UNIVERSITÄT
KOBLENZ · LANDAU
Institut für Informatik

FB 4
Informatik

Linkless Normal Form for ALC Concepts

Claudia Schon

Nr. 12/2010

Universität Koblenz-Landau • Campus Koblenz • Universitätsstraße 1 • 56070 Koblenz
Telefon +49 261 287-0 • http://www.uni-koblenz.de

Arbeitsberichte des Fachbereichs Informatik

ISSN (Print): 1864-0346
ISSN (Online): 1864-0850

Kontaktdaten der Verfasser

Claudia Schon
Institut für Informatik
Fachbereich Informatik
Universität Koblenz-Landau
Universitätsstraße 1
D-56070 Koblenz
EMail: schon@uni-koblenz.de

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

Claudia Schon

University of Koblenz-Landau, Germany

Abstract. Knowledge compilation is a common technique for propositional logic knowledge bases. A given knowledge base is transformed into a normal form, for which queries can be answered efficiently. This precompilation step is expensive, but it only has to be performed once. We apply this technique to concepts defined in the Description Logic $\mathcal{ALC}$. We introduce a normal form called linkless normal form for $\mathcal{ALC}$ concepts and discuss an efficient satisfiability test for concepts given in this normal form. Furthermore, we will show how to efficiently calculate uniform interpolants of precompiled concepts w.r.t. a given signature.

1 Introduction

Knowledge compilation is a technique, which was 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 systems, that can be queried more efficiently after this precompilation. An overview about techniques for propositional knowledge bases is given in [7].

There are several techniques for Description Logics which are related to knowledge compilation techniques. An overview on precompilation techniques for Description Logics such as structural subsumption, normalization and absorption is given in [10]. 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. However these algorithms typically have problems with more expressive Description Logics. Especially general negation, which is an important feature in the application of Description Logics, is a problem for those algorithms. In contrast to structural subsumption algorithms, our approach is able to handle general negation without problems. Absorption ([15]) and normalization ([3]) have the aim of increasing the performance of tableau based reasoning procedures. Unlike those approaches, we extend the use of preprocessing. We suggest to transform the concept into a normal form called linkless normal form allowing an efficient consistency test, not requiring a tableau procedure.

With regards to Description Logics, knowledge compilation has firstly been investigated in [14], where $\mathcal{FL}$ concepts are approximated by $\mathcal{FL}^-$concepts. Recently, [4] introduced a normal form called prime implicate normal form for $\mathcal{ALC}$ concepts which allows for a polynomial subsumption check. So far, however, prime implicate normal form has not been extended for Tboxes yet. 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, the so called linkless graph. Using this linkless graph, certain subsumption queries can be answered in polynomial time. However a disadvantage of the precompilation of concepts into linkless graphs is, that the linkless graph provides no possibility to see the result of the precompilation as a concept. In this paper we remedy this situation by presenting linkless concepts as the result of the precompilation process. This makes the whole precompilation process more comprehensible and makes certain properties of precompiled concepts more obvious.

In this paper we will consider the Description Logic $\mathcal{ALC}$ [2] and adopt the notion of linkless formulas, as it was introduced in [13, 12]. Firstly, we are presenting the basics of the Description Logics $\mathcal{ALC}$ and $\mathcal{ALE}$. Then we are defining 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 answer certain subsumption queries using precompiled concepts. This precompilation is closely related to the precompilation presented in [9] and [8], where the result of the precompilation is a graph structure, the so called linkless graph. Further this linkless graph is used to answer queries. However, there is no possibility to see the result of the precompilation as a concept. In contrast to that, the result of the precompilation presented in this paper is a concept. This makes the whole precompilation process much more comprehensible. Furthermore, presenting the result of the precompilation as a concept facilitates to see certain properties of precompiled concepts. This makes it easier to develop an operator to calculate uniform interpolation of precompiled concepts w.r.t. a given signature.

2 Preliminaries

At first we introduce syntax and semantics of the Description Logics $\mathcal{ALE}$ and $\mathcal{ALC}$ ([2]). 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 $\langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$, where $\Delta^{\mathcal{I}}$ is a nonempty set which is 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:

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

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. Further 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 $C^{\mathcal{I}} \subset D^{\mathcal{I}}$ ($C^{\mathcal{I}} = D^{\mathcal{I}}$). 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 $A \sqsubseteq B$ and a TBox $\mathcal{T}$ we often want to know if $A \sqsubseteq B$ holds w.r.t. $\mathcal{T}$, which we denote by $A \sqsubseteq_{\mathcal{T}} B$ and call it a query to the TBox $\mathcal{T}$. Further $A \sqsubseteq_{\mathcal{T}} B$ holds, if $A \sqsubseteq B$ is true in all models of $\mathcal{T}$. Another way to show that $A \sqsubseteq_{\mathcal{T}} B$ holds is to show that $\mathcal{T}$ together with $A \sqcap \neg B$ is unsatisfiable.

In the following, unless stated otherwise, by the term concept, we denote $\mathcal{ALC}$ concepts given in NNF, i.e., negation occurs only in front of concept names. By *concept literal*, we denote a concept name or a negated concept name. Further by *literal* we denote a concept literal or a role restriction. By concepts occurring on the topmost level of a concept C in NNF, we understand each literal occurring in C, which is not in the scope of a role restriction. A concept C in NNF is in disjunctive normal form (DNF), iff $C = (\bigsqcup_{i=1}^{n}(\bigsqcap_{j=1}^{m} L_{i,j}))$ where $L_{i,j}$ is a literal. Note that this definition of DNF only affects the topmost level of a concept. For example the concept $(E \sqcap \neg B) \sqcup \exists R.(A \sqcap (D \sqcup E))$ is in DNF, even though the concept in the scope of the existential role restriction does not have a special structure. Each concept can be transformed into DNF by transforming it into NNF and then using distributive as well as DeMorgan's law.

In the sequel we will analyse conjunctive paths through a concept. Therefore we give a short definition of a path:

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

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

For example the concept: $C = (\exists R.(D \sqcup E) \sqcup \neg A) \sqcap \forall R.D \sqcap \forall R.E \sqcap B$ has the 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 in NNF. We call concepts B and D conjunctively combined in C, if C has a path which contains both B and D or if C contains $QR.A$, $Q \in \{\exists, \forall\}$ and B and D are conjunctively combined in A.*

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. To ensure that this precompilation preserves equivalence, the concept first has to be transformed into the so called *propagated ∃-normal form*, which bases on the *∀-normal form*. The ∀-normal form and propagated ∃-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 [**?**]. However transforming a concept into the normal form used in [**?**] in general produces a larger blowup than the precompilation into linkless normal form.

3.1 ∀-normal form

The ∀-normal form (∀-NF) restricts occurrences of universal role restrictions on the topmost level of a concept. It exploits the fact that $\forall R.A \sqcap \forall R.B$ is equivalent to $\forall R.(A \sqcap B)$ and demands all conjunctively combined occurrences of universal role restrictions w.r.t. the same role to be summarized.

Definition 3. *A concept is in ∀-normal form (∀-NF), if it is in NNF and the topmost level of the concept does not contain conjunctively combined $\forall R.B_1$ and $\forall R.B_2$.*

For example the concept

$$C = \exists R.(B \sqcup E) \sqcap \forall R.\neg B \sqcap (E \sqcup D \sqcup \forall R.F) \tag{1}$$

is not in ∀-NF, since the two universal role restrictions $\forall R.\neg B$ and $\forall R.F$ are conjunctively combined.

Theorem 1. *For every concept there is an equivalent concept which is in ∀-NF.*

Proof. A concept C can be transformed into an equivalent concept C' in ∀-NF by first transforming it into NNF and then using the following algorithm:

1. If C doesn't contain any role restrictions, then $C' = C$.
2. If C contains conjunctively combined universal role restrictions $\forall R.B_1$ and $\forall R.B_2$ at the topmost level:
   - transform C into DNF and
   - use commutativity together with the rule $\forall R.B_1 \sqcap \forall R.B_2 \equiv \forall R.(B_1 \sqcap B_2)$.

□

Concept C from (1) can be transformed into ∀-NF by first transforming C into DNF and then combining all conjunctively combined universal role restrictions referring to the same role. The resulting concept is:

$$(\exists R.(B \sqcup E) \sqcap \forall R.\neg B \sqcap E) \sqcup (\exists R.(B \sqcup E) \sqcap \forall R.\neg B \sqcap D) \sqcup (\exists R.(B \sqcup E) \sqcap \forall R.(\neg B \sqcap F)) \tag{2}$$

However going the whole way to DNF results in a concept, which is larger than necessary. It is possible to create a more succinct version of the ∀-NF of a concept by expanding the concept only as far as necessary. Where necessary means, that C is gradually expanded only until the ∀-NF is reached. For the concept considered above, it is possible to calculate a more succinct ∀-NF which is:

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

3.2 ∃-normal form

Next we introduce the ∃−normal form which imposes some restrictions on the occurrences of existentially quantified role restrictions on the topmost level of a concept.

Definition 4. *A concept C is in ∃-normal form (∃-NF), if C is in ∀-NF and further each $\exists R.B$ occurring on the topmost level of C is conjunctively combined with at most one role restriction of the form $\forall R.A$.*

Note that Definition 4 restricts *occurrences* of $\exists R.A$ in C. This means that for example the concept $D = (\exists R.B \sqcap \forall R.(E \sqcup F)) \sqcup (\exists R.B \sqcap \forall R.\neg E)$ is in ∃-NF, because the claimed condition holds for each occurrence of $\exists R.B$ in D.

Theorem 2. *For every concept there is an equivalent concept which is in ∃-NF.*

The ∃-NF of a concept can be calculated by an algorithm which is similar to the one given in the proof of Theorem 1. As in the case of the ∀-NF this can lead to an ∃-NF, which is larger than necessary. However by partially expanding the concept only as far as necessary it is possible to produce a more succinct ∃-NF. The ∃-NF for concept C given in (3) is:

$$\big(\exists R.(B \sqcup E) \sqcap \forall R.\neg B \sqcap (E \sqcup D)\big) \sqcup \big(\exists R.(B \sqcup E) \sqcap \forall R.(\neg B \sqcap F)\big) \tag{4}$$

With the help of the following lemma, we are able to combine existential and universal role restrictions occurring in a concept.

Lemma 1. *For $\mathcal{ALC}$ concepts A, B holds:* $\exists R.A \sqcap \forall R.B \equiv \exists R.(A \sqcap B) \sqcap \forall R.B$

Definition 5. *Let C be a concept in ∃-NF. The result of applying Lemma 1 to all existential role restrictions occurring on the topmost level of C, which are conjunctively combined with a universal role restriction is called the* propagated *∃-NF of C. Further C is in* complete propagated *∃-NF, if C is in propagated ∃-NF and for all $QR.B$ occurring in C, $Q \in \{\exists, \forall\}$, B is in complete propagated ∃-NF as well.*

For every concept in ∃-NF there is an equivalent concept in propagated ∃-NF. The result of this is called the *propagated* ∃-NF for C. The propagated ∃-NF of concept given in (4) is:

$$\big(\exists R.((B \sqcup E) \sqcap \neg B) \sqcap \forall R.\neg B \sqcap (E \sqcup D)\big) \sqcup \big(\exists R.((B \sqcup E) \sqcap \neg B \sqcap F) \sqcap \forall R.(\neg B \sqcap F)\big) \quad (5)$$

Now we are able to give details on the way, a concept can be precompiled.

4 Linkless Concepts

The core of our precompilation technique is the removal of so called links ([13]). Intuitively a link is a contradictory part of a concept, which can be removed from the concept preserving equivalence.

Definition 6. *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 C is in NNF and there is no path in C which contains 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 part. Further if all paths of a formula contain a link, the formula is unsatisfiable. The special structure of linkless formulas i.e. formulas without links in propositional logic allows us to decide satisfiability in constant time and it is possible to enumerate models very efficiently. One possibility to remove links from a formula is to use *path dissolution* [13]. The idea of this algorithm is to eliminate 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.

Path dissolution can be used for Description Logics as well. We use a bijection between concepts and propositional logic formulas. This bijection, called *prop*, maps each concept name 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 the worst case, the removal of links can cause an exponential blowup.

Definition 7. *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}.*

Note that if *prop*(C) is unsatisfiable, *fulldissolvent*$(C) = \bot$.

Theorem 3. *Let C be a concept. Then fulldissolvent$(C) \equiv C$.*

Theorem 3 follows from the fact [Murray & Rosenthal 93], that path dissolution preserves equivalence in the propositional case.

In general, a path p is inconsistent, if the conjunction of its elements is inconsistent. For a concept given in propagated ∃-NF, a path p is inconsistent, iff p contains a link or p contains $\exists R.A$ for which the concept A is inconsistent.

The aim is now, to develop a normal form for concepts, which has the same nice properties as the linkless normal form 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 claim the input concept to be 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 answer queries. Asking queries can introduce new existential role restrictions, which we need to be able to combine with universal role restrictions occurring in the precompiled concept very efficiently during querytime. Therefore it is advantageous to have precompiled versions of concepts occurring in the scope of universal role restrictions.

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

Definition 8. *A concept C is in* linkless normal form *(linkless NF), if it is in propagated $\exists$-NF, top-level linkless and for all $QR.B$ occurring in C, B is in linkless NF and further C is simplified according Fig. 1.*

$$\top \sqcap D = D \qquad \top \sqcup D = \top \qquad \bot \sqcap D = \bot \qquad \bot \sqcup D = D \qquad \exists R.\bot = \bot$$

Fig. 1. Simplifications

A concept, which is given in linkless NF is also called linkless.

The following algorithm calculates the linkless NF for a given concept:

Algorithm 4 *Let C be a concept in NNF. The concept linkless(C) can be calculated as follows:*

1. *Transform C into complete propagated $\exists$-NF.*
2. *Substitute C by fulldissolvent(C).*
3. *For all role restrictions $QR.B$ on the topmost level of C, replace B by linkless(B) and simplify the result according to Fig. 1.*

Note that the precompilation, i.e. the application of algorithm 4 preserves equivalence. Performing the simplifications of Fig. 1 during the precompilation ensures, that a linkless concept can only be inconsistent, if it is $\bot$. Dissolution only removes inconsistent paths from the concept. It does not introduce new paths. Hence all levels of a linkless concept are in propagated $\exists$-NF. The linkless NF of the example concept C, which is given in propagated $\exists$-NF at the end of Section 3.2 is:

$$\big(\exists R.(E \sqcap \neg B) \sqcap \forall R.\neg B \sqcap (E \sqcup D)\big) \sqcup \big(\exists R.(E \sqcap \neg B \sqcap F) \sqcap \forall R.(\neg B \sqcap F)\big) \quad (6)$$

5 Properties of Linkless Concepts

We now analyse different properties of linkless concepts in order to clarify the benefits of the linkless NF for concepts. We first take a look at closure properties, then consider consistency, next take a look at query answering and in the end concentrate on uniform interpolation.

5.1 Closure Properties

For query answering it is interesting to analyse closure properties of linkless concepts. From the structure of linkless concepts follows the next theorem:

Theorem 5. *Let C_1 and C_2 be linkless concepts. Then $C_1 \sqcup C_2$, $\forall R.C_1$ and $\exists R.C_1$ are linkless as well.*

It is easy to see that linkless concepts are not closed under negation, since the negation of a linkless concept generally is not in NNF. Further linkless concepts are not closed under conjunction. Take the linkless concepts A and $\neg A$ as an example: $A \sqcap \neg A$ is not linkless.

5.2 Consistency

Due to the fact, that the simplifications of Fig. 1 are preformed during the precompilation, the following theorem holds.

Theorem 6. *Let C be a linkless concept. Then C can only be inconsistent, if it is $\bot$.*

Therefore the consistency of linkless concepts can be tested in constant time.

Proof. Let C be a linkless concept. We prove Theorem 6 by induction on the nesting depth of roles in C.

Induction basis: C does not contain any roles. Since C is linkless, C is simplified according to Fig. 1. Therefore the assertion holds.

Induction hypothesis: For linkless concepts C with a maximal nesting depth n of roles holds: C is unsatisfiable, iff $C = \bot$. This means that for $n > 0$ C is satisfiable.

Induction step: Let C be linkless and $n + 1$ its nesting depth of roles. We have to show that C must be satisfiable. We assume C to be unsatisfiable. Since C is in NNF this means that, all paths in C must be inconsistent. In general there are three reasons for a path p to be inconsistent:

1. p contains a link:
   However since C is linkless, this is not possible.

2. p contains $\exists R.A$ and $\forall R.B$ with $A \sqcap B$ unsatisfiable:
   Since C is linkless and therefore in propagated $\exists$-NF, it follows that A has to be unsatisfiable. Further A has to be linkless. Moreover the maximal nesting depth of roles in A is n. From the induction hypothesis follows, that $A = \bot$. However this is not possible, because a linkless concept C does not contain occurrences of $\exists R.\bot$, since C is simplified according Fig. 1.
3. p contains $\bot$:
   Since C is simplified according to Fig. 1, this means that path p only contains $\bot$ and nothing else. However this is a contradiction to the assumption that C has a nesting depth of roles greater than 0.

Since none of the reasons for a path to be inconsistent is possible, it follows that C has to have satisfiable paths and hence is satisfiable. □

5.3 Tractable Query Answering

Given a linkless concept C subsumption queries $\models C \sqsubseteq E$ can be answered in linear time, if E has a certain structure. In general, a subsumption $C \sqsubseteq E$ holds, iff $C \sqcap \neg E$ is unsatisfiable. In order to save clerical work, we consider subsumption queries $C \sqsubseteq \neg D$ which hold iff $C \sqcap D$ is unsatisfiable. The next definition specifies, for which concepts D this property holds.

Definition 9. *A consistent $\mathcal{ALE}$ concept D is called a query concept, if D is in complete propagated $\exists$-NF and for all $QR.B$ occurring in D, B is consistent.*

This definition shows, that query concepts have to be transformed. Even though the precompilation of the query has to be done during query-time, this is not too harmful, because it is reasonable to expect the query to be rather small. In the following we understand an $\mathcal{ALE}$ concept to be the set of its conjuncts.

Given a linkless concept C, in order to find out whether a subsumption query $C \sqsubseteq \neg D$ holds, we have to check the satisfiability of $C \sqcap D$. However linkless concepts are not closed under conjunction. So the concept $C \sqcap D$ doesn't need to be linkless. Therefore we have to define an operator, which allows us to conjunctively combine the linkless concept C with the query concept D resulting in a linkless concept. The operator used here is an enhancement of the conditioning operator introduced in [5] for propositional logic formulas. Intuitively, conditioning a linkless concept C by a query concept D means, that we assume D to be true and simplify C according to this assumption.

Definition 10. *Let C be a linkless concept and D be an query 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 $\forall R.E$:*
$$C|D = \begin{cases} \bot, & \text{if } \exists R.B' \in D \text{ with } E|B' = \bot. \\ \forall R.(E|B), & \text{if } \forall R.B \in D \text{ and there is no } \exists R.B' \in D \text{ with } E|B' = \bot. \\ \forall R.E, & \text{if } \forall R.B \notin D \text{ and there is no } \exists R.B' \in D \text{ with } E|B' = \bot. \end{cases}$$
5. *If C has the form $\exists R.E$:*
$$C|D = \begin{cases} \bot, & \text{if } \forall R.B \in D \text{ and } E|B = \bot. \\ \exists R.(E|B), & \text{if } \forall R.B \in D \text{ and } E|B \neq \bot. \\ \exists R.E, & \text{otherwise} \end{cases}$$

Conditioning a concept C by a query concept D can be done in time linear to the size of C.

Lemma 2. *Let C be a linkless concept and D a query concept. Then $C|D \sqcap D = C \sqcap D$.*

Proof. We prove this by induction on the structure of concept C:

Induction basis: C is a concept literal.

a.) $C \in D$:
Then $C|D \sqcap D \overset{\text{Def. 10}}{=} \top \sqcap D = D$. Further $C \sqcap D = D$, because $C \in D$.
b.) $\overline{C} \in D$:
Then $C|D \sqcap D \overset{\text{Def. 10}}{=} \bot \sqcap D = \bot$. Further $C \sqcap D = \bot$, because $\overline{C} \in D$.
c.) $\overline{C} \notin D$ and $C \notin D$:
Then $C|D \sqcap D \overset{\text{Def. 10}}{=} C \sqcap D$.

Induction hypothesis: For linkless concepts C_1, C_2, holds for all query concepts D: $C_i|D \sqcap D = C_i \sqcap D$ $(i \in \{1,2\})$. In the following $\overset{\text{IH}}{=}$ means, that this equality follows from the induction hypothesis.

Induction step: We now have to show, that the assertion holds for the linkless concepts $C_1 \sqcap C_2$, $C_1 \sqcup C_2$, $\forall R.C_1$ and $\exists R.C_1$.

1.) $C = C_1 \sqcap C_2$
  a.) $C_1|D = \bot$ or $C_2|D = \bot$. W.l.o.g. let $C_1|D = \bot$. Then $C_1|D \sqcap D = \bot \sqcap D = \bot$. Further $C_1|D \sqcap D \overset{\text{IH}}{=} C_1 \sqcap D = \bot$.

Then

$$\begin{aligned} C|D \sqcap D &\overset{\text{Def. 10}}{=} \bot \sqcap D \\ &= \bot \\ &= \bot \sqcap C_2 \\ &= (C_1 \sqcap D) \sqcap C_2 \\ &= C \sqcap D \end{aligned}$$

The case $C_2|D = \bot$ is in the same manner.

b.) $C_1|D = \top$ or $C_2|D = \top$. W.l.o.g. let $C_1|D = \top$. Then

$$\begin{aligned} C|D \sqcap D &\overset{\text{Def. 10}}{=} C_2|D \sqcap D \\ &\overset{\text{IH}}{=} C_2 \sqcap D \\ &= (\top \sqcap D) \sqcap (C_2 \sqcap D) \\ &= (C_1|D \sqcap D) \sqcap (C_2 \sqcap D) \\ &\overset{\text{IH}}{=} (C_1 \sqcap D) \sqcap (C_2 \sqcap D) \\ &= (C_1 \sqcap C_2) \sqcap D \\ &= C \sqcap D \end{aligned}$$

The case $C_2|D = \top$ is in the same manner.

c.) Otherwise:

$$\begin{aligned} C|D \sqcap D &\overset{\text{Def. 10}}{=} C_1|D \sqcap C_2|D \sqcap D \\ &= (C_1|D \sqcap D) \sqcap (C_2|D \sqcap D) \\ &\overset{\text{IH}}{=} (C_1 \sqcap D) \sqcap (C_2 \sqcap D) \\ &= (C_1 \sqcap C_2) \sqcap D \\ &= C \sqcap D \end{aligned}$$

2.) $C = C_1 \sqcup C_2$

a.) $C_1|D = \top$ or $C_2|D = \top$. W.l.o.g. let $C_1|D = \top$. Then $C|D \sqcap D \overset{\text{Def. 10}}{=} \top \sqcap D = D$. Further

$$\begin{aligned} C \sqcap D &= (C_1 \sqcup C_2) \sqcap D \\ &= (C_1 \sqcap D) \sqcup (C_2 \sqcap D) \\ &\overset{\text{IH}}{=} (C_1|D \sqcap D) \sqcup (C_2 \sqcap D) \\ &= (\top \sqcap D \sqcup (C_2 \sqcap D) \\ &= D \sqcup (C_2 \sqcap D) \\ &= D \end{aligned}$$

The case $C_2|D = \top$ is in the same manner.

b.) $C_1|D = \bot$ or $C_2|D = \bot$. W.l.o.g. let $C_1|D = \bot$. Then $C|D \sqcap D \overset{\text{Def. 10}}{=} C_2|D \sqcap D \overset{\text{IH}}{=} C_2 \sqcap D$. Further

$$\begin{aligned} C \sqcap D &= (C_1 \sqcup C_2) \sqcap D \\ &= (C_1 \sqcap D) \sqcup (C_2 \sqcap D) \\ &\overset{\text{IH}}{=} (C_1|D \sqcap D) \sqcup (C_2 \sqcap D) \\ &= (\bot \sqcap D) \sqcup (C_2 \sqcap D) \\ &= \bot \sqcup (C_2 \sqcap D) \\ &= C_2 \sqcap D \end{aligned}$$

The case $C_2|D = \bot$ is in the same manner.

c.) Otherwise:

$$\begin{aligned} C|D \sqcap D &\overset{\text{Def. 10}}{=} (C_1|D \sqcup C_2|D) \sqcap D \\ &= (C_1|D \sqcap D) \sqcup (C_2|D \sqcap D) \\ &\overset{\text{IH}}{=} (C_1 \sqcap D) \sqcup (C_2 \sqcap D) \\ &= C \sqcap D \end{aligned}$$

3.) $C = \forall R.C_1$

a.) Let D be $D_1 \sqcap \ldots \sqcap D_{i-1} \sqcap \exists R.B' \sqcap D_i \sqcap \ldots \sqcap D_n$ with $C_1|B' = \bot$. According to the induction hypothesis, $C_1|B' \sqcap B' = C_1 \sqcap B' = \bot$. Then $C|D \sqcap D \overset{\text{Def. 10}}{=} \bot \sqcap D = \bot$. Further

$$\begin{aligned} C \sqcap D &= \forall R.C_1 \sqcap D_1 \sqcap \ldots \sqcap D_{i-1} \sqcap \exists R.B' \sqcap D_{i+1} \sqcap \ldots \sqcap D_n \\ &= \exists R.(B' \sqcap C_1) \sqcap \forall R.C_1 \sqcap D_1 \sqcap \ldots \sqcap D_{i-1} \sqcap D_{i+1} \sqcap \ldots \sqcap D_n \\ &\overset{\text{IH}}{=} \exists R.\bot \sqcap \forall R.C_1 \sqcap D_1 \sqcap \ldots \sqcap D_{i-1} \sqcap D_{i+1} \sqcap \ldots \sqcap D_n \\ &= \bot \end{aligned}$$

b.) Let further D be $D_1 \sqcap \ldots \sqcap D_{i-1} \sqcap \forall R.B \sqcap D_{i+1} \sqcap \ldots \sqcap D_n$ and there is no $\exists R.B' \in D$, with $C_1|B' = \bot$. Then

$$\begin{aligned} C|D \sqcap D &\overset{\text{Def. 10}}{=} \forall R.C_1|B \sqcap D \\ &= \forall R.C_1|B \sqcap D_1 \sqcap \ldots \sqcap D_{i-1} \sqcap \forall R.B \sqcap D_{i+1} \sqcap \ldots \sqcap D_n \\ &= \forall R.(C_1|B \sqcap B) \sqcap D \\ &\overset{\text{IH}}{=} \forall R.(C_1 \sqcap B) \sqcap D \end{aligned}$$

Further

$$\begin{aligned} C \sqcap D &\overset{\text{Def. 10}}{=} \forall R.C_1 \sqcap D \\ &= \forall R.C_1 \sqcap D_1 \sqcap \ldots \sqcap D_{i-1} \sqcap \forall R.B \sqcap D_{i+1} \sqcap \ldots \sqcap D_n \\ &= \forall R.(C_1 \sqcap B) \sqcap D \end{aligned}$$

c.) There is no $\forall R.B$ in D and there is no $\exists R.B'$ in D with $C_1|B' = \bot$. Then $C|D \sqcap D \overset{\text{Def. 10}}{=} \forall R.C_1 \sqcap D = C \sqcap D$.

4.) $C = \exists R.C_1$:

a.) Let D be $D_1 \sqcap \ldots \sqcap D_{i-1} \sqcap \forall R.B \sqcap D_{i+1} i \sqcap \ldots \sqcap D_n$ and $C_1|B = \bot$. Then $C|D \sqcap D \overset{\text{Def. 10}}{=} \bot \sqcap D = \bot$ and further

$$\begin{aligned} C \sqcap D &= \exists R.C_1 \sqcap D_1 \sqcap \ldots \sqcap D_{i-1} \sqcap \forall R.B \sqcap D_{i+1} \sqcap \ldots \sqcap D_n \\ &= \exists R.(C_1 \sqcap B) \sqcap D \\ &\overset{\text{IH}}{=} \exists R.(C_1|B \sqcap B) \sqcap D \\ &= \exists R.(\bot \sqcap B) \sqcap D \\ &= \exists R.\bot \sqcap D \\ &= \bot \end{aligned}$$

b.) Let D be $D_1 \sqcap \ldots \sqcap D_{i-1} \sqcap \forall R.B \sqcap D_{i+1} i \sqcap \ldots \sqcap D_n$ and $C_1|B \neq \bot$. Then

$$\begin{aligned} C|D \sqcap D &\overset{\text{Def. 10}}{=} \exists R.C_1|B \sqcap D_1 \sqcap \ldots \sqcap D_{i-1} \sqcap \forall R.B \sqcap D_{i+1} i \sqcap \ldots \sqcap D_n \\ &= \exists R.(C_1|B \sqcap B) \sqcap D \\ &\overset{\text{IH}}{=} \exists R.(C_1 \sqcap B) \sqcap D \\ &= C \sqcap D \end{aligned}$$

c.) Otherwise: $C|D \sqcap D \overset{\text{Def. 10}}{=} \exists R.C_1 \sqcap D = C \sqcap D$.

□

Lemma 3. *Let C_1 and C_2 be linkless concepts with $C_1 \sqsubseteq C_2$ and D be a query concept. Then $C_1|D \sqsubseteq C_2|D$.*

Proof. We prove this by induction on the structure of concept C_2. In the following we assume $C_1 \neq \bot$. It's save to make this assumption, because for $C_1 = \bot$, $C_1|D = \bot|D = \bot \sqsubseteq C_2|D$ holds, irrespective of C_2.

Induction basis: C_2 is a concept literal. Then $C_1 \sqsubseteq C_2$ implies $C_1 = C_2$ and further $C_1|D = C_2|D$ which implies $C_1|D \sqsubseteq C_2|D$.

Induction hypothesis: The assertion holds for linkless concepts B_1 and B_2.

Induction step: We now have to show, that the assertion holds for linkless concepts $B_1 \sqcap B_2$, $B_1 \sqcup B_2$, $\forall R.B_1$ and $\exists R.B_1$.

1.) $C_2 = B_1 \sqcap B_2$
Since $C_1 \sqsubseteq C_2 = B_1 \sqcap B_2$, it also has to hold that $C_1 \sqsubseteq B_1$ and $C_1 \sqsubseteq B_2$. According to the induction hypothesis, this implies $C_1|D \sqsubseteq B_1|D$ and $C_1|D \sqsubseteq B_2|D$.

a.) $B_1|D = \bot$ or $B_2|D = \bot$. W.l.o.g let $B_1|D = \bot$. Then $C_2|D \overset{\text{Def. 10}}{=} \bot$. According to the induction hypothesis, $C_1|D \sqsubseteq B_1|D = \bot$ which implies $C_1|D = \bot$. This leads to $\bot = C_1|D \sqsubseteq C_2|D = \bot$. The case $B_2|D = \bot$ is in the same manner.

b.) $B_1|D = \top$ or $B_2|D = \top$. W.l.o.g let $B_1|D = \top$. Then $C_2|D \stackrel{\text{Def. 10}}{=} B_2|D$. This leads to $C_1|D \sqsubseteq B_2|D = C_2|D$. The case $B_2|D = \top$ is in the same manner.

c.) Otherwise:
Then $C_2|D = B_1|D \sqcap B_2|D$. According to the induction hypothesis, both $C_1|D \sqsubseteq B_1|D$ and $C_1|D \sqsubseteq B_2|D$ hold, which implies $C_1|D \sqsubseteq B_1|D \sqcap B_2|D = C_2|D$.

2.) $C_2 = B_1 \sqcup B_2$
Since $C_1 \sqsubseteq C_2 = B_1 \sqcup B_2$, it also has to hold that $C_1 \sqsubseteq B_1$ or $C_1 \sqsubseteq B_2$. W.l.o.g. we assume $C_1 \sqsubseteq B_1$. According to the induction hypothesis, this implies $C_1|D \sqsubseteq B_1|D$.

a.) $B_1|D = \top$ or $B_2|D = \top$. W.l.o.g let $B_1|D = \top$. Then $C_2|D \stackrel{\text{Def. 10}}{=} \top$. This leads to $C_1|D \sqsubseteq \top = C_2|D$. The case $B_2|D = \top$ is in the same manner.

b.) (i) Let $B_1|D = \bot$. Since $C_1|D \sqsubseteq B_1|D$, it follows that $C_1|D = \bot$. This leads to $C_1|D = \bot \sqsubseteq C_2|D$.

(ii) Let $B_2|D = \bot$. Then $C_2|D \stackrel{\text{Def. 10}}{=} B_1|D$, which leads to $C_1|D \sqsubseteq B_1|D = C_2|D$.

c.) Otherwise:
$C_2|D = B_1|D \sqcup B_2|D$. Further $C_1|D \sqsubseteq B_1|D \sqsubseteq B_1|D \sqcup B_2|D = C_2|D$.

The case $C_1 \sqsubseteq B_2$ is in the same manner.

3.) $C_2 = \forall R.B_1$.
Since $C_1 \sqsubseteq C_2 = \forall R.B_1$ and C_1 is linkless, every path in C_1 has to contain a concept of the form $\forall R.B_{C_1}$ with $B_{C_1} \sqsubseteq B_1$. According to the induction hypothesis, $B_{C_1}|D \sqsubseteq B_1|D$ holds for all query concepts D.

a.) Let $\exists R.B_D \in D$ with $B_1|B_D = \bot$. Since every path of C_1 contains a concept of the form $\forall R.B_{C_1}$ with $B_{C_1} \sqsubseteq B_1$, it follows that $B_{C_1}|B_D \sqsubseteq B_1|B_D = \bot$ and further $B_{C_1}|B_D = \bot$. This means, that every path of $C_1|D$ contains $\bot$ which leads to $C_1|D = \bot$. Further $C_2|D \stackrel{\text{Def. 10}}{=} \bot$ and $C_1|D = \bot \sqsubseteq \bot = C_2|D$.

b.) Let $\forall R.B_D \in D$ and there is no $\exists R.B'_D \in D$ with $B_1|B'_D = \bot$. As mentioned above, every path in C_1 has to contain a concept of the form $\forall R.B_{C_1}$ with $B_{C_1} \sqsubseteq B_1$. This means that concept C_1 can be every linkless concept which can be constructed by the following syntax rule

$$C_1 \to \forall R.B_{C_1} | C_1 \sqcap E | C_1 \sqcup C_1$$

Where E and B_{C_1} are arbitrary linkless concepts with $B_{C_1} \sqsubseteq B_1$. We have to show that the assertion holds by a separate induction on the structure of C_1:
Induction basis: $C_1 = \forall R.B_{C_1}$. Then $C_1|D = \forall R.B_{C_1}|D = \forall R.(B_{C_1}|B_D) \sqsubseteq \forall R.(B_1|B) = C_2|D$.
Induction hypothesis: The assertion holds for C_1' and C_1'' which can be constructed according to the above mentioned syntax rule.
Induction step: We have to show that the assertion also holds for linkless $C_1' \sqcap E$ and $C_1' \sqcup C_1''$.

i. $C_1 = C_1' \sqcap E$. According to the induction hypothesis $C_1'|D \sqsubseteq C_2|D$.

A. $C_1'|D = \bot$ or $E|D = \bot$. Then $C_1|D \overset{\text{Def. 10}}{=} \bot \sqsubseteq C_2|D$.

B. $E|D = \top$. Then $C_1|D \overset{\text{Def. 10}}{=} C_1'|D \sqsubseteq C_2|D$.

C. $C_1'|D = \top$. Since $\top = C_1'|D \sqsubseteq C_2|D$ holds according to the induction hypothesis, $C_2|D$ must be $\top$ and therefore $C_1|D \sqsubseteq C_2|D$.

D. $C_1'|D \neq \top$, $C_1'|D \neq \bot$, $E|D \neq \top$ and $E|D \neq \bot$: This leads to $C_1|D = C_1'|D \sqcap E|D \sqsubseteq C_2|D$.

ii. $C_1 = C_1' \sqcup C_1''$. According to the induction hypothesis $C_1'|D \sqsubseteq C_2|D$ and $C_1''|D \sqsubseteq C_2|D$.

A. $C_1'|D = \top$ or $C_1''|D = \top$ (W.l.o.g. $C_1'|D = \top$): Since $\top = C_1'|D \sqsubseteq C_2|D$ holds according to the induction hypothesis, $C_2|D$ must be $\top$ and this implies $C_1|D \sqsubseteq C_2|D$.

B. $C_1'|D = \bot$ or $C_1''|D = \bot$ (W.l.o.g. $C_1'|D = \bot$): Then $C_1|D = C_1''|D \sqsubseteq C_2|D$

C. $C_1'|D \neq \top$, $C_1'|D \neq \bot$, $C_1''|D \neq \top$ and $C_1''|D \neq \bot$: This leads to $C_1|D = C_1'|D \sqcup C_1''|D \sqsubseteq C_2|D$.

c.) Otherwise: Then $C_2|D = \forall R.B_1$. A similar induction as in case b.) proves this case.

4.) $C_2 = \exists R.B_1$:
Since $C_1 \sqsubseteq C_2 = \exists R.B_1$ and C_1 is linkless, every path in C_1 has to contain a concept of the form $\exists R.B_{C_1}$ with $B_{C_1} \sqsubseteq B_1$. According to the induction hypothesis, $B_{C_1}|D \sqsubseteq B_1|D$ holds for all query concepts D. In this case concept C_1 can be every linkless concept which constructed by the following syntax rule

$$C_1 \rightarrow \exists R.B_{C_1} | C_1 \sqcap E | C_1 \sqcup C_1$$

Where E and B_{C_1} are arbitrary linkless concepts with $B_{C_1} \sqsubseteq B_1$.

a.) $\forall R.B_D \in D$ with $B_1|B_D = \bot$. Since every path of C_1 contains a concept of the form $\exists R.B_{C_1}$ with $B_{C_1} \sqsubseteq B_1$, it follows from the induction hypothesis that $B_{C_1}|B_D \sqsubseteq B_1|B_D = \bot$ which implies $B_{C_1}|B_D = \bot$. This means, that every path of $C_1|D$ contains $\bot$ which leads to $C_1|D = \bot$. Further $C_2|D \overset{\text{Def. 10}}{=} \bot$ and $C_1|D = \bot \sqsubseteq \bot = C_2|D$.

b.) Let $\forall R.B_D \in D$ and $B_1|B_D \neq \bot$. As mentioned above, every path in C_1 has to contain a concept of the form $\exists R.B_{C_1}$ with $B_{C_1} \sqsubseteq B_1$. In this case, concept C_1 can be every linkless concept which can be constructed by the following syntax rule

$$C_1 \rightarrow \exists R.B_{C_1} | C_1 \sqcap E | C_1 \sqcup C_1$$

Where E and B_{C_1} are arbitrary linkless concepts with $B_{C_1} \sqsubseteq B_1$. We have to show that the assertion holds by a separate induction on the structure of C_1:
Induction basis: $C_1 = \exists R.B_{C_1}$. Then $C_1|D = (\exists R.B_{C_1})|D = \exists R.(B_{C_1}|B_D) \sqsubseteq \exists R.(B_1|B_D) = C_2|D$.

c.) Otherwise: Then $C_2|D = \exists R.B_1$. An induction similar to the one given in 4.) b.) proves this case.

□

Lemma 4. *Let C be a linkless concept and D be a query concept. Then $C|D$ is satisfiable, iff $C \sqcap D$ is satisfiable. Furthermore $C|D$ is linkless.*

Proof. The satisfiability of $C \sqcap D$ implies the satisfiability of $C|D$ by Lemma 2. We show the other direction and the linkless property by induction on the structure of concept C:

Induction basis: C is a concept literal.

1.) $C \in D$:
Then $C|D = \top$, which is satisfiable. Then $C \sqcap D$ is be satisfiable as well. Besides this $C|D = \top$ is linkless.
2.) $\overline{C} \in D$:
Then $C|D = \bot = C \sqcap D$. Furthermore $\bot$ is linkless.
3.) $\overline{C} \notin D$ and $C \notin D$:
Then $C|D = C$, which is satisfiable. Further $C \sqcap D$ is be satisfiable, since D is satisfiable by definition and C is a concept literal not occurring in D. Besides this C is linkless.

Induction hypothesis: For linkless concepts C_1, C_2 and all query concepts D holds: If $C_i|D$ is satisfiable, $C_i \sqcap D$ is satisfiable as well and $C_i|D$ is linkless ($i \in \{1, 2\}$).

Induction step: We have to show, that the assertion holds for linkless concepts $C_1 \sqcap C_2$, $C_1 \sqcup C_2$, $\forall R.C_1$ and $\exists R.C_1$.

1.) $C = C_1 \sqcap C_2$, C linkless:
  (a) $C_1|D = \bot$ or $C_2|D = \bot$. W.l.o.g. $C_1|D = \bot$:
  Then $C|D = \bot$ which is unsatisfiable and linkless. Furthermore, according to the induction hypothesis, $C_1 \sqcap D$ has to be unsatisfiable. This also implies the unsatisfiability of $C \sqcap D = C_1 \sqcap C_2 \sqcap D$.
  (b) $C_1|D = \top$ or $C_2|D = \top$. W.l.o.g. $C_1|D = \top$.
  According to the induction hypothesis $C_1 \sqcap D$ is satisfiable.
  Then $C|D \overset{\text{Def. 10}}{=} C_2|D$, which according to the induction hypothesis, is satisfiable if $C_2 \sqcap D$ is satisfiable.
  Further, $C \sqcap D = (C_1 \sqcap C_2) \sqcap D = C_1 \sqcap D \sqcap C_2$. Since $C = C_1 \sqcap C_2$ is linkless, $C_1 \sqcap C_2$ is satisfiable. Further $C_1 \sqcap D$ is satisfiable. Since $C_1 \sqcap C_2$ is linkless, it is not possible that a contradiction in $C_1 \sqcap C_2 \sqcap D$ is caused by all three conjuncts. This is ensured, because $C_1 \sqcap C_2$ is in propagated $\exists$-NF. Therefore $C_1 \sqcap D \sqcap C_2$ is satisfiable, iff $C_2 \sqcap D$ is satisfiable.
  Furthermore, according to the induction hypothesis, $C|D = C_2|D$ is linkless.

(c) $C_i|D \neq \bot$ and $C_i|D \neq \top$ for $i \in \{1, 2\}$:
Then $C|D = C_1|D \sqcap C_2|D$. If $C|D$ is satisfiable, both $C_i|D$ have to be satisfiable. According to the induction hypothesis, this implies the satisfiability off $C_i \sqcap D$ ($i \in \{1, 2\}$).
Further $C \sqcap D = (C_1 \sqcap C_2) \sqcap D = (C_1 \sqcap D) \sqcap (C_2 \sqcap D)$. Since C_1, C_2, D, $C_1 \sqcap C_2$, $C_1 \sqcap D$ and $C_2 \sqcap D$ are all satisfiable, $C_1 \sqcap C_2 \sqcap D$ is satisfiable as well. This is the case, because $C_1 \sqcap C_2$ is linkless and therefore it is not possible that a contradiction in $C_1 \sqcap C_2 \sqcap D$ is caused by all three conjuncts. This is ensured, because $C_1 \sqcap C_2$ is in propagated ∃-NF.

In addition to that $C|D = C_1|D \sqcap C_2|D$ is linkless. This is the case, since $C_1 \sqcap C_2$ is linkless and the conditioning with D does not introduce links. Further we have to show that $C_1|D \sqcap C_2|D$ is in propagated ∃-NF. Conditioning with D does not introduce any universally quantified roles and therefore $C_1|D \sqcap C_2|D$ is in ∀-NF and ∃-NF. We still have to show, that $C_1|D \sqcap C_2|D$ is in *propagated* ∃-NF. Let's assume that, w.l.o.g. $\exists R.A$ occurs in C_1 and $\forall R.B$ in C_2. Since $C_1 \sqcap C_2$ is linkless, $A \equiv A \sqcap B$ has to hold, which means $A \sqsubseteq B$. It follows from Lemma 2, that $A|D \sqsubseteq B|D$ and therefore $C_1|D \sqcap C_2|D = C|D$ is in propagated ∃-NF.

2.) $C = C_1 \sqcup C_2$

(a) $C_1|D = \top$ or $C_2|D = \top$. W.l.o.g. $C_1|D = \top$.
Then $C|D = (C_1 \sqcup C_2)|D \stackrel{\text{Def. 10}}{=} \top$ is satisfiable and linkless. According to the induction hypothesis $C_1 \sqcap D$ is satisfiable, which implies $C \sqcap D = (C_1 \sqcup C_2) \sqcap D = (C_1 \sqcap D) \sqcup (C_2 \sqcap D)$ to be satisfiable as well.

(b) $C_1|D = \bot$ or $C_2|D = \bot$. W.l.o.g. $C_1|D = \bot$.
Then $C|D \stackrel{\text{Def. 10}}{=} C_2|D$. If $C|D = C_2|D$ is satisfiable, $C_2 \sqcap D$ has to be satisfiable as well, according to the induction hypothesis. Further $C \sqcap D = (C_1 \sqcup C_2) \sqcap D = (C_1 \sqcap D) \sqcup (C_2 \sqcap D)$. Since $C_1 \sqcap D$ is unsatisfiable according to the induction hypothesis, $C \sqcap D$ is satisfiable, iff $C_2 \sqcap D$ is satisfiable.
In addition to that, $C|D = C_2|D$ is linkless according to the induction hypothesis.

(c) $C_i|D \neq \top$ and $C_i|D \neq \bot$ for $i \in \{1, 2\}$.
Then $C|D \stackrel{\text{Def. 10}}{=} C_1|D \sqcup C_2|D$, which is satisfiable, iff $C_1|D$ or $C_2|D$ is satisfiable. W.l.o.g. let $C_1|D$ be satisfiable. By induction hypothesis, this implies the satisfiability of $C_1 \sqcap D$, which leads to the satisfiability of $C \sqcap D = (C_1 \sqcup C_2) \sqcap D = (C_1 \sqcap D) \sqcup (C_2 \sqcap D)$.
Furthermore both $C1|D$ and $C_2|D$ are linkless according to the induction hypothesis. Hence $C|D = C_1|D \sqcup C_2|D$ is linkless as well.

3.) $C = \forall R.C_1$

(a) Let D be $D_1 \sqcap D_2 \sqcap \ldots \sqcap D_{i-1} \sqcap \exists R.B' \sqcap D_{i+1} \sqcap \ldots \sqcap D_n$ with $C_1|B' = \bot$. Then $C|D \overset{\text{Def. 10}}{=} \bot$ which is unsatisfiable and linkless. Further

$$\begin{aligned} C \sqcap D &= \forall R.C_1 \sqcap D \\ &= \forall R.C_1 \sqcap D_1 \sqcap D_2 \sqcap \ldots \sqcap D_{i-1} \sqcap \exists R.B' \sqcap D_{i+1} \sqcap \ldots \sqcap D_n \\ &= \exists R.(C_1 \sqcap B') \sqcap \forall R.C_1 \sqcap D_1 \sqcap D_2 \sqcap \ldots \sqcap D_{i-1} \sqcap D_{i+1} \sqcap \ldots \sqcap D_n \\ &\overset{\text{IH}}{=} \exists R.\bot \sqcap \forall R.C_1 \sqcap D_1 \sqcap D_2 \sqcap \ldots \sqcap D_{i-1} \sqcap D_{i+1} \sqcap \ldots \sqcap D_n \\ &= \bot \end{aligned}$$

which is unsatisfiable as well.

(b) Let D be $D_1 \sqcap D_2 \sqcap \ldots \sqcap D_{i-1} \sqcap \forall R.B \sqcap D_{i+1} \sqcap \ldots \sqcap D_n$ and there is no $\exists R.B'$ in D with $C_1|B' = \bot$.
Then $C|D \overset{\text{Def. 10}}{=} \forall R.(C_1|B)$ which is satisfiable and according to the induction hypothesis is linkless. Further

$$\begin{aligned} C \sqcap D &= \forall R.C_1 \sqcap D \\ &= \forall R.C_1 \sqcap D_1 \sqcap D_2 \sqcap \ldots \sqcap D_{i-1} \sqcap \forall R.B \sqcap D_{i+1} \sqcap \ldots \sqcap D_n \\ &= \forall R.(C_1 \sqcap B) \sqcap D \end{aligned}$$

which is satisfiable, since we claimed that there is no $\exists R.B'$ in D with $C_1|B' = \bot$.

(c) $\forall R.B \notin D$ and there is no $\exists R.B'$ in D with $C_1|B' = \bot$.
Then $C|D \overset{\text{Def. 10}}{=} \forall R.C_1$, which is satisfiable and according to the induction hypothesis is linkless. Further $C \sqcap D = \forall R.C_1 \sqcap D$ has to be satisfiable, since D is a query concept and for all existential role restriction $\exists R.B_i$ in D holds $C_1|B_i \neq \bot$ which implies the satisfiability of $C_1 \sqcap B_i$.

4.) $C = \exists R.C_1$

(a) Let D be $D_1 \sqcap D_2 \sqcap \ldots \sqcap D_{i-1} \sqcap \forall R.B \sqcap D_{i+1} \sqcap \ldots \sqcap D_n$ and $C_1|B = \bot$. Since D is a query concept, it is ensured that no D_j, $j \in \{1, \ldots, i-1, i+1, \ldots, n\}$ has the form $\forall R.B'$.
From the induction hypothesis follows, that $C_1 \sqcap B$ is unsatisfiable. Further $C|D \overset{\text{Def. 10}}{=} \bot$, which is unsatisfiable.
Then

$$\begin{aligned} C \sqcap D &= \exists R.C_1 \sqcap D_1 \sqcap D_2 \sqcap \ldots \sqcap D_{i-1} \sqcap \forall R.B \sqcap D_{i+1} \sqcap \ldots \sqcap D_n \\ &= \exists R.(C_1 \sqcap B) \sqcap D_1 \sqcap D_2 \sqcap \ldots \sqcap D_{i-1} \sqcap \forall R.B \sqcap D_{i+1} \sqcap \ldots \sqcap D_n \\ &\overset{\text{IH}}{=} \exists R.\bot \sqcap D \\ &= \bot \end{aligned}$$

which is not satisfiable.
Further $C|D = \bot$ is linkless.

(b) Let $\forall R.B \in D$ and $C_1|B \neq \bot$.
Then $C|D \overset{\text{Def. 10}}{=} \exists R.(C_1|B)$ is satisfiable. Further it follows from the

induction hypothesis, that $C_1 \sqcap B$ is satisfiable.
Let D be $D_1 \sqcap D_2 \sqcap \ldots \sqcap D_{i-1} \sqcap \forall R.B \sqcap D_{i+1} \sqcap \ldots \sqcap D_n$. Since D is a query concept, it is ensured that no D_j, $j \in \{1, \ldots, i-1, i+1, \ldots, n\}$ has the form $\forall R.B'$.Then

$$\begin{aligned} C \sqcap D &= \exists R.C_1 \sqcap D \\ &= \exists R.C_1 \sqcap D_1 \sqcap D_2 \sqcap \ldots \sqcap D_{i-1} \sqcap \forall R.B \sqcap D_{i+1} \sqcap \ldots \sqcap D_n \\ &= \exists R.(C_1 \sqcap B) \sqcap D \end{aligned}$$

Since both $C_1 \sqcap B$ and D are satisfiable and further $\forall R.B' \notin D$ for all $B' \neq B$, it follows that $C \sqcap D$ is satisfiable.
From the induction hypothesis follows, that $C_1|B$ is linkless. Hence $C|D = \exists R.(C_1|B)$ is linkless as well.

(c) Let $\forall R.B \notin D$.
Then $C|D \overset{\text{Def. 10}}{=} \exists R.C_1$, which is satisfiable, iff C_1 is satisfiable.
Further $C \sqcap D = \exists R.C_1 \sqcap D$ is satisfiable, iff C_1 is satisfiable. This follows from the fact that the query concept D is satisfiable and further $\forall R.B \notin D$.
In addition to that, $C|D = \exists R.C_1$ is linkless, because C_1 is linkless according to the induction hypothesis.

The next theorem follows directly from Lemma 4. It shows how to use the conditioning operator for an efficient subsumption check.

Theorem 7. *Let C be a linkless concept and D be a query concept. Then $C \sqsubseteq \neg D$ holds, iff $C|D$ is unsatisfiable.*

Corollary 1. *Let C be a linkless concept and D be a query concept. Then it can be decided in linear time, if $C \sqsubseteq \neg D$ holds.*

Corollary 1 is a direct consequence of Theorem 7 and the fact that conditioning can be performed in linear time. On page 7 in Section 4, our example concept C is given in linkless NF. We now want to check, if the subsumption $\models C \sqsubseteq E \sqcup \exists R.F$ holds. Negating the right side of the subsumption, leads to the query concept $\neg E \sqcap \forall R.\neg F$. With the help of Definition 10, we can calculate the result of conditioning C by the query concept: $C|\neg E \sqcap \forall R.\neg F = \exists R.(E \sqcap \neg B) \sqcap \forall R.\neg B \sqcap D$. Since this concept is satisfiable, the subsumption $\models C \sqsubseteq E \sqcup \exists R.F$ does not hold.

5.4 Uniform Interpolation

Another interesting transformation for precompiled theories mentioned in [6] 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, the uniform interpolant of a concept C w.r.t. a set of atomic concept symbols Φ is the concept D, which does not contain any atomic symbols from Φ and is indistinguishable from C regarding the consequences 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 11. *Let C be a concept and Φ a set of atomic concept symbols. Then the concept D is called uniform interpolant of C w.r.t. Φ or short Φ-interpolant of C, iff the following conditions hold:*

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

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

Definition 12. *Let C be a linkless concept and Φ be a set of atomic concept symbols. Then $UI(C, \Phi)$ is the concept obtained by substituting each occurrence of A and $\neg A$ in C by $\top$, iff $A \in \Phi$.*

From the way $UI(C, \Phi)$ is constructed, the following lemma is obvious:

Lemma 5. *Let C and Φ be defined as in the previous definition and C_1 and C_2 be linkless concepts. Then*

1. *If C does not contain any symbols from Φ, then $C = UI(C, \Phi)$.*
2. $UI(C_1 \odot C_2, \Phi) = UI(C_1, \Phi) \odot UI(C_2, \Phi)$ *with* $\odot \in \{\sqcap, \sqcup\}$
3. $UI(QR.C, \Phi) = QR.UI(C, \Phi)$, *for* $Q \in \{\exists, \forall\}$

Lemma 6. *Let C be a concept and Φ be a set of atomic concept symbols, such that $UI(C, \Phi)$ is defined. Then holds: C is satisfiable, iff $UI(C, \Phi)$ is satisfiable.*

Proof. We assume that C is satisfiable. Then there has to be a consistent path in C. During the construction of $UI(C, \Phi)$, the only thing that is done are substitutions by $\top$. This can never make a consistent path inconsistent. Therefore there has to be a consistent path in $UI(C, \Phi)$ as well, which makes $UI(C, \Phi)$ satisfiable.

We show the other direction of the equivalence by contraposition: We assume that C is unsatisfiable. Since C is linkless, the only way that C is unsatisfiable is that $C = \bot$. Therefore $UI(C, \Phi) = UI(\bot, \Phi) = \bot$ for all Φ, which is unsatisfiable.

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

Theorem 8. *Let C be a linkless concept and Φ a set of atomic concept symbols. Then holds:*

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

The second and the third assertion of Theorem 8 follows directly from the way, $UI(C, \Phi)$ is constructed.

Proof. (of the first assertion of Theorem 8) In order to show that $UI(C,\Phi)$ is the Φ-interpolant of C, we have to show that $UI(C,\Phi)$ has the three properties given in definition 11.

- The property, that $UI(C,\Phi)$ contains only atomic concept symbols which occur in C but not in Φ follows directly from the way, $UI(C,\Phi)$ is constructed.
- The property, that $\models C \sqsubseteq UI(C,\Phi)$ means, that $C \sqcap \neg UI(C,\Phi)$ is unsatisfiable, which follows directly from Lemma 6.
- The third property is that, for all concepts E not containing symbols from Φ holds: $\models C \sqsubseteq E$, iff $\models UI(C,\Phi) \sqsubseteq E$. We prove this by showing that for all concepts E, $C \sqcap \neg E$ is unsatisfiable iff $UI(C,\Phi) \sqcap \neg E$ is unsatisfiable.

  First note that, if C or $\neg E$ is unsatisfiable, then $C \sqcap \neg E$ and $UI(C,\Phi) \sqcap \neg E$ are unsatisfiable as well and the assertion holds. Therefore we assume C and $\neg E$ to be satisfiable.
  1. If $C \sqcap \neg E$ is satisfiable, then there is a satisfiable path p in $C \sqcap nnf(\neg E)$. We show that this path p corresponds to a satisfiable path p' in $UI(C,\Phi) \sqcap nnf(\neg E)$. Let p_C denote the subpath of p, passing through C and $p_{\neg E}$ be the subpath of p passing through $nnf(\neg E)$. If we consider $UI(C,\Phi) \sqcap \neg E$, we know that the construction of $UI(C,\Phi)$ only changes the subpath p_C of p such that each occurrence of A and $\neg A$ with $A \in \Phi$ is substituted by $\top$. This can not cause path p to become unsatisfiable, so the resulting path p' is satisfiable as well. Therefore $UI(C,\Phi) \sqcap nnf(\neg E)$ and $UI(C,\Phi) \sqcap \neg E$ is satisfiable.
  2. If $C \sqcap \neg E$ is unsatisfiable, then all paths in $C \sqcap nnf(\neg E)$ are unsatisfiable. However due to the assumption that both C and $\neg E$ are satisfiable, we know that the subpaths p_C and $p_{\neg E}$ are satisfiable. Therefore the contradiction in a path has to be constructed from both elements from p_C and $p_{\neg E}$ and has to use symbols not occurring in Φ. However these symbols are not affected by the construction of UI. Thus the contradictions are still contained in the paths of $UI(C,\Phi) \sqcap nnf(\neg E)$, which implies that $UI(C,\Phi) \sqcap \neg E$ has to be unsatisfiable.

Given for example the set $\Phi = \{E, D\}$, we can calculate the Φ-interpolant of the linkless concept C from the example given in Section 4 on page 7: $UI(C,\Phi) = (\exists R.(\top \sqcap \neg B) \sqcap \forall R.\neg B \sqcap (\top \sqcup \top)) \sqcup (\exists R.(\top \sqcap \neg B \sqcap F) \sqcap \forall R.(\neg B \sqcap F))$ which can be simplified according to Fig. 1 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 Conclusion / Future Work

This paper presents a precompilation of $\mathcal{ALC}$ concepts into a NF called linkless concepts, which allows for an efficient satisfiability test, subsumption test and uniform interpolation. In [8] we presented a method to transform $\mathcal{ALC}$ concepts and Tboxes into a special structure called *linkless graph* which is closely related to the idea of the precompilation presented in this paper. The precompilation into linkless graphs is implemented and first promising results can be found in [8]. Since the two precompilation techniques are closely related, the experimental results given in [8] can be seen as experimental results for the precompilation presented in this paper as well. A disadvantage of the precompilation of concepts into linkless graphs is, that there is no possibility to see the result of the precompilation as a concept. This paper remedies this situation by presenting linkless concepts as the result of the precompilation process. This makes the whole precompilation process more comprehensible and makes certain properties of precompiled concepts more obvious.

Next, we will extend the linkless NF to handle $\mathcal{ALC}$ Tboxes. We expect this step to be manageable, since the linkless graphs are already developed for $\mathcal{ALC}$ Tboxes. However, when constructing the uniform interpolant for a precompiled TBox, things get more complicated, since uniform interpolants for $\mathcal{ALC}$ Tboxes need not exist. We are planning to focus our research on this area as well.

References

1. Franz Baader, Ralf Küsters, and Ralf Molitor. Computing least common subsumers in description logics with existential restrictions. pages 96–101. Morgan Kaufmann, 1999.
2. Baader, F. and Calvanese, D. and McGuinness, D. and Nardi, D. and Patel-Schneider, P., editor. *The Description Logic Handbook*. Cambridge University Press, 2003.
3. P. Balsiger and A. Heuerding. Comparison of theorem provers for modal logics - introduction and summary. In *TABLEAUX*, volume 1397 of *LNCS*, pages 25–26. Springer, 1998.
4. M. Bienvenu. Prime implicate normal form for alc concepts. In *Proc. of the 21st International Workshop on Description Logics (DL2008)*, 2008.
5. A. Darwiche. Decomposable Negation Normal Form. *Journal of the ACM*, 48(4), 2001.
6. Adnan Darwiche and Pierre Marquis. A knowledge compilation map. *Journal of Artificial Intelligence Research*, 17:229–264, 2002.
7. Darwiche, A. and Marquis, P. A knowlege compilation map. *Journal of Artificial Intelligence Research*, 17:229–264, 2002.
8. Ulrich Furbach, Heiko Günther, and Claudia Obermaier. A knowledge compilation technique for alc tboxes. In *Pro. of the Twenty-Second International FLAIRS Conference*, 2009. To appear.
9. Ulrich Furbach and Claudia 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. [2], pages 306–346.
11. B. Konev, D. Walther, and F. Wolter. Forgetting and uniform interpolation in large-scale description logic terminologies. In *Proc. of IJCAI 09*, 2009.
12. N. Murray and E. Rosenthal. Tableaux, Path Dissolution, and Decomposable Negation Normal Form for Knowledge Compilation. In *Proceedings of TABLEAUX 2003*, volume 1397 of *LNCS*. Springer, 2003.

13. Murray, N. and Rosenthal, E. Dissolution: Making paths vanish. *J. ACM*, 40(3):504–535, 1993.
14. Selman, B. and Kautz, H. Knowledge compilation and theory approximation. *J. ACM*, 43(2):193–224, 1996.
15. D. Tsarkov and I. Horrocks. Fact++ description logic reasoner: System description. In *Proc. of the Int. Joint Conf. on Automated Reasoning*, pages 292–297. Springer, 2006.

Bisher erschienen

Arbeitsberichte aus dem Fachbereich Informatik

(http://www.uni-koblenz-landau.de/koblenz/fb4/publications/Reports/arbeitsberichte)

Claudia Schon, Linkless Normal Form for ALC Concepts, Arbeitsberichte aus dem Fachbereich Informatik 12/2010

Alexander Hug, Informatik hautnah erleben, Arbeitsberichte aus dem Fachbereich Informatik 11/2010

Marc Santos, Harald F.O. von Kortzfleisch, Shared Annotation Model – Ein Datenmodell für kollaborative Annotationen, Arbeitsberichte aus dem Fachbereich Informatik 10/2010

Gerd Gröner, Steffen Staab, Categorization and Recognition of Ontology Refactoring Pattern, Arbeitsberichte aus dem Fachbereich Informatik 9/2010

Daniel Eißing, Ansgar Scherp, Carsten Saathoff, Integration of Existing Multimedia Metadata Formats and Metadata Standards in the M3O, Arbeitsberichte aus dem Fachbereich Informatik 8/2010

Stefan Scheglmann, Ansgar Scherp, Steffen Staab, Model-driven Generation of APIs for OWL-based Ontologies, Arbeitsberichte aus dem Fachbereich Informatik 7/2010

Daniel Schmeiß, Ansgar Scherp, Steffen Staab, Integrated Mobile Visualization and Interaction of Events and POIs, Arbeitsberichte aus dem Fachbereich Informatik 6/2010

Rüdiger Grimm, Daniel Pähler, E-Mail-Forensik – IP-Adressen und ihre Zuordnung zu Internet-Teilnehmern und ihren Standorten, Arbeitsberichte aus dem Fachbereich Informatik 5/2010

Christoph Ringelstein, Steffen Staab, PAPEL: Syntax and Semantics for Provenance-Aware Policy Definition, Arbeitsberichte aus dem Fachbereich Informatik 4/2010

Nadine Lindermann, Sylvia Valcárcel, Harald F.O. von Kortzfleisch, Ein Stufenmodell für kollaborative offene Innovationsprozesse in Netzwerken kleiner und mittlerer Unternehmen mit Web 2.0, Arbeitsberichte aus dem Fachbereich Informatik 3/2010

Maria Wimmer, Dagmar Lück-Schneider, Uwe Brinkhoff, Erich Schweighofer, Siegfried Kaiser, Andreas Wieber, Fachtagung Verwaltungsinformatik FTVI Fachtagung Rechtsinformatik FTRI 2010, Arbeitsberichte aus dem Fachbereich Informatik 2/2010

Max Braun, Ansgar Scherp, Steffen Staab, Collaborative Creation of Semantic Points of Interest as Linked Data on the Mobile Phone, Arbeitsberichte aus dem Fachbereich Informatik 1/2010

Marc Santos, Einsatz von „Shared In-situ Problem Solving“ Annotationen in kollaborativen Lern- und Arbeitsszenarien, Arbeitsberichte aus dem Fachbereich Informatik 20/2009

Carsten Saathoff, Ansgar Scherp, Unlocking the Semantics of Multimedia Presentations in the Web with the Multimedia Metadata Ontology, Arbeitsberichte aus dem Fachbereich Informatik 19/2009

Christoph Kahle, Mario Schaarschmidt, Harald F.O. von Kortzfleisch, Open Innovation: Kundenintegration am Beispiel von IPTV, Arbeitsberichte aus dem Fachbereich Informatik 18/2009

Dietrich Paulus, Lutz Priese, Peter Decker, Frank Schmitt, Pose-Tracking Forschungsbericht, Arbeitsberichte aus dem Fachbereich Informatik 17/2009

Andreas Fuhr, Tassilo Horn, Andreas Winter, Model-Driven Software Migration Extending SOMA, Arbeitsberichte aus dem Fachbereich Informatik 16/2009

Eckhard Großmann, Sascha Strauß, Tassilo Horn, Volker Riediger, Abbildung von grUML nach XSD soamig, Arbeitsberichte aus dem Fachbereich Informatik 15/2009

Kerstin Falkowski, Jürgen Ebert, The STOR Component System Interim Report, Arbeitsberichte aus dem Fachbereicht Informatik 14/2009

Sebastian Magnus, Markus Maron, An Empirical Study to Evaluate the Location of Advertisement Panels by Using a Mobile Marketing Tool, Arbeitsberichte aus dem Fachbereich Informatik 13/2009

Sebastian Magnus, Markus Maron, Konzept einer Public Key Infrastruktur in iCity, Arbeitsberichte aus dem Fachbereich Informatik 12/2009

Sebastian Magnus, Markus Maron, A Public Key Infrastructure in Ambient Information and Transaction Systems, Arbeitsberichte aus dem Fachbereich Informatik 11/2009

Ammar Mohammed, Ulrich Furbach, Multi-agent systems: Modeling and Virification using Hybrid Automata, Arbeitsberichte aus dem Fachbereich Informatik 10/2009

Andreas Sprotte, Performance Measurement auf der Basis von Kennzahlen aus betrieblichen Anwendungssystemen: Entwurf eines kennzahlengestützten Informationssystems für einen Logistikdienstleister, Arbeitsberichte aus dem Fachbereich Informatik 9/2009

Gwendolin Garbe, Tobias Hausen, Process Commodities: Entwicklung eines Reifegradmodells als Basis für Outsourcingentscheidungen, Arbeitsberichte aus dem Fachbereich Informatik 8/2009

Petra Schubert et. al., Open-Source-Software für das Enterprise Resource Planning, Arbeitsberichte aus dem Fachbereich Informatik 7/2009

Ammar Mohammed, Frieder Stolzenburg, Using Constraint Logic Programming for Modeling and Verifying Hierarchical Hybrid Automata, Arbeitsberichte aus dem Fachbereich Informatik 6/2009

Tobias Kippert, Anastasia Meletiadou, Rüdiger Grimm, Entwurf eines Common Criteria-Schutzprofils für Router zur Abwehr von Online-Überwachung, Arbeitsberichte aus dem Fachbereich Informatik 5/2009

Hannes Schwarz, Jürgen Ebert, Andreas Winter, Graph-based Traceability – A Comprehensive Approach. Arbeitsberichte aus dem Fachbereich Informatik 4/2009

Anastasia Meletiadou, Simone Müller, Rüdiger Grimm, Anforderungsanalyse für Risk-Management-Informationssysteme (RMIS), Arbeitsberichte aus dem Fachbereich Informatik 3/2009

Ansgar Scherp, Thomas Franz, Carsten Saathoff, Steffen Staab, A Model of Events based on a Foundational Ontology, Arbeitsberichte aus dem Fachbereich Informatik 2/2009

Frank Bohdanovicz, Harald Dickel, Christoph Steigner, Avoidance of Routing Loops, Arbeitsberichte aus dem Fachbereich Informatik 1/2009

Stefan Ameling, Stephan Wirth, Dietrich Paulus, Methods for Polyp Detection in Colonoscopy Videos: A Review, Arbeitsberichte aus dem Fachbereich Informatik 14/2008

Tassilo Horn, Jürgen Ebert, Ein Referenzschema für die Sprachen der IEC 61131-3, Arbeitsberichte aus dem Fachbereich Informatik 13/2008

Thomas Franz, Ansgar Scherp, Steffen Staab, Does a Semantic Web Facilitate Your Daily Tasks?, Arbeitsberichte aus dem Fachbereich Informatik 12/2008

Norbert Frick, Künftige Anfordeungen an ERP-Systeme: Deutsche Anbieter im Fokus, Arbeitsberichte aus dem Fachbereicht Informatik 11/2008

Jürgen Ebert, Rüdiger Grimm, Alexander Hug, Lehramtsbezogene Bachelor- und Masterstudiengänge im Fach Informatik an der Universität Koblenz-Landau, Campus Koblenz, Arbeitsberichte aus dem Fachbereich Informatik 10/2008

Mario Schaarschmidt, Harald von Kortzfleisch, Social Networking Platforms as Creativity Fostering Systems: Research Model and Exploratory Study, Arbeitsberichte aus dem Fachbereich Informatik 9/2008

Bernhard Schueler, Sergej Sizov, Steffen Staab, Querying for Meta Knowledge, Arbeitsberichte aus dem Fachbereich Informatik 8/2008

Stefan Stein, Entwicklung einer Architektur für komplexe kontextbezogene Dienste im mobilen Umfeld, Arbeitsberichte aus dem Fachbereich Informatik 7/2008

Matthias Bohnen, Lina Brühl, Sebastian Bzdak, RoboCup 2008 Mixed Reality League Team Description, Arbeitsberichte aus dem Fachbereich Informatik 6/2008

Bernhard Beckert, Reiner Hähnle, Tests and Proofs: Papers Presented at the Second International Conference, TAP 2008, Prato, Italy, April 2008, Arbeitsberichte aus dem Fachbereich Informatik 5/2008

Klaas Dellschaft, Steffen Staab, Unterstützung und Dokumentation kollaborativer Entwurfs- und Entscheidungsprozesse, Arbeitsberichte aus dem Fachbereich Informatik 4/2008

Rüdiger Grimm: IT-Sicherheitsmodelle, Arbeitsberichte aus dem Fachbereich Informatik 3/2008

Rüdiger Grimm, Helge Hundacker, Anastasia Meletiadou: Anwendungsbeispiele für Kryptographie, Arbeitsberichte aus dem Fachbereich Informatik 2/2008

Markus Maron, Kevin Read, Michael Schulze: CAMPUS NEWS – Artificial Intelligence Methods Combined for an Intelligent Information Network, Arbeitsberichte aus dem Fachbereich Informatik 1/2008

Lutz Priese,Frank Schmitt, Patrick Sturm, Haojun Wang: BMBF-Verbundprojekt 3D-RETISEG Abschlussbericht des Labors Bilderkennen der Universität Koblenz-Landau, Arbeitsberichte aus dem Fachbereich Informatik 26/2007

Stephan Philippi, Alexander Pinl: Proceedings 14. Workshop 20.-21. September 2007 Algorithmen und Werkzeuge für Petrinetze, Arbeitsberichte aus dem Fachbereich Informatik 25/2007

Ulrich Furbach, Markus Maron, Kevin Read: CAMPUS NEWS – an Intelligent Bluetooth-based Mobile Information Network, Arbeitsberichte aus dem Fachbereich Informatik 24/2007

Ulrich Furbach, Markus Maron, Kevin Read: CAMPUS NEWS - an Information Network for Pervasive Universities, Arbeitsberichte aus dem Fachbereich Informatik 23/2007

Lutz Priese: Finite Automata on Unranked and Unordered DAGs Extented Version, Arbeitsberichte aus dem Fachbereich Informatik 22/2007

Mario Schaarschmidt, Harald F.O. von Kortzfleisch: Modularität als alternative Technologie- und Innovationsstrategie, Arbeitsberichte aus dem Fachbereich Informatik 21/2007

Kurt Lautenbach, Alexander Pinl: Probability Propagation Nets, Arbeitsberichte aus dem Fachbereich Informatik 20/2007

Rüdiger Grimm, Farid Mehr, Anastasia Meletiadou, Daniel Pähler, Ilka Uerz: SOA-Security, Arbeitsberichte aus dem Fachbereich Informatik 19/2007

Christoph Wernhard: Tableaux Between Proving, Projection and Compilation, Arbeitsberichte aus dem Fachbereich Informatik 18/2007

Ulrich Furbach, Claudia Obermaier: Knowledge Compilation for Description Logics, Arbeitsberichte aus dem Fachbereich Informatik 17/2007

Fernando Silva Parreiras, Steffen Staab, Andreas Winter: TwoUse: Integrating UML Models and OWL Ontologies, Arbeitsberichte aus dem Fachbereich Informatik 16/2007

Rüdiger Grimm, Anastasia Meletiadou: Rollenbasierte Zugriffskontrolle (RBAC) im Gesundheitswesen, Arbeitsberichte aud dem Fachbereich Informatik 15/2007

Ulrich Furbach, Jan Murray, Falk Schmidsberger, Frieder Stolzenburg: Hybrid Multiagent Systems with Timed Synchronization-Specification and Model Checking, Arbeitsberichte aus dem Fachbereich Informatik 14/2007

Björn Pelzer, Christoph Wernhard: System Description:“E-KRHyper“, Arbeitsberichte aus dem Fachbereich Informatik, 13/2007

Ulrich Furbach, Peter Baumgartner, Björn Pelzer: Hyper Tableaux with Equality, Arbeitsberichte aus dem Fachbereich Informatik, 12/2007

Ulrich Furbach, Markus Maron, Kevin Read: Location based Informationsystems, Arbeitsberichte aus dem Fachbereich Informatik, 11/2007

Philipp Schaer, Marco Thum: State-of-the-Art: Interaktion in erweiterten Realitäten, Arbeitsberichte aus dem Fachbereich Informatik, 10/2007

Ulrich Furbach, Claudia Obermaier: Applications of Automated Reasoning, Arbeitsberichte aus dem Fachbereich Informatik, 9/2007

Jürgen Ebert, Kerstin Falkowski: A First Proposal for an Overall Structure of an Enhanced Reality Framework, Arbeitsberichte aus dem Fachbereich Informatik, 8/2007

Lutz Priese, Frank Schmitt, Paul Lemke: Automatische See-Through Kalibrierung, Arbeitsberichte aus dem Fachbereich Informatik, 7/2007

Rüdiger Grimm, Robert Krimmer, Nils Meißner, Kai Reinhard, Melanie Volkamer, Marcel Weinand, Jörg Helbach: Security Requirements for Non-political Internet Voting, Arbeitsberichte aus dem Fachbereich Informatik, 6/2007

Daniel Bildhauer, Volker Riediger, Hannes Schwarz, Sascha Strauß, „grUML – Eine UML-basierte Modellierungssprache für T-Graphen“, Arbeitsberichte aus dem Fachbereich Informatik, 5/2007

Richard Arndt, Steffen Staab, Raphaël Troncy, Lynda Hardman: Adding Formal Semantics to MPEG-7: Designing a Well Founded Multimedia Ontology for the Web, Arbeitsberichte aus dem Fachbereich Informatik, 4/2007

Simon Schenk, Steffen Staab: Networked RDF Graphs, Arbeitsberichte aus dem Fachbereich Informatik, 3/2007

Rüdiger Grimm, Helge Hundacker, Anastasia Meletiadou: Anwendungsbeispiele für Kryptographie, Arbeitsberichte aus dem Fachbereich Informatik, 2/2007

Anastasia Meletiadou, J. Felix Hampe: Begriffsbestimmung und erwartete Trends im IT-Risk-Management, Arbeitsberichte aus dem Fachbereich Informatik, 1/2007

„Gelbe Reihe“
(http://www.uni-koblenz.de/fb4/publikationen/gelbereihe)

Lutz Priese: Some Examples of Semi-rational and Non-semi-rational DAG Languages. Extended Version, Fachberichte Informatik 3-2006

Kurt Lautenbach, Stephan Philippi, and Alexander Pinl: Bayesian Networks and Petri Nets, Fachberichte Informatik 2-2006

Rainer Gimnich and Andreas Winter: Workshop Software-Reengineering und Services, Fachberichte Informatik 1-2006

Kurt Lautenbach and Alexander Pinl: Probability Propagation in Petri Nets, Fachberichte Informatik 16-2005

Rainer Gimnich, Uwe Kaiser, and Andreas Winter: 2. Workshop "Reengineering Prozesse" – Software Migration, Fachberichte Informatik 15-2005

Jan Murray, Frieder Stolzenburg, and Toshiaki Arai: Hybrid State Machines with Timed Synchronization for Multi-Robot System Specification, Fachberichte Informatik 14-2005

Reinhold Letz: FTP 2005 – Fifth International Workshop on First-Order Theorem Proving, Fachberichte Informatik 13-2005

Bernhard Beckert: TABLEAUX 2005 – Position Papers and Tutorial Descriptions, Fachberichte Informatik 12-2005

Dietrich Paulus and Detlev Droege: Mixed-reality as a challenge to image understanding and artificial intelligence, Fachberichte Informatik 11-2005

Jürgen Sauer: 19. Workshop Planen, Scheduling und Konfigurieren / Entwerfen, Fachberichte Informatik 10-2005

Pascal Hitzler, Carsten Lutz, and Gerd Stumme: Foundational Aspects of Ontologies, Fachberichte Informatik 9-2005

Joachim Baumeister and Dietmar Seipel: Knowledge Engineering and Software Engineering, Fachberichte Informatik 8-2005

Benno Stein and Sven Meier zu Eißen: Proceedings of the Second International Workshop on Text-Based Information Retrieval, Fachberichte Informatik 7-2005

Andreas Winter and Jürgen Ebert: Metamodel-driven Service Interoperability, Fachberichte Informatik 6-2005

Joschka Boedecker, Norbert Michael Mayer, Masaki Ogino, Rodrigo da Silva Guerra, Masaaki Kikuchi, and Minoru Asada: Getting closer: How Simulation and Humanoid League can benefit from each other, Fachberichte Informatik 5-2005

Torsten Gipp and Jürgen Ebert: Web Engineering does profit from a Functional Approach, Fachberichte Informatik 4-2005

Oliver Obst, Anita Maas, and Joschka Boedecker: HTN Planning for Flexible Coordination Of Multiagent Team Behavior, Fachberichte Informatik 3-2005

Andreas von Hessling, Thomas Kleemann, and Alex Sinner: Semantic User Profiles and their Applications in a Mobile Environment, Fachberichte Informatik 2-2005

Heni Ben Amor and Achim Rettinger: Intelligent Exploration for Genetic Algorithms – Using Self-Organizing Maps in Evolutionary Computation, Fachberichte Informatik 1-2005