Linkless Normal Form for $\mathcal{ALC}$ Concepts and TBoxes

Claudia Schon

University of Koblenz-Landau, Germany

Abstract. In this paper we introduce a normal form for $\mathcal{ALC}$ concepts and TBoxes called *linkless normal form*. We investigate properties of concepts given in this normal form such as an efficient satisfiability test and the calculation of uniform interpolants. We further show a way to approximate a TBox by a concept in linkless normal form, which allows us to check certain subsumptions efficiently. This makes the linkless normal form interesting from the viewpoint of knowledge compilation. Furthermore, we show how to use the approximation of a TBox in linkless normal form to efficiently construct an approximation of a uniform interpolant of a TBox w.r.t. a given signature.

1 Introduction

Knowledge compilation is a technique originally developed for dealing with the computational intractability of propositional reasoning. It has been used in various AI systems for compiling knowledge bases offline into representations, which can be queried more efficiently. An overview of techniques for propositional knowledge bases is given in [7].

Several techniques for Description Logics, such as structural subsumption, normalization and absorption, are related to knowledge compilation. To perform a subsumption check on two concepts, structural subsumption algorithms [2] transform both concepts into a normal form and compare the structure of these normal forms. In contrast to structural subsumption, our approach is able to handle general negation. Absorption [18] and normalization [4] have the aim of increasing the performance of tableau based reasoning procedures. Unlike those approaches, we extend the use of preprocessing, allowing an efficient consistency test without requiring a tableau procedure.

With regards to Description Logics, knowledge compilation has first been investigated in [16], where $\mathcal{FL}$ concepts are approximated by $\mathcal{FL}^-$concepts. Recently, [5] introduced a normal form called *prime implicate normal form* for $\mathcal{ALC}$ concepts which allows a polynomial subsumption check. So far, however, the prime implicate normal form has not been extended for TBoxes. Another approach to precompile both $\mathcal{ALC}$ concepts and TBoxes is presented in [9] and [8]. There, the result of the precompilation is represented as a graph structure. Using this graph, certain subsumptions can be checked in polynomial time. However, a disadvantage of the precompilation of concepts into these graphs is that the graph provides no possibility to see the result of the precompilation as a concept. In this paper we remedy this situation by presenting concepts as the result of the precompilation process. This clarifies the whole precompilation process and emphasizes certain properties of precompiled concepts. For example, it will be simple to develop an operator to calculate uniform interpolants of precompiled concepts w.r.t. a given signature.

In this paper we will consider the Description Logic $\mathcal{ALC}$ [2] and adopt the notion of linkless formulas, as introduced in [14, 13]. First, we present the basics of the Description Logics $\mathcal{ALC}$ and $\mathcal{ALE}$. Then we define some normal forms used to introduce the idea of our precompilation. Afterwards we will discuss properties of precompiled concepts and introduce a method to efficiently check certain subsumptions using precompiled concepts.

2 Preliminaries

At first we introduce syntax and semantics of the Description Logics $\mathcal{ALE}$ and $\mathcal{ALC}$ [3]. Complex $\mathcal{ALE}$ concepts C and D are formed from atomic concepts and atomic roles according to the following syntax rule:

$$C, D \rightarrow A \mid \top \mid \bot \mid \neg A \mid C \sqcap D \mid \exists R.C \mid \forall R.C$$

where A is an atomic concept and R is an atomic role. $\mathcal{ALC}$ has the additional rules $C, D \rightarrow \neg C \mid C \sqcup D$. Next we consider the semantics of $\mathcal{ALC}$ concepts. An interpretation $\mathcal{I}$ is a pair $\langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$. $\Delta^{\mathcal{I}}$ is a nonempty set (the domain of the interpretation) and $\cdot^{\mathcal{I}}$ is an interpretation function assigning to each atomic concept A a set $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ and to each atomic role R a binary relation $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. We extend the interpretation function to complex concepts by the following inductive definitions:

$$\begin{aligned}
&\top^{\mathcal{I}} = \Delta^{\mathcal{I}} \\
&\bot^{\mathcal{I}} = \emptyset \\
&(\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}} \\
&(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}} \\
&(C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}} \\
&(\exists R.C)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} \mid \exists b\ (a, b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\} \\
&(\forall R.C)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} \mid \forall b\ (a, b) \in R^{\mathcal{I}} \rightarrow b \in C^{\mathcal{I}}\}
\end{aligned}$$

A concept C is satisfiable if there is an interpretation $\mathcal{I}$ with $C^{\mathcal{I}} \neq \emptyset$. We call such an interpretation a model for C. A terminological axiom has the form $C \sqsubseteq D$ or $C \equiv D$ where C, D are concepts and an axiom $C \sqsubseteq D$ ($C \equiv D$) is satisfied by an interpretation $\mathcal{I}$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ ($C^{\mathcal{I}} = D^{\mathcal{I}}$). A TBox consists of a finite set of terminological axioms and is called satisfiable if there is an interpretation satisfying all its axioms. Given an axiom $C \sqsubseteq D$ and a TBox $\mathcal{T}$ we often want to know if $C \sqsubseteq D$ w.r.t. $\mathcal{T}$, which we denote by $C \sqsubseteq_{\mathcal{T}} D$. $C \sqsubseteq_{\mathcal{T}} D$ holds iff $C \sqsubseteq D$

is true in all models of $\mathcal{T}$. Another way to show that $C \sqsubseteq_{\mathcal{T}} D$ holds is to show that $(C \sqcap \neg D)^{\mathcal{I}} = \emptyset$ for every model $\mathcal{I}$ of $\mathcal{T}$.

In the following, unless stated otherwise, by the term *concept*, we denote $\mathcal{ALC}$ concepts given in negation normal form (NNF), i.e., negation occurs only in front of atomic concepts. By the term *role restriction* we denote a concept of the form $QR.C$ with $Q \in \{\exists, \forall\}$ and by *concept literal*, we denote an atomic concept or a negated atomic concept. Further by *literal* we denote a concept literal or a role restriction and $\overline{C}$ means the complement of a concept literal C. By concepts occurring on the topmost level of a concept C, we understand each literal occurring in C, that is not in the scope of a role restriction. Further we say that two concepts C_1 and C_2 occur on the same level in concept C if they occur in the scope of the same role restrictions in C.

A concept C is in disjunctive normal form (DNF) iff C has the form $C = (\bigsqcup_{i=1}^{n}(\bigsqcap_{j=1}^{m} L_{i,j}))$ where $L_{i,j}$ is a literal and $L_{i,j} \neq L_{i,k}$ for all $i, j, k,\ j \neq k$. Note that this definition of DNF only affects the topmost level of a concept. Each concept can be transformed into DNF using the distributive law.

In the sequel we will analyze conjunctive paths through a concept.

Definition 1. *For a concept C, the set of its paths is defined as follows:*

$$\begin{aligned}
&\mathit{paths}(\bot) = \emptyset\\
&\mathit{paths}(\top) = \{\emptyset\}\\
&\mathit{paths}(C) = \{\{C\}\},\ \textit{if } C \textit{ is a literal}\\
&\mathit{paths}(C_1 \sqcup C_2) = \mathit{paths}(C_1) \cup \mathit{paths}(C_2)\\
&\mathit{paths}(C_1 \sqcap C_2) = \{X \cup Y | X \in \mathit{paths}(C_1) \textit{ and } Y \in \mathit{paths}(C_2)\}
\end{aligned}$$

For example the concept: $C = (\exists R.(D \sqcup E) \sqcup \neg A) \sqcap \forall R.D \sqcap \forall R.E \sqcap B$ has two different paths $p_1 = \{\exists R.(D \sqcup E), \forall R.D, \forall R.E, B\}$ and $p_2 = \{\neg A, \forall R.D, \forall R.E, B\}$.

Definition 2. *Let C be a concept. We call distinct concepts D and E conjunctively combined in C if C has a path containing both D and E or if C contains $QR.F$, $Q \in \{\exists, \forall\}$ and D and E are conjunctively combined in F.*

By $|C|$ we denote the number of subconcepts occurring in C. For a TBox $\mathcal{T}$, $|\mathcal{T}|$ is the sum of all $|C|$ where C is the left or right hand side of an axiom in $\mathcal{T}$. By $\mathit{depth}(C)$ we denote the maximal depth of nested role restrictions occurring in C, e.g. $\mathit{depth}(\exists R.\forall S.(A \sqcup \exists S.B)) = 3$. The size of a concept C, denoted by $\mathit{size}(C)$, is the number of atomic concepts, role restrictions, negations and connectives used in C. For example the size of $A \sqcup \exists R.(B \sqcap \neg A)$ is 7. Note that the size of a concept C is in the same order of magnitude as the number of subconcepts of C.

3 Normal Forms

In the precompilation introduced in this paper, we will first precompile the topmost level of a given concept and in the next step, we will recursively perform the precompilation on subconcepts occurring in the scope of a role restriction. In the following definition, A, B, B_1, B_2, C_1 and C_2 denote concepts.

Definition 3. *A concept C is in $\forall$-normal form ($\forall$-NF) if its topmost level does not contain conjunctively combined concepts of the form $\forall R.B_1$ and $\forall R.B_2$ for the same role R. Further C is in $\exists$-normal form ($\exists$-NF) if C is in $\forall$-NF and each $\exists R.B$ occurring on the topmost level of C is conjunctively combined with at most one role restriction of the form $\forall R.A$. If C is in $\exists$-NF and for all concepts of the form $\exists R.C_1$ and $\forall R.C_2$ occurring conjunctively combined on the topmost level of C, C_1 is equivalent to $C_1 \sqcap C_2$, we say that C is in* propagated *$\exists$-NF. Furthermore C is in* completely propagated *$\exists$-NF if C is in propagated $\exists$-NF and for all $QR.B$ occurring in C, $Q \in \{\exists, \forall\}$, B is in complete propagated $\exists$-NF as well.*

Note that Def. 3 restricts occurrences of $\exists R.A$ in C. This means that for example the concept $D = (\exists R.B \sqcap \forall R.F) \sqcup (\exists R.B \sqcap \forall R.\neg E)$ is in $\exists$-NF, because the claimed condition holds for each occurrence of $\exists R.B$ in D.

For example the concept

$$C = \exists R.(B \sqcup E) \sqcap \forall R.\neg B \sqcap (E \sqcup D \sqcup \forall R.F)$$

is not in $\forall$-NF, since the two universal role restrictions $\forall R.\neg B$ and $\forall R.F$ are conjunctively combined. C can be transformed into $\forall$-NF. The resulting concept is:

$$\exists R.(B \sqcup E) \sqcap \big((\forall R.\neg B \sqcap (E \sqcup D)) \sqcup \forall R.(\neg B \sqcap F)\big).$$

Transformation to $\exists$-NF leads to:

$$\big(\exists R.(B \sqcup E) \sqcap \forall R.\neg B \sqcap (E \sqcup D)\big) \sqcup \big(\exists R.(B \sqcup E) \sqcap \forall R.(\neg B \sqcap F)\big).$$

A completely propagated $\exists$-NF of concept C is:

$$\big(\exists R.((B \sqcup E) \sqcap \neg B) \sqcap \forall R.\neg B \sqcap (E \sqcup D)\big) \sqcup \big(\exists R.((B \sqcup E) \sqcap \neg B \sqcap F) \sqcap \forall R.(\neg B \sqcap F)\big) \quad (1)$$

One way to transform a concept into $\forall$-NF is using the idea of path dissolution [14]. Usually path dissolution is used to remove unsatisfiable paths from a propositional logic formula. In the following we give the intuition how to use path dissolution for the transformation of concepts into $\forall$-NF. For better understanding of the idea of path dissolution, please note that for propositional logic formulas a path is satisfiable if it does not contain complementary literals. Furthermore two propositional logic formulas are equivalent if they have the same set of satisfiable paths. Given a propositional logic formula F and a set of literals $L = \{L_1, \ldots, L_n\}$, $n \geq 1$, the term *conjunctive path extension* of L in F ($CPE(L, F)$) denotes a formula in NNF whose paths are exactly those paths of F containing an element of L. Further the term *conjunctive path complement* of L in F ($CC(L, F)$) denotes a formula in NNF whose paths are exactly those paths of F containing no element of L. Neither $CPE(L, F)$ nor $CC(L, F)$ have to be in DNF. [14] gives an algorithm to compute both $CPE(L, F)$ and $CC(L, F)$ for a given formula F and a set of literals L.

Path dissolution can be also used to remove conjunctively combined universal role restrictions $\forall R.D$ and $\forall R.E$ from a concept C. For this, we use a bijection between concepts and propositional logic formulas. This bijection, called *prop*, maps each atomic concept A to a propositional logic variable a, further $\sqcap$ ($\sqcup$) to $\wedge$ ($\vee$), $\top$ ($\bot$) to *true* (*false*) and $QR.C$ to a propositional logic variable Q_r_c with $Q \in \{\exists, \forall\}$. In order to remove one occurrence of conjunctively combined universal role restrictions $\forall R.D$ and $\forall R.E$ from a concept C, we first determine the smallest subconcept $G \sqcap H$ of C modulo commutativity, which contains the conjunctive combination of $\forall R.D$ and $\forall R.E$. Modulo commutativity of $\sqcap$ means, that there is a subconcept $D_1 \sqcap \ldots \sqcap D_n$ of C with $n \geq 2$ and there are distinct D_i, D_j in $\{D_1, \ldots, D_n\}$ with $G = D_i$ and $H = D_j$. W.l.o.g. we assume that $\forall R.D$ occurs in G and $\forall R.E$ occurs in H. Next we use the bijection and construct the propositional logic formula $prop(G \sqcap H) = G' \wedge H'$. We construct $CPE(\{\forall\_r\_d\}, G')$, $CC(\{\forall\_r\_d\}, G')$, $CPE(\{\forall\_r\_e\}, H')$ and $CC(\{\forall\_r\_e\}, H')$. It is obvious, that

$$\begin{aligned} G' \wedge H' \equiv & (CC(\{\forall\_r\_d\}, G') \wedge CC(\{\forall\_r\_e\}, H')) \\ & \vee (CC(\{\forall\_r\_d\}, G') \wedge CPE(\{\forall\_r\_e\}, H')) \\ & \vee (CPE(\{\forall\_r\_d\}, G') \wedge CC(\{\forall\_r\_e\}, H')) \\ & \vee (CPE(\{\forall\_r\_d\}, G') \wedge CPE(\{\forall\_r\_e\}, H')) \end{aligned} \quad (2)$$

Note that only the last disjunct of formula (2) contains conjunctively combined occurrences of $\forall\_r\_d$ and $\forall\_r\_e$ and further every path in the last disjunct contains both $\forall\_r\_d$ and $\forall\_r\_e$. Next we use the bijection to map the right side of formula (2) back to a concept N. Since every path in $(CPE(\forall\_r\_d, H) \wedge CPE(\forall\_r\_e, G))$ contains both $\forall\_r\_d$ and $\forall\_r\_e$, we can combine them to $\forall R.(D \sqcap E)$ in the concept N. Next we substitute the result of this for $G \sqcap H$ in C. After this step, the number of conjunctively combined concepts of the form $\forall R.C_1$ and $\forall R.C_2$ in C decreased by one.

In this way, all conjunctively combined concepts of the form $\forall R.C_1$ and $\forall R.C_2$ in C can be removed step by step, leading to a concept in $\forall$-NF. Note that the result of this transformation does not necessarily have to be in DNF. Only in the worst case, the result is in DNF which means an exponential blowup occurred. In [14] many optimizations are introduced which help to keep the result of dissolution as succinct as possible.

In a similar way, we can use dissolution to transform a concept into $\exists$-NF.

The $\forall$-normal form and propagated $\exists$-normal form introduced here are closely related to the normalization rules used in [1] to compute the least common subsumer of $\mathcal{ALE}$ concept descriptions. Another related approach is the normal form used for the calculation of uniform interpolants in [19]. The complete propagated $\exists$-NF is closely related to the notion of standard formula of degree d in multi-modal logic introduced in [12]. However only in the worst case, the blowup produced by transforming a concept into complete propagated $\exists$-NF corresponds to the blowup produced to transform a formula into a standard formula of degree d or into the normal form used in [19]. The reason for that is the fact, that concepts given in complete propagated $\exists$-NF are allowed to have a NNF structure and are not supposed to be transformed to DNF. In contrast to that, the normal form used in [19] is based on the DNF.

Now we are able to introduce the linkless normal form.

4 Linkless Concepts

The core of our precompilation technique is the removal of so called links [14]. Intuitively a link is a contradictory part of a concept, which can be removed from the concept preserving equivalence.

Definition 4. *For a given concept C a* link *is a set of two complementary concept literals occurring in a path of C. A concept C is called* top-level linkless *if there is no path in C containing a link.*

The idea of links was first introduced for propositional logic formulas. If a formula contains a link, this means that the formula has a contradictory path. Further if all paths of a formula contain a link, the formula is unsatisfiable. The special structure of linkless formulas in propositional logic allows us to decide satisfiability in constant time and it is possible to enumerate models very efficiently. We can remove links from a formula with the help of *path dissolution* [14] by eliminating paths containing a link. The result of removing all links from a propositional logic formula F is called *full dissolvent* of F. Further path dissolution simplifies away all occurrences of *true* and *false* in a formula. In the worst case, the removal of links can cause an exponential blowup. Path dissolution can be used for Description Logics as well. We use the bijection between concepts and propositional logic formulas introduced in Section 3.

Definition 5. *Let C be a concept mapped to $prop(C)$. Then $fulldissolvent(C)$ is the concept obtained by mapping the full dissolvent of $prop(C)$ back to a concept using $prop^{-1}$.*

From the fact, that path dissolution preserves equivalence in the propositional case [14] it follows, that $fulldissolvent(C) \equiv C$. Note that if $prop(C)$ is unsatisfiable, $fulldissolvent(C) = \bot$. In general, a path p is inconsistent if the conjunction of its elements is inconsistent. For a concept given in propagated $\exists$-NF, a path p is inconsistent iff p contains a link or p contains $\exists R.A$ with an inconsistent concept A. Our aim is now to develop a normal form for concepts, which has the same nice properties as the linkless normal form known for propositional logic formulas. The idea of this normal form is to remove links from a concept not only from the topmost level of the concept but from *all levels* of the concept. For our precompilation we assume that the input concept is in propagated $\exists$-NF and in the first step of our precompilation, we remove all links from the concept. The concept resulting from this step can still be inconsistent. Take $\exists R.(\neg B \sqcap B) \sqcap \forall R.B$ as an example. Therefore, in the second step of the precompilation we precompile all subconcepts occurring in the scope of an existential role restriction. Further we precompile all subconcepts occurring in the scope of an universal role restriction. This last step is necessary when we want to check subsumptions. Checking subsumptions can introduce new existential role restrictions we need to be able to combine with universal role restrictions occurring in the precompiled concept very efficiently. Therefore it is advantageous to have precompiled versions of subconcepts occurring in the scope of universal role restrictions.

The result of the precompilation is given in the next definition.

Definition 6. *A concept C is in* linkless normal form *(linkless NF) if it is in propagated ∃-NF, top-level linkless and for all $QR.B$ occurring in C, B is in linkless NF and further C is simplified according the following simplifications:*
$\top \sqcap D \to D \qquad \top \sqcup D \to \top \qquad \bot \sqcap D \to \bot \qquad \bot \sqcup D \to D \qquad \exists R.\bot \to \bot$

A concept is given in linkless NF is also called linkless. The following algorithm calculates the linkless NF for a given concept:

Algorithm 1 *Let C be a concept. The concept linkless(C) can be recursively calculated as follows:*

1. *Transform C into propagated ∃-NF.*
2. *Substitute C by fulldissolvent(C).*
3. *Simplify the result according to the simplifications given in Def. 6.*
4. *For all role restrictions $QR.B$ on the topmost level of C, replace B by linkless(B).*

Note that the precompilation, i.e. the application of algorithm 1 preserves equivalence. Therefore every concept can be transformed into an equivalentl inkless NF. Like in the propositional case, the removal of links can cause an exponential blowup. The linkless NF of the example concept, which is given in propagated ∃-NF in (1) is:

$$\big(\exists R.(E \sqcap \neg B) \sqcap \forall R.\neg B \sqcap (E \sqcup D)\big) \sqcup \big(\exists R.(E \sqcap \neg B \sqcap F) \sqcap \forall R.(\neg B \sqcap F)\big) \quad (3)$$

5 Properties of Linkless Concepts

From the structure of linkless concepts follows for linkless concepts C_1 and C_2, that the concepts $C_1 \sqcup C_2$, $\forall R.C_1$ and $\exists R.C_1$ are linkless as well. It is easy to see that linkless concepts are not closed under negation and conjunction.

The consistency of linkless concepts can be tested in constant time.

Theorem 1. *A linkless concept C can only be inconsistent if $C = \bot$.*

The proof of Theorem 1 can be found in [15] and uses the fact, that the simplifications given in Def. 6 are performed during the precompilation.

5.1 Tractable Subsumption Checking

In general, a subsumption $C \sqsubseteq E$ holds iff $C \sqcap \neg E$ is unsatisfiable. To simplify notation we consider subsumptions $C \sqsubseteq \neg D$, which hold iff $C \sqcap D$ is unsatisfiable. Given a linkless concept C a subsumption $C \sqsubseteq \neg D$ can be answered in time linear to $\mathit{size}(C) \cdot \mathit{size}(D)$ if D has a certain structure. The next definition specifies, for which concepts D we can check subsumptions efficiently.

Definition 7. *A consistent $\mathcal{ALE}$ concept D is called a q-concept if D is in complete propagated $\exists$-NF and for all $QR.B$ occurring in D, B is consistent.*

Since it is reasonable to expect concept D in a subsumption check $C \sqsubseteq \neg D$ to be rather small, a possible exponential blowup produced by the transformation of D into propagated $\exists$-NF is not too harmful.

Given a linkless concept C, in order to check a subsumption $C \sqsubseteq \neg D$, we have to check the satisfiability of $C \sqcap D$. However linkless concepts are not closed under conjunction. Therefore, we have to define an operator, which allows us to conjunctively combine the linkless concept C with the q-concept D resulting in a linkless concept. The operator used here is an enhancement of the conditioning operator introduced in [6] in propositional logic. Intuitively, conditioning a linkless concept C by a q-concept D means, that we assume D to be true and simplify C according to this assumption. In the following we understand a q-concept D to be the set of its conjuncts.

Definition 8. *Let C be a linkless concept and D be a q-concept. Then C conditioned with D, denoted by $C|D$, is defined as:*

1. *If C is a concept literal:*
$$C|D = \begin{cases} \top, \text{ if } C \in D \\ \bot, \text{ if } \overline{C} \in D \\ C, \text{ otherwise} \end{cases}$$
2. *If C has the form $C_1 \sqcap C_2$:*
$$C|D = \begin{cases} \bot, & \text{if } C_1|D = \bot \text{ or } C_2|D = \bot \\ C_i|D, & \text{if } C_j|D = \top,\ (i,j \in \{1,2\}, i \neq j) \\ C_1|D \sqcap C_2|D, & \text{otherwise} \end{cases}$$
3. *If C has the form $C_1 \sqcup C_2$:*
$$C|D = \begin{cases} \top, & \text{if } C_1|D = \top \text{ or } C_2|D = \top \\ C_i|D, & \text{if } C_j|D = \bot,\ (i,j \in \{1,2\}, i \neq j) \\ C_1|D \sqcup C_2|D, & \text{otherwise} \end{cases}$$
4. *If C has the form $\forall R.E$:*
$$C|D = \begin{cases} \bot, & \text{if there is } \exists R.B' \in D \text{ with } E|B' = \bot. \\ \forall R.(E|B), & \text{if there is } \forall R.B \in D \text{ and there is no } \exists R.B' \in D \\ & \quad \text{with } E|B' = \bot. \\ \forall R.E, & \text{if there is no } \forall R.B \in D \text{ and there is no } \exists R.B' \in D \\ & \quad \text{with } E|B' = \bot. \end{cases}$$
5. *If C has the form $\exists R.E$:*
$$C|D = \begin{cases} \bot, & \text{if there is } \forall R.B \in D \text{ and } E|B = \bot. \\ \exists R.(E|B), & \text{if there is } \forall R.B \in D \text{ and } E|B \neq \bot. \\ \exists R.E, & \text{otherwise} \end{cases}$$

Conditioning a concept C by a q-concept D has complexity $O(\mathit{size}(C) \cdot \mathit{size}(D))$. Note that $C|D$ is linkless.

Lemma 1. *Let C be a linkless concept and D a q-concept. Then $(C|D) \sqcap D \equiv C \sqcap D$. Further $C|D$ is satisfiable iff $C \sqcap D$ is satisfiable.*

Corollary 1. *Let C be a linkless concept and D be a q-concept. Then it can be decided in time linear to $\mathit{size}(C) \cdot \mathit{size}(D)$ if $C \sqsubseteq \neg D$ holds.*

The proof of Lemma 1 can be found in [15]. Corollary 1 is a direct consequence of Lemma 1, the fact that $C \sqsubseteq \neg D$ iff $C \sqcap D$ is unsatisfiable and the complexity of conditioning in the propositional case [6]. In (3) our example concept C is given in linkless NF. We now want to check if the subsumption $C \sqsubseteq E \sqcup \exists R.F$ holds. Negating the right side of the subsumption, leads to the q-concept $\neg E \sqcap \forall R.\neg F$. With the help of Def. 8, we can calculate: $C|\neg E \sqcap \forall R.\neg F = \exists R.(E \sqcap \neg B) \sqcap \forall R.\neg B \sqcap D$. Since this concept is satisfiable, the subsumption $C \sqsubseteq E \sqcup \exists R.F$ does not hold.

5.2 Uniform Interpolation

Another interesting transformation for precompiled theories mentioned in [7] is uniform interpolation. With regard to ontologies, uniform interpolation has many applications [11], e.g. re-use of ontologies, predicate hiding and ontology versioning. Intuitively, a uniform interpolant of a concept C w.r.t. a set of atomic concepts Φ is a concept D that does not contain any atomic concepts from Φ and is indistinguishable from C regarding the superconcepts and subsumers that do not use symbols from Φ. So the idea of uniform interpolation is to forget all symbols given in Φ without changing the meaning of C.

Definition 9. *Let C be a concept and Φ a set of atomic concepts. Then the concept D is called uniform interpolant of C w.r.t. Φ, or Φ-interpolant of C for short, iff the following conditions hold:*

- *D contains only atomic concepts which occur in C but not in Φ.*
- *$\models C \sqsubseteq D$.*
- *For all concepts E not containing symbols from Φ it holds: $\models C \sqsubseteq E$ iff $\models D \sqsubseteq E$.*

We will now present an operator to compute a uniform interpolant of a linkless concept w.r.t. a set of concept symbols.

Definition 10. *Let C be a linkless concept and Φ be a set of atomic concepts. Then $UI(C, \Phi)$ is the concept obtained by substituting each occurrence of A and $\neg A$ in C by $\top$ iff $A \in \Phi$.*

The next theorem states that uniform interpolants of linkless concept w.r.t. a set of concept symbols can be calculated efficiently.

Theorem 2. *Let C be a linkless concept and Φ a set of atomic concepts. Then the following hold:*

1. *$UI(C,\Phi)$ is a Φ-interpolant of C,*
2. *$UI(C,\Phi)$ can be calculated in time linear in the size of C and*
3. *if $UI(C,\Phi)$ is simplified according to the simplifications given in Def. 6, then $UI(C,\Phi)$ is linkless.*

The second and the third assertion follow directly from the way $UI(C,\Phi)$ is constructed. The proof of the first assertion can be found in [15].

Given for example the set $\Phi = \{E, D\}$, we can calculate the Φ-interpolant of the linkless concept C from the example given in (3): $UI(C,\Phi) = (\exists R.(\top \sqcap \neg B) \sqcap \forall R.\neg B \sqcap (\top \sqcup \top)) \sqcup (\exists R.(\top \sqcap \neg B \sqcap F) \sqcap \forall R.(\neg B \sqcap F))$ which can be simplified according to the simplifications given in Def. 6 to the concept $(\exists R.\neg B \sqcap \forall R.\neg B) \sqcup (\exists R.(\neg B \sqcap F) \sqcap \forall R.(\neg B \sqcap F))$.

6 Linkless TBoxes

In order to extend the linkless NF for TBoxes, we first have to consider TBox approximations. We transform a TBox $\mathcal{T} = \{A_1 \sqsubseteq B_1, \ldots, A_n \sqsubseteq B_n\}$ into a metaconstraint [10] by first transforming each assertion $A_i \sqsubseteq B_i$ into an equivalent assertion $\top \sqsubseteq \neg A_i \sqcup B_i$ and then conjoining all assertions. This leads to the one single assertion $\top \sqsubseteq C_\mathcal{T}$ with $C_\mathcal{T} = \bigsqcap_{A_i \sqsubseteq B_i \in \mathcal{T}} (\neg A_i \sqcup B_i)$, stating that every element in the domain has to belong to the concept $C_\mathcal{T}$. Now subsumptions w.r.t. the TBox $\mathcal{T}$ can be checked using $C_\mathcal{T}$. If we want to know if a concept D is satisfiable w.r.t. $\mathcal{T}$, we can decide this by checking the satisfiability of $D \sqcap C_\mathcal{T} \sqcap \forall U.C_\mathcal{T}$. Where U is the transitive closure of the union of all roles occurring in $\mathcal{T}$. Since $C_\mathcal{T} \sqcap \forall U.C_\mathcal{T}$ is not an $\mathcal{ALC}$ concept, we need an approximation of it [19]. This approximation can be transformed into linkless NF in order to get an approximation of a linkless version of the TBox.

Definition 11. *For a TBox $\mathcal{T}$ and $n \geq 0$ the n-th approximation of $\mathcal{T}$ is defined as:*

$$C_\mathcal{T}^{(n)} = \bigsqcap_{k=0}^{n} \bigsqcap_{R_1,\ldots,R_k \in \mathcal{R}} \forall R_1 \ldots \forall R_k . C_\mathcal{T}$$

with $\mathcal{R}$ the set of all roles occurring in $\mathcal{T}$.

Given for example the TBox $\mathcal{T} = \{A \sqsubseteq (B \sqcup \exists R.D), B \sqsubseteq \forall R'.D\}$ we get
$C_\mathcal{T}^{(0)} = C_\mathcal{T} = (\neg A \sqcup B \sqcup \exists R.D) \sqcap (\neg B \sqcup \forall R'.D)$
$C_\mathcal{T}^{(1)} = C_\mathcal{T}^{(0)} \sqcap \forall R.C_\mathcal{T} \sqcap \forall R'.C_\mathcal{T}$
$C_\mathcal{T}^{(2)} = C_\mathcal{T}^{(1)} \sqcap \forall R.\forall R.C_\mathcal{T} \sqcap \forall R.\forall R'.C_\mathcal{T} \sqcap \forall R'.\forall R.C_\mathcal{T} \sqcap \forall R'.\forall R'.C_\mathcal{T}$

The next theorem follows directly from Lemma 9 in [17] and states how the approximation $C_\mathcal{T}^{(n)}$ of a TBox $\mathcal{T}$ can be used to check subsumptions w.r.t. $\mathcal{T}$.

Theorem 3. *Let $\mathcal{T}$ be a TBox and D a concept. If $n \geq 2^{|D|+|\mathcal{T}|}$, then D is satisfiable w.r.t $\mathcal{T}$ iff $D \sqcap C_\mathcal{T}^{(n)}$ is satisfiable.*

Since subsumption checks can be transformed into a satisfiability test, we can use $C_\mathcal{T}^{(n)}$ to check subsumptions as well. Now we extend our normal form to TBoxes. The simple idea is to make $C_{\mathcal{T}}^{(n)}$ linkless for a certain number n in order to be able to check subsumptions w.r.t $\mathcal{T}$. We choose $n \geq 2^{|D|+|\mathcal{T}|}$. The higher we choose n the higher the size of the q-concept can be. It is reasonable to assume that q-concepts are small compared to the size of the TBox. Therefore this is rather nonrestrictive.

In the previous section we considered uniform interpolation. For TBoxes, uniform interpolation is even more interesting. In many applications, only a small subset of the signature of a given TBox is used. This leads to the idea of uniform interpolation where all atomic concepts that are not of interest are removed preserving the meaning of the original TBox within the atomic concepts of interest. However in general, uniform interpolation for $\mathcal{ALC}$ TBoxes need not exist [19].

Definition 12. *Let $\mathcal{T}$ be a TBox and Φ a set of atomic concepts. Then the TBox $\mathcal{T}'$ is called a uniform interpolant of $\mathcal{T}$ w.r.t. Φ or short Φ-interpolant of $\mathcal{T}$ iff the following conditions hold:*

- *$\mathcal{T}'$ contains only atomic concepts occurring in $\mathcal{T}$ but not in Φ.*
- *$\mathcal{T} \models \mathcal{T}'$.*
- *For all concept inclusions $C \sqsubseteq D$ not containing symbols from Φ: $\mathcal{T} \models C \sqsubseteq D$ implies $\mathcal{T}' \models C \sqsubseteq D$.*

From Theorem 3 and Proposition 5.1 in [19] follows:

Corollary 2. *Let $\mathcal{T}$ be a TBox, Φ a set of atomic concepts, D a q-concept containing only atomic concepts occurring in $\mathcal{T}$ but not in Φ and $n \geq 2^{|D|+|\mathcal{T}|}$. Then D is satisfiable w.r.t. $\mathcal{T}$ iff D is satisfiable w.r.t $UI(linkless(C_{\mathcal{T}}^{(n)}), \Phi)$.*

We call $UI(linkless(C_{\mathcal{T}}^{(n)}), \Phi)$ nth approximation of a Φ-interpolant of $\mathcal{T}$. If we know the maximal size of concepts we want to test the satisfiability w.r.t. the Φ-interpolant of $\mathcal{T}$, we can choose n as suggested in [19] accordingly and use the nth approximation of the Φ-interpolant instead of the Φ-interpolant itself.

Note that, given a linkless version of $C_{\mathcal{T}}^{(n)}$, the UI operator introduced in the previous section can be used to calculate the approximation of the Φ-interpolant in time linear to the size of the linkless $C_{\mathcal{T}}^{(n)}$.

7 Conclusion / Future Work

This paper presents a precompilation of $\mathcal{ALC}$ concepts and TBoxes into a normal form called linkless normal form, which allows for an efficient satisfiability test, subsumption test and uniform interpolation. In future work, we would like to extend our normal form to handle more expressive description logics. We expect that especially transitive roles will prove challenging. Another interesting point for future work is to try out other target languages known from the field of knowledge compilation for propositional logic. This could be done using the propagated ∃-NF as a basis and then transforming the result into a target language known from propositional logic. We expect a comparison of the results for different target languages to be very interesting.

References

1. F. Baader, R. Küsters, and R. Molitor. Computing least common subsumers in description logics with existential restrictions. In Thomas Dean, editor, *IJCAI*, pages 96–103. Morgan Kaufmann, 1999.
2. F. Baader and W. Nutt. Basic description logics. In Baader, F. and Calvanese, D. and McGuinness, D. and Nardi, D. and Patel-Schneider, P. [3], pages 43–95.
3. Baader, F. and Calvanese, D. and McGuinness, D. and Nardi, D. and Patel-Schneider, P., editor. *The Description Logic Handbook*. Cambridge University Press, 2003.
4. P. Balsiger and A. Heuerding. Comparison of theorem provers for modal logics - introduction and summary. In H. de Swart, editor, *TABLEAUX*, volume 1397 of *Lecture Notes in Computer Science*, pages 25–26. Springer, 1998.
5. M. Bienvenu. Prime implicates and prime implicants in modal logic. In *Proceedings of the Twenty-Second Conference on Artificial Intelligence (AAAI-07)*, pages 397–384, 2007.
6. A. Darwiche. Decomposable negation normal form. *Journal of the ACM*, 48(4), 2001.
7. A. Darwiche and P. Marquis. A knowledge compilation map. *Journal of Artificial Intelligence Research*, 17:229–264, 2002.
8. U. Furbach, H. Günther, and C. Obermaier. A knowledge compilation technique for ALC TBoxes. In C. Lane and H. Guesgen, editors, *FLAIRS Conference*. AAAI Press, 2009.
9. U. Furbach and C. Obermaier. Precompiling ALC Tboxes and query answering. In *Proceedings of the 4th Workshop on Contexts and Ontologies*, 2008.
10. I. Horrocks. Implementation and optimization techniques. In Baader, F. and Calvanese, D. and McGuinness, D. and Nardi, D. and Patel-Schneider, P. [3], pages 306–346.
11. B. Konev, D. Walther, and F. Wolter. Forgetting and uniform interpolation in large-scale description logic terminologies. In C. Boutilier, editor, *IJCAI*, pages 830–835, 2009.
12. M. Kracht. *Tools and techniques in Modal Logic*. Elsevier Science, 1999.
13. N. Murray and E. Rosenthal. Tableaux, path dissolution, and decomposable negation normal form for knowledge compilation. In M. Mayer and F. Pirri, editors, *Proceedings of TABLEAUX 2003*, volume 1397 of *LNCS*. Springer, 2003.
14. Murray, N. and Rosenthal, E. Dissolution: Making paths vanish. *Journal of the ACM*, 40(3):504–535, 1993.
15. C. Schon. Linkless normal form for ALC concepts. Reports of the Faculty of Informatics 12/2010, Universität Koblenz-Landau, 2010. Available at http://www.uni-koblenz.de/FB4/Publications/Reports.
16. Selman, B. and Kautz, H. Knowledge compilation and theory approximation. *J. ACM*, 43(2):193–224, 1996.
17. B. ten Cate, W. Conradie, M. Marx, and Y. Venema. Definitorially complete description logics. In P. Doherty, J. Mylopoulos, and C. Welty, editors, *Proc. of KR 2006*, pages 79–89. AAAI Press, 2006.
18. D. Tsarkov and I. Horrocks. Fact++ description logic reasoner: System description. In U. Furbach and N. Shankar, editors, *IJCAR*, volume 4130 of *LNCS*, pages 292–297. Springer, 2006.
19. K. Wang, Z. Wang, R. Topor, J. Pan, and G. Antoniou. Concept and role forgetting in ALC ontologies. In *Proc. of ISWC2009*, 2009.