Knowledge Compilation for Description Logics

Ulrich Furbach and Claudia Obermaier

Universität Koblenz-Landau
D56070 Koblenz, Germany
{uli,obermaie}@uni-koblenz.de

1 Introduction

Knowledge compilation is a technique for dealing with computational intractability of propositional reasoning. Transforming a given knowledge base into a special normal form ([MR03,DH05]) is aiming at efficient reasoning. We propose to apply this technique to knowledge bases defined in Description Logics. For this, we introduce a normal form, called linkless concept descriptions. We present an algorithm, which can be used to transform a given concept description into an equivalent linkless concept description. The normal form supports a linear satisfiability check and efficient projection operators. Techniques like structural subsumption algorithms ([BN03]), normalization ([BH98] and absorption ([Hor98]) are related to our approach. Further [SK96] introduces the idea of approximating \( F\mathcal{L} \) concept descriptions by \( F\mathcal{L}^\sim \) concept descriptions.

2 Linkless Concepts

In the following we only consider \( A\mathcal{LC} \) concept descriptions which are given in Negation Normal Form (NNF). The term concept literal denotes an atomic concept or a negated atomic concept, role literal denotes a concept description of the form \( \forall R.E \) or \( \exists R.E \) with \( E \) a concept in NNF. By the term literal we mean a concept literal or a role literal.

**Definition 1.** For a given concept \( C \), the set of its c-paths is defined as follows:

- \( c\text{-paths}(C) = \{\{C\}\} \), if \( C \) is a literal
- \( c\text{-paths}(C_1 \cap C_2) = \{X \cup Y | X \in c\text{-paths}(C_1) \text{ and } Y \in c\text{-paths}(C_2)\} \)
- \( c\text{-paths}(C_1 \cup C_2) = c\text{-paths}(C_1) \cup c\text{-paths}(C_2) \)

In propositional logic a link ([MR93]) means that the formula has a contradictory part. Furthermore in Description Logics it is possible to construct a contradiction as \( \exists R.C \cap \forall R.\neg C \) by using role restrictions.
Definition 2. For a given concept $C$ a link is either a concept link or a role link. A concept link is a set of two complementary concept literals occurring in a c-path of $C$. A role link is a set $\{\exists R.D, \forall R.E_1, \ldots, \forall R.E_n\}$ of role literals occurring in a c-path of $C$, where all c-paths in $D \sqcap E_1 \sqcap \ldots \sqcap E_n$ contain $\bot$ or a concept link or a role link and no subset of $\{\exists R.D, \forall R.E_1, \ldots, \forall R.E_n\}$ is a role link. A c-path is called inconsistent, if it contains a link. Otherwise it is called consistent.

We regard $\bot$ and $\top$ as a complementary pair of concept literals. Note that a set of consistent c-paths uniquely determines a class of semantically equivalent concept descriptions.

Definition 3. We call a concept description $C$ linkless, if all c-paths in $C$ are consistent and for each occurrence of $QR.E$ in $C$ with $Q \in \{\exists, \forall\}$ the concept $E$ is linkless as well.

3 Transformation

Now we introduce a method to transform a concept into an equivalent linkless concept by eliminating all inconsistent c-paths. This method is closely related to path dissolution ([MR93]) used for propositional logic.

Definition 4. Let $G$ be a concept description and $A$ be a set of literals where each element of $A$ occurs in $G$. The c-path extension (c-path complement) of $A$ in $G$, denoted by $CPE(A, G)$ ($CPC(A, G)$), is a concept $G'$ containing exactly those c-paths in $G$ which (do not) contain $A$.

One possibility to get $CPE(A, G)$ and $CPC(A, G)$ is to construct the disjunction of all respective c-paths in $G$.

Definition 5. Given a concept description $G = G_1 \sqcap G_2$ which contains the link $A$. Further $A$ is neither a link for $G_1$ nor $G_2$. The positive part \(^1 \text{L}\) of the link occurs in $G_1$ and the negative part $\text{L}$ occurs in $G_2$. The dissolvent of $G$ and $A$ denoted by $\text{Diss}(A, G)$, is $\text{Diss}(A, G) = (CPE(L, G_1) \sqcap CPC(L, G_2)) \sqcup (CPC(L, G_1) \sqcap CPC(\text{L}, G_2)) \sqcup (CPC(L, G_1) \sqcap CPE(\text{L}, G_2))$

$\text{Diss}(A, G)$ removes exactly those c-paths from $G$ which contain the link $A$. Since these c-paths are inconsistent, $\text{Diss}(A, G)$ is equivalent to $G$.

\(^1\) The positive (negative) part of a concept link denotes its positive (negative) concept literal. Further the positive (negative) part of a role link denotes the existentially (universally) quantified elements of the role link.
Lemma 1. Let $A$ and $G$ be defined as in Definition 5. Then the following holds:
\[
\text{Diss}(A, G) \equiv (G_1 \cap CPC(L, G_2)) \cup (CPC(L, G_1) \cap CPE(L, G_2))
\]
\[
\text{Diss}(A, G) \equiv (CPC(L, G_1) \cap CPC(L, G_2)) \cup (CPC(L, G_1) \cap G_2)
\]

Example 1. We consider the concept $G = (\exists R.(\neg E \sqcup \neg B) \sqcup D) \sqcap (A \sqcup \forall R.E) \sqcap (C \sqcup \forall R.B)$ which contains the role link $\{\exists R.(\neg E \sqcup \neg B), \forall R.B, \forall R.E\}$. According to this role link we divide $G$ into $G_1 = \exists R.(\neg E \sqcup \neg B)$ and $G_2 = (A \sqcup \forall R.E) \sqcap (C \sqcup \forall R.B)$. In the next step we construct:
\[
\begin{align*}
CPE(\{\exists R.(\neg E \sqcup \neg B)\}, G_1) &= \exists R.(\neg E \sqcup \neg B) \\
CPC(\{\exists R.(\neg E \sqcup \neg B)\}, G_1) &= D \\
CPE(\{\forall R.E, \forall R.B\}, G_2) &= \forall R.E \sqcap \forall R.B \\
CPC(\{\forall R.E, \forall R.B\}, G_2) &= (A \sqcap C) \sqcup (A \sqcap \forall R.B) \sqcup (\forall R.E \sqcap C)
\end{align*}
\]

We use Lemma 1 to remove the role link. This leads to the linkless concept description $G'$, which is equivalent to $G$.
\[
G' = ((\exists R.(\neg E \sqcup \neg B) \sqcup D) \sqcap ((A \sqcap C) \sqcup (A \sqcap \forall R.B) \sqcup (\forall R.E \sqcap C)) \sqcup (D \sqcap \forall R.E \sqcap \forall R.B))
\]

Next we give an algorithm to remove all links in the way it is described above. In the following definition $G[G_1/G_2]$ denotes the concept one obtains by substituting all occurrences of $G_1$ in $G$ by $G_2$.

**Definition 6.** Let $G$ be a concept description.

- $\text{make\_linkless}(G) = G$, if $G$ is linkless.
- $\text{make\_linkless}(G) = \text{make\_linkless}(G[H/Diss(A,H)])$, where $H$ is a subconcept of $G$ and $A$ is a link in $H$, such that $\text{Diss}(A,H)$ is defined.
- $\text{make\_linkless}(G) = \text{make\_linkless}(G[B/make\_linkless(B)])$, where $B$ with $Q \in \{\exists, \forall\}$ is a subconcept of $G$, $B$ is not linkless and $G$ contains neither concept nor role links.

**Theorem 1.** Let $G$ be a concept description. Then $\text{make\_linkless}(G)$ is equivalent to $G$ and is linkless.

Note that in the worst case this transformation leads to an exponential blowup of the concept description.

**4 Properties of Linkless Concept Descriptions**

In this section we consider the properties of linkless concepts in order to understand why it is desirable to transform a concept into a linkless concept.
4.1 Satisfiability

Definition 7. For a linkless concept description \( C \) the predicate \( \text{Sat}(C) \) is defined as follows:

1. \( \text{Sat}(C) = \begin{cases} 
\text{true}, & \text{if } C \text{ is a concept literal, of the form } \forall R.D \text{ or } \top; \\
\text{false}, & \text{if } C \text{ is } \bot. 
\end{cases} \)
2. \( \text{Sat}(\exists R. D) = \text{Sat}(D) \).
3. \( \text{Sat}(C = \bigwedge_i \alpha_i) = \text{true}, \) iff \( \text{Sat}(\alpha_i) = \text{true} \) for all \( i \).
4. \( \text{Sat}(C = \bigvee_i \alpha_i) = \text{true}, \) iff \( \text{Sat}(\alpha_i) = \text{true} \) for at least one \( i \).

The predicate \( \text{Sat} \) can be directly transformed into an algorithm which checks the satisfiability of a linkless concept description in linear time. If we further assume that the following simplifications are applied to exhaustion after each step during the transformation of a concept \( C \) into a linkless concept \( C' \), \( \text{Sat}(C') \) can be calculated in constant time: \( \top \sqcap C = C, \top \sqcup C = \top, \bot \sqcap C = \bot, \bot \sqcup C = C, \exists R. \bot = \bot, \forall R. \top = \top. \) This is obvious since after these simplification a linkless concept \( C' \) description can only be inconsistent, if \( C' = \bot \).

4.2 Subsumption Queries

For subsumption checks an operator called conditioning ([Dar01]) is very helpful. In Definition 8, \( \overline{C} \) denotes the complement of a concept \( C \) and can be calculated simply by transforming \( \neg C \) in NNF.

Definition 8. Let \( C \) be a linkless concept description and \( \alpha = C_1 \sqcap \ldots \sqcap C_n \) where \( C_i \) is either a concept literal or has the form \( \exists R.C'_i \) or \( \forall R.C'_i \) with \( C'_i \) a concept literal. Then \( C \) conditioned by \( \alpha \) denoted by \( C|\alpha \) is the concept description one gets by replacing each occurrence of \( C_i \) in \( C \) by \( \top \) and each occurrence of \( \overline{C'_i} \) by \( \bot \) and simplifying the result as mentioned in the previous section.

The conditioning operation is linear in the size of the concept description \( C \). Further \( C|\alpha \cap \alpha \) and equivalent to \( C \cap \alpha \) and is obviously linkless.

Proposition 1. Let \( C \) be a linkless concept description and \( D \) be a disjunction of concept literals and role restrictions \( QR.B \) with \( Q \in \{\exists, \forall\} \) where \( B \) is a concept literal. Then \( C \sqsubseteq D \) can be tested in linear time.

Because of the structure proposition 1 claims for the concept \( D \), it is sufficient to calculate \( \text{Sat}(C|\neg D \cap \neg D) \). The conditioning as well as the \( \text{Sat} \) operator are linear, therefore the subsumption \( C \sqsubseteq D \) can be checked in linear time as well.
5 Precompilation of TBoxes

Our approach can be extended to handle unfoldable TBoxes. First we unfold the TBox as described in [BN03]. Next we transform the right side of each concept description with the \texttt{make\_linkless} operator into an equivalent linkless concept description. Then subsumption queries \( C \sqsubseteq D \) can be tested by unfolding \( C \sqcap \neg D \) according to the precompiled TBox. In general the result of this unfolding is not linkless. However if it is linkless, we can use the \texttt{Sat} predicate for a linear satisfiability test. Further, if after the unfolding of \( C \sqcap \neg D \), \( C \) is linkless and \( \neg D \) has the form claimed for \( \alpha \) in Definition 8, the subsumption check \( C \sqsubseteq D \) can be performed in linear time by using the conditioning operator. Precompiling every single concept description of the TBox separately makes it easy to extend the precompiled TBox as new concept descriptions can be added without precompiling the \textit{whole} TBox again.

6 Future Work / Conclusion

The extension of our approach on TBoxes which are not unfoldable as well as the extension of our normal form to more expressive Description Logics is one of our future interests. Further we want to investigate how to project linkless concept descriptions on sets of literals, since projection is a very helpful technique for combining different TBoxes.

References


