

Prime Implicate Normal Form for \mathcal{ALC} Concepts

Meghyn Bienvenu
IRIT-UPS, 118 route de Narbonne,
31062 Toulouse Cedex, France

IRIT Research Report IRIT/RR-2008-6-FR

April 2008

Abstract

In this paper, we present a normal form for concept expressions in the description logic \mathcal{ALC} which is based on a recently introduced notion of prime implicate for the modal logic \mathcal{K} . We show that concepts in prime implicate normal form enjoy a number of interesting properties. For one thing, they do not contain any unnecessary atomic concepts or roles. Not only does this make the concept more readable but it also helps us to identify the parts of a concept which are relevant to a given subject matter. Another feature of concepts in prime implicate normal form is that they can be easily approximated over a sublanguage or up to a fixed depth. These operations may prove useful when a concept description is too large to be fully understood or when data needs to be exchanged between systems using different languages. Perhaps the most remarkable property of prime implicate normal form is that subsumption between \mathcal{ALC} concepts in this form can be carried out in quadratic time using a simple structural subsumption algorithm reminiscent of those used for less expressive description logics. This property makes prime implicate normal form interesting for the purposes of knowledge compilation. Of course, in order to take advantage of all of these nice properties, we need a way to transform concepts into equivalent concepts in prime implicate normal form. We provide a sound and complete algorithm for putting concepts into prime implicate normal form, and we investigate the spatial complexity of this transformation, showing there to be an at most doubly-exponential blowup in concept size. At the end of the paper, we compare prime implicate normal form to two other normal forms for \mathcal{ALC} concepts that have been proposed in the literature, discussing the relative merits of the different approaches.

1 Introduction

It is well-known that standard reasoning tasks are intractable for propositional logic, and the complexity of reasoning increases as one moves up to more expressive logics. Researchers have investigated a variety of strategies for coping with the high computational complexity of reasoning. Some have looked into restricted languages for which efficient reasoning is possible. Others have focused their efforts on the development of reasoning algorithms which perform well in practice, even if the worst-case complexity remains high. Still others have advocated the use of knowledge compilation [8], in which a knowledge base is put into a normal form which admits polytime querying, the idea being that the cost of the initial preprocessing will be offset by the computational savings made on later queries.

In the description logics community, the first two strategies have been privileged, while the third strategy, knowledge compilation, has remained largely unexplored. The likely explanation for this phenomenon is not a lack of interest on the part of this community but the simple fact that there have been no normal forms proposed in the literature which yield tractable reasoning for any reasonably expressive description logic.

Our paper aims to help remedy this situation by showing how prime implicate normal form, a well-studied normal form for propositional logic which has been influential in AI, can be extended to concept expressions in the description logic \mathcal{ALC} . The starting point for this paper is our recent study [4, 5] of prime implicates for the modal logic \mathcal{K} , a known notational variant of \mathcal{ALC} . While the definition of prime implicates proposed in [4, 5] does not immediately yield a suitable notion of prime implicate normal form, it will play a key role in the definition we propose later in this paper. Concepts in our normal form can be shown to be much better behaved computationally than arbitrary \mathcal{ALC} concepts: we can test in constant time whether a concept in prime implicate normal form is satisfiable or tautologous and in quadratic time whether two concepts in prime implicate normal form are equivalent or if one subsumes the other. It is also easy to approximate concepts in prime implicate normal form over a sublanguage or up to a specified depth. Finally, concepts in prime implicate normal form do not contain any unnecessary atomic concepts or roles nor do they contain redundant or irrelevant subconcepts, properties which make them easier for humans to read and understand.

Our paper is organized as follows. In the first two sections, we recall the basics of the description logic \mathcal{ALC} and the notion of prime implicates in \mathcal{ALC} . In the following section, we propose a definition of prime implicate

normal form for \mathcal{ALC} concepts, and we show that concepts in this form support a variety of polynomial-time queries and transformations. We then introduce an algorithm for putting concepts into prime implicate normal form and give some results concerning the spatial complexity of this transformation. At the end of the paper, we provide a detailed comparison of prime implicate normal form to two other normal forms for \mathcal{ALC} concepts, and then we conclude with a discussion of future work.

2 Preliminaries

In this section, we recall the syntax and semantics of the description logic \mathcal{ALC} as well as other useful notions and notations. We start off with the syntax of concept expressions in \mathcal{ALC} :

Definition 1 (Syntax of \mathcal{ALC}). Concept expressions in \mathcal{ALC} are defined as follows:

$$C ::= \top \mid \perp \mid A \mid \neg C \mid C \sqcap C \mid C \sqcup C \mid \forall R.C \mid \exists R.C$$

where A ranges over atomic concept names, R ranges over atomic role names, and C ranges over the set of \mathcal{ALC} concepts.

In analogy with classical logic, we will say that $C_1 \sqcap C_2$ is a conjunction (or intersection) of concepts, and we will call C_1 and C_2 conjuncts of $C_1 \sqcap C_2$. Likewise, we will say that $D_1 \sqcup D_2$ is a disjunction (or union) of concepts and that D_1 and D_2 are its disjuncts. Where convenient we will abuse notation and treat conjunction and disjunction as n -ary connectives. A concept is said to be in *negation normal form* (NNF) if negation only appears directly before atomic concepts. The *length* of a concept C , written $|C|$, is defined to be the total number of occurrences of atomic concept and role names in C . The (role) *depth* of a concept C , noted $\delta(C)$, is defined to be the maximum number of nested $\exists R$ or $\forall R$ appearing in C . For example, the depth of the concept $(A \sqcup \forall R.(A \sqcup \forall S.A)) \sqcap \exists R.\exists S.\exists S.A$ is 3. We will call a concept *propositional* if it has depth 0. We define a *signature* to be any set of atomic roles and concepts. We define the *signature of a concept* C , written $\text{sig}(C)$, to be the set of atomic concepts and roles which appear in C . For example, the signature of the concept $\forall R.A \sqcap \exists S.B$ is $\{R, S, A, B\}$.

The meaning of \mathcal{ALC} concepts is defined via a model-theoretic semantics. An interpretation (model) \mathcal{I} is a pair $\langle \Omega^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$, where $\Omega^{\mathcal{I}}$ is a non-empty set and $\cdot^{\mathcal{I}}$ is a function mapping each atomic concept A to a set $A^{\mathcal{I}} \subseteq \Omega^{\mathcal{I}}$ and

each atomic role R to a relation $R^{\mathcal{I}} \subseteq \Omega^{\mathcal{I}} \times \Omega^{\mathcal{I}}$. We extend $\cdot^{\mathcal{I}}$ to complex concepts as follows:

$$\begin{aligned}
\top^{\mathcal{I}} &= \Omega^{\mathcal{I}} \\
\perp^{\mathcal{I}} &= \emptyset \\
(\neg C)^{\mathcal{I}} &= \Omega^{\mathcal{I}} \setminus C^{\mathcal{I}} \\
(C \sqcap D)^{\mathcal{I}} &= C^{\mathcal{I}} \cap D^{\mathcal{I}} \\
(C \sqcup D)^{\mathcal{I}} &= C^{\mathcal{I}} \cup D^{\mathcal{I}} \\
(\forall R.C)^{\mathcal{I}} &= \{a \in \Omega^{\mathcal{I}} \mid \forall b. (a, b) \in R^{\mathcal{I}} \Rightarrow b \in C^{\mathcal{I}}\} \\
(\exists R.C)^{\mathcal{I}} &= \{a \in \Omega^{\mathcal{I}} \mid \exists b. (a, b) \in R^{\mathcal{I}} \text{ and } b \in C^{\mathcal{I}}\}
\end{aligned}$$

A concept C is said to be *satisfiable* if there is some interpretation \mathcal{I} for which $C^{\mathcal{I}} \neq \emptyset$. If there is no such interpretation, then C is said to be *unsatisfiable*, and we write $\models C \sqsubseteq \perp$. A concept C is said to be *tautologous*, written $\models \top \sqsubseteq C$, just in the case that $\neg C$ is unsatisfiable. We say that a concept C is *subsumed* by D (or that D *subsumes* C), written $\models C \sqsubseteq D$, if for every model \mathcal{I} we have $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$. Concepts C and D are said to be *equivalent*, written $\models C \equiv D$ if C subsumes D and D subsumes C .

We finish this section with the definition of \mathcal{L} - and n -interpolants.

Definition 2 (\mathcal{L} -interpolant). A concept C is said to be the uniform interpolant of a concept D with respect to the signature \mathcal{L} , or simply the \mathcal{L} -interpolant of D , if and only if $\text{Sig}(C) \subseteq \mathcal{L}$, $\models D \sqsubseteq C$, and $\models C \sqsubseteq E$ for every concept E such that $\text{Sig}(E) \subseteq \mathcal{L}$ and $\models D \sqsubseteq E$.

\mathcal{L} -interpolants (cf. [18]) give us the finest approximation of a concept over a given signature. This notion is closely related to the notion of forgetting which has been extensively studied in propositional setting (cf. [13]): the result of forgetting a set S of atomic concepts and roles from a concept C is precisely the $\text{Sig}(C) \setminus S$ -interpolant of C .

Another type of approximation which makes sense for description logics is approximation up to a specified depth. This notion has not been studied in propositional logic for obvious reasons.

Definition 3 (n -interpolant). A concept C is the n -interpolant of a concept D if and only if $\delta(C) \leq n$, $\models D \sqsubseteq C$, and $\models C \sqsubseteq E$ for every concept E such that $\delta(E) \leq n$ and $\models D \sqsubseteq E$.

It can be trivially shown that \mathcal{L} - and n -interpolants are unique up to equivalence, which is why we can say that a concept is *the* \mathcal{L} - or n -interpolant of another concept.

3 Prime Implicates of \mathcal{ALC} Concepts

In this section, we define prime implicates for \mathcal{ALC} concepts and point out some of their key properties. All of the definitions and results in this section first appeared in [4] for the modal logic \mathcal{K} . We have adapted them to \mathcal{ALC} using the well-known correspondence [17] between formulae in \mathcal{K} and concept expressions in \mathcal{ALC} . For more details and for proofs of the results in this section, refer to [5].

Definition 4 (Literal/Clausal/Cubal Concepts). We define literal, clausal, and cubal concepts as follows:

$$\begin{aligned}
 L & ::= \top \mid \perp \mid A \mid \neg A \mid \forall R.D \mid \exists R.D \\
 Cl & ::= L \mid Cl \sqcup Cl \\
 Cb & ::= L \mid Cb \sqcap Cb \\
 D & ::= \top \mid \perp \mid A \mid \neg A \mid D \sqcap D \mid D \sqcup D \mid \forall R.D \mid \exists R.D
 \end{aligned}$$

where A ranges over atomic concept names, R ranges over atomic role names, and L , Cl , and Cb range respectively over the sets of *literal concepts*, *clausal concepts*, and *cubal concepts*. We will use the term *atomic literal concept* to refer to concepts of the form A or $\neg A$ where A is an atomic concept name.

In order to aid the presentation, we introduce the notation $Prop(Cl)$ to refer to the set of propositional literal concepts which are disjuncts of the clausal concept Cl , and we will use $\exists R(Cl)$ (respectively $\forall R(Cl)$) to refer to the set of concepts C such that $\exists R.C$ (respectively $\forall R.C$) is a disjunct of Cl . For example, if $Cl = A \sqcup \exists R.A \sqcup \exists R.B \sqcup \forall S.B$, then $Prop(Cl) = \{A\}$, $\exists R(Cl) = \{A, B\}$, $\forall S(Cl) = \{B\}$, and $\exists S(Cl) = \forall R(Cl) = \emptyset$. Abusing notation, we will also use $Prop(Cb)$, $\exists R(Cb)$, and $\forall R(Cb)$ to refer respectively to the propositional literal concepts which are conjuncts of the cubal concept Cb , the concepts C such that $\exists R(C)$ is a conjunct of Cb , and the concepts C such that $\forall R.C$ is a conjunct of Cb . If C is either a clausal or cubal concept, then we will use $Roles_{\exists}(C)$ (respectively $Roles_{\forall}(C)$) to refer to the set of roles such that $\exists R(C) \neq \emptyset$ (respectively $\forall R(C) \neq \emptyset$).

Sometimes it will prove convenient to treat clausal concepts as sets of literal concepts. We use the notation $Cl \setminus \{L_1, \dots, L_n\}$ to refer to the clausal concept obtained by removing the concepts L_1, \dots, L_n from the union of concepts in Cl . Likewise, we write $Cl \cup \{L_1, \dots, L_n\}$ for the clausal concept which is obtained from Cl by adding the literal concepts L_1, \dots, L_n to the union.

We now recall some basic properties of clausal concepts.

Proposition 5. *Every concept C is equivalent to an intersection of clausal concepts $Cl_1 \sqcap \dots \sqcap Cl_n$ such that $\delta(Cl_i) \leq \delta(C)$ and $Sig(Cl_i) \subseteq Sig(C)$ for all i .*

Proposition 6. *Let P_1, \dots, P_k be atomic literal concepts, let E, F_1, \dots, F_n be concept expressions in \mathcal{ALC} , and let R be a role. Then*

1. $\models \exists R.F_1 \sqcup \dots \sqcup \exists R.F_n \equiv \exists R.(F_1 \sqcup \dots \sqcup F_n)$
2. $\models \exists R.E \sqcup \forall R.F_1 \sqcup \dots \sqcup \forall R.F_n \equiv \exists R.E \sqcup \forall R.(F_1 \sqcup E) \sqcup \dots \sqcup \forall R.(F_n \sqcup E)$
3. $\models \top \sqsubseteq P_1 \sqcup \dots \sqcup P_k \sqcup \exists R.E \sqcup \forall R.F_1 \sqcup \dots \sqcup \forall R.F_n \Leftrightarrow$
 $\models \top \sqsubseteq P_1 \sqcup \dots \sqcup P_k$ or $\models \top \sqsubseteq E \sqcup F_i$ for some i

Proposition 7. *Let C and D be clausal concepts. If $\not\models \top \sqsubseteq D$, then $\models C \sqsubseteq D$ just in the case the following three conditions hold:*

1. $Prop(C) \subseteq Prop(D)$
2. If $R \in Roles_{\exists}(C)$, then $\models \bigsqcup_{E \in \exists R(C)} E \sqsubseteq \bigsqcup_{E \in \exists R(D)} E$ (or simply $\models \bigsqcup_{E \in \exists R(C)} E \sqsubseteq \perp$ if $R \notin Roles_{\exists}(D)$)
3. If $F \in \forall R(C)$, then there is some $G \in \forall R(D)$ such that $\models F \sqsubseteq G \sqcup (\bigsqcup_{E \in \exists R(D)} E)$ (or simply $\models F \sqsubseteq G$ if $R \notin Roles_{\exists}(D)$)

With a notion of clause in hand, we can define prime implicates just as in propositional logic.

Definition 8 (Prime implicate). A clausal concept Cl is an implicate of a concept C if and only if $\models C \sqsubseteq Cl$. A clausal concept Cl is a *prime implicate* of C if and only if:

1. Cl is an implicate of C
2. If Cl' is an implicate of C such that $\models Cl' \sqsubseteq Cl$, then $\models Cl \sqsubseteq Cl'$

This definition yields the standard notion of prime implicates when restricted to the propositional fragment of \mathcal{ALC} . It can also be shown to satisfy a number of properties of the propositional definition, most notably finiteness, covering, and equivalence:

Proposition 9 (Finiteness). *The number of prime implicates of a concept is finite modulo logical equivalence.*

Proposition 10 (Covering). *Every implicate of a concept subsumes some prime implicate of the concept.*

Proposition 11 (Equivalence). *Every concept is equivalent to the intersection of its prime implicates.*

Proposition 12 (Distribution). *Every prime implicate of a disjunction of concepts is equivalent to some disjunction of prime implicates of its disjuncts.*

In Figure 1, we present the prime implicate generation algorithm GEN-PI from [4, 5] since it will prove useful later in the paper when we design a procedure for putting a concept into prime implicate normal form. We briefly recall the functioning of GEN-PI. In the first step, we check whether the input concept C is unsatisfiable, outputting \perp if this is the case. For satisfiable C , we set \mathcal{T} equal to a set of satisfiable cubal concepts whose disjunction is equivalent to C . We know from the distribution property (Proposition 12) that every prime implicate of C is equivalent to some disjunction of prime implicates of the cubal concepts in \mathcal{T} . In Step (2) we construct a set $\Delta(T)$ of clausal concepts for each $T \in \mathcal{T}$ in such a way that every prime implicate of T is equivalent to some element in $\Delta(T)$. This means that in Step (3) we are guaranteed that every prime implicate of the input concept is equivalent to some candidate prime implicate in *Candidates*. During the comparison phase in Step (4), non-prime candidates are eliminated, and exactly one prime implicate for each equivalence class will be retained.

Proposition 13. *The algorithm GEN-PI always terminates, and it outputs exactly the set of prime implicates of the input concept.*

By examining the structure of the prime implicates generated by GEN-PI, it is possible to place upper bounds on the length of a concept's prime implicates and on the number of prime implicates a concept can possess:

Proposition 14. *The length of the smallest representation of a prime implicate of a concept C cannot exceed $2^n |C|$ where n is the number of distinct literal concepts which are subconcepts of C .*

Proof. Let C be some concept, and let n be the number of distinct literal concepts which are subconcepts of C . We remark that each of the cubal concepts in $\text{DNF}(C)$ is a conjunction of distinct literal subconcepts of C . It follows then that the number of elements in $\text{DNF}(C)$ cannot exceed 2^n and that each element in $\text{DNF}(C)$ has at most n conjuncts. Moreover, the literal concepts which appear as conjuncts in the elements in $\text{DNF}(C)$ all appear in C outside the scope of the role restrictions, which means that the elements in $\text{DNF}(C)$ must all have length of at most $|C|$. But in that case, each of

the concepts in $\Delta(T)$ for some T must have length at most $|C|$ (this follows from the definition of $\Delta(T)$), so each of the concepts in *Candidates* must have length at most $2^n |C|$. As every prime implicate of C is equivalent to some concept in *Candidates*, we can conclude that the length of the smallest representation of a prime implicate of C is bounded above by $2^n |C|$. \square

Proposition 15. *The number of non-equivalent prime implicates of a concept C is bounded above by n^{2^n} where n is the number of literal concepts which are subconcepts of C .*

Proof. We have seen in the proof of Proposition 14 that the number of elements in $\text{DNF}(C)$ cannot exceed 2^n and that each element in $\text{DNF}(C)$ has at most n conjuncts, where n is the number of literal subconcepts of C . As the number of elements in $\Delta(T)$ is bounded by the number of conjuncts of T , it follows that $|\Delta(T)| \leq n$ for all $T \in \text{DNF}(C)$. That means that there are at most n^{2^n} different concepts in *Candidates*, corresponding to the n choices of an element in $\Delta(T)$ for each of the at most 2^n elements in $\text{DNF}(C)$. As every prime implicate of C is equivalent to one of the concepts in *Candidates*, there can be no more than n^{2^n} non-equivalent prime implicates of C . \square

We can also bound the signature and depth of the prime implicates generated by GEN-PI.

Proposition 16. *The signatures of the concepts outputted by GEN-PI are contained in the signature of the inputted concept. The depths of the concepts outputted by GEN-PI are no greater than the depth of the inputted concept. Every literal subconcept which appears behind the role restrictions in some concept outputted by GEN-PI is also a literal subconcept of C .*

Proof. The cubal concepts in $\text{DNF}(C)$ are conjunctions of literal subconcepts of C , so we must have $\text{Sig}(T) \subseteq \text{Sig}(C)$ and $\delta(T) \leq \delta(C)$ for every $T \in \text{DNF}(C)$. Moreover, we know from the definition of $\Delta(T)$ that $\text{Sig}(D) \subseteq \text{Sig}(T) \subseteq \text{Sig}(C)$ and $\delta(D) \leq \delta(T) \leq \delta(C)$ for every $D \in \Delta(T)$. It is also easy to see from the definition that if E is such that $\exists R.E$ or $\forall R.E$ is a conjunct of some element in $\Delta(T)$, then E is a conjunction of subconcepts of C , so each of the literal subconcepts appearing in E must also appear in C . As the concepts of *Candidates* are disjunctions of the elements in the $\Delta(T)$, it follows that the concepts in *Candidates* have signatures contained in $\text{Sig}(C)$ and depths at most $\delta(C)$ and that the literal subconcepts behind their role restrictions also appear in C . This is enough to prove the result since every concept outputted by GEN-PI belongs to *Candidates*. \square

Function GEN-PI(C): returns the set of prime implicates of C

- (1) If C is unsatisfiable, return \perp . Otherwise, set $\mathcal{T} = \text{DNF}(C)$.
- (2) For each $T \in \mathcal{T}$: For each $R \in \text{Roles}_\forall$, set F_R equal to the conjunction of elements in $\forall R(T)$. Set $\Delta(T)$ equal to the set

$$\begin{aligned} & \text{Prop}(T) \cup \{\exists R.(D \sqcap F_R) \mid R \in \text{Roles}_\exists \cap \text{Roles}_\forall \ \& \ D \in \exists R(T)\} \\ & \cup \{\exists R.D \mid R \in \text{Roles}_\exists \ \& \ R \notin \text{Roles}_\forall \ \& \ D \in \exists R(T)\} \cup \{\forall R.F_R \mid R \in \text{Roles}_\forall\} \end{aligned}$$

- (3) Set $\text{Candidates} = \{\sqcup_{T \in \mathcal{T}} G_T \mid G_T \in \Delta(T)\}$.
- (4) For each $Cl_j \in \text{Candidates}$: remove Cl_j from Candidates if $Cl_k \models Cl_j$ for some $Cl_k \in \text{Candidates}$ with $k < j$, or if both $Cl_j \models Cl_k$ and $Cl_k \not\models Cl_j$ for $k > j$.
- (5) Return Candidates .

Function DNF(C): returns a set of cubal concepts whose union is equivalent to the input concept C

- (1) Set $\Sigma = \{\{\text{NNF}(C)\}\}$.
- (2) Apply the following rules until no rule is applicable:
 - \sqcap -rule If $S \in \Sigma$ and $S = \{D_1 \sqcap D_2\} \cup S'$, set $\Sigma = \Sigma \setminus S \cup \{S' \cup \{D_1\} \cup \{D_2\}\}$
 - \sqcup -rule If $S \in \Sigma$ and $S = \{D_1 \sqcup D_2\} \cup S'$, set $\Sigma = \Sigma \setminus S \cup \{S' \cup \{D_1\}\} \cup \{S' \cup \{D_2\}\}$
- (3) Return $\{\prod_{\sigma \in S} \sigma \mid S \in \Sigma \text{ and } S \text{ is consistent}\}$

Figure 1: Algorithm GEN-PI for prime implicate generation and helper function DNF which rewrites concepts as unions of satisfiable cubal concepts. The function NNF which is invoked by DNF is not defined here because it is the standard procedure for putting \mathcal{ALC} concepts into NNF.

4 Prime Implicate Normal Form

In this section, we introduce prime implicate normal form for \mathcal{ALC} concepts, show some of the nice properties it satisfies, and give an algorithm for putting concepts into prime implicate normal form. We also give some results concerning the size of concepts in prime implicate normal form.

4.1 Definition of Prime Implicate Normal Form

In propositional logic, a formula is said to be in prime implicate normal form if it is the conjunction of its prime implicates. Formulae in prime

implicate normal form enjoy a number of interesting properties, which make this normal form useful in variety of AI applications.

Unfortunately, if we extend the definition of prime implicate normal form to \mathcal{ALC} concepts in the obvious way, we find that concepts in prime implicate normal form fail to satisfy most of the nice properties of the propositional case. For example, subsumption between two concepts in prime implicate normal form is no easier than between arbitrary \mathcal{ALC} concepts. To see why, consider any pair of concepts C_1 and C_2 in negation normal form. The concepts $\exists R.C_1$ and $\exists R.C_2$ are their own prime implicates and hence would be in prime implicate normal form if we used the naive definition. As C_1 subsumes C_2 just in the case that $\exists R.C_1$ subsumes $\exists R.C_2$, we can reduce subsumption between arbitrary concepts in NNF to subsumption between concepts in prime implicate normal form. As the former problem is known to be PSPACE-complete (cf. [17]), it follows that the latter is PSPACE-complete as well.

We remark, however, that the problem appears to stem from the fact that the concepts behind the role restrictions are left undecomposed. It seems then that we should require not only that the original concept be represented by its prime implicates but also that the sub-concepts appearing in the prime implicates be themselves represented by their prime implicates. This intuition is at the heart of our definition of prime implicate normal form for \mathcal{ALC} concepts:

Definition 17 (Prime Implicate Normal Form). A concept C is in *prime implicate normal form* if and only if it satisfies one of the following conditions:

1. $C = \perp$
2. $C = \top$
3. $\not\sqsubseteq C \sqsubseteq \perp$ and $\not\sqsubseteq \top \sqsubseteq C$ and $C = Cl_1 \sqcap \dots \sqcap Cl_p$ where
 - (a) $\not\sqsubseteq Cl_i \sqsubseteq Cl_j$ for $i \neq j$
 - (b) each prime implicate of C is equivalent to some conjunct Cl_i
 - (c) every Cl_i is a prime implicate of C such that
 - i. if D is a disjunct of Cl_i , then $\not\sqsubseteq Cl_i \equiv Cl_i \setminus \{D\}$
 - ii. $|\exists R(Cl_i)| \leq 1$ for every role R
 - iii. if $E \in \exists R(Cl_i) \cup \forall R(Cl_i)$ for some R , then E is in prime implicate normal form

iv. if $E \in \exists R(Cl_i)$ and $F \in \forall R(Cl_i)$, then $\models F \equiv F \sqcup E$

Let us briefly go over the different points of the definition. The first two items state that all unsatisfiable concepts must be represented as \perp and all tautologous concepts must be represented as \top . All other concepts are to be represented by a conjunction of their prime implicates, but we place some strong restrictions on how the prime implicates themselves are represented. First, we require that they contain no unnecessary disjuncts (part (i) of 3c). We also stipulate that they contain at most one existential restriction per role (part (ii)) and that the concepts appearing behind the existential and universal restrictions be themselves in prime implicate normal form (part (iii)). Finally, we demand that if a prime implicate contains disjuncts $\exists R.E$ and $\forall R.F$ then E and F are such that $\models E \sqsubseteq F$ (part (iv)). This requirement may seem a little less intuitive than the others, but it ensures that if a universal restriction is subsumed by a clausal concept, then it is subsumed by some universal restriction appearing in the clausal concept¹. This property is crucial since it allows our subsumption algorithm to treat universal restrictions separately from the existential restrictions.

We will show later in the paper that this definition is well-founded by proving that every concept can be rewritten as an equivalent concept in prime implicate normal form. We first motivate the interest of doing so by exhibiting some of the desirable properties of concepts in prime implicate normal form.

4.2 Properties of Prime Implicate Normal Form

In this section, we show that prime implicate normal form has some nice properties which make it an interesting target language for knowledge compilation.

4.2.1 Tractable subsumption

The most important criterion when choosing a normal form for compilation is the set of polynomial time queries that the normal form supports. In [8], the authors enumerate a set of queries which they then use to compare different normal forms for propositional logic. Of the eight queries they consider, only four are well-defined for \mathcal{ALC}^2 : satisfiability-testing, tautology-testing,

¹This does not hold in general: $\models \forall R.A \sqsubseteq \exists R.A \sqcup \forall R.B$ but $\not\models \forall R.A \sqsubseteq \forall R.B$.

²For example, clausal entailment is ill-defined since there are many possible definitions of clauses in \mathcal{ALC} , and model counting makes little sense since every concept has infinitely many distinct models.

subsumption, and equivalence-testing. We show that for concepts in prime implicate normal form, all four queries are computable in polynomial time.

For satisfiability and tautology-testing, there is really nothing to prove since by definition a concept C in prime implicate normal form is unsatisfiable just in the case that $C = \perp$ and is tautologous just in the case that $C = \top$. It follows that these tasks can be carried out in constant-time.

For subsumption and equivalence, we provide in Figure 2 a structural subsumption algorithm Π -SUBSUME which decides subsumption between concepts in prime implicate normal form. Let us explain briefly the functioning of Π -SUBSUME. The first two steps treat limit cases where one or both of the concepts is unsatisfiable or tautologous. For all other pairs of concepts, we proceed to Step 3, in which we perform a structural comparison of the two concepts. We know from the equivalence property (Proposition 11) that a concept C_1 is subsumed by a concept C_2 just in the case that C_1 is subsumed by each of the prime implicates of C_2 . Moreover, it follows from the covering property (Proposition 10) that C_1 is subsumed by a prime implicate D of C_2 if and only if some prime implicate of C_1 is subsumed by D . As concepts in prime implicate normal form are conjunctions of their prime implicates, testing whether C_2 subsumes C_1 comes down to testing whether each conjunct of C_2 subsumes some conjunct of C_1 . If we hadn't placed any requirements on the form of the conjuncts, then this problem would be as hard as subsumption in general. But since C_1 and C_2 are in prime implicate normal form, their conjuncts have a particular structure which makes subsumption easy to test. We first check that the propositional literals in the first conjunct all appear in the second conjunct. We then call Π -SUBSUME on sub-concepts appearing in the two conjuncts in order to ensure that each of the existential and universal restrictions appearing in the first conjunct is subsumed by an existential or universal restriction in the second. The algorithm performs these checks on each possible pair of conjuncts and returns **no** if it finds some conjunct of C_2 which does not subsume any conjunct of C_1 . If no such conjunct is found, the algorithm returns **yes** since every conjunct of C_2 has been shown to subsume a conjunct of C_1 , which means that C_2 subsumes C_1 .

Lemma 18. *If C_1 and C_2 are both in prime implicate normal form, then the algorithm Π -SUBSUME outputs **yes** on input (C_1, C_2) if $\models C_1 \sqsubseteq C_2$.*

Proof. The proof is by induction on $\min(\delta(C_1), \delta(C_2))$. We begin with the base case where $\models C_1 \sqsubseteq C_2$ and $\min(\delta(C_1), \delta(C_2)) = 0$, i.e. where one or both of C_1 and C_2 is propositional. There are three possibilities: either $\models C_1 \sqsubseteq \perp$, or $\models \top \sqsubseteq C_2$, or neither $\models C_1 \sqsubseteq \perp$ nor $\models \top \sqsubseteq C_2$. In the first

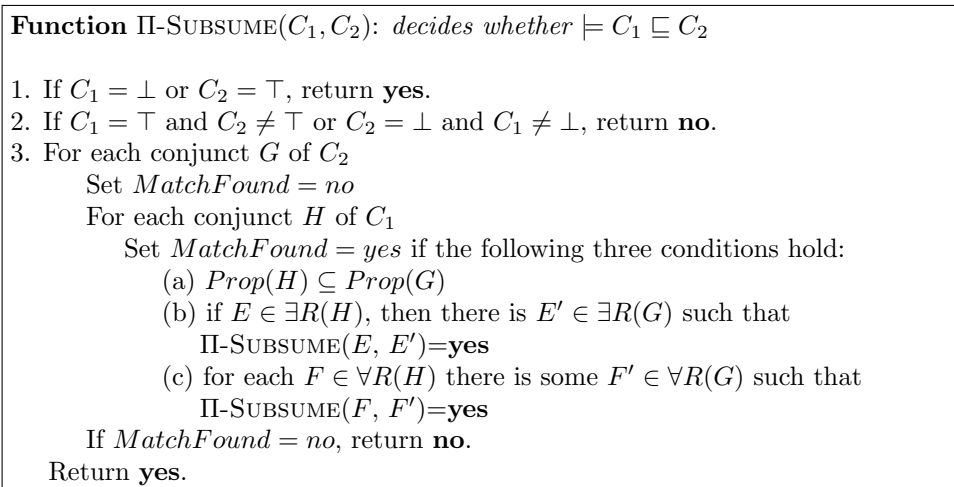


Figure 2: Algorithm for deciding subsumption between concepts in prime implicate normal form.

case, C_1 must be \perp (otherwise C_1 would not be in prime implicate normal form), so the algorithm will return **yes** in Step 1. Similarly, in the second case, we must have $C_2 = \top$, so the algorithm returns **yes** in the first step.

Let us then concentrate on the third case in which $\not\models C_1 \sqsubseteq \perp$ and $\not\models \top \sqsubseteq C_2$. Since $\models C_1 \sqsubseteq C_2$, it follows that we must also have $\not\models C_2 \sqsubseteq \perp$ and $\not\models \top \sqsubseteq C_1$. This means that the conditions for Steps 1 and 2 of $\Pi\text{-SUBSUME}$ are not satisfied, so we will proceed to Step 3. Now since $\models C_1 \sqsubseteq C_2$, it must be the case that C_1 is subsumed by every conjunct of C_2 . As the conjuncts of C_2 are all clausal concepts (since C_2 is in prime implicate normal form), it follows from Proposition 10 that every conjunct in C_2 subsumes some prime implicate of C_1 . But since C_1 is in prime implicate normal form, every prime implicate of C_1 is equivalent to some conjunct of C_1 . This means that for every conjunct G of C_2 there must be some conjunct H of C_1 such that $\models H \sqsubseteq G$. If C_1 is propositional, then so are all its conjuncts, so $\models H \sqsubseteq G$ just in the case that $\text{Prop}(H) \subseteq \text{Prop}(G)$ (by Proposition 7). It follows that when the algorithm considers the conjuncts G and H , it will set *MatchFound* = *yes*. If instead it is C_2 which is propositional, then G is also propositional, so every disjunct of H must be either a propositional literal which belongs to G or a literal concept of the form $\exists R.D$ where D is unsatisfiable (otherwise we would not have $\models H \sqsubseteq G$). But since H is in prime implicate normal form it cannot have any unsatisfiable disjuncts, so

H must be composed only of propositional literals which appear in G . This means that the algorithm will mark $MatchFound = yes$ when considering the pair of concepts G and H . Thus, in either case, we have that for each conjunct G of C_2 , there is some conjunct H of C_1 for which we will mark $MatchFound = yes$, so Π -SUBSUME will return **yes**.

We have just shown that Π -SUBSUME returns **yes** whenever the input concepts C_1 and C_2 are such that $\models C_1 \sqsubseteq C_2$ and $\min(\delta(C_1), \delta(C_2)) = 0$. Now let us suppose that the result holds whenever we $\min(\delta(C_1), \delta(C_2)) \leq k$ and then show that the result still holds when the minimum depth is $k + 1$.

Let C_1 and C_2 be concepts in prime implicate normal form such that $\models C_1 \sqsubseteq C_2$ and $\min(\delta(C_1), \delta(C_2)) = k + 1$. As C_1 and C_2 both have positive depth, it follows that they can be neither unsatisfiable nor tautologous (since in that case they would be equal to either \perp or \top , both of which have depth zero). That means that the algorithm will proceed directly to Step 3. Let G be some conjunct of C_2 . Now since $\models C_1 \sqsubseteq C_2$, we must have $\models C_1 \sqsubseteq G$. But since C_2 is in prime implicate normal form, G must be a clausal concept, and so Proposition 10 tells us that there is some prime implicate P of C_1 such that $\models P \sqsubseteq G$. The concept C_1 is also in prime implicate normal form, so there must be some conjunct H of C_1 such that $\models P \equiv H$ and hence such that $\models H \sqsubseteq G$. As G and H are both clausal concepts, and G is non-tautologous, by Proposition 7 we must have:

- (a) $Prop(H) \subseteq Prop(G)$
- (b) If $R \in Roles_{\exists}(H)$, then $\models \bigsqcup_{E \in \exists R(H)} E \sqsubseteq \bigsqcup_{F \in \exists R(G)} F$ (or just $\models \bigsqcup_{E \in \exists R(H)} E \sqsubseteq \perp$ if $R \notin Roles_{\exists}(G)$)
- (c) If $E \in \forall R(H)$, then there is some $F \in \forall R(G)$ such that $\models E \sqsubseteq F \sqcup (\bigsqcup_{J \in \exists R(G)} J)$ (or $\models E \sqsubseteq F$ if $R \notin Roles_{\exists}(G)$)

Statement (a) means that the first condition of the algorithm is satisfied for the pair G and H . As for the second condition, let us suppose that $R \in Roles_{\exists}(H)$. As C_1 is in prime implicate normal form, there must be exactly one element in $\exists R(H)$ and this element must be satisfiable (otherwise H would contain an unnecessary disjunct). Let E be this concept. Now because of (b) and the fact E is satisfiable, $\exists R(G)$ must be non-empty and E must be subsumed by the disjunction of the elements in $\exists R(G)$. But C_2 is also in prime implicate normal form, so there must be a single element in $\exists R(G)$, call it E' . We thus have $\models E \sqsubseteq E'$. Because C_1 and C_2 are concepts in prime implicate normal form with $\min(\delta(C_1), \delta(C_2)) = k + 1$, it follows that E and E' are also in prime implicate normal form and $\min(\delta(E), \delta(E')) \leq k$. This

means the induction hypothesis applies, so $\Pi\text{-SUBSUME}(E, E') = \mathbf{yes}$, and hence the second condition of the algorithm is satisfied for the pair G and H . Finally, we remark that because of (c) and conditions 3(b)ii and 3(b)iv of Definition 17 (which apply to G and H since we have assumed C_1 and C_2 are in prime implicate normal form) it follows that for each disjunct $\forall R.F$ of H there is some disjunct $\forall R.F'$ of G such that $\models F \sqsubseteq F'$. Now F and F' are concepts in prime implicate normal form (by 3(b)iii of Definition 17) such that $\min(\delta(E), \delta(E')) \leq k$ and $\models F \sqsubseteq F'$, so according to the induction hypothesis, it must be the case that $\Pi\text{-SUBSUME}(F, F') = \mathbf{yes}$. This means that G and H satisfy the third and final condition of the algorithm. We have thus shown that for every conjunct G of C_2 there is some conjunct H of C_1 such that the three conditions of Step 3 are satisfied, so the algorithm will return **yes** on input (C_1, C_2) . \square

Lemma 19. *If C_1 and C_2 are both in prime implicate normal form, then the algorithm $\Pi\text{-SUBSUME}$ outputs **no** on input (C_1, C_2) if $\not\models C_1 \sqsubseteq C_2$.*

Proof. The proof is by induction on $\min(\delta(C_1), \delta(C_2))$. We begin with the base case where $\not\models C_1 \sqsubseteq C_2$ and $\min(\delta(C_1), \delta(C_2)) = 0$, i.e. where one or both of C_1 and C_2 is propositional. If $\models \top \sqsubseteq C_1$ and $\not\models \top \sqsubseteq C_2$, then $C_1 = \top$ and $C_2 \neq \top$ since C_1 and C_2 are assumed to be in prime implicate normal form), so the algorithm will return **no** in the second step. Likewise, if $\models C_2 \sqsubseteq \perp$ and $\not\models C_1 \sqsubseteq \perp$, then we must have $C_1 \neq \perp$ and $C_2 = \perp$, so the algorithm returns **no** in Step 2. If neither of these cases holds, then C_1 and C_2 must both be satisfiable and non-tautologous, so the algorithm proceeds to Step 3. As $\not\models C_1 \sqsubseteq C_2$, it must be the case that there is some conjunct G of C_2 such that $\not\models H \sqsubseteq G$ for every conjunct H of C_1 . If it is C_1 that is propositional, then it follows from Proposition 7 that $\text{Prop}(H) \not\subseteq \text{Prop}(G)$ for every conjunct H of C_1 . If it is C_2 that is propositional, then for each conjunct H of C_1 either $\text{Prop}(H) \not\subseteq \text{Prop}(G)$ or H contains universal or existential restrictions. In either case, we find that each conjunct H of C_1 violates at least one of the conditions in Step 3. This means that algorithm does not set $\text{MatchFound} = \text{yes}$ when examining the conjunct G and hence returns **no**.

We have thus shown that $\Pi\text{-SUBSUME}$ returns **no** whenever C_1 and C_2 are concepts in prime implicate normal form such that $\not\models C_1 \sqsubseteq C_2$ and $\min(\delta(C_1), \delta(C_2)) = 0$. We will now suppose that the same statement holds whenever $\min(\delta(C_1), \delta(C_2)) \leq k$ and will show that the result remains true when the minimal depth is $k + 1$.

Let C_1 and C_2 be concepts in prime implicate normal form such that $\not\models C_1 \sqsubseteq C_2$ and $\min(\delta(C_1), \delta(C_2)) = k + 1$. Since C_1 and C_2 are in prime

implicate normal form and have positive depth, C_1 and C_2 can be neither unsatisfiable nor tautologous, so the algorithm proceeds directly to Step 3. As $\not\sqsubseteq C_1 \sqsubseteq C_2$, there must be some conjunct G of C_2 such that $\not\sqsubseteq H \sqsubseteq G$ for every conjunct H of C_1 . According to Proposition 7, this means that for every conjunct H of C_1 we have one of the following:

- (a) $Prop(H) \not\sqsubseteq Prop(G)$
- (b) $R \in Roles_{\exists}(H)$ and either $R \notin Roles_{\exists}(G)$ and $\not\sqsubseteq \bigsqcup_{E \in \exists R(H)} E \sqsubseteq \perp$ or $R \in Roles_{\exists}(G)$ and $\not\sqsubseteq \bigsqcup_{E \in \exists R(H)} E \sqsubseteq \bigsqcup_{E \in \exists R(G)} E$
- (c) For some $E \in \forall R(H)$, there is no $F \in \forall R(G)$ such that $\models E \sqsubseteq F \sqcup (\bigsqcup_{J \in \exists R(G)} J)$ (or $\models E \sqsubseteq F$ if $R \notin Roles_{\exists}(G)$)

If (a) holds, then the first condition of Step 3 is violated. If (b) holds, then either $\exists R(G) = \emptyset$ or $\not\sqsubseteq E \sqsubseteq E'$, where $E \in \exists R(H)$ and $E' \in \exists R(G)$ (remember that since C_1 and C_2 are in prime implicate normal form, the clausal concepts G and H can have at most one existential restriction per role). In the first case, the second condition of Step 3 is violated since $\exists R(G)$ is empty. In the second case, the condition is also violated since E and E' are concepts in prime implicate normal form such that $\not\sqsubseteq E \sqsubseteq E'$ and $\min(\delta(E), \delta(E')) = k$, so according to the induction hypothesis $\Pi\text{-SUBSUME}(E, E') = \mathbf{no}$. Finally, if (c) holds, then for some disjunct $\forall R.F$ of H and every disjunct $\forall R.F'$ of G we have $\not\sqsubseteq F \sqsubseteq F' \sqcup J$ where $J \in \exists R(G)$ (or simply $\not\sqsubseteq F \sqsubseteq F'$ if $\exists R(G)$ is empty). But since G is in prime implicate normal form, if $J \in \exists R(G)$ then $\models F \equiv F' \sqcup J$. So we get that $\not\sqsubseteq F \sqsubseteq F'$, and hence by the induction hypothesis (which applies since F and F' are in prime implicate normal form and $\min(\delta(F), \delta(F')) = k$) that $\Pi\text{-SUBSUME}(F, F')$ returns \mathbf{no} . We have thus shown that for every conjunct H of C_1 at least one of the three conditions of Step 3 will not be satisfied for the pair G and H . This means that when the algorithm has finished its examination of the conjunct G , the variable *MatchFound* will still be set to *no*, so $\Pi\text{-SUBSUME}$ will return \mathbf{no} . \square

Lemma 20. *The algorithm $\Pi\text{-SUBSUME}$ terminates in linear time in $|C_1| |C_2|$ (hence at most quadratic time in $|C_1| + |C_2|$) when given concepts C_1 and C_2 as input.*

Proof. The algorithm $\Pi\text{-SUBSUME}$ compares at most once each pair of symbols from C_1 and C_2 , and the comparison takes constant time, yielding an overall complexity which is linear in $|C_1| |C_2|$. \square

Proposition 21. *Subsumption of concepts in prime implicate normal form can be decided in quadratic time in the size of the input.*

Proof. Direct consequence of Lemmas 18, 19, and 20. □

Corollary 22. *Equivalence of concepts in prime implicate normal form can be decided in quadratic time in the size of the input.*

Our subsumption algorithm requires that both concepts be in prime implicate normal form. However, it is not always necessary for the second concept to be in prime implicate normal form to obtain polynomial time subsumption, as the following proposition demonstrates:

Proposition 23. *Let C be a concept in prime implicate normal form, and let D be a disjunction of atomic literal concepts and concepts of the form $\exists R.Cl$ or $\forall R.Cl$ where Cl is a propositional clausal concept. Then it can be decided in linear time in $|C|$ whether $\models C \sqsubseteq D$.*

Proof. Consider the following procedure:

Step 1 Apply the following modifications to D :

- (a) For each R : if $\exists R(D) = \{E_1, \dots, E_m\}$ where $m > 1$, replace D by $D \setminus \{\exists R.E_1, \dots, \exists R.E_m\} \cup \{\exists R.\top\}$ if $E_1 \sqcup \dots \sqcup E_m$ is a tautologous propositional clause and by $D \setminus \{\exists R.E_1, \dots, \exists R.E_m\} \cup \{\exists R.(E_1 \sqcup \dots \sqcup E_m)\}$ otherwise.
- (b) For each R : if $\exists R(D) = \{E\}$ and $\forall R(P) = \{F_1, \dots, F_n\}$, replace D by $D \setminus \{\forall R.F_1, \dots, \forall R.F_n\} \cup \{\forall R.(F_1 \sqcup E), \dots, \forall R.(F_n \sqcup E)\}$.
- (c) If $Prop(D)$ contains two complementary atomic literal concepts, or if there is some disjunct $\forall R.Cl$ where Cl is a tautologous propositional clause, replace D by \top .
- (d) Remove any unnecessary disjuncts from D and from the clausal concepts appearing in the role restrictions of D .

Step 2 Return $\Pi\text{-SUBSUME}(C, D)$

We claim that if C and D are as described in the statement of the proposition, then $\models C \sqsubseteq D$ if and only if the above procedure returns **yes**. We first consider the case where $\models \top \sqsubseteq D$. There are two ways this can happen: either the propositional part of D is tautologous, in which case the propositional part contains two opposing atomic literal concepts, or there is some universal role restriction $\forall R.E$ and existential restrictions $\exists R.F_1, \dots$,

$\exists R.F_n$ such that $\models \top \sqsubseteq E \sqcup F_1 \sqcup \dots \sqcup F_n$. In the first case, we find that $Prop(D)$ contains two opposing atomic literal concepts, so we will change D to \top in Step 1 and return **yes** in Step 2 because Π -SUBSUME always returns **yes** when the second concept is \top . In the second case, there is either a pair $\forall R.E$ and $\exists R.F$ or a pair $\exists R.E$ and $\exists R.F$ of disjuncts of D such that $E \sqcup F$ contains either \top or a pair of atomic literal concepts. But in that case, we will have transformed D into \top in Step 1, either in part (a) if the pair consists of two existential restrictions or in part (c) if a universal restriction is involved. This means that the algorithm will return **yes** in Step 2. Now consider the case where D is not a tautology. It can be easily verified that at the end of Step 1 the concept D is in prime implicate normal form: D has no redundant literals, there is at most one existential restriction per role, the concepts behind the role restrictions are also in prime implicate normal form (since they are propositional clauses with no repeated literals), and the concepts behind the existential restrictions have been added to the concepts behind the universal restrictions. Note that all the modifications in Step 1 are equivalence-preserving so C is subsumed by the original D just in the case it is subsumed by the modified D . In Step 2, we call Π -SUBSUME to decide whether C is subsumed by D . Because of Proposition 21 we know that Π -SUBSUME(C, D) returns **yes** just in the case that D subsumes C , so the answer returned in Step 2 will be correct.

The only step of the procedure which concerns C is Step 2 in which we call Π -SUBSUME to test whether the modified concept D subsumes C . We know from Lemma 20 that the Π -SUBSUME terminates in time which is linear in $|C|$. \square

If we are only concerned by the complexity with respect to the size of the first input concept, then we could extend the algorithm in the previous proof in order to treat the entire class of \mathcal{ALC} concepts in NNF which do not contain conjunction. Unfortunately, we find that the complexity in the second argument is no longer polynomial. The reason for this is the problem of deciding whether an arbitrary \mathcal{ALC} concept in NNF without disjunction is unsatisfiable is an NP-complete task [9], which means that the dual problem of deciding whether an arbitrary \mathcal{ALC} concept in NNF without conjunction is a tautology must also be NP-complete. Whereas it seems rather natural to measure the complexity in just the first argument when the second “query” argument is assumed to have significantly smaller size, this no longer makes sense when the complexity in the second argument is so much higher than the complexity in the first argument.

We also point out that the above category of concepts is just one example

of a tractable class of query concepts. There are a variety of other syntactic conditions which could be placed on query concepts in order to guarantee polynomial subsumption.

4.2.2 Polynomial Transformations

As we noted earlier, the notions of \mathcal{L} - and n -interpolants correspond to the finest approximations of concepts over a given signature or up to a specified depth. \mathcal{L} - and n -interpolants allow us to focus in on just part of a concept, which may prove useful when a concept description is too large to be fully understood or when data needs to be exchanged between systems using different signatures. We show in this subsection that it is easy to transform concepts in prime implicate normal form into their \mathcal{L} - and n -interpolants.

In Figure 3, we present an algorithm for computing the \mathcal{L} -interpolant of a concept in prime implicate normal form.

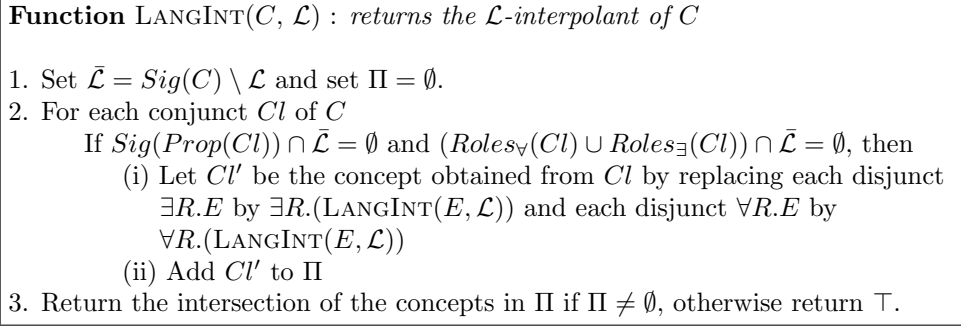


Figure 3: Algorithm for generating the \mathcal{L} -interpolants of concepts in prime implicate normal form.

Lemma 24. *If C is a concept in prime implicate normal form, then the output of LANGINT(C, \mathcal{L}) is an \mathcal{L} -interpolant of C .*

Proof. The proof is by induction on the depth of the input concept C . The base case is when $\delta(C) = 0$, i.e. when C is a propositional concept. In this case, the algorithm simply returns the intersection of the conjuncts of C whose signatures are contained in \mathcal{L} , or \top if there are no such concepts. Let D be some concept such that $\text{Sig}(D) \subseteq \mathcal{L}$ and $\models C \sqsubseteq D$. Because of Proposition 5 we can suppose without loss of generality that D is a conjunction of clausal concepts. We know from Proposition 10 that if a clausal concept Cl subsumes C , then there is some prime implicate of C , hence some conjunct

of C (since C is in prime implicate normal form), which is subsumed by Cl . It follows that every conjunct of D subsumes some conjunct of C . We remark that C is a conjunction of propositional clausal concepts and that a propositional clausal concept containing atomic concepts outside \mathcal{L} cannot be subsumed by a non-tautologous concept with signature contained in \mathcal{L} . This means that if there are no conjuncts of C with signature contained in \mathcal{L} , then D must be a tautologous concept, and if such conjuncts exist, then each of the conjuncts in D subsumes at least one such conjunct. In the first case, we find that D subsumes \top , and in the second case, D subsumes the intersection of the conjuncts of C whose signatures are contained in \mathcal{L} . In both cases, we find that the output of $\text{LANGINT}(C, \mathcal{L})$ is a concept with signature in \mathcal{L} which subsumes C and is subsumed by every concept D in \mathcal{L} which subsumes C , so it must be the \mathcal{L} -interpolant of C .

Let us next assume that the result holds for every concept in prime implicate normal form with depth at most k and show that the result still holds for concepts with depth $k + 1$. Our first step will be to show that the following statements hold for every clausal concept Cl in prime implicate normal form with depth at most $k + 1$:

1. If Cl does not satisfy the conditions in Step 2 of LANGINT , then the \mathcal{L} -interpolant of Cl is \top
2. If Cl does satisfy the conditions in Step 2 of LANGINT , then the concept obtained from Cl by replacing disjuncts of the form $\exists R.E$ and $\forall R.E$ by $\exists R.(\text{LANGINT}(E, \mathcal{L}))$ and $\forall R.(\text{LANGINT}(E, \mathcal{L}))$ is the \mathcal{L} -interpolant of Cl

We begin with statement (1). Let Cl be a clausal concept in prime implicate normal form which does not satisfy the conditions in Step 2. There are two possibilities: either $\text{Sig}(\text{Prop}(Cl)) \cap \bar{\mathcal{L}} \neq \emptyset$ or $(\text{Roles}_{\forall}(Cl) \cup \text{Roles}_{\exists}(Cl)) \cap \bar{\mathcal{L}} \neq \emptyset$. If $\text{Sig}(\text{Prop}(Cl)) \cap \bar{\mathcal{L}} = \emptyset$, then there is some atomic concept $A \notin \mathcal{L}$ such that A or $\neg A$ is a disjunct of Cl . But we know that the only concepts with signature in \mathcal{L} which subsume $(\neg)A$ are tautologous concepts, so Cl cannot be subsumed by any non-tautologous concepts with signature contained in \mathcal{L} , i.e. \top is the \mathcal{L} -interpolant of Cl . If instead we have $(\text{Roles}_{\forall}(Cl) \cup \text{Roles}_{\exists}(Cl)) \cap \bar{\mathcal{L}} = \emptyset$, then there is some satisfiable disjunct $\exists R.E$ or $\forall R.E$ of Cl with $R \notin \mathcal{L}$. But a satisfiable concept $\exists R.E$ or $\forall R.E$ cannot imply any non-tautologous concepts which do not contain R by Proposition 7. It follows that every concept which subsumes Cl and has signature contained in \mathcal{L} is tautologous, so \top is the \mathcal{L} -interpolant of Cl .

We now show (2). Let Cl be a clausal concept in prime implicate normal form of depth at most $k + 1$ such that $Sig(Prop(Cl)) \cap \bar{\mathcal{L}} = \emptyset$ and $(Roles_{\forall}(Cl) \cup Roles_{\exists}(Cl)) \cap \bar{\mathcal{L}} = \emptyset$, and let Cl' be the concept obtained from Cl by replacing disjuncts of the form $\exists R.E$ and $\forall R.E$ respectively by $\exists R.(LANGINT(E, \mathcal{L}))$ and $\forall R.(LANGINT(E, \mathcal{L}))$. We remark that Cl' has the same propositional disjuncts as Cl , and its existential and universal disjuncts have the same roles as those in Cl . We also note that the concepts appearing behind the universal and existential role restrictions have the form $LANGINT(E, \mathcal{L})$ where E is a concept in prime implicate normal form with depth at most k . Applying the induction hypothesis, we find that for each such concept E , $LANGINT(E, \mathcal{L})$ is the \mathcal{L} -interpolant of E . In particular, that means that $LANGINT(E, \mathcal{L})$ has signature contained in \mathcal{L} . It follows that Cl' also has signature contained in \mathcal{L} . It also means that each E is subsumed by $LANGINT(E, \mathcal{L})$, from which can deduce that Cl' subsumes Cl . We now need to show that Cl' is subsumed by every concept which subsumes Cl and has signature in \mathcal{L} . Let D be such a concept. Clearly every propositional disjunct in Cl' must be subsumed by D since Cl and Cl' have the same propositional disjuncts and $\models Cl \sqsubseteq D$. Every existential disjunct of Cl' is equal to $\exists R.(LANGINT(E, \mathcal{L}))$ for some disjunct $\exists R.E$ of Cl . As $\exists R.(LANGINT(E, \mathcal{L}))$ is the \mathcal{L} -interpolant of $\exists R.E$, it follows that $\exists R.(LANGINT(E, \mathcal{L}))$ must be subsumed by D since D subsumes $\exists R.E$ and $Sig(D) \subseteq \mathcal{L}$. It follows that every existential disjunct of Cl' must be subsumed by D . Likewise, we remark that $\forall R.(LANGINT(E, \mathcal{L}))$ is the \mathcal{L} -interpolant of $\forall R.E$, so every universal restriction which is a disjunct of Cl' must be subsumed by D . As every disjunct of Cl' is subsumed by D , it follows that Cl' must subsumed by D as well, so Cl' is the \mathcal{L} -interpolant of Cl . This together with statement (1) tells us that the output of $LANGINT(C, k + 1)$ is equivalent to the intersection of the \mathcal{L} -interpolants of the conjuncts of C .

Now let G be a concept such that $Sig(G) \subseteq \mathcal{L}$ and $\models C \sqsubseteq G$. We can assume without loss of generality that G is a conjunction of clausal concepts since any concept is equivalent to some concept of this form and with the same or smaller signature (Proposition 5). Then since C is subsumed by G , it follows that C is subsumed by each of the clausal concepts which are conjuncts of G . By the Covering Property (Proposition 10), we know that every clausal concept which subsumes C subsumes some prime implicate of C . As we have assumed C to be in prime implicate normal form, this means that every conjunct of G subsumes some conjunct of C . This together with the fact that the conjuncts of G have signature contained in \mathcal{L} means that each of the conjuncts of G subsumes the \mathcal{L} -interpolant of some conjunct of

C . It follows then that the concept G subsumes the intersection of the \mathcal{L} -interpolants of the conjuncts of C . But then G must subsume the output of $\text{LANGINT}(C, \mathcal{L})$ since we have shown above that the output of $\text{LANGINT}(C, \mathcal{L})$ is precisely the intersection of the \mathcal{L} -interpolants of the conjuncts of C . We have thus demonstrated that the output of $\text{LANGINT}(C, \mathcal{L})$ is a concept with signature in \mathcal{L} which subsumes C and is subsumed by every concept with signature in \mathcal{L} which subsumes C , i.e. the output of $\text{LANGINT}(C, \mathcal{L})$ is the \mathcal{L} -interpolant of C . \square

Lemma 25. *The algorithm LANGINT runs in linear time in the size of the input concept.*

Proof. LANGINT terminates in linear time with respect to the size of the input concept because all the algorithm does is scan the input concept a single time in order to remove those disjunctive sub-concepts which violate the syntactic requirements set forth in Step 2. \square

Proposition 26. *The \mathcal{L} -interpolant of a concept C in prime implicate normal form can be generated in linear time in the size of C .*

Proof. Follows directly from Lemmas 24 and 25. \square

In Figure 4, we introduce an algorithm DEPTHINT for generating the n -interpolant of a concept in prime implicate normal form.

Function $\text{DEPTHINT}(C, n)$: *returns the n -interpolant of C*

1. If $n = 0$, return the intersection of the propositional conjuncts of C if such conjuncts exist, and \top otherwise. If $n > 0$, set $\Pi = \emptyset$.
2. For each conjunct Cl of C , add to Π the concept obtained from Cl by replacing each disjunct $\exists R.E$ by $\exists R.(\text{DEPTHINT}(E, n - 1))$ and each disjunct $\forall R.E$ by $\forall R.(\text{DEPTHINT}(E, n - 1))$.
3. Return the intersection of concepts in Π .

Figure 4: Algorithm for generating the n -interpolants of concepts in prime implicate normal form.

Lemma 27. *If C is a concept in prime implicate normal form, then the output of $\text{DEPTHINT}(C, n)$ is the n -interpolant of C .*

Proof. The proof is by induction on the value of n . The base case is when $n = 0$. In this case, the algorithm returns the conjunction of the propositional concepts which are conjuncts of C , or \top if there are no such concepts. As C is assumed to be in prime implicate normal form, we know that the clausal concepts which are conjuncts of C do not contain any unsatisfiable $\exists R$ disjuncts. This means that every propositional prime implicate of C is equivalent to some propositional conjunct of C . It follows that if there are no propositional conjuncts, then there are no propositional prime implicates, so the only concepts of depth 0 which subsume C are tautologous concepts. In this case, the result holds since \top is a concept of depth 0 which is subsumed by all tautologous concepts, i.e. \top is the 0-interpolant of C . Let us next consider the case where C contains at least one propositional conjunct. We know from Proposition 10 that if a propositional clausal concept Cl subsumes C , then there is some prime implicate Cl' of C which is subsumed by Cl . Moreover, we can assume that Cl' is propositional, since any clausal concept which is subsumed by a propositional clausal concept is itself equivalent to a propositional clausal concept. But we have shown above that every propositional prime implicate of C is equivalent to some propositional conjunct of C , so every propositional implicate Cl of C must be subsumed by some propositional conjunct of C . As every concept of depth 0 is equivalent to a conjunction of propositional clausal concepts, it follows that for every concept D of depth 0, if $\models C \sqsubseteq D$, then D subsumes the conjunction of the propositional conjuncts in C . We have thus shown that the conjunction of propositional conjuncts of C is a concept of depth 0 which subsumes C and is subsumed by every concept of depth 0 which subsumes C , so it is the 0-interpolant of C .

Let us now assume that the result holds whenever $n \leq k$. We will begin by proving the following: if Cl is a clausal concept in prime implicate normal form, then the concept obtained from Cl by replacing disjuncts of the form $\exists R.E$ and $\forall R.E$ by $\exists R.DEPTHINT(E, k)$ and $\forall R.DEPTHINT(E, k)$ is the $k+1$ -interpolant of Cl . Let Cl be a clausal concept in prime implicate normal form, let Cl' be the clausal concept obtained from Cl by replacing disjuncts $\exists R.E$ and $\forall R.E$ by $\exists R.DEPTHINT(E, k)$ and $\forall R.DEPTHINT(E, k)$, and let D be some clausal concept of depth at most $k+1$ such that $\models Cl \sqsubseteq D$. Now every propositional disjunct in Cl' must be subsumed by D since Cl and Cl' have the same propositional disjuncts and $\models Cl \sqsubseteq D$. Every existential disjunct of Cl' is equal to $\exists R.DEPTHINT(E, k)$ for some disjunct $\exists R.E$ of Cl . It follows from the induction hypothesis that $\exists R.DEPTHINT(E, k)$ is the $k+1$ -interpolant of $\exists R.E$, which means that $\exists R.DEPTHINT(E, k)$ must be subsumed by D since D subsumes $\exists R.E$ and $\delta(D) \leq k+1$. It follows

that every existential disjunct of C' must be subsumed by D . Similarly, we can see that $\forall R.\text{DEPTHINT}(E, k)$ is the $k + 1$ -interpolant of $\forall R.E$, so every universal restriction in C' must be subsumed by D . As every disjunct of C' is subsumed by D , it follows that C' is subsumed by D too. We have thus shown that modifying a clausal concept C in prime implicate normal form so that disjuncts $\exists R.E$ and $\forall R.E$ are replaced by $\exists R.\text{DEPTHINT}(E, k)$ and $\forall R.\text{DEPTHINT}(E, k)$ yields the $k + 1$ -interpolant of C . As the conjuncts of C are clausal concepts in prime implicate normal form, it follows that the set of concepts in Π at the end of Step 2 is precisely the set of $k + 1$ -interpolants of the conjuncts of C .

Now let G be a concept of depth at most $k + 1$ such that $\models C \sqsubseteq G$. We can assume without loss of generality that G is a conjunction of clausal concepts since any concept is equivalent to some concept of this form and with less or equal depth (Proposition 5). It follows that C is subsumed by each of the clausal concepts which are conjuncts of G . By Proposition 10 we know that every clausal concept which subsumes C subsumes some prime implicate of C , and hence subsumes some conjunct of C (because C is assumed to be in prime implicate normal form). But since all the conjuncts of G have depth at most $k + 1$, it follows that each of them subsumes the $k + 1$ -interpolant of some conjunct of C . This means that the concept G subsumes the intersection of the $k + 1$ -interpolants of the conjuncts of C . As we have shown above that the output of $\text{DEPTHINT}(C, k + 1)$ is exactly the intersection of the $k + 1$ -interpolants of the conjuncts of C , it follows that G subsumes the output of $\text{DEPTHINT}(C, k + 1)$. We have thus shown that the output of $\text{DEPTHINT}(C, k + 1)$ is a concept with depth at most $k + 1$ which subsumes C and which is subsumed by every concept of depth at most $k + 1$ which subsumes C . In other words, the output of $\text{DEPTHINT}(C, k + 1)$ is the $k + 1$ -interpolant of C . \square

Lemma 28. *The algorithm DEPTHINT runs in linear time in the size of the input concept.*

Proof. DEPTHINT scans through the input concept C to locate those concepts appearing at depth n in C and to remove their non-propositional conjuncts. As the algorithm makes a single pass through C , it terminates in linear time in the length of C . \square

Proposition 29. *The n -interpolant of a concept C in prime implicate normal form can be generated in linear time in $|C|$.*

Proof. Follows directly from Lemmas 27 and 28. \square

4.2.3 Other Properties

Another interesting property of concepts in prime implicate normal form is that they do not contain any unnecessary concept or roles names. This is an important property if we are going to be presenting information to a user, since the presence of irrelevant atomic concepts and roles can make a concept difficult to understand.

Proposition 30. *If C is a concept in prime implicate normal form, then for every concept D such that $\models C \equiv D$ we have $Sig(C) \subseteq Sig(D)$.*

Proof. The proof is by induction on the depth of C . For the base case, let C be a concept in prime implicate normal form such that $\delta(C) = 0$, and suppose for a contradiction that there is some D such that $\models C \equiv D$ and $Sig(C) \not\subseteq Sig(D)$. We can assume without loss of generality that D is in prime implicate normal form since we show in Proposition 33 of the following section that every concept can be rewritten as an equivalent concept in prime implicate normal form, and this transformation does not add any new symbols. Now since $Sig(C) \not\subseteq Sig(D)$ and $\delta(C) = 0$, there must be some atomic concept A which appears in C but not in D . This means that there is some propositional clause Cl which is a conjunct of C and which mentions A . Because D is in prime implicate normal form and $\models C \equiv D$, it follows that there is some conjunct Cl' of D such that $\models Cl \equiv Cl'$. Now as the signature of C is non-empty and C is in prime implicate normal form, C must be non-tautologous, so D and Cl' must also be non-tautologous. But as either A or $\neg A$ appears as a disjunct in Cl , and both $\models Cl \sqsubseteq Cl'$ and $\not\models \top \sqsubseteq Cl'$, it follows by Proposition 7 that either A or $\neg A$ is a literal in Cl' as well, contradicting the fact that A is not part of the signature of D . We can thereby conclude that the result holds whenever $\delta(C) = 0$.

Next suppose that the result holds whenever the concept has depth at most k , and let C be some concept in prime implicate normal form of depth $k + 1$. Suppose for a contradiction that there is some D such that $\models C \equiv D$ and $Sig(C) \not\subseteq Sig(D)$. Again because of Proposition 33 we can assume without loss of generality that D is in prime implicate normal form. Now since $Sig(C) \not\subseteq Sig(D)$, there is some atomic concept or role which appears in C but not D . Let Cl be a conjunct of C which contains this symbol. As C and D are in prime implicate normal form and $\models C \equiv D$, it follows that there is some conjunct Cl' of D such that $\models Cl \equiv Cl'$. We remark that since C is in prime implicate normal form and its signature is non-empty, C , D , and Cl' must all non-tautologous. Now we know that there is either an atomic concept or role which is in Cl but not in D , hence not in Cl' .

We consider first the case where there is some atomic concept A which is in $Sig(Cl) \setminus Sig(Cl')$. There are three possibilities: either $(\neg)A$ is a disjunct in Cl , or $(\neg)A$ appears in E for some disjunct $\exists R.E$ of Cl , or $(\neg)A$ appears in E for some disjunct $\forall R.E$ of Cl . In the first case, we have a contradiction since $(\neg)A$ cannot be a disjunct of Cl' (otherwise $A \in Sig(D)$), which means that $\not\models Cl \sqsubseteq Cl'$ (by Proposition 7). If $(\neg)A$ appears in a disjunct $\exists R.E$ of Cl , then since $\models Cl \equiv Cl'$, it follows by Proposition 7 and Definition 17 that there is some disjunct $\exists R.E'$ in Cl' such that $\models E \equiv E'$. But E is a concept in prime implicate normal form of depth at most k , so by the induction hypothesis we can conclude that $Sig(E) \subseteq Sig(E')$, contradicting the fact that A does not appear in D . Consider then the third case in which A appears in $\forall R.E$. Then because Cl and Cl' are both in prime implicate normal form and $\models Cl \equiv Cl'$, it follows from Proposition 7 and Definition 17 that there is some disjunct $\forall R.E'$ of Cl' such that $\models E \equiv E'$. Applying the induction hypothesis (which applies since E is a concept in prime implicate normal form of depth at most k), we find that $Sig(E) \subseteq Sig(E')$, and hence that $Sig(E) \subseteq Sig(D)$, which contradicts the assumption that A is not in the signature of D .

Finally, let us consider the case where there is a role S which belongs to the signature of Cl but not to the signature of Cl' . Now if there is a disjunct $\exists S.E$ in Cl , we have a contradiction since by Proposition 7 we must have $\not\models Cl \equiv Cl'$ because $\not\models \top \sqsubseteq Cl'$, $\exists S.E$ is satisfiable, and there is no disjunct $\exists S.E'$ in Cl' . Similarly we can show that there can be no disjunct of the form $\forall S.E$ in Cl . It must then be the case that there is a disjunct $\exists R.E$ or $\forall R.E$ such that $R \neq S$ and E mentions S . In either case, we find (using precisely the same arguments as used in the previous paragraph) that there is a sub-concept E in prime implicate normal form appearing in Cl with depth at most k , and a sub-concept E' of Cl' such that $\models E \equiv E'$. By applying the induction hypothesis, we find that $Sig(E) \subseteq Sig(E')$, so S must appear in D , contradicting our earlier assumption. We have thus shown that for every concept C in prime implicate normal form and for every D such that $\models C \equiv D$ there can be no symbol which appears in C but not in D . \square

In [3], we introduced a restricted form of prime implicates, in which concepts are required to mention non-trivially the symbols of a given signature. This type of prime implicate is useful when one wants to characterize the information that a concept contains about a certain topic of interest. Here is the formal definition:

Definition 31 (About- \mathcal{L} -prime implicate). Let \mathcal{L} be a signature. A clausal concept Cl is an *about- \mathcal{L} -prime implicate* of a concept C if and only if Cl

is a prime implicate of C and for every concept E such that $\models E \equiv Cl$ we have $\mathcal{L} \subseteq \text{Sig}(E)$.

One of the consequences of Proposition 30 is that calculating the about- \mathcal{L} -prime implicates of a concept C in prime implicate normal form is as simple as selecting those conjuncts of C which mention every symbol in the signature \mathcal{L} . This means that for concepts in prime implicate normal form we can easily pick out the information which is relevant to a given subject matter.

Proposition 32. *The about- \mathcal{L} -prime implicates of a concept C in prime implicate normal form can be generated in linear time in $|C|$.*

Proof. Consider the algorithm which examines each of the conjuncts of C and returns only those conjuncts which contain all the atomic concepts and roles belonging to \mathcal{L} . This algorithm can clearly be implemented so as to run in linear time in the size of C . We claim that every outputted concept is an about- \mathcal{L} -prime implicate and that every about- \mathcal{L} -prime implicate is equivalent to one of the outputted concepts. The proof of the first claim is straightforward: every outputted concept D is a conjunct of C and hence a prime implicate of C and in prime implicate normal form, from which it follows by Lemma 30 that there is no concept E equivalent to D and such that $\mathcal{L} \not\subseteq \text{Sig}(E)$. For the second claim, suppose for a contradiction that there is some about- \mathcal{L} -prime implicate Cl which is not equivalent to any of the outputted concepts. As Cl is also a prime implicate of C (by definition), it follows that there is some conjunct Cl' of C such that $\models Cl \equiv Cl'$. But we also know that Cl' must not contain all of the symbols in \mathcal{L} (otherwise it would be one of the outputted concepts), contradicting the fact that Cl is an about- \mathcal{L} -prime implicate of C . \square

4.3 Computing Prime Implicate Normal Form

We have seen in the last subsection that concepts in prime implicate normal form enjoy some nice properties, but in order to take advantage of them, we need a method for putting concepts into prime implicate normal form.

We present in Figure 5 the algorithm PINF which transforms a given concept as an equivalent concept in prime implicate normal form. The first step of our algorithm is to check whether the inputted concept is unsatisfiable or tautologous, in which case we return respectively \perp or \top . For all other concepts, we continue on to Step 2, where we use GEN-PI to generate the set of prime implicates of the inputted concept, which we then modify in

Step 3 so that they satisfy all the conditions of Definition 17. We first check to see whether there are multiple existential restrictions for a single role, in which case we group them together as a single existential restriction. Next we make sure that the concepts behind the universal restrictions are in the proper form by unioning them with the concept behind the existential restriction (if there is one). We then check if each of the concepts in the clausal concept is necessary, and we remove all literal concepts which are found to be redundant. After that, we consider the concepts appearing behind a universal or existential restriction, and we put each of them into prime implicate normal form. Finally, in Step 4, we return the intersection of these modified prime implicates.

Function $\text{PINF}(C)$: *returns a concept which is in prime implicate normal form and is equivalent to C*

1. If $\models C \sqsubseteq \perp$, return \perp . If $\models \top \sqsubseteq C$, return \top .
2. Set $\Sigma = \text{GEN-PI}(C)$.
3. For each P in Σ
 - (i) For each role R : if $\exists R(P) = \{D_1, \dots, D_m\}$ where $m > 1$, replace P by $P \setminus \{\exists R.D_1, \dots, \exists R.D_m\} \cup \{\exists R.(D_1 \sqcup \dots \sqcup D_m)\}$.
 - (ii) For each role R : if $\exists R(P) = \{D\}$ and $\forall R(P) = \{E_1, \dots, E_n\}$, replace P by $P \setminus \{\forall R.E_1, \dots, \forall R.E_n\} \cup \{\forall R.(E_1 \sqcup D), \dots, \forall R.(E_n \sqcup D)\}$.
 - (iii) For each disjunct D in P : if $\models P \equiv P \setminus \{D\}$, replace P by $P \setminus \{D\}$.
 - (iv) For each $D \in \cup_R(\exists R(P) \cup \forall R(P))$, replace D by $\text{PINF}(D)$.
4. Return $\prod_{P \in \Sigma} P$.

Figure 5: Algorithm for putting a concept into prime implicate normal form.

Proposition 33. *The algorithm PINF always terminates, and the concept it returns is a concept in prime implicate normal form which is equivalent to the inputted concept, has a signature contained in the signature of the inputted concept, and has depth at most that of the inputted concept.*

Proof. The proof is by induction on the depth of the inputted concept C . If C has depth 0, then either $\models C \sqsubseteq \perp$ or $\models \top \sqsubseteq C$, or C is neither unsatisfiable nor tautologous. In the first case, the result trivially holds. In the latter case, we will continue on to Step 2 where we set Σ equal to the output of $\text{GEN-PI}(C)$. Because of Proposition 13 we know that every element in Σ is a prime implicate of C and that all prime implicates of C are equivalent to some element in Σ . It follows then from Proposition 11 that C

is equivalent to conjunction of the elements in Σ . Moreover, we know from Proposition 16 that the signatures of the concepts in Σ are all contained in the signature of C and that the depths of the elements in Σ are bounded above by $\delta(C)$. As C is assumed to be propositional, the only modification we may make to Σ in Step 3 is to eliminate repeated literals appearing in the prime implicates. It follows then that the algorithm terminates and returns a concept in prime implicate normal form which is equivalent to C , has a signature contained in $Sig(C)$, and has depth at most $\delta(C)$.

Suppose next that the result holds whenever the inputted concept has depth at most k , and let C be a concept of depth $k + 1$. Clearly the result holds if $\models C \sqsubseteq \perp$ or $\models \top \sqsubseteq C$. Suppose then that C is neither unsatisfiable nor tautologous. In Step 2, we set Σ equal to the output of $GEN-PI(C)$. By Proposition 13, we know that the elements of Σ are precisely the prime implicates of C , so C must be equivalent to the conjunction of elements in Σ by Proposition 11. We also know from Proposition 16 that the signatures of the concepts in Σ are all contained in the signature of C and that the depths of the elements in Σ cannot exceed $\delta(C)$. Thus all we need to show is that the operations performed on the concepts in Σ in Step 3 are equivalence-, signature-, and depth-preserving. For (i) and (ii), this follows directly from Proposition 6, and for (iii), this is obvious. For (iv), this follows from the induction hypothesis since we apply the function $PINF$ to concepts with depth at most k . We have thus shown that the concept outputted by $PINF(C)$ is equivalent to C , has signature contained in $Sig(C)$, and depth at most $\delta(C)$. We now verify that $PINF(C)$ is in prime implicate normal form. Clearly, $PINF(C)$ is a conjunction of clausal concepts, since the elements in Σ are originally clausal concepts, and the modifications in Step 3 do not change this. As we have shown the operations in Step 3 to be equivalence-preserving, it follows that the conjuncts of $PINF(C)$ are all prime implicates of C and that each prime implicate of C is equivalent to some conjunct of $PINF(C)$. Moreover, the conjuncts all satisfy the other conditions of Definition 5. We have $|\exists R(C)| \leq 1$ for every role R because of (i) of Step 3. Because of Step 3 (ii), we know that for if there are disjuncts $\exists R.D$ and $\forall R.E$, then $\models D \sqsubseteq E$. We also know that there is no redundant disjuncts since all unnecessary disjuncts were eliminated in Step 3 (iii). Finally, we can be sure that all of the concepts appearing behind an existential or universal role restriction are in prime implicate normal form because of part (iv) of Step 3. We have thus shown that $PINF(C)$ is in prime implicate normal form, completing the proof. \square

4.4 Spatial Complexity of Prime Implicate Normal Form

In the current subsection, we investigate the spatial complexity of prime implicate normal form in order to determine how much more space is needed to represent a concept in prime implicate normal form.

It is well-known that in propositional logic the transformation to prime implicate normal form can result in an exponential blowup in the size of the formula (cf. [7]). The blowup can be never be more than singly-exponential since there are at most 3^n distinct clauses on n variables.

Proposition 34. *Every propositional concept built from n atomic concepts is equivalent to a concept in prime implicate normal form whose length is singly-exponential in n .*

We now prove that for arbitrary concepts the transformation to prime implicate normal form involves an at most doubly-exponential blowup in concept length.

Proposition 35. *Every concept C is equivalent to a concept in prime implicate normal form whose length is at most doubly-exponential in $|C|$.*

Proof. We will use $f_n(k)$ to denote the maximal length of the output of PINF when the input concept has depth k and n distinct literal subconcepts. We know from Proposition 34 that there exists some polynomial q such that every propositional concept built using at most m atomic concepts is equivalent to some propositional concept in prime implicate normal form with length at most $2^{q(m)}$. As the number of atomic concepts appearing in a concept can never exceed the number of distinct literal subconcepts appearing in the concept, it follows that there exists some polynomial function p such that $f_n(0) \leq 2^{p(n)}$.

Now that we have obtained an upper bound on $f_n(0)$, we try to obtain an upper bound on $f_n(k+1)$ in terms of $f_n(k)$. Consider some concept C with depth $k+1$ and having at most n distinct literal subconcepts. The output of $\text{PINF}(C)$ is a conjunction of clausal concepts, one for each prime implicate of C . We know by Proposition 15 that there can be no more than n^{2^n} distinct prime implicates of C , so there can be at most n^{2^n} conjuncts in the output of $\text{PINF}(C)$. We also know that each of the prime implicates outputted by $\text{GEN-PI}(C)$ has at most 2^n disjuncts (cf. proof of Proposition 14). In Step 3, equivalence-preserving modifications are made to the prime implicates outputted by $\text{GEN-PI}(C)$, but none of these modifications may increase the number of disjuncts. It follows then that the conjuncts of the output of $\text{PINF}(C)$ all have at most 2^n disjuncts.

We now want to place a bound on the size of the disjuncts appearing in the conjuncts of $\text{PINF}(C)$. Consider some conjunct P of $\text{PINF}(C)$, and let Q be the element of $\text{GEN-PI}(C)$ which was turned into P via the modifications in Step 3 of PINF . Besides the propositional disjuncts which have length at most 1, there are two types of disjuncts which may appear in P : existential restrictions of the form $\exists R.\text{PINF}(D_1 \sqcup \dots \sqcup D_r)$ where $\exists R(Q) = \{D_1, \dots, D_r\}$, and universal restrictions of the form $\forall R.\text{PINF}(E \sqcup D_1 \sqcup \dots \sqcup D_r)$ where $E \in \forall R(Q)$ and $\exists R(Q) = \{D_1, \dots, D_r\}$. Now since the elements in $\exists R(Q) \cup \forall R(Q)$ are all subconcepts of C , every literal subconcept of one of the concepts in $\exists R(Q) \cup \forall R(Q)$ is also a literal subconcept of C . That means that if $E \in \forall R(Q)$ and $\exists R(Q) = \{D_1, \dots, D_r\}$, then all the literal subconcepts appearing in $D_1 \sqcup \dots \sqcup D_r$ or $E \sqcup D_1 \sqcup \dots \sqcup D_r$ also appear in C . As we have assumed there to be at most n distinct literal subconcepts in C , it follows that there can be no more than n distinct literal subconcepts in $D_1 \sqcup \dots \sqcup D_r$ or $E \sqcup D_1 \sqcup \dots \sqcup D_r$. We also know that the disjuncts of Q have depth at most $k+1$ (Proposition 16), which means that any concept of the form $D_1 \sqcup \dots \sqcup D_n$ or $E \sqcup D_1 \sqcup \dots \sqcup D_r$ where $E \in \forall R(P)$ and $\exists R(P) = \{D_1, \dots, D_r\}$ must have depth no greater than k . We can thus conclude that $|\text{PINF}(D_1 \sqcup \dots \sqcup D_n)| \leq f_n(k)$ and $|\text{PINF}(E \sqcup D_1 \sqcup \dots \sqcup D_r)| \leq f_n(k)$ where $E \in \forall R(P)$ and $\exists R(P) = \{D_1, \dots, D_r\}$, which means that any disjunct in Q must have length at most $f_n(k) + 1$ (the extra 1 is for the role in the role restriction).

Putting all of this together, we find the following relationship between $f_n(k+1)$ and $f_n(k)$:

$$f_n(k+1) \leq n^{2^n} (2^n (f_n(k) + 1))$$

Here the n^{2^n} gives the maximal number of conjuncts, 2^n gives the maximal number of disjuncts per conjunct, and $f_n(k)+1$ gives the maximal size of the disjuncts. Using standard techniques for solving first-order linear recurrence relations, we arrive at the following:

$$f_n(k) \in O((n^{2^n} 2^n)^k f_n(0))$$

It is not hard to see that this expression is no more than doubly-exponential in n . Now suppose that C is a concept with n literal subconcepts and depth k . We know that the size of $\text{PINF}(C)$ is bounded above by $f_n(k)$. As the number of literal subconcepts in a concept C can never exceed $|C|$, we must have $n \leq |C|$. We also know that the depth of C is bounded by the length of C , i.e. $k = \delta(C) \leq |C|$. This means that the above expression is at most doubly-exponential in $|C|$, so $|\text{PINF}(C)|$ must also be at most than doubly-exponential in $|C|$. \square

We now prove that this upper bound is optimal by showing that in some cases the transformation to prime implicate normal form may involve a doubly-exponential blowup in concept length.

Proposition 36. *There exist concepts C such that the smallest equivalent concept in prime implicate normal form has length which is doubly-exponential in the length of C .*

Proof. In [5], it has been shown that there exists a concept C such that the number of non-equivalent prime implicates of C is doubly-exponential in $|C|$. Any concept in prime implicate normal form which is equivalent to C must have a doubly-exponential number of conjuncts, and hence length which is doubly-exponential in $|C|$. \square

5 Related Work

Most of the subsumption algorithms that have been proposed for subpropositional description logics involve a normalization step in which concepts are put into some type of normal form. This is the case for instance for the description logics \mathcal{FL}_0 [14], CLASSIC [6], \mathcal{ALN} [15], \mathcal{ALE} [1], and \mathcal{EL}^{++} [2]. There has been relatively little work however on normal forms for more expressive description logics like \mathcal{ALC} which support full disjunction. Two notable exceptions are the disjunctive form introduced for the mu-calculus in [12] and adapted to \mathcal{ALC} in [18] and the linkless normal form recently proposed in [10]. In this section, we examine some of the properties of these normal forms and compare them to our own.

5.1 Disjunctive Form

Disjunctive form was first introduced in [12] as a normal form for mu-calculus formulae. In more recent work [18], B. ten Cate and colleagues have used disjunctive form as a normal form for concepts in \mathcal{ALC} :

Definition 37 (Disjunctive Form). If R is a role and Φ a set of \mathcal{ALC} concepts, then $\nabla R.\Phi$ stands for the following concept:

$$\bigsqcap_{C \in \Phi} \exists R.C \sqcap \forall R. \bigsqcup_{C \in \Phi} C$$

The set of \mathcal{ALC} concepts in *disjunctive form* is generated by the following recursive definition:

$$C ::= \top \mid \perp \mid \pi \sqcap \nabla R_1.\Phi_1 \sqcap \dots \sqcap \nabla R_n.\Phi_n \mid C \sqcup C$$

where π is a consistent conjunction of atomic literal concepts, R_1, \dots, R_n are distinct roles, and Φ_1, \dots, Φ_n are finite sets of concepts in disjunctive form.

Disjunctive form can be seen as a description of a concept's models. Each of the disjuncts $\pi \sqcap \nabla R_1.\Phi_1 \sqcap \dots \sqcap \nabla R_n.\Phi_n$ represents a possible model in which the root of model satisfies π , there is at least one R_i -successor satisfying each of the concepts in Φ_i , and all R_i -successors satisfy at least of the concepts in Φ_i .

In [18], the authors prove that it is possible to generate the \mathcal{L} -interpolants of concepts in disjunctive form in linear time and that the transformation to disjunctive form involves an only singly-exponential blowup in concept size. These properties together mean that disjunctive form can be used to produce singly-exponential size \mathcal{L} -interpolants. Indeed, the authors use disjunctive form to provide a tight upper bound on the length of \mathcal{L} -interpolants.

We can also show that the n -interpolants of concepts in disjunctive form can be produced in linear time, which means that disjunctive form can be used to generate singly-exponential-sized n -interpolants.

Proposition 38. *The n -interpolant of a concept C in disjunctive form can be generated in linear time in the size of C .*

Proof. Define $f(C, n)$ as follows:

$$\begin{aligned} f(\top, n) &= \top \\ f(\perp, n) &= \perp \\ f(D_1 \sqcup D_2, n) &= f(D_1, n) \sqcup f(D_2, n) \\ f(\pi \sqcap \prod_i \nabla R_i.\Phi_i, n) &= \begin{cases} \perp & \text{if } \models \prod_i \nabla R_i.\Phi_i \sqsubseteq \perp \\ \pi & \text{if } n = 0 \\ \pi \sqcap \prod_i \nabla R_i.\{f(G, n-1) \mid G \in \Phi_i\} & \text{if } n > 1 \end{cases} \end{aligned}$$

Satisfiability of concepts in disjunctive form can be computed in linear time [12]. This means that $f(C, n)$ can be generated in linear time from C . We now show that if C is in disjunctive form then $f(C, n)$ is indeed the n -interpolant of C . The proof is by induction on n . The base case is $n = 0$. The result trivially holds for propositional C , since they are left unchanged by f . Suppose then that C is of the form $\pi \sqcap \nabla R.\Phi$, and let $D = P_1 \sqcup \dots \sqcup P_j$ be a propositional clausal concept such that $\models C \sqsubseteq D$. Then we must have $\models \top \sqsubseteq \neg C \sqcup D$, which gives us

$$\models \top \sqsubseteq \neg\pi \sqcup \neg(\nabla R.\Phi) \sqcup P_1 \sqcup \dots \sqcup P_j$$

It follows from Proposition 6 that either $\models \top \sqsubseteq \neg\pi \sqcup P_1 \sqcup \dots \sqcup P_j$ or $\models \top \sqsubseteq \neg(\nabla R.\Phi)$. In the first case, we have $\models \pi \sqsubseteq P_1 \sqcup \dots \sqcup P_j$, and in the

second case, we have $\models \nabla R.\Phi \sqsubseteq \perp$. In either case, we find that $f(C, 0)$ is subsumed by D , and hence by every concept of depth 0 which subsumes C . Using similar reasoning, we can show the same result holds for arbitrary concepts of depth 0 in disjunctive form. As $f(C, 0)$ clearly has depth 0 and subsumes C , it follows that $f(C, 0)$ must be the 0-interpolant of C .

Next suppose that the result holds whenever $n \leq k$, and consider the case where $n = k + 1$. Now as before, the result trivially holds for propositional concepts. We consider the case where C is of the form $\nabla R.\Phi$. In this case, we get either $f(C, k + 1) = \perp$ if $\nabla R.\Phi$ is unsatisfiable (which is correct since \perp is the $k + 1$ -interpolant of any unsatisfiable concept), and otherwise we have $f(C, k + 1) = \nabla R.\{f(G, n) \mid G \in \Phi\} = \bigcap_{G \in \Phi} \exists R.f(G, k) \sqcap \forall R.\bigcup_{G \in \Phi} f(G, k)$. Now suppose that $D = P_1 \sqcup \dots \sqcup P_j \sqcup \exists R.E \sqcup \forall R.F_1 \sqcup \dots \sqcup \forall R.F_m$ is some clausal concept with depth at most $k + 1$ which subsumes C . Then we must have $\models \top \sqsubseteq \neg C \sqcup D$, and thus

$$\models \top \sqsubseteq \bigcup_{G \in \Phi} \forall R.\neg G \sqcup \exists R.\bigcap_{G \in \Phi} \neg G \sqcup P_1 \sqcup \dots \sqcup P_j \sqcup \exists R.E \sqcup \forall R.F_1 \sqcup \dots \sqcup \forall R.F_m$$

It follows from Proposition 6 that one of the following holds: $\models \top \sqsubseteq P_1 \sqcup \dots \sqcup P_j$, $\models \top \sqsubseteq (\bigcap_{G \in \Phi} \neg G) \sqcup E \sqcup F_i$ for some i , or $\models \top \sqsubseteq (\bigcap_{G \in \Phi} \neg G) \sqcup E \sqcup \neg H$ for some $H \in \Phi$. In the first case, D is tautologous, in the second case, $\forall R.\bigcup_{G \in \Phi} G \sqsubseteq \exists R.E \sqcup \forall R.F_i$ for some i , and in the third case, $\models \exists R.H \sqsubseteq \exists R.E$ for some $H \in \Phi$. We thus find that either $\exists R.H$ is subsumed by D for some $H \in \Phi$ or $\forall R.\bigcup_{G \in \Phi} G$ is subsumed by D . But since D has depth at most $k + 1$, it must be the case that the $k + 1$ -interpolant of some $\exists R.H$ or the $k + 1$ -interpolant of $\forall R.\bigcup_{G \in \Phi} G$ is subsumed by D . Because of the induction hypothesis, we know that the k -interpolant of H is $f(H, k)$ for $H \in \Phi$, and the k -interpolant of $\bigcup_{G \in \Phi} G$ is $\bigcup_{G \in \Phi} f(G, k)$ (since interpolation distributes over \sqcup). It follows that the $k + 1$ -interpolant of $\exists R.H$ is $\exists R.f(H, k)$ and the $k + 1$ -interpolant of $\forall R.\bigcup_{G \in \Phi} G$ is $\forall R.\bigcup_{G \in \Phi} f(G, k)$. We thus find that $f(C, k + 1)$ is subsumed by D for every clausal concept D of depth at most $k + 1$ which subsumes C . This implies $f(C, k + 1)$ is the $k + 1$ -interpolant of C since every concept can be rewritten as a conjunction of clausal concepts of lesser or equal depth (Proposition 5). Using similar reasoning, it can be shown that $f(C, k + 1)$ is the $k + 1$ -interpolant of C for concepts C of the form $\pi \sqcap \prod_i \nabla R_i.\Phi_i$. This is enough to show the result for arbitrary C in disjunctive form with depth $k + 1$ since (a) every such concept is a disjunction of concepts of the form $\pi \sqcap \prod_i \nabla R_i.\Phi_i$ of depth at most $k + 1$, and (b) the $k + 1$ -interpolant of a disjunction of concepts is equivalent to the disjunction of the $k + 1$ -interpolants of the disjuncts. We

have thus shown that the above procedure can be used to generate the n -interpolants of concepts in disjunctive form in linear time. \square

Most of the other nice properties of prime implicate normal form can be shown not to hold for disjunctive form. Concepts in disjunctive form may contain unnecessary atomic concepts. Consider for instance the concept $A \sqcup \neg A$ which is in disjunctive form but is equivalent to \top which does not mention A . Moreover, while satisfiability-testing of concepts in disjunctive form is easy (it is shown in [12] to be decidable in linear time), we can show that tautology-testing, subsumption, and equivalence-testing cannot be carried out efficiently (unless $P=NP$):

Proposition 39. *Deciding whether a concept in disjunctive form is a tautology is co-NP-hard.*

Proof. Any propositional formula ϕ in DNF can be transformed in linear time into an equivalent concept C_ϕ in disjunctive form by simply removing any unsatisfiable disjuncts from ϕ . This means that if we were able to test in polynomial time whether a concept in disjunctive form is a tautology, then we could do the same for propositional DNF formulae. As the DNF tautology problem is known to be co-NP-complete (cf. [11]), it follows that testing whether a concept in disjunctive form is a tautology is a co-NP-hard problem. \square

As both subsumption and equivalence-testing can be used to identify tautologies, these tasks must also be co-NP-hard:

Corollary 40. *Subsumption between concepts in disjunctive form is co-NP-hard.*

Corollary 41. *Deciding equivalence of concepts in disjunctive form is co-NP-hard.*

5.2 Linkless Normal Form

In [10], Furbach and Obermaier extend linkless normal form (cf. [16]) from propositional logic to \mathcal{ALC} concepts. Unlike disjunctive form which was never intended to be used for compilation, linkless normal form was specifically introduced for the purpose of compiling description logic knowledge bases. For completeness, we recall their definition of linkless normal form:

Definition 42 (C-path). The set of c -paths of a concept in NNF is defined as follows:

$c\text{-paths}(C) = \{\{C\}\}$, if C is a literal concept

$c\text{-paths}(C_1 \sqcap C_2) = \{X \cup Y \mid X \in c\text{-paths}(C_1), Y \in c\text{-paths}(C_2)\}$

$c\text{-paths}(C_1 \sqcup C_2) = c\text{-paths}(C_1) \cup c\text{-paths}(C_2)$

Definition 43 (Link). A *link* is either a concept link or a role link:

- A *concept link* of C is a set of complementary atomic literal concepts appearing in a c-path of C
- A *role link* of C is a set $\{\exists R.D, \forall R.E_1, \dots, \forall R.E_n\}$ occurring in a c-path of C and where all c-paths in $D \sqcap E_1 \sqcap \dots \sqcap E_n$ contain \perp or a concept link or role link and no proper subset of $\{\exists R.D, \forall R.E_1, \dots, \forall R.E_n\}$ is a role link

Definition 44 (Linkless normal form). A concept C is *linkless* (or in linkless normal form) if it is in NNF and there is no c-path in C which contains a link and if for each occurrence of $QR.E$ in C with $Q \in \{\exists, \forall\}$ the concept E is linkless as well.

Deciding whether a concept in linkless normal form is unsatisfiable can be done in polynomial time [10]. The same cannot be said for tautology-testing:

Proposition 45. *Deciding whether a concept in linkless normal form is a tautology is co-NP-hard.*

Proof. The proof is almost identical to that of Proposition 39. We can transform any propositional formula ϕ in DNF into an equivalent concept C_ϕ in linkless normal form in linear time by removing any unsatisfiable disjuncts from ϕ , which means that tautology-testing for linkless concepts can be reduced to the co-NP-complete DNF tautology problem, making the former problem co-NP-hard. \square

It follows that subsumption and equivalence-testing cannot be efficiently tested either:

Corollary 46. *Subsumption between concepts in linkless normal form is co-NP-hard.*

Corollary 47. *Deciding equivalence of concepts in linkless normal form is co-NP-hard.*

In [10], it is shown that for a certain class of queries subsumption can be carried out in linear time. The class of queries is rather restricted, consisting of clauses composed of atomic literal concepts and existential and universal restrictions followed by a single atomic literal concept. It is unclear whether their results can be extended larger classes of queries, like the class of queries that we proposed in Proposition 23.

The authors of [10] conjecture that \mathcal{L} -interpolants of concepts in linkless form can be generated efficiently, but this remains an open question. The complexity of n -interpolant generation is also unknown.

Finally, concepts in linkless form can contain unnecessary atomic concepts or roles. For example, the concepts $(A \sqcap B) \sqcup (A \sqcap \neg B)$ and $A \sqcap \forall R.(A \sqcup \neg A)$ are both in linkless normal form but are equivalent to the concept A which contains neither B nor R .

5.3 Discussion

The results in this section suggest that prime implicate normal form is better suited than both disjunctive form and linkless normal form for the purposes of knowledge compilation, as prime implicate normal form supports the same class of polynomial transformations and a wider range of polynomial time queries. In particular, the fact that subsumption is polynomial between concepts in prime implicate normal form means that we can test whether an arbitrary query concept subsumes a concept in prime implicate normal form by first putting the (presumably small) query concept into prime implicate normal form and then using structural subsumption. For the other two normal forms, there is currently no procedure for posing arbitrary queries to a compiled concept.

On the other hand, disjunctive form and linkless normal form have the advantage of a lower spatial complexity. This means that if one is using a normal form for the sole purpose of generating \mathcal{L} - and n -interpolants, then disjunctive form is more appropriate since it produces singly-exponential-sized interpolants, whereas those obtained using prime implicate normal form may have doubly-exponential size.

6 Conclusion and Future Work

The main contribution of this paper is the introduction of prime implicate normal form as a new normal form for concepts in the description logic \mathcal{ALC} . We showed that prime implicate normal form has a number of interesting

properties which make it suitable for the purposes of knowledge compilation, some of which are not satisfied by other normal forms proposed in the literature. We also provided an algorithm for transforming concepts into equivalent concepts in prime implicate normal form and proved that the transformation involves an at most doubly-exponential blowup in concept size.

In future work we would like to implement our prime implicate normal form transformation and our structural subsumption algorithm to see what kind of performance they give in practice. This should help us to identify the type of situations in which the benefits gained by a concept being in prime implicate normal form outweigh the cost of putting it in this form.

Another interesting question for future research is how our normal form can be extended to handle even more expressive description logics. We expect that the extension to languages with nominals should be straightforward, but that number restrictions and inverse roles will prove more challenging.

We also want to address what is probably the most important limitation of our work, namely the fact that our normal form treats concept expressions rather than sets of axioms (commonly known as TBoxes). We expect that the extension of prime implicates and prime implicate normal form to TBoxes will be highly non-trivial. In spite of this, we feel that this question is worth pursuing since it could potentially provide description logic practitioners with a new tool for dealing with the high complexity of TBox reasoning.

References

- [1] F. Baader, R. Kusters, and R. Molitor. Computing least common subsumers in description logics with existential restrictions. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI'99)*, pages 96–103, 1999.
- [2] Franz Baader, Sebastian Brandt, and Carsten Lutz. Pushing the envelope. In *Proceedings of IJCAI'05*, pages 364–369, 2005.
- [3] M. Bienvenu. Consequence finding in \mathcal{ALC} . In *Proceedings of the Twentieth International Workshop on Description Logics (DL2007)*, volume 250 of *CEUR Workshop Proceedings*, 2007.

- [4] 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.
- [5] M. Bienvenu. Prime implicates and prime implicants in modal logic: Extended and revised version. Research report IRT/RR–2007-17–FR, IRT, Université Paul Sabatier, Toulouse, 2007.
- [6] A. Borgida and P. Patel-Schneider. A semantics and complete algorithm for subsumption in the CLASSIC description logic. *Journal of Artificial Intelligence Research*, 1:277–308, 1994.
- [7] A. Chandra and G. Markowsky. On the number of prime implicants. *Discrete Mathematics*, 24:7–11, 1978.
- [8] A. Darwiche and P. Marquis. A knowledge compilation map. *Journal of Artificial Intelligence Research*, 17:229–264, 2002.
- [9] F. M. Donini, B. Hollunder, M. Lenzerini, A. Marchetti Spaccamela, D. Nardi, and W. Nutt. The complexity of existential qualification in concept languages. *Artificial Intelligence*, 53:309–327, 1992.
- [10] Ulrich Furbach and Claudia Obermaier. Knowledge compilation for description logics. In *Proceedings of the 3rd Workshop on Knowledge Engineering and Software Engineering (KESE)*, 2007.
- [11] M. Garey and D. Johnson. *Computers and intractability. A guide to the theory of NP-completeness*. W. H. Freeman, 1979.
- [12] D. Janin and I. Walukeiwicz. Automata for the modal mu-calculus and related results. In *Proceedings of the Twentieth International Symposium on the Mathematical Foundations of Computer Science (MFCS'95)*, volume 969 of *Lecture Notes in Computer Science*, pages 552–562. Springer, 1995.
- [13] J. Lang, P. Liberatore, and P. Marquis. Propositional independence: Formula-variable independence and forgetting. *Journal of Artificial Intelligence Research*, 18:391–443, 2003.
- [14] H. Levesque and R. Brachman. Expressiveness and tractability in knowledge representation and reasoning. *Computational Intelligence*, 3(78-93), 1987.

- [15] R. Molitor. Structural subsumption for ALN. LTCS-Report 98-03, RWTH Aachen, 1998.
- [16] N. Murray and E. Rosenthal. Dissolution: Making paths vanish. *Journal of the ACM*, 40(3):504–535, 1993.
- [17] Klaus Schild. A correspondence theory for terminological logics: preliminary report. In *Proceedings of IJCAI-91, 12th International Joint Conference on Artificial Intelligence*, pages 466–471, 1991.
- [18] B. ten Cate, W. Conradie, M. Marx, and Y. Venema. Definitorially complete description logics. In *Proceedings of the Tenth International Conference on Principles of Knowledge Representation and Reasoning (KR06)*, pages 79–89, 2006.