Next Steps for Description Logics of Minimal Knowledge and Negation as Failure

Peihong Ke and Ulrike Sattler

School of Computer Science, the University of Manchester, UK
{kep,sattler}@cs.man.ac.uk

1 Introduction

The more DLs are being used in applications such as the Semantic Web [2], biology, and the clinical sciences, the more certain expressive weaknesses are commented upon. A recurring set of these comments is due to the fact that only few DLs and even fewer DL reasoners support forms of defeasible reasoning. For example, Rector describes in [12, 16] how useful statements such as "the heart of a human is normally located on the left hand side of the body" could be for the clinical sciences, and OWL design patterns[1] have been developed to work around the lack of such statements.

Various combinations of DLs with nonmonotonic formalisms have been investigated so far. DL-MKNF, the combination of DLs with minimal knowledge and negation as failure (MKNF) [9] is introduced in [4]. DL-MKNF extends DLs with two *modal operators* and is considered to be a unified framework for nonmonotonic extensions of DLs since various nonmonotonic logics can be embedded into MKNF [9]; these include default logic [13] and autoepistemic logic [10]. The combination of DLs with default logic was introduced [1], implemented in Pellet [7], and its translation into DL-MKNF was explained in [4]. The combination of DLs with circumscription [3] provides a powerful and flexible alternative way for nonmonotonic reasoning in DLs since its entailment relation is parametrized with a set of concepts to be circumscribed. Hence we can pick different modes of defeasibility without changing our knowledge base. Decidablity and complexity are known for various DLs with circumscription [3], but no calculus or implementation is known. The integration of DLs with logic programming (LP) using MKNF [11] is closely related to DL-MKNF. They differ in expressive power since LP rules can make use of arbitrarily connected variables, yet these variables are all quantified in the same way. In contrast, DL-MKNF allows modal operators appearing in existential and universal restrictions. An exact comparison of this relationship is part of our future work.

A tableau algorithm for the combination of the basic DL $\mathcal{ALC}$ [15] with MKNF ($\mathcal{ALCK}_{\mathcal{NF}}$) has been described in [4]. As mentioned in [4], $\mathcal{ALCK}_{\mathcal{NF}}$ can capture certain kinds of defaults and integrity constraints (ICs). For example, our example default regarding the location of the heart in humans can be formalised

[1] See http://odps.sourceforge.net/odp/html/

as follows:

$$\mathbf{K}HumanHeart \sqsubseteq \mathbf{A}\exists Locates.Right \sqcup \mathbf{K}\exists Locates.Left.$$

The IC "anybody known to be a student should be known to be either male or female" can be formalised by $\mathbf{K}Student \sqsubseteq \mathbf{A}Man \sqcup \mathbf{A}Woman$.

Our work is an extension of the work described in [4]. We first present a translation that takes an $\mathcal{ALCK_{NF}}$ KB with *nested modal operators* into an equivalent *flat* $\mathcal{ALCK_{NF}}$ KB, i.e., a KB without nested modal operators. As a consequence, we can restrict our attention to flat KBs and thus simplify the "general" algorithm from [4]: our general algorithm is based on partitioning a set of so-called *slim* modal atoms which can be roughly seen as a subset of a set of modal atoms in [4]. Secondly, we have specified a *minimality check* that is computationally less expensive than the original one. Finally, we present a goal-directed tableau algorithm for computing models of an $\mathcal{ALCK_{NF}}$ KB based on our general algorithm which is, due to the optimised minimality check, "exponentially cheaper" than its counterpart from [4]. Moreover, since we can restrict our attention to flat KBs, we can also avoid the four $S5$ tableau rules from [4] and thus design what we believe to be a more readable algorithm. In [6], the interested reader can find full proofs.

2 Preliminaries

We briefly recall the syntax and semantics of $\mathcal{ALCK_{NF}}$ from [4] and introduce some notation.

2.1 Syntax and Semantics of $\mathcal{ALCK_{NF}}$

$\mathcal{ALCK_{NF}}$ is defined as an extension of $\mathcal{ALC}$ with the two modal operators $\mathbf{K}$ and $\mathbf{A}$ allowed in concepts and roles.

Definition 1 ($\mathcal{ALCK_{NF}}$ Syntax [4]) *The $\mathcal{ALCK_{NF}}$ syntax is defined as follows:*

$$C ::= \top \mid \bot \mid C_a \mid C_1 \sqcap C_2 \mid C_1 \sqcup C_2 \mid \neg C \mid \exists R.C \mid \forall R.C \mid \mathbf{K}C \mid \mathbf{A}C$$
$$R ::= R_a \mid \mathbf{K}R_a \mid \mathbf{A}R_a,$$

where C_a denotes an atomic $\mathcal{ALC}$ concept, C_1 and C_2 denote arbitrary $\mathcal{ALCK_{NF}}$ concepts, and R_a denotes an atomic role.

An $\mathcal{ALCK_{NF}}$ KB Σ is a tuple $\langle \mathcal{A}, \mathcal{T} \cup \Gamma \rangle$, where $\mathcal{T}$ is an $\mathcal{ALC}$ TBox and Γ is a modal $\mathcal{ALCK_{NF}}$ TBox*, i.e., an $\mathcal{ALCK_{NF}}$ TBox containing modal operators. Assertions in the ABox $\mathcal{A}$ are either $\mathcal{ALC}$ assertions or* modal $\mathcal{ALCK_{NF}}$ assertions*, i.e., assertions containing modal operators.*

The operator $\mathbf{K}$ is interpreted as "known". The operator $\mathbf{A}$ is simply a rewriting of $\neg$**not**, i.e., the negation of negation as failure. The operator $\mathbf{A}$ is interpreted as "already known" and thus understood as the autoepistemic operator L in autoepistemic logic [10].

The semantics of $\mathcal{ALCK}_{\mathcal{NF}}$ is obtained based on the following assumptions on $\mathcal{ALC}$ interpretations [4]: (1): the set of names of individuals is a countable-infinite set; (2): all $\mathcal{ALC}$ interpretations are defined over the same domain $\triangle$ and each individual name in every interpretation maps to the same domain element.

Definition 2 ($\mathcal{ALCK}_{\mathcal{NF}}$ Semantics [4]) *An $\mathcal{ALCK}_{\mathcal{NF}}$ interpretation is a triple $(\mathcal{I}, \mathcal{M}, \mathcal{N})$, where $\mathcal{I}$ is* an $\mathcal{ALC}$ interpretation $(\triangle, \cdot^{\mathcal{I}})$, *and $\mathcal{M}$ and $\mathcal{N}$ are sets of $\mathcal{ALC}$ interpretations. Atomic concepts and roles are interpreted in $\mathcal{I}$ as usual. Non-atomic and modal concepts are interpreted over $(\mathcal{I}, \mathcal{M}, \mathcal{N})$ as follows:*

$$
\begin{aligned}
(\top)^{\mathcal{I},\mathcal{M},\mathcal{N}} &= \triangle \\
(\bot)^{\mathcal{I},\mathcal{M},\mathcal{N}} &= \emptyset \\
(\neg C)^{\mathcal{I},\mathcal{M},\mathcal{N}} &= \triangle \setminus (C)^{\mathcal{I},\mathcal{M},\mathcal{N}} \\
(C_1 \sqcap C_2)^{\mathcal{I},\mathcal{M},\mathcal{N}} &= (C_1)^{\mathcal{I},\mathcal{M},\mathcal{N}} \cap (C_2)^{\mathcal{I},\mathcal{M},\mathcal{N}} \\
(C_1 \sqcup C_2)^{\mathcal{I},\mathcal{M},\mathcal{N}} &= (C_1)^{\mathcal{I},\mathcal{M},\mathcal{N}} \cup (C_2)^{\mathcal{I},\mathcal{M},\mathcal{N}} \\
(\exists R.C)^{\mathcal{I},\mathcal{M},\mathcal{N}} &= \{d \in \triangle \mid \exists d'.(d,d') \in (R)^{\mathcal{I},\mathcal{M},\mathcal{N}} \text{ and } d' \in (C)^{\mathcal{I},\mathcal{M},\mathcal{N}}\} \\
(\forall R.C)^{\mathcal{I},\mathcal{M},\mathcal{N}} &= \{d \in \triangle \mid \forall d'.(d,d') \in (R)^{\mathcal{I},\mathcal{M},\mathcal{N}} \text{ implies } d' \in (C)^{\mathcal{I},\mathcal{M},\mathcal{N}}\} \\
(\mathbf{K}C)^{\mathcal{I},\mathcal{M},\mathcal{N}} &= \textstyle\bigcap_{\mathcal{J}\in\mathcal{M}} (C)^{\mathcal{J},\mathcal{M},\mathcal{N}} \\
(\mathbf{A}C)^{\mathcal{I},\mathcal{M},\mathcal{N}} &= \textstyle\bigcap_{\mathcal{J}\in\mathcal{N}} (C)^{\mathcal{J},\mathcal{M},\mathcal{N}} \\
(\mathbf{K}R_a)^{\mathcal{I},\mathcal{M},\mathcal{N}} &= \textstyle\bigcap_{\mathcal{J}\in\mathcal{M}} (R_a)^{\mathcal{J},\mathcal{M},\mathcal{N}} \\
(\mathbf{A}R_a)^{\mathcal{I},\mathcal{M},\mathcal{N}} &= \textstyle\bigcap_{\mathcal{J}\in\mathcal{N}} (R_a)^{\mathcal{J},\mathcal{M},\mathcal{N}}
\end{aligned}
$$

A concept inclusion $C \sqsubseteq D$ is satisfied *in $(\mathcal{I}, \mathcal{M}, \mathcal{N})$, written $(\mathcal{I}, \mathcal{M}, \mathcal{N}) \models C \sqsubseteq D$, iff $C^{\mathcal{I},\mathcal{M},\mathcal{N}} \subseteq D^{\mathcal{I},\mathcal{M},\mathcal{N}}$ holds. An ABox assertion $C(a)$ is* satisfied *in $(\mathcal{I}, \mathcal{M}, \mathcal{N})$, written $(\mathcal{I}, \mathcal{M}, \mathcal{N}) \models C(a)$, iff $a \in C^{\mathcal{I},\mathcal{M},\mathcal{N}}$ holds, and, $R(a, b)$ is* satisfied *in $(\mathcal{I}, \mathcal{M}, \mathcal{N})$, written $(\mathcal{I}, \mathcal{M}, \mathcal{N}) \models R(a, b)$, iff $(a, b) \in R^{\mathcal{I},\mathcal{M},\mathcal{N}}$ holds.*

An inclusion $C \sqsubseteq D$ is satisfied *in* an $\mathcal{ALCK}_{\mathcal{NF}}$ structure $(\mathcal{M}, \mathcal{N})$, *written $(\mathcal{M}, \mathcal{N}) \models C \sqsubseteq D$, iff $C \sqsubseteq D$ is satisfied in $(\mathcal{I}, \mathcal{M}, \mathcal{N})$ for each $\mathcal{I} \in \mathcal{M}$. An assertion $C(a)$ is* satisfied *in $(\mathcal{M}, \mathcal{N})$, written $(\mathcal{M}, \mathcal{N}) \models C(a)$, iff $C(a)$ is satisfied in $(\mathcal{I}, \mathcal{M}, \mathcal{N})$ for each $\mathcal{I} \in \mathcal{M}$. An assertion $R(a, b)$ is* satisfied *in $(\mathcal{M}, \mathcal{N})$, written $(\mathcal{M}, \mathcal{N}) \models R(a, b)$, iff $R(a, b)$ is satisfied in $(\mathcal{I}, \mathcal{M}, \mathcal{N})$ for each $\mathcal{I} \in \mathcal{M}$. A TBox $\mathcal{T}$ (resp. Γ) is* satisfied *in $(\mathcal{M}, \mathcal{N})$, written $(\mathcal{M}, \mathcal{N}) \models \mathcal{T}$ (resp. $(\mathcal{M}, \mathcal{N}) \models \Gamma$), iff all inclusions in $\mathcal{T}$ (resp. Γ) are satisfied in $(\mathcal{M}, \mathcal{N})$. An ABox $\mathcal{A}$ is* satisfied *in $(\mathcal{M}, \mathcal{N})$, written $(\mathcal{M}, \mathcal{N}) \models \mathcal{A}$, iff all assertions in $\mathcal{A}$ are satisfied in $(\mathcal{M}, \mathcal{N})$. A KB Σ is* satisfied *in $(\mathcal{M}, \mathcal{N})$, written $(\mathcal{M}, \mathcal{N}) \models \Sigma$, iff, $\mathcal{T}$, Γ, and $\mathcal{A}$ are satisfied in $(\mathcal{M}, \mathcal{N})$.*

K and **A** are interpreted in the same way but on different sets. The following definition distinguishes their meanings through the maximality condition.

Definition 3 ($\mathcal{ALCK}_{\mathcal{NF}}$ Model [4]) *A set of $\mathcal{ALC}$ interpretations $\mathcal{M}$ is* a model for an $\mathcal{ALCK}_{\mathcal{NF}}$ KB Σ *iff the structure $(\mathcal{M}, \mathcal{M})$ satisfies Σ, and, for each set of interpretations $\mathcal{M}'$, if $\mathcal{M} \subset \mathcal{M}'$, then $(\mathcal{M}', \mathcal{M})$ does not satisfy Σ.*

A KB Σ is satisfiable *if there exists a model for Σ. An inclusion $C \sqsubseteq D$* is a consequence of *a KB Σ, written $\Sigma \models C \sqsubseteq D$, iff $(\mathcal{M}, \mathcal{M}) \models C \sqsubseteq D$ holds for every model $\mathcal{M}$ of Σ. Analogously, an assertion $C(a)$ (resp. $R(a, b)$)* is a consequence of *Σ, written $\Sigma \models C(a)$ (resp. $\Sigma \models R(a, b)$), iff $(\mathcal{M}, \mathcal{M}) \models C(a)$*

holds (resp. $(\mathcal{M}, \mathcal{M}) \models R(a, b)$ *holds) for every model* $\mathcal{M}$ *of* Σ*. Two* $\mathcal{ALCK}_{\mathcal{NF}}$ *KBs* Σ_1 *and* Σ_2 *are* equivalent*, written* $\Sigma_1 \equiv \Sigma_2$*, if, for any set of* $\mathcal{ALC}$ *interpretations* $\mathcal{M}$*,* $\mathcal{M}$ *is a model for* Σ_1 *iff* $\mathcal{M}$ *is a model for* Σ_2*.*

Notations. An $\mathcal{ALCK}_{\mathcal{NF}}$ concept S is *subjective* if each $\mathcal{ALC}$ subconcept of S lies within the scope of at least one modal operator. An $\mathcal{ALCK}_{\mathcal{NF}}$ concept O is *objective* if none of the $\mathcal{ALC}$ subconcepts of O lies within the scope of a modal operator. In other words, objective concepts are $\mathcal{ALC}$ concepts. An $\mathcal{ALCK}_{\mathcal{NF}}$ concept C is an *atomic modal concept* if C is of the form $\mathbf{M}D$, where D is an $\mathcal{ALC}$ concept. An $\mathcal{ALCK}_{\mathcal{NF}}$ concept C is in *negation normal form (NNF)* if negation occurs only in front of atomic modal concepts. Two concepts C_1 and C_2 are *equivalent*, written $C_1 \equiv C_2$, iff $C_1^{(\mathcal{I},\mathcal{M},\mathcal{N})} = C_2^{(\mathcal{I},\mathcal{M},\mathcal{N})}$ holds for any $\mathcal{ALCK}_{\mathcal{NF}}$ interpretation $(\mathcal{I}, \mathcal{M}, \mathcal{N})$. For two $\mathcal{ALC}$ KBs Σ and Σ', $\Sigma \models \Sigma'$ iff Σ entails all axioms and assertions in Σ'.

In the remainder of this paper, we assume that all concepts are in NNF, $\dot{\neg}C$ denotes the NNF of $\neg C$, $\mathbf{M}$ denotes either $\mathbf{K}$ or $\mathbf{A}$, Σ denotes an $\mathcal{ALCK}_{\mathcal{NF}}$ KB $\Sigma = \langle \mathcal{A}, \mathcal{T} \cup \Gamma \rangle$, R_a denotes an $\mathcal{ALC}$ role, C_a, D_a, E_a, and F_a denote $\mathcal{ALC}$ concepts, and C, D, and E denote arbitrary $\mathcal{ALCK}_{\mathcal{NF}}$ concepts.

3 Syntax Restrictions

Since an $\mathcal{ALCK}_{\mathcal{NF}}$ model $\mathcal{M}$ is in general infinite, in [4], $\mathcal{M}$ is represented in terms of an $\mathcal{ALC}$ KB $\Sigma_{\mathcal{M}}$ such that $\mathcal{M} = \{\mathcal{I} \mid \mathcal{I} \models \Sigma_{\mathcal{M}}\}$. Hence it is crucial to make sure that each model $\mathcal{M}$ is *$\mathcal{ALC}$-representable* (i.e., there is an $\mathcal{ALC}$ KB $\Sigma_{\mathcal{M}}$ s.t. $\mathcal{M} = \{\mathcal{I} \mid \mathcal{I} \models \Sigma_{\mathcal{M}}\}$) and the corresponding $\Sigma_{\mathcal{M}}$ is finite. In [4], to ensure $\mathcal{ALC}$-representability of $\mathcal{ALCK}_{\mathcal{NF}}$ models, an $\mathcal{ALCK}_{\mathcal{NF}}$ KB is restricted to a *subjectively quantified KB*. A subjectively quantified KB is further restricted to a *simple* KB to ensure the termination of the tableau algorithm.

In this section, we first loosen the definition of a subjectively quantified KB. Our notion still ensures $\mathcal{ALC}$-representability of $\mathcal{ALCK}_{\mathcal{NF}}$ models.

Definition 4 (New Subjectively Quantified KBs) *A* subjectively quantified $\mathcal{ALCK}_{\mathcal{NF}}$ KB Σ *is an* $\mathcal{ALCK}_{\mathcal{NF}}$ *KB such that each concept* C *of the form* $\exists R.D$ *or* $\forall R.D$ *occurring in* Σ *satisfies one of the conditions:* R *is an* $\mathcal{ALC}$ *role and* D *is an* $\mathcal{ALC}$ *concept or* R *is of the form* $\mathbf{M}R_a$ *and* D *is subjective.*

If R is of the form $\mathbf{M}R_a$, in [4], D is required to be the form of $\mathbf{M}D'$ or $\neg\mathbf{M}D'$. We only require D to be subjective.

Next, we loosen the definition of a simple KB [4] by allowing more concepts to occur in ABoxes. Again, this preserves $\mathcal{ALC}$-representability of $\mathcal{ALCK}_{\mathcal{NF}}$ models and termination.

Definition 5 (New Simple KBs) Σ *is* simple *if* Σ *is subjectively quantified and satisfies the following conditions:*

1. *only axioms of the form* $\mathbf{K}C_a \sqsubseteq D$ *are contained in* Γ*, where* C_a *is an* $\mathcal{ALC}$ *concept and* D *is a subjectively quantified concept such that no* $\mathbf{K}$ *operator occurs in existential and universal restrictions; and*
2. *for each* $\mathbf{K}C_a \sqsubseteq D \in \Gamma$*,* $\mathcal{T} \not\models \top \sqsubseteq C_a$ *holds.*

Although we restrict Σ to a simple $\mathcal{ALCK}_{\mathcal{NF}}$ KB, Σ is still expressive enough for nonmonotonic applications such as defaults and integrity constraints.

4 Flattening an Simple $\mathcal{ALCK}_{\mathcal{NF}}$ KB

In this section, we will sketch how to *flatten* a KB with nested modalities, and start with some notation. An $\mathcal{ALCK}_{\mathcal{NF}}$ concept C *contains nested modal operators* if there is a modal operator in C lying in the scope of another modal operator. An $\mathcal{ALCK}_{\mathcal{NF}}$ concept F is *flat* if F is subjective and contains no nested modal operators. An $\mathcal{ALCK}_{\mathcal{NF}}$ role R is *flat* if R is of the form $\mathbf{M}R_a$. An $\mathcal{ALCK}_{\mathcal{NF}}$ ABox $\mathcal{A}$ is *flat* if C is flat for each $C(a) \in \mathcal{A}$ and R is flat for each $R(a,b) \in \mathcal{A}$. A modal $\mathcal{ALCK}_{\mathcal{NF}}$ TBox Γ is *flat* if D is flat for each $\mathbf{K}C_a \sqsubseteq D \in \Gamma$. An $\mathcal{ALCK}_{\mathcal{NF}}$ KB Σ is *flat* if both $\mathcal{A}$ and Γ are flat.

In this section, we introduce a method to equivalently translate an arbitrary $\mathcal{ALCK}_{\mathcal{NF}}$ KB to a flat one, which we believe has the following two advantages:

1. Working on flat KBs simplifies the "general" algorithm because we focus on a smaller set of atoms, so-called *slim* modal atoms, which are a subset of modal atoms in [4];
2. Flattening an $\mathcal{ALCK}_{\mathcal{NF}}$ KB replaces the four $S5$ tableau rules[2] in [4] with a pre-processing step which we believe makes our $\mathcal{ALCK}_{\mathcal{NF}}$ tableau algorithm easier to understand and "trims" input $\mathcal{ALCK}_{\mathcal{NF}}$ KBs into a uniform format which we hope will make our tableau algorithm easier to optimise and implement.

In Table 1, we present some equivalences, with which $\mathcal{ALCK}_{\mathcal{NF}}$ concepts of the form $\mathbf{M}C$ can be simplified to ones without nested modal operators [6]. First-order versions of Equivalence 2 and 3 are described in [11].

Theorem 1 *All equivalences from Table 1 hold.*

Concepts such as $\mathbf{KA}C_a$ and $\mathbf{K}(C_a \sqcup \mathbf{A}D_a)$ can be translated into $\mathbf{A}C_a$ and $\mathbf{K}C_a \sqcup \mathbf{A}D_a$ using Equivalence 1 and 2 in **Table** 1, respectively. However, concepts such as $\mathbf{K}(C_a \sqcup (D_a \sqcap \mathbf{K}E_a))$ or $\mathbf{K}((C_a \sqcap \mathbf{K}D_a) \sqcup (E_a \sqcap \mathbf{K}F_a))$ cannot be easily translated into equivalent flat concepts: we first need to translate $C_a \sqcup (D_a \sqcap \mathbf{K}E_a)$ and $(C_a \sqcap \mathbf{K}D_a) \sqcup (E_a \sqcap \mathbf{K}F_a)$ into Conjunctive Normal Form (CNF) using equivalences 4 to 8. As shown in [11], after translating $C_a \sqcup (D_a \sqcap \mathbf{K}E_a)$ to its CNF $(C_a \sqcup D_a) \sqcap (C_a \sqcup \mathbf{K}E_a)$, we can flatten $\mathbf{K}((C_a \sqcup D_a) \sqcap (C_a \sqcup \mathbf{K}E_a))$ as follows: $\mathbf{K}((C_a \sqcup D_a) \sqcap (C_a \sqcup \mathbf{K}E_a)) \equiv \mathbf{K}(C_a \sqcup D_a) \sqcap \mathbf{K}(C_a \sqcup \mathbf{K}E_a) \equiv \mathbf{K}(C_a \sqcup D_a) \sqcap (\mathbf{K}C_a \sqcup \mathbf{K}E_a)$.

[2] There are the $\mathbf{M}$-rule-1, the $\mathbf{M}$-rule-2, the $\neg\mathbf{M}$-rule-1, and the $\neg\mathbf{M}$-rule-2.

1	$\mathbf{M}S \equiv S$
2	$\mathbf{M}(A_1 \sqcap A_2) \equiv \mathbf{M}A_1 \sqcap \mathbf{M}A_2$
3	$\mathbf{M}(S \sqcup A) \equiv S \sqcup \mathbf{M}A$
4	$(A_1 \sqcup A_2) \sqcap A_3 \equiv (A_1 \sqcap A_3) \sqcup (A_2 \sqcap A_3)$ (Distributive Law 1)
5	$(A_1 \sqcap A_2) \sqcup A_3 \equiv (A_1 \sqcup A_3) \sqcap (A_2 \sqcup A_3)$ (Distributive Law 2)
6	$\neg(A_1 \sqcap A_2) \equiv \neg A_1 \sqcup \neg A_2$ (De Morgan's Law 1)
7	$\neg(A_1 \sqcup A_2) \equiv \neg A_1 \sqcap \neg A_2$ (De Morgan's Law 2)
8	$\neg\neg A \equiv A$ (Double Negative Law)

Note: S means a subjective $\mathcal{ALCK}_{\mathcal{NF}}$ concept;
A_1, A_2, A_3, and A mean arbitrary $\mathcal{ALCK}_{\mathcal{NF}}$ concepts.

Table 1. Equivalences of $\mathcal{ALCK}_{\mathcal{NF}}$ expressions.

Theorem 2 *Let $D_1 \equiv D_2$ be an equivalence and $\Sigma[D_1 \to D_2]$ the KB obtained by replacing an occurrence of D_1 in Σ with D_2. Then $\Sigma \equiv \Sigma[D_1 \to D_2]$ holds.*

In order to flatten Σ, we need to make sure that C is flat for each $C(a) \in \mathcal{A}$ and E is flat for each $\mathbf{K}D_a \sqsubseteq E \in \Gamma$. We can flatten concepts of the form $\mathbf{K}C$, but, in Σ, we may have $\mathcal{ALC}$ assertions, $\mathcal{ALCK}_{\mathcal{NF}}$ assertions such as $D(a)$, or modal axioms such as $\mathbf{K}C_a \sqsubseteq D$, where D is not of the form $\mathbf{M}C$. For flattening Σ, we make use of our algorithm for flattening concepts of the form $\mathbf{M}C$ in [6] by translating Σ to an equivalent KB $\Sigma' = \langle \mathcal{A}', \mathcal{T} \cup \Gamma' \rangle$, where
$\mathcal{A}' := \{\mathbf{M}R_a(a,b) \mid \mathbf{M}R_a(a,b) \in \mathcal{A}\} \cup \{\mathbf{K}R_a(a,b) \mid R_a(a,b) \in \mathcal{A}\} \cup \{\mathbf{K}C(a) \mid C(a) \in \mathcal{A}\}$ and
$\Gamma' := \{\mathbf{K}C_a \sqsubseteq \mathbf{K}D \mid \mathbf{K}C_a \sqsubseteq D \in \Gamma\}$.

Theorem 3 *Concepts of the form $\mathbf{M}C$ can be translated into an equivalent flat concept. Σ and flattened Σ' are equivalent.*

By flattening an $\mathcal{ALCK}_{\mathcal{NF}}$ KB, we may have to translate concepts into their CNF, which might result in an exponential blow-up of the size of $\mathcal{ALCK}_{\mathcal{NF}}$ KB. We believe that this transformation into CNF will rarely be required: for example, KBs resulting from defaults or integrity constraints are "naturally" flat. It will be part of our future work to look into this more closely.

5 Computing Models for a Flat $\mathcal{ALCK}_{\mathcal{NF}}$ KB

In this section, we first sketch our general algorithm for computing models for a flat $\mathcal{ALCK}_{\mathcal{NF}}$ KB. Then we present a "goal-direct" tableau algorithm whose correctness is based on the correctness of our general algorithm.

5.1 The General Algorithm

As described in [4], the general algorithm is an extension of the propositional MBNF[3] reasoning algorithm [14]. The algorithm for computing the models of

[3] Minimal belief and negation as failure. MBNF has a minor difference from MKNF.

an $\mathcal{ALCK}_{\mathcal{NF}}$ KB is to "blindly" guess a partition of a set of so-called atoms (which may be infinite) and check whether such a partition satisfies a certain set of conditions. This algorithm may not terminate and is not practical at all, but it is the foundation for the tableau algorithm.

Our general algorithm is developed based on the one in [4], but differs in the following two aspects.

Firstly, our algorithm computes an $\mathcal{ALCK}_{\mathcal{NF}}$ model of a flat $\mathcal{ALCK}_{\mathcal{NF}}$ KB Σ by checking whether there is a *partition* of *slim* modal atoms for Σ (denoted with $\mathsf{SMA}_{\triangle}(\Sigma)$) satisfying a certain set of conditions.

A set of slim modal atoms is defined based on a revised version of a set of modal atoms in [4]: The set of *modal atoms $MA_{\triangle}(\Sigma)$ of Σ w.r.t. a domain $\triangle$* is defined inductively as follows:

1. if $C(a)/R(a,b) \in \mathcal{A}$, then $C(a)/R(a,b) \in \mathsf{MA}_{\triangle}(\Sigma)$;
2. if $\neg\mathbf{M}C(x) \in \mathsf{MA}_{\triangle}(\Sigma)$, then $\mathbf{M}C(x) \in \mathsf{MA}_{\triangle}(\Sigma)$;
3. if $C \sqcup D(x) \in \mathsf{MA}_{\triangle}(\Sigma)$, then $\{C(x), D(x)\} \subseteq \mathsf{MA}_{\triangle}(\Sigma)$;
4. if $C \sqcup D(x) \in \mathsf{MA}_{\triangle}(\Sigma)$, then $\{C(x), \dot{\neg}C(x), D(x), \dot{\neg}D(x)\} \subseteq \mathsf{MA}_{\triangle}(\Sigma)$;
5. if $\exists\mathbf{M}R.C(x) \in \mathsf{MA}_{\triangle}(\Sigma)$, then $\{\mathbf{M}R(a,y), C(y) \mid y \in \triangle\} \subseteq \mathsf{MA}_{\triangle}(\Sigma)$;
6. if $\forall\mathbf{M}R.C(x) \in \mathsf{MA}_{\triangle}(\Sigma)$, then $\{\mathbf{M}R(a,y), C(y) \mid y \in \triangle\} \subseteq \mathsf{MA}_{\triangle}(\Sigma)$; and
7. if $\mathbf{K}C \sqsubseteq D \in \Gamma$, then $\{\mathbf{K}C(x), D(x) \mid x \in \triangle\} \subseteq \mathsf{MA}_{\triangle}(\Sigma)$.

The set of *slim* modal atoms $\mathsf{SMA}_{\triangle}(\Sigma)$ for Σ w.r.t. $\triangle$, which is a subset of $\mathsf{MA}_{\triangle}(\Sigma)$ defined above, is defined as follows:

$$\mathsf{SMA}_{\triangle}(\Sigma) = \{\mathbf{M}C(x) \mid \mathbf{M}C(x) \in \mathsf{MA}_{\triangle}(\Sigma)\} \cup \{\mathbf{M}R(x,y) \mid \mathbf{M}R(x,y) \in \mathsf{MA}_{\triangle}(\Sigma)\}.$$

The set $\mathsf{SMA}_{\triangle}(\Sigma)$ in our algorithm can be roughly seen as a subset of the set of modal atoms in [4]. In particular, the "value" of an atom in $\mathsf{SMA}_{\triangle}(\Sigma)$ is independent from the "values" of other $\mathsf{SMA}_{\triangle}(\Sigma)$ atoms. Hence we do not need to maintain these dependencies and check that they do not cause inconsistencies.

Secondly, we modify the *minimality condition*, and this modification will improve the complexity of the tableau algorithm: instead of "comparing" a partition of $\mathsf{MA}_{\triangle}(\Sigma)$ with the partitions of $\mathsf{MA}_{\triangle}(\Sigma')$ in [4], where Σ' depends on Σ and the given partition, we prove that it is sufficient to "compare" a partition of $\mathsf{SMA}_{\triangle}(\Sigma)$ with other partitions of $\mathsf{SMA}_{\triangle}(\Sigma)$.

5.2 The Tableau Algorithm

Like the tableau algorithm in [4], our tableau algorithm computes models of an $\mathcal{ALCK}_{\mathcal{NF}}$ KB by means of so-called *branches* which correspond to finite subsets of (slim) modal atoms.

Our tableau algorithm is different from the one in [4] in the following aspects:

1. The tableau algorithm in [4] has 11 tableau rules, while ours has only 5 rules. Apart from the four $S5$ tableau rules being replaced with a pre-preprocessing stage, we have a $\sqcup$-rule which can be treated as a natural combination of the $\sqcup$-rule and the mcut-rule in [4] and a $\forall$-rule which can be treated as a combination of the $\forall$-rule and the $\mathbf{K}R$-rule in [4]. We believe that a compact set of tableau rules makes the tableau algorithm more natural and readable;

2. Our tableau algorithm employs a different minimality condition and can therefor be said to be "exponentially cheaper" than the one in [4].

In this section, $\Sigma = \langle \mathcal{A}, \mathcal{T} \cup \Gamma \rangle$ represents a flat and simple $\mathcal{ALCK}_{\mathcal{NF}}$ KB. The tableau algorithm generates a set of so-called *branches*. A branch satisfying a certain set of conditions corresponds to a model of Σ. The algorithm starts from $\mathcal{A}$. Γ is taken into account in the trigger-rule (see Figure 1). $\mathcal{T}$ is considered implicitly through the so-called *objective knowledge* of a branch and in the set of conditions which a branch must satisfy.

The $\mathcal{ALCK}_{\mathcal{NF}}$ tableau algorithm looks similar to a standard DL tableau algorithm. As mentioned in [4], it has the following features: (1): A DL reasoner is used as an underlying reasoner; (2): Only modal assertions and axioms are "decomposed" by the tableau rules. $\mathcal{ALC}$ assertions and axioms are "pushed down" to a DL reasoner.

We first introduce some definitions about branches[4] for Σ. The *initial branch* $\mathcal{B}_0(\Sigma)$ *for* Σ is the set $\mathcal{A}$. A *branch for* Σ is a set of $\mathcal{ALCK}_{\mathcal{NF}}$ assertions obtained from $\mathcal{B}_0(\Sigma)$ by applying the tableau rules from Figure 1. A branch $\mathcal{B}$ for Σ is *completed* if no rules from Figure 1 is applicable to $\mathcal{B}$.

The tableau rules are introduced in Figure 1. We discuss them briefly here.

1. The $\sqcap$-rule and the $\exists$-rule is analogous to usual DL tableau rules and are the same as in [4];
2. The $\sqcup$-rule is a bit different from a usual tableau rule for disjunction. For a disjunction $C \sqcup D(a)$ in a DL tableau rule, we add either C or D to the label of a node. In contrast, for a disjunctive assertion $C \sqcup D(a)$ in $\mathcal{ALCK}_{\mathcal{NF}}$, we need to know both the "values" of $C(a)$ and $D(a)$ because we need them for the minimality check at the end of the tableau algorithm. Note that the fourth case is to generate a clash when there is an inconsistency on $C \sqcup D(a)$;
3. The first half of the $\forall$-rule is similar to a usual DL rule for universal quantification. In the second part, in order to make this algorithm correct, we consider the relationship between $\mathbf{K}R(x, y)$ and $\mathbf{A}R(x, y)$. More precisely, $\mathbf{A}R(x, y)$ is an element of $\mathsf{SMA}_{\triangle}(\Sigma)$ and is "hidden" in $\forall \mathbf{A}R.C(x)$. The "value" of $\mathbf{A}R(x, y)$ in the tableau algorithm is by default "false" until the appearance of $\mathbf{K}R(x, y)$ "supporting" it to be "true".
4. The trigger-rule is needed in order to take in to account Γ. A DL reasoner is employed here to check if $C_a(x)$ can be entailed from "what have been known so far", i.e., from the $\mathbf{K}$-*objective-knowledge* $Ob_{\mathbf{K}}(\mathcal{B})$ (defined below) which represents the knowledge "extracted" from $\mathcal{B}$ and $\mathcal{T}$.

The following definitions and Theorem 4 are similar to their counterparts in [4], but they differ in that our algorithm works on a flat KB and therefor these definitions can be based on slim modal atoms.

Definition 6 ($(P_{\mathcal{B}}, N_{\mathcal{B}})$) *The* partition $(P_{\mathcal{B}}, N_{\mathcal{B}})$ associated with a branch $\mathcal{B}$ *is defined as follows: for* $\mathbf{M} \in \{\mathbf{K}, \mathbf{A}\}$,
$P_{\mathcal{B}} = \{\mathbf{M}C(x) \mid \mathbf{M}C(x) \in \mathcal{B}\} \cup \{\mathbf{M}R(x, y) \mid \mathbf{M}R(x, y) \in \mathcal{B}\}$*; and*
$N_{\mathcal{B}} = \{\mathbf{M}C(x) \mid \neg\mathbf{M}C(x) \in \mathcal{B}\}$.

[4] The definitions have counterparts in [4] with minor changes.

$\sqcap$-rule if $C \sqcap D(x) \in \mathcal{B}$ and $\{C(x), D(x)\} \not\subseteq \mathcal{B}$, then add $C(x)$ and $D(x)$ to $\mathcal{B}$.

$\sqcup$-rule if $C \sqcup D(x) \in \mathcal{B}$, we distinguish the following four cases:
- if $\{C(x), \dot{\neg}C(x), D(x), \dot{\neg}D(x)\} \cap \mathcal{B} = \emptyset$, then add S to $\mathcal{B}$, where $S = \{C(x), D(x)\}$, $S = \{C(x), \dot{\neg}D(x)\}$, or $S = \{\dot{\neg}C(x), D(x)\}$;
- if $\{C(x), \dot{\neg}C(x)\} \cap \mathcal{B} \neq \emptyset$ and $\{D(x), \dot{\neg}D(x)\} \cap \mathcal{B} = \emptyset$, then add either $D(x)$ or $\dot{\neg}D(x)$ to $\mathcal{B}$;
- if $\{D(x), \dot{\neg}D(x)\} \cap \mathcal{B} \neq \emptyset$ and $\{C(x), \dot{\neg}C(x)\} \cap \mathcal{B} = \emptyset$, then add either $C(x)$ or $\dot{\neg}C(x)$ to $\mathcal{B}$;
- if $\{\dot{\neg}C(x), \dot{\neg}D(x)\} \subseteq \mathcal{B}$, then add $C(x)$ to $\mathcal{B}$.

$\forall$-rule We distinguish the following two cases:
- if $\forall \mathbf{M}R.C(x) \in \mathcal{B}$, then for each $\mathbf{M}R(x,y) \in \mathcal{B}$, if $C(y) \notin \mathcal{B}$, add $C(y)$ to $\mathcal{B}$;
- if $\forall \mathbf{A}R.C(x) \in \mathcal{B}$, then for each $\mathbf{K}R(x,y) \in \mathcal{B}$, if $\mathbf{A}R(x,y) \notin \mathcal{B}$, add $\mathbf{A}R(x,y)$ to $\mathcal{B}$.

$\exists$-rule if $\exists \mathbf{M}R.C(x) \in \mathcal{B}$ and $\{\mathbf{M}R(x,y), C(y)\} \not\subseteq \mathcal{B}$ for any $y \in \mathcal{O}_{\mathcal{B}}$, then add $\mathbf{M}R(x,z)$ and $C(z)$ to $\mathcal{B}$, for some $z \in \mathcal{O}_{\mathcal{B}} \cup \{\iota\}$, where $\iota \notin \mathcal{O}_{\mathcal{B}}$.

***trigger*-rule** if $\mathbf{K}C_a \sqsubseteq D \in \Gamma$, $x \in \mathcal{O}_{\mathcal{B}}$, $Ob_{\mathbf{K}}(\mathcal{B}) \models C_a(x)$, and $\{\mathbf{K}C_a(x), D(x)\} \not\subseteq \mathcal{B}$, then add $\mathbf{K}C_a(x)$ and $D(x)$ to $\mathcal{B}$.

Fig. 1. The new $\mathcal{ALCK}_{\mathcal{NF}}$ tableau rules.

Intuitively, $P_{\mathcal{B}}$ is the set of atoms which are believed to be true and $N_{\mathcal{B}}$ is the set of atoms which are believed to be false. The set $P_{\mathcal{B}} \cup N_{\mathcal{B}}$ is a finite subset of $\mathsf{SMA}_{\triangle}(\Sigma)$ [6]. The (infinite) atoms in $\mathsf{SMA}_{\triangle}(\Sigma) \setminus (P_{\mathcal{B}} \cup N_{\mathcal{B}})$ are believed to be false.

Definition 7 (Objective Knowledge) *Let $\mathcal{B}$ be a branch for Σ. For $\mathbf{M} \in \{\mathbf{K}, \mathbf{A}\}$, the $\mathcal{ALC}$ KB $Ob_{\mathbf{M}}(\mathcal{B}) = \langle \mathcal{T}, \{C(x) \mid \mathbf{M}C(x) \in P_{\mathcal{B}}\} \cup \{R_a(x,y) \mid \mathbf{M}R_a(x,y) \in P_{\mathcal{B}}\}\rangle$ is called the* **M**-objective-knowledge *of $\mathcal{B}$.*

Next, we define what it means for a branch to be without (more or less) obvious inconsistencies.

Definition 8 (Open Branch) *Let $\mathcal{B}$ be a branch for Σ. $\mathcal{B}$ is* open *if, for $\mathbf{M} \in \{\mathbf{K}, \mathbf{A}\}$, $Ob_{\mathbf{M}}(\mathcal{B})$ is satisfiable and $Ob_{\mathbf{M}}(\mathcal{B}) \not\models C(x)$ for each $\mathbf{M}C(x) \in N_{\mathcal{B}}$.*

If there is an open and completed branch $\mathcal{B}$ for Σ, then there is an $\mathcal{ALCK}_{\mathcal{NF}}$ structure $(\mathcal{M}, \mathcal{N})$ that satisfies Σ.

Definition 9 (Pre-preferred Branch) *Let $\mathcal{B}$ be a branch for Σ. $\mathcal{B}$ is* pre-preferred *if the following conditions hold: (1): $\mathcal{B}$ is open and completed; (2): $Ob_{\mathbf{K}}(\mathcal{B}) \models Ob_{\mathbf{A}}(\mathcal{B})$; and (3): $Ob_{\mathbf{K}}(\mathcal{B}) \not\models C(x)$ for each $\mathbf{A}C(x) \in N_{\mathcal{B}}$.*

If there is a pre-preferred branch $\mathcal{B}$ for Σ, then there is an $\mathcal{ALCK}_{\mathcal{NF}}$ structure $(\mathcal{M}, \mathcal{M})$ that satisfies Σ.

A pre-preferred $\mathcal{B}$ branch might not "identify" a model for Σ, because $\mathcal{B}$ might not satisfy the minimality condition. In order to check the minimality condition up to the renaming of the variables introduced by the $\exists$-rule, we use the following definitions.[5] Let S be a KB or a set of assertions. $\mathcal{O}_S$ represents the

[5] These definitions have counterparts in [4] with minor changes.

set of individuals appearing in S. Let $\mathcal{B}$ be a branch for Σ, $f : \mathcal{O}_{\mathcal{B}} \setminus \mathcal{O}_{\Sigma} \rightarrow \triangle \setminus \mathcal{O}_{\Sigma}$ an injection, and $\triangle$ the domain of the interpretations for Σ. We denote with $f(\mathcal{B})$ the branch obtained from $\mathcal{B}$ by replacing each occurrence of x with $f(x)$ for each $x \in \mathcal{O}_{\mathcal{B}} \setminus \mathcal{O}_{\Sigma}$. $f(\mathcal{B})$ is called a *renamed branch of* $\mathcal{B}$.

Definition 10 (Minimality Condition for $\mathcal{B}$) *Let $\mathcal{B}$ be a completed branch for Σ. $\mathcal{B}$ satisfies the* minimality condition *if there does not exists a completed and open branch $\mathcal{B}'$ for Σ and an injection $g : \mathcal{O}_{\mathcal{B}'} \setminus \mathcal{O}_{\Sigma} \rightarrow \mathcal{O}_{\mathcal{B}} \setminus \mathcal{O}_{\Sigma}$ such that $|\mathcal{O}_{\mathcal{B}'}| \leq |\mathcal{O}_{\mathcal{B}}|$ and all of the following conditions hold:*

1. $Ob_{\mathbf{K}}(\mathcal{B}) \models Ob_{\mathbf{K}}(g(\mathcal{B}'))$;
2. $Ob_{\mathbf{K}}(g(\mathcal{B}')) \not\models Ob_{\mathbf{K}}(\mathcal{B})$;
3. $Ob_{\mathbf{K}}(\mathcal{B}) \models Ob_{\mathbf{A}}(g(\mathcal{B}'))$; *and*
4. $Ob_{\mathbf{K}}(\mathcal{B}) \not\models C(x)$ *for each* $\mathbf{A}C(x) \in N_{g(\mathcal{B}')}$.

Intuitively, the minimality condition corresponds to the maximality condition in Definition 3. We have the condition $|\mathcal{O}_{\mathcal{B}'}| \leq |\mathcal{O}_{\mathcal{B}}|$ since $\mathcal{B}$ will not violate the minimality condition because of a branch $\mathcal{B}'$ which is "bigger" (i.e., $|\mathcal{O}_{\mathcal{B}'}| > |\mathcal{O}_{\mathcal{B}}|$) than $\mathcal{B}$. We take renamed branches of $\mathcal{B}'$ into account to guarantee that $\mathcal{B}$ faithfully meets the minimality condition under the condition of changing the variables introduced by the $\exists$-rule. Our minimality check is "exponentially cheaper" than the one in [4] because our minimality condition "compares" $\mathcal{B}$ with other branches for Σ while the one in [4] "compares" $\mathcal{B}$ with all the branches for Σ' (generating all branches for Σ' is exponential because of the $\sqcup$-rule and the mcut-rule in [4]), where Σ' is gained from Σ and $\mathcal{B}$.

Definition 11 (Preferred Branch) *Let $\mathcal{B}$ be a completed branch for Σ. $\mathcal{B}$ is* preferred *if $\mathcal{B}$ is pre-preferred and $\mathcal{B}$ satisfies the minimality condition.*

If there is a preferred branch $\mathcal{B}$ for Σ, then there is an $\mathcal{ALCK}_{\mathcal{NF}}$ structure $(\mathcal{M}, \mathcal{M})$ satisfies Σ, and, for each set of interpretations $\mathcal{M}'$, if $\mathcal{M} \subset \mathcal{M}'$ holds, then $(\mathcal{M}', \mathcal{M})$ does not satisfy Σ, i.e., $\mathcal{M}$ is a model for Σ.

To state the completeness of our algorithm, we need to define what it means for a branch to represent a model: let $\mathcal{B}$ be a completed branch for Σ and $\mathcal{M}$ a model for Σ. $\mathcal{B}$ *represents* $\mathcal{M}$ if $\mathcal{B}$ is preferred and there exists an injection $f : \mathcal{O}_{\mathcal{B}} \setminus \mathcal{O}_{\Sigma} \rightarrow \triangle \setminus \mathcal{O}_{\Sigma}$ such that $\mathcal{M} = \{\mathcal{I} \mid \mathcal{I} \models Ob_{\mathbf{K}}(f(\mathcal{B}))\}$ holds.

Theorem 4 (Correctness) *If $\mathcal{B}$ is a preferred branch for Σ, then $\mathcal{M} = \{\mathcal{I} \mid \mathcal{I} \models Ob_{\mathbf{K}}(\mathcal{B})\}$ is a model for Σ. If $\mathcal{M}$ is a model for Σ, then there exists a completed branch $\mathcal{B}$ for Σ representing $\mathcal{M}$. The tableau algorithm terminates.*

6 Future Work

We are currently implementing and optimising our tableau algorithm, and hope to be able to report on its evaluation soon. As in [7], incremental ABox reasoning [5] will be extremely useful for our implementation. It prevents the underlying DL reasoner from performing reasoning from scratch every time, and we believe that the "mostly increasing" nature of our algorithm will be well-suited to profit from incremental reasoning. Regarding optimisations, we have developed a DL version of the *possible true choices* technique [8] to avoid "stupid guesses" in the nondeterministic $\sqcup$-rule.

References

1. Franz Baader and Bernhard Hollunder. Embedding defaults into terminological knowledge representation formalisms. pages 306–317, 1992.
2. Tim Berners-Lee, James A. Hendler, and Ora Lassila. The semantic web. *Scientific American*, 284(5):34–43, May 2001.
3. Piero Bonatti, Carsten Lutz, and Frank Wolter. Description logics with circumscription. In *Knowledge Representation*, pages 400–410, 2006.
4. Francesco M. Donini, Daniele Nardi, and Riccardo Rosati. Description logics of minimal knowledge and negation as failure. *ACM Transactions on Computational Logic*, 3(2):177–225, 2002.
5. Christian Halaschek-Wiener, Bijan Parsia, Evren Sirin, and Aditya Kalyanpur. Description logic reasoning for dynamic ABoxes. volume 189 of *CEUR Workshop Proceedings*, 2006.
6. Peihong Ke and Ulrike Sattler. Next steps for description logics of minimal knowledge and negation as failure. Technical report, Department of Computer Science, University of Manchester, 2008.
7. Vladimir Kolovski, Bijan Parsia, and Yarden Katz. Implementing owl defaults. *OWL: Experiences and Directions 2006*, 2006.
8. Nicola Leone, Pasquale Rullo, and Francesco Scarcello. Disjunctive stable models: Unfounded sets, fixpoint semantics, and computation. *Information and Computation*, 135(2):69–112, 1997.
9. Vladimir Lifschitz. Minimal belief and negation as failure. *Artificial Intelligence*, 70(1–2):53–72, 1994.
10. Robert C. Moore. Semantical considerations on nonmonotonic logic. *Artificial Intelligence*, 25:75–94, 1985.
11. Boris Motik and Riccardo Rosati. Closing semantic web ontologies. Technical report, Department of Computer Science, University of Manchester, 2006.
12. Alan L. Rector. Defaults, context, and knowledge: Alternatives for OWL-indexed knowledge bases. In Russ B. Altman, A. Keith Dunker, Lawrence Hunter, Tiffany A. Jung, and Teri E. Klein, editors, *Pacific Symposium on Biocomputing*, pages 226–237. World Scientific, 2004.
13. Raymond Reiter. A logic for default reasoning. *Artificial Intelligence*, 13:81–32, 1980.
14. Riccardo Rosati. Reasoning with minimal belief and negation as failure: Algorithms and complexity. pages 430–435, 1997.
15. Manfred Schmidt-Schauß and Gert Smolka. Attributive concept descriptions with complements. 48(1):1–26, 1991.
16. Robert Stevens, Mikel Egaña Aranguren, Katty Wolstencroft, Ulrike Sattler, Nick Drummond, and Mathew Horridge. Managing OWL's limitations in modelling biomedical knowledge. *International Journal of Human Computer Studies - special issue on the limits of ontologies*, 2005.