a, b\rangle : S \in \mathcal{A}_{pc}$ with $S \preceq R$, or there is a sequence $\langle a, o_1\rangle : S_1, \langle o_1, o_2\rangle : S_2, \ldots, \langle o_n, b\rangle : S_{n+1}$ in $\mathcal{A}_{pc}$ and a role $S \in \mathcal{RN}_+$ such that $S_i \preceq S \preceq R$ for all $i = 1, \ldots, n+1$. In the first case the assertion $b : C_1$ is in $\mathcal{A}_{pc}$ by Proposition 3. In the second case it is easy to see that every o_i has the assertion $\forall R.C_1$ by Proposition 3, so $(\forall R.C_1) \in \mathcal{L}(\Sigma_{pc}, o_n)$ and $C_1 \in \mathcal{L}(\Sigma_{pc}, b)$. Therefore $b^{\overline{\mathcal{I}}} \in C_1^{\mathcal{I}}$, so $b^{\overline{\mathcal{I}}} \in C_1^{\overline{\mathcal{I}}}$ by induction hypothesis.
- If $x \in \Delta_b \setminus \{b^{\overline{\mathcal{I}}}\}$ then, for Proposition 6 there should be a transitive role $S \preceq R$ such that $(a^{\overline{\mathcal{I}}}, b^{\overline{\mathcal{I}}}), (b^{\overline{\mathcal{I}}}, x) \in S^{\overline{\mathcal{I}}} \subseteq R^{\overline{\mathcal{I}}}$. As for the previous case the concept $(\forall R.C_1)$ should be in the label of b, so $x \in C_1^{\mathcal{I}_b}$. Using the previous Proposition 7 it can be concluded that $x \in C_1^{\overline{\mathcal{I}}}$.

□

4.2.3 Satisfiability of the knowledge base

It is clear that a model for the knowledge base is a model for every individual concept. Given the results in the previous sections, Propositions 9 shows that the union interpretation is a model for the precompleted knowledge base.

Proposition 9 *An $\mathcal{ALCH}_{\mathcal{R}^+}$ precompleted knowledge base $\Sigma_{pc} = (\mathcal{T}_{pc}, \mathcal{A}_{pc})$ is satisfiable if and only if for each individual name $o \in \mathcal{O}$ the concept expression $\bigsqcap \mathcal{L}(\Sigma_{pc}, o)$ (see Definition 2) is satisfiable with respect the terminology $\mathcal{T}_{pc}$.*

PROOF.

$\Rightarrow$ Let be $\mathcal{I}$ a model for Σ_{pc}; then for any individual name o and for any concept C in $\mathcal{L}(\Sigma_{pc}, o)$ the individual $o^{\mathcal{I}}$ is in the extension $C^{\mathcal{I}}$ (see Definition 2). Therefore $o^{\mathcal{I}} \in (\bigsqcap \mathcal{L}(\Sigma_{pc}, o))^{\mathcal{I}}$ so its interpretation is non–empty with respect the terminology $\mathcal{T}_{pc}$.

$\Leftarrow$ If for every named individual o the concept expression $\bigsqcap \mathcal{L}(\Sigma_{pc}, o)$ is satisfiable with respect to the terminology $\mathcal{T}_{pc}$, there is a tree structured model $\mathcal{I}_o$ (Proposition 4); therefore the union interpretation $\overline{\mathcal{I}} = (\overline{\Delta}, \cdot^{\overline{\mathcal{I}}})$ can be defined as Definition 3. Verifying that interpretation $\overline{\mathcal{I}}$ satisfies every assertion in Σ_{pc} it can be proved that it is a model for Σ_{pc}.

- Let be $a : C$ a concept assertion in $\mathcal{A}_{pc}$, then by Proposition 8 $a^{\overline{\mathcal{I}}} \in C^{\overline{\mathcal{I}}}$.
- Let be $\langle a, b\rangle : R$ a role assertion in $\mathcal{A}_{pc}$, then $(a^{\overline{\mathcal{I}}}, b^{\overline{\mathcal{I}}}) \in R^{\overline{\mathcal{I}}}$ by construction of $R^{\overline{\mathcal{I}}}$ in Definition 3.
- Let be $\top \sqsubseteq C$ an axiom in $\mathcal{T}_{pc}$, then $\overline{\mathcal{I}}$ satisfies the axiom if and only if every element in the domain $\overline{\Delta}$ is in the extension of C with respect to $\overline{\mathcal{I}}$. Let be x an element of $\overline{\Delta}$, then it 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 $x \in \Delta_a \setminus \{a^{\overline{\mathcal{I}}}\}$

  - If $x = a^{\overline{\mathcal{I}}}$ then $x \in C^{\overline{\mathcal{I}}}$ because of Proposition 8 ($a : C$ is in the precompleted ABox $\mathcal{A}_{pc}$ by Proposition 3).
  - If $x \neq a^{\overline{\mathcal{I}}}$, then $x \in C^{\overline{\mathcal{I}}}$ by Proposition 7.

- If $R \in \mathcal{RN}_+$ then $R^{\overline{\mathcal{I}}}$ is transitive by construction as in Definition 3.
- If $S \sqsubseteq R$ is an axiom in $\mathcal{T}_{pc}$, then $S^{\overline{\mathcal{I}}} \subseteq R^{\overline{\mathcal{I}}}$ by construction as in Definition 3.

□

4.3 Knowledge base satisfiability

As summarised in Theorem 1, the satisfiability of a general knowledge base can be verified by checking the satisfiability of its precompletions via a terminological reasoner.

Theorem 1 *The satisfiability checking problem for an $\mathcal{ALCH}_{\mathcal{R}^+}$ knowledge base can be reduced to the concept satisfiability with respect to a terminology.*

PROOF.
A knowledge base Σ is satisfiable if and only if one of the precompletions $\Sigma_1, \ldots, \Sigma_n$ generated by the algorithm in Definition 1 is satisfiable (Proposition 2). Moreover, the problem of checking the satisfiability of each one of the precompletions can be solved by a concept satisfiability checking for each named individual in the knowledge base (Proposition 9).

□

This means that we now have a KB satisfiability algorithm: the precompletion rules of Figure 1 provide an algorithmic way to enumerate the precompletions of a knowledge base, while the satisfiability of each precompletion is verified using an external terminological system as FaCT (see Horrocks [1998]).

5 Discussions

Two different approaches to ABox reasoning with general terminologies have previously been presented. The first, based on the so called *encoding technique* (see De Giacomo and Lenzerini [1996]), deals with very expressive DL languages and transforms the knowledge base into a terminology. In this approach the individuals are represented by concept names, while a set of additional axioms involving those names ensures that the generated terminology is satisfiable if and only if the original KB is satisfiable. This method has not lead to any practical implementations. The encoding requires a very expressive underlying DL language, for which no efficient reasoner is yet available; it is also the case that the size of the generated terminology, although polynomial, soon become unmanageable when the number of individuals grows. In the second approach the standard tableau technique has been extended for treating general inclusion axioms (see Buchheit et al. [1993], Haarslev et al. [1998]). A possible problem with this approach is that it is not clear that the optimisations which are necessary to obtain a usable terminological system are still applicable under such an extension. Also, since the work has a different scope, the DL language considered in Buchheit et al. [1993] includes neither transitive roles nor role hierarchy, although it is possible that the technique can be modified to include these two extensions.

The technique we have presented in this report can be implemented straightforwardly because it relies only on the availability of an efficient terminological reasoner (such as FaCT). Because of the modularity of the approach, we need have no worries about any adverse interaction with the optimisation techniques used by the reasoner.

There are other possible advantages to our approach that can only be verified by testing after implementation. For instance the fact that the algorithm relies heavily on the topology of the relation between individuals suggests good behaviour for large "loosely" connected KBs. Other possible good target applications are those where the knowledge base is "nearly" a model, with only a small percentage of the individuals having complex concept assertions; for example KBs automatically derived from a database.

References

Alessandro Artale, Enrico Franconi, Nicola Guarino, and Luca Pazzi. Part-whole relations in object-centered systems: An overview. *Data and Knowledge Engineering (DKE)*, 20:347–383, 1996.

Martin Buchheit, Francesco M. Donini, and Andrea Schaerf. Decidable reasoning in terminological knowledge representation systems. *Journal of Artificial Intelligence Research*, 1:109–138, 1993.

Giuseppe De Giacomo and Maurizio Lenzerini. TBox and ABox reasoning in expressive description logics. In *Proceedings of the Fifth International Conference on the Principles of Knowledge Representation and Reasoning (KR-96)*, pages 316–327. Morgan Kaufmann Publishers, 1996.

F.M. Donini, M. Lenzerini, D. Nardi, and A. Schaerf. Deduction in concept languages: From subsumption to instance checking. *Journal of Logic and Computation*, 4(4):423–452, 1994.

Enrico Franconi, Giuseppe De Giacomo, Robert M. MacGregor, Werner Nutt, and Christopher A. Welty, editors. *Proceeding of the 1998 International Workshop on Description Logics (DL'98)*, 1998. CEUR Pubblication.

Volker Haarslev, Ralf Moeller, and Anni-Yasmin Turhan. Implementing an alcrp(d) abox reasoner - progress report. In Franconi et al. [1998].

Bernhard Hollunder. *Algorithmic Foundations of Terminological Knowledge Representation Systems*. PhD thesis, Universität des Saarlandes, 1994.

I. Horrocks. Using an expressive description logic: FaCT or fiction? In *Proc. of the 6th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR'98)*, 1998.

I. Horrocks and G. Gough. Description logics with transitive roles. LRI, Universitè PARIS-SUD, Cente d'Orsay, 1997.

I. Horrocks and U. Sattler. A description logic with transitive and inverse roles and role hierarchies. In Franconi et al. [1998].