ABox Reasoning with Transitive Roles and Axioms

Sergio Tessaris and Graham Gough
University of Manchester

1 Introduction

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 [1]). Several previous papers (see [8, 9]) 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 paper extends the above work by presenting an algorithm for reasoning within a knowledge representation which also includes ABox assertions (Proofs are presented in outline only; an extended version with the complete proofs is in [10]).

The approach used is based on an extension of the so–called *precompletion* technique [6, 4] 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 paper 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 [5]). 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.

1.1 $\mathcal{ALCH}_{\mathcal{R}^+}$ knowledge bases

The DL $\mathcal{ALCH}_{\mathcal{R}^+}$ extends $\mathcal{ALC}$ with transitively closed primitive roles and simple role axioms. Valid concept expressions are defined by the syntax:

$$\begin{aligned} C \quad ::= \quad & \top \mid \bot \mid A \mid \neg A \mid C_1 \sqcap C_2 \mid C_1 \sqcup C_2 \mid \\ & \forall R.C \mid \exists R.C \end{aligned}$$

where A is a concept name chosen from the set $\mathcal{CN}$ and R a role name from $\mathcal{RN}$; the subset $\mathcal{RN}_+ \subseteq \mathcal{RN}$ is the set of transitive roles. A standard Tarski style model theoretic semantics provides the meaning for the expressions by means of interpretations $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ (see [7]). The set $\Delta^{\mathcal{I}}$ is the domain, and $\cdot^{\mathcal{I}}$ is an interpretation function which 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 $\Delta^{\mathcal{I}}$. In addition, transitive roles must satisfy the condition $R^{\mathcal{I}} = (R^{\mathcal{I}})^+$, where $\cdot^+$ indicates transitive closure.

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 concept axioms of the form $\top \sqsubseteq C$ and role axioms of the form $S \sqsubseteq R$; while the ABox contains assertions about a set of individual names $\mathcal{O}$. These assertions are of the form $a : C$ or $\langle a, b \rangle : R$, where a,b are individual names in $\mathcal{O}$. The semantics of a KB is given by an interpretation which maps individual names to different elements of the interpretation domain:

TBox	$\top \sqsubseteq C$	$\top^{\mathcal{I}} \subseteq C^{\mathcal{I}}$
	$S \sqsubseteq R$	$S^{\mathcal{I}} \subseteq R^{\mathcal{I}}$
ABox	$a : C$	$a^{\mathcal{I}} \in C^{\mathcal{I}}$
	$\langle a, b\rangle : R$	$(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$

Note that restricting the syntax of concept 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 $\top \sqsubseteq C_2 \sqcup \neg C_1$. 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 (it is a model for the knowledge base).

2 KB satisfiability

The proof is divided in two parts: in the first 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 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 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.

To check the satisfiability of a precompletion we show that both 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,[1] then their models can be combined in a model for the precompleted knowledge base.

2.1 Precompletions of a KB

The definition of a precompletion for a knowledge base $\Sigma = (\mathcal{T}, \mathcal{A})$ is given in a procedural way as a new KB $\Sigma_{pc} = (\mathcal{T}, \mathcal{A}_{pc})$ where the ABox $\mathcal{A}_{pc}$ is obtained by extending $\mathcal{A}$ using the nondeterministic syntactic rules in Figure 1 as long as they are applicable.[2]

[1] A concept is satisfiable if there exists an interpretation (the model), satisfying the assertions in the terminology, in which the extension of the concept is not empty.

[2] In the rules the symbol $\preceq$ represents the partial order induced by the role inclusion assertions.

$\mathcal{A} \rightarrow_{\sqsubseteq} \{o : C\} \cup \mathcal{A}$
 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 $\mathcal{A}$, $D = C_1$ 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 $R \in \mathcal{RN}_+$, $S \preceq R \preceq T$,
 and $o' : C$ is not in $\mathcal{A}$.

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

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. In fact the only rule that does not introduce a smaller assertion in the knowledge base is the $\rightarrow_{\forall+}$, but its applicability is bounded by the number of role assertions which is invariant. The number of precompletions of a KB can be exponential because of the presence of a nondeterministic rule, however the size of each precompletion is polynomial with respect to the size of the original KB.

The set of all precompletions of a knowledge base characterises its satisfiability:

Proposition 1 *A knowledge base $\Sigma = (\mathcal{T}, \mathcal{A})$ is satisfiable if and only if it has a satisfiable precompletion.*

The "if" direction is trivial because Σ is included in all its precompletions, so a model for a precompletion is a model for the knowledge base as well. The "only if" part is proved by using a technique similar to that used in [9]. A model $\mathcal{I}$ of Σ which witnesses its satisfiability is used to deterministically guide the application of precompletion rules to an unique satisfiable precompletion. For this purpose the only nondeterministic rule is transformed into its deterministic counterpart:

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

Satisfiability is preserved by rule application since 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. Therefore, given the fact that the terminology does not change, $\mathcal{I}$ is a model for the generated precompletion.

2.2 Precompletion satisfiability

Given a precompleted knowledge base Σ_{pc}, we can construct, for each individual o in the KB, the *individ-*

ual concept $\mathcal{C}(\Sigma_{pc}, o)$, for o. This is defined to be the conjunction of all the concept expressions in the set $\{C \mid o : C \in \mathcal{A}\}$, or $\top$ if there are not assertions about o. In Proposition 2 we claim that a precompleted knowledge base is satisfiable if and only if all these individual concepts are satisfiable. It is clear that a model for the knowledge base is a model for every individual concept, we now show how, given models for the individual concepts, we can build a model for the knowledge base.

If each individual concept $\mathcal{C}(\Sigma_{pc}, o)$ 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. Interpretations can be infinite, but without loss of generality it can be assumed that their interpretation domains are pairwise disjoint, as well as having a tree structure,[3] whose root elements (indicated by $\widehat{\mathcal{I}_o}$) are in the extension of the corresponding individual concept $\mathcal{C}(\Sigma_{pc}, o)$ (as shown in [7]).

The *union interpretation* $\overline{\mathcal{I}} = (\overline{\Delta}, \cdot^{\overline{\mathcal{I}}})$ for the precompleted knowledge base is defined over a domain consisting of the union of all the domains from each individual model ($\overline{\Delta} = \bigcup_{o \in \mathcal{O}} \Delta_o$). On this domain the interpretation function is defined in terms of the individual interpretation functions for each $o \in \mathcal{O}$, $A \in \mathcal{CN}$, $R \in \mathcal{RN} \setminus \mathcal{RN}_+$ and $R_+ \in \mathcal{RN}_+$:

$$\begin{array}{lcl} o^{\overline{\mathcal{I}}} & \mapsto & \widehat{\mathcal{I}_o} \\ A^{\overline{\mathcal{I}}} & \mapsto & \bigcup_{o \in \mathcal{O}} A^{\mathcal{I}_o} \\ R^{\overline{\mathcal{I}}} & \mapsto & \bigcup_{o \in \mathcal{O}} R^{\mathcal{I}_o} \cup \overline{R}^{\overline{\mathcal{I}}} \cup \bigcup_{S \preceq R} S^{\overline{\mathcal{I}}} \\ R_+^{\overline{\mathcal{I}}} & \mapsto & (\bigcup_{o \in \mathcal{O}} R_+^{\mathcal{I}_o} \cup \overline{R}_+^{\overline{\mathcal{I}}} \cup \bigcup_{S \preceq R_+} S^{\overline{\mathcal{I}}})^+ \end{array}$$

New pairs are added to the interpretation of roles by interpreting the role assertions in the knowledge base as

$$\overline{R}^{\overline{\mathcal{I}}} = \{(a^{\overline{\mathcal{I}}}, b^{\overline{\mathcal{I}}}) \mid \langle a, b\rangle : R \in \mathcal{A}_{pc}\}$$

and ensuring that transitivity and role hierarchy are satisfied.

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.

- First, new links are added only to root nodes (those associated to individual names); in fact for each pair in $(x, y) \in R^{\overline{\mathcal{I}}}$, $x \in (\Delta_o \setminus \{o^{\overline{\mathcal{I}}}\})$ implies that (x, y) is in $R^{\mathcal{I}_o}$. An important consequence of this property is that if a role connects two individuals of different individual domains, then the first element of the pair must be a root individual.
- Second, there cannot be a connection between elements from different individual domains without a path passing through a root node. That is, given two distinct individuals u and v, whenever there is a pair $(u^{\overline{\mathcal{I}}}, x) \in R^{\overline{\mathcal{I}}}$ with $x \in \Delta_v$, there should be a transitive role S included in R such that both pairs $(u^{\overline{\mathcal{I}}}, v^{\overline{\mathcal{I}}})$, $(v^{\overline{\mathcal{I}}}, x)$ are in $S^{\overline{\mathcal{I}}}$, or $x = v^{\overline{\mathcal{I}}}$. 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.

It is easy to see that interpretation of roles satisfies the role assertions in the KB, because of their simple form (see Section 1.1); 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, given the simple knowledge base $\{\langle a, a\rangle : R\}$, an individual model for it can be $\Delta_a = \{0\}$ 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 union interpretation this 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; i.e. for any concept expression D, $D^{\mathcal{I}_o} \setminus \{o^{\overline{\mathcal{I}}}\} \subseteq D^{\overline{\mathcal{I}}}$ for each individual name o.

For root individuals this property does not hold for arbitrary concept expressions; however, the property is still valid for concept expressions which appear as assertions in the precompleted KB. In fact, for each assertion $o : D \in \mathcal{A}_{pc}$ the individual $a^{\overline{\mathcal{I}}}$ is in the extension $D^{\overline{\mathcal{I}}}$. 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. At this point the arguments are enough to support the satisfiability result for precompleted knowledge bases.

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

[3] Viewing the model as a graph, whose arcs are labelled by the roles and the nodes by the set of primitive concepts the element belong to.

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. The proof is simply the combination of Proposition 2 and Proposition 1.

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.*

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 [7].

3 Discussions

Two different approaches to ABox reasoning with general terminologies have previously been presented. The first, based on the so called *encoding technique* (see [3]), 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 [2, 11]). 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 [2] 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 paper 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

[1] A. Artale, E. Franconi, N. Guarino, and L. Pazzi. Part-whole relations in object-centered systems: An overview. *Data and Knowledge Engineering (DKE)*, 1996.

[2] M. Buchheit, F.M. Donini, and A. Schaerf. Decidable reasoning in terminological knowledge representation systems. *Journal of Artificial Intelligence Research*, 1993.

[3] G. De Giacomo and M. Lenzerini. TBox and ABox reasoning in expressive description logics. In *Proc. of KR'96*, 1996.

[4] F.M. Donini, M. Lenzerini, D. Nardi and A. Schaerf Deduction in Concept Languages: From subsumption to instance checking. *Journal of Logic and Computation*, 1994.

[5] E. Franconi, G. De Giacomo, R.M. MacGregor, W. Nutt, and C.A. Welty, editors. *Proc. of DL'98*, 1998.

[6] B. Hollunder. *Algorithmic Foundations of Terminological Knowledge Representation Systems.* PhD thesis, Universität des Saarlandes, 1994.

[7] I. Horrocks. Using an expressive description logic: FaCT or fiction? In *Proc. of KR'98*, 1998.

[8] I. Horrocks and G. Gough. Description logics with transitive roles. In *Proc. of DL'97*, 1997.

[9] I. Horrocks and U. Sattler. A description logic with transitive and inverse roles and role hierarchies. In *Proc. of DL'98*, 1998.

[10] S. Tessaris. Precompletion algorithm for $\mathcal{ALCH}_{\mathcal{R}^+}$ knowledge bases with axioms. Technical report, Computer Science Dept. – University of Manchester, 1999. To appear.

[11] V. Haarslev, R. Moeller and A. Turhan Implementing an ALCRP(D) ABox Reasoner - Progress Report In *Proc. of DL'98*, 1998.