Semantic Difference in ALN

Alain Léger, Christophe Rey, Farouk Toumani

Semantic difference in $\mathcal{ALN}$

Alain Léger[1], Christophe Rey[2], and Farouk Toumani[2]

[1] France Télécom R&D, Rennes, France
alain.leger@francetelecom.com
[2] LIMOS, CNRS, Université Blaise Pascal, Clermont-Ferrand, France
christophe.rey@univ-bpclermont.fr,ftoumani@isima.fr

Abstract. We study the semantic difference operator defined in [16] for $\mathcal{ALN}$. We give a polynomial-time algorithm to compute it. We compare it with the syntactic difference operator defined in [7], for which we also give a polynomial-time algorithm for $\mathcal{ALN}$.

1 Introduction

Among non standard reasonings for description logics, the subtraction or difference operation addresses the problem of computing descriptions that are part of one concept and not part of another one. Two definitions of the difference operation have been used in the literature, namely the *semantic* and the *syntactic* difference. The semantic difference between two descriptions B and A, noted $B - A$, has been defined in [16] as the most general descriptions C such that $C \sqcap A \equiv B$. Two general kinds of applications of the semantic difference have been mentioned in [16]: removing specific information from a description and description decomposition. Such an inference mechanisms can, for example, be useful in tutorial systems that have to explain concepts to users. Recently, motivated by two applications in the areas of *semantic web service discovery* and *querying e-catalog communities*, we have used the semantic difference operation to define a new more flexible concept rewriting approach, called *best covering concepts* using terminologies [10, 4, 5].

However, the difference operation suffers from some drawbacks. First, in the languages that provide full negation the difference $B - A$ is always equal to $\neg(A \sqcap \neg B)$, a description which is not very useful in practice [16, 7]. Second, in many description logics, e.g., $\mathcal{ALN}$, the difference operations is not semantically unique, i.e., it yields to a set of descriptions that are not semantically equivalent to each other. In this case, the semantic difference is a set-valued operation and gives rise to two main difficulties: *(i)* computation of the semantic difference, and *(ii)* manipulation of the results in practical cases.

To cope with these limitations, a *syntactic* difference operation has been proposed in [7]. In this case, the difference $B-A$ yield to a syntactically minimal[3] description C such that $C \sqcap A \equiv B \sqcap A$. The syntactic difference has been used in [7] to measure the accuracy of an approximation of a given concept.

[3] That is a description containing less syntactic redundancies possible.

In contrast with its semantic version, the syntactic difference operation always produce a unique description and hence it is usually easier to compute. However, as it can be expected, the result of a syntactic operation is less accurate than the one produced by the semantic one. The overall conclusion is that the choice of a difference operator, i.e., semantic v.s. syntactic, is application dependent.

In this paper, we investigate the problem of computing the difference descriptions in the context of the $\mathcal{ALN}$ language. The motivation is the extension of our previous work on the concept covering problem [10] to $\mathcal{ALN}$. We first recall some useful results about $\mathcal{ALN}$ in section 2. Then, in section 3, we provide a polynomial-time algorithm to compute the semantic difference in $\mathcal{ALN}$. In section 4, we provide another polynomial-time algorithm to compute the syntactic difference in $\mathcal{ALN}$, then we compare both operators, and justify why we have chosen the semantic one in the study of concept covers. We conclude in section 5.

2 Preliminaries

We assume the reader familiar with the $\mathcal{ALN}$ description logic. Let C an $\mathcal{ALN}$-concept description. The normal form of C, noted $\widehat{C}$, is obtained as usual by applying the set of normalization rules given in [2, 12] that aim at removing redundancies and making explicit all implicit inconsistencies due to interactions between constructors. For more conveniency, we define in this paper a set-oriented representation of (normalized) $\mathcal{ALN}$-descriptions. To this end, we introduce below the notions of $\mathcal{ALN}$-clause and clausal form.

Let P be an atomic concept, the negation of an atomic concept or a number restriction. An $\mathcal{ALN}$-clause, or more simply a clause, is either: *(i)* a description P, or *(ii)* a description of the form $\forall R_1.(...(\forall R_n.P))$. In the following, a clause of the form $\forall R_1.(...(\forall R_n.P)...)$ is written $\forall R_1...R_n.P$.

Let C be an $\mathcal{ALN}$-description. A clausal form of C, noted $\widehat{\mathcal{G}}_C^{\#}$, is the set made of all the clauses that appear in the description obtained by recursively applying the following rule on $\widehat{C}$: $(\forall R.(E \sqcap F)) \xrightarrow{\equiv} (\forall R.E) \sqcap (\forall R.F)$.

Example 1. The following example shows the normal form and the clausal form of an $\mathcal{ALN}$-descriptions C.

$C \equiv\ \forall T.((\geq 4\ R) \sqcap (\exists S) \sqcap (\leq 1\ R)) \sqcap (\geq 2\ R) \sqcap (\forall Q.A) \sqcap (\forall Q.(\forall R.(\forall S.(B \sqcap \neg B))))$
$\widehat{C} =\ (\leq 0\ T) \sqcap (\geq 2\ R) \sqcap (\forall Q.(A \sqcap \forall R.(\leq 0\ S)))$
$\widehat{\mathcal{G}}_C^{\#} = \{\leq 0\ T, \geq 2\ R, \forall Q.A, \forall QR. \leq 0\ S\}$

Clausal forms enable a set-oriented representation of concept descriptions that is easy to understand and manipulate (especially from an algorithmic point of view). Moreover, previous results regarding subsumption and lcs in $\mathcal{ALN}$, achieved using different formal frameworks such as description graphs [14, 11] or automata theory [1, 11], can be easily translated in our context as shown below.

Theorem 1 (Structural subsumption and lcs in $\mathcal{ALN}$). *Let C and D two $\mathcal{ALN}$-descriptions. There is :*

1. $C \sqsubseteq D \Leftrightarrow \forall c_D \in \widehat{\mathcal{G}}_D^{\#}, \exists c_C \in \widehat{\mathcal{G}}_C^{\#} \mid c_C \sqsubseteq c_D$
2. $\widehat{\mathcal{G}}_{lcs(C,D)}^{\#} = \{c_1 \mid (c_1, c_2) \in (\widehat{\mathcal{G}}_C^{\#} \times \widehat{\mathcal{G}}_D^{\#}) \cup (\widehat{\mathcal{G}}_D^{\#} \times \widehat{\mathcal{G}}_C^{\#})$ *and* $c_2 \sqsubseteq c_1\}$

We recall that, building $\widehat{C}$ from an $\mathcal{ALN}$-description C can be achieved in polynomial time in the size of C [6]. So $\widehat{\mathcal{G}}_C^{\#}$ can also be computed in polynomial time in the size of C. Consequently, testing subsumption between two $\mathcal{ALN}$ descriptions C and D using theorem 1 can also be achieved in polynomial time in the sizes of the inputs. Moreover, it is shown in [11] that the lcs of two $\mathcal{ALN}$ descriptions always exists and it can be computed in polynomial time in the sizes of the inputs.

In order to study the semantic difference in $\mathcal{ALN}$, we need to recall the notion of weak approximation defined in [8, 9]. The $\mathcal{L}_2$-description D is a weak approximation of the $\mathcal{L}_1$-description C if D is the maximal description w.r.t. subsumption being subsumed by C. In this case, we write $D = Approx_{\uparrow}(C)$. In fact, as will be seen later, in our work we are only interested in computing the weak approximation of the negation of an $\mathcal{ALN}$-clause (which is an $\mathcal{ALEN}$-description) by an $\mathcal{ALN}$description. To this end, we reuse the following result [8, 9]:

Lemma 1 (Weak approximation of $\forall R_1 R_2 ... R_n.P$). *Let C be a $\mathcal{ALN}$-clause, i.e. $C \equiv \forall R_1 R_2 ... R_n.P$ with P an atomic concept, the negation of an atomic concept or a number restriction. There is:*

$$\widehat{\mathcal{G}}_{Approx_{\uparrow}(\neg C)}^{\#} = \{\ \forall R_1 R_2 ... R_n.(\neg P),\ \forall R_1 R_2 .....R_{n-1}.(\geq 1\ R_n),$$
$$\forall R_1 R_2 .....R_{n-2}.(\geq 1\ R_{n-1}), ..., \forall R_1.(\geq 1\ R_2), (\geq 1\ R_1)\ \ \}$$

All previous recalls and following results about semantic and syntactic difference in $\mathcal{ALN}$ can be extended to take into account an $\mathcal{ALN}$-terminology $\mathcal{T}$ containing either concept definitions of the form $A \equiv C$ or atomic concept inclusions of the form $A \sqsubseteq C$, with A an atomic concept and C an $\mathcal{ALN}$-description. This is due to the fact that testing the subsumption of two $\mathcal{ALN}$-descriptions wrt $\mathcal{T}$ amounts to testing the subsumption of the same but unfolded descriptions wrt to the empty terminology (i.e. $(C \sqsubseteq_{\mathcal{T}} D) \Leftrightarrow (\mathcal{T}(C) \sqsubseteq \mathcal{T}(D))$). Thus, to take into account concept definitions, a first unfolding step is mandatory (which can lead to an exponential blow-up [15]). Taking into account atomic concept inclusions can be achieved by replacing them by concept definitions adding a new atomic concept. For example, $A \sqsubseteq P_1 \sqcap \forall R.P_2$ would be replaced by $A \equiv P_1 \sqcap \forall R.P_2 \sqcap A'$. In the sequel, we suppose that we work on unfolded descriptions, so we do not talk about terminologies any more.

3 Semantic difference in $\mathcal{ALN}$

Given two concept descriptions B and A, the semantic difference $B - A$ amounts to computing all the maximal, w.r.t. subsumption, descriptions C s.t. $C \sqcap A$ is equivalent to B. So C can be seen as (i) what has to be added to A in order to get back B, and as (ii) the rest of B after removing its common information with A. The fact that C must be maximal with respect to subsumption ensures that

there is no semantic redundancy in C, which means (i) C is only what is strictly necessary to add to A to get B, and (ii) C describes what is really specific in B w.r.t. A. Primarily defined with the constraint $B \sqsubseteq A$, the semantic difference is generalized to all couples of descriptions using their least common subsumer, if it exists [16]. The formal definition of the semantic difference is now given.

Definition 1 (Semantic difference [16]). *Let $\mathcal{L}$ a description logic, B and A two $\mathcal{L}$-concept descriptions such that $B \sqsubseteq A$. The semantic difference between B and A, noted $B - A$, is defined by :*
$B - A := Max_{\sqsubseteq}\{C | C \sqcap A \equiv B\}$
If the lcs always exists between two $\mathcal{L}$-descriptions (for example in $\mathcal{ALN}$), then this definition is extended to couples of descriptions B and A with $B \not\sqsubseteq A$ by :
$B - A := B - lcs(B, A)$

Note that this definition is independent of $\mathcal{L}$, and the result of a semantic difference may be a set of descriptions. Whereas, in [16] the semantic difference is especially studied for languages having a special property[4] ensuring a unique description in the result, we study here the semantic difference for $\mathcal{ALN}$. The difference of two $\mathcal{ALN}$ descriptions may lead to potentially numerous non equivalent $\mathcal{ALN}$ descriptions (see example 2). This is due to the possibility to decompose the empty concept $\bot$ into non trivial conjunctions. Up to our knowledge, this is the first time that the semantic difference is studied for a language that implies a non unique difference.

Example 2. Let us consider the following two $\mathcal{ALN}$ descriptions:
$Q \equiv A \sqcap \forall R_1.(B \sqcap \leq 4R_2) \sqcap \leq 0R_3$
$S \equiv A \sqcap \forall R_1.(B \sqcap \forall R_4.C) \sqcap \forall R_3.(D \sqcap \forall R_5.E \sqcap \leq 2R_6)$
The lcs of Q and S is:
$lcs(Q, S) \equiv A \sqcap \forall R_1.B \sqcap \forall R_3.(D \sqcap \forall R_5.E \sqcap \leq 2R_6)$
Hence, the semantic difference between Q and S is given by the set:
$Q - S = \{\forall R_1. \leq 4R_2 \sqcap \forall R_3 \forall R_5.\neg E \sqcap \forall R_3. \geq 1R_5,$
$\forall R_1. \leq 4R_2 \sqcap \forall R_3.\neg D,$
$\forall R_1. \leq 4R_2 \sqcap \forall R_3. \geq 3R_6\}$

We now see algorithm **compute$\mathcal{ALN}$SemDiff** to compute the semantic difference of two $\mathcal{ALN}$-descriptions A and B. Due to lack of space, its detailed form is given in [13], but its underlying principle are given in lemma 2 and its soundness and completeness is given in theorem 2. This is the main contribution of this paper.

Lemma 2 (Building one description of the semantic difference). *Let B and A two $\mathcal{ALN}$-descriptions such that $B \sqsubseteq A$. Let C be an $\mathcal{ALN}$-description in $B - A$. Let P be any atomic concept, the negation of any atomic concept or any number restriction. $\widehat{\mathcal{G}}_C^{\#}$ can be built as follows:*

[4] This is the so-called structural subsumption property in the sense of [16] which is stronger than the usual notion of a structural subsumption algorithm.

First, if $\widehat{\mathcal{G}}_A^{\#} = \widehat{\mathcal{G}}_B^{\#}$, *then* $\widehat{\mathcal{G}}_C^{\#}$ *must be* $\{\top\}$. *Else,* $\widehat{\mathcal{G}}_C^{\#}$ *is initialized at* $\emptyset$, *and then, for all* c_B *in* $\widehat{\mathcal{G}}_B^{\#}$ *:*

- ***Inconsistency case:***

if $c_B = \forall R_1 R_2 ... R_{n-1}.(\leq 0\ R_n)$, *with* $n \geq 0$ *(if* $n = 0$ *then* $c_B = \bot$*)*
and $\exists c_A \in \widehat{\mathcal{G}}_A^{\#} \mid c_A = \forall R_1 R_2 ... R_n R_{n+1} ... R_{n+m}.P$, $m \geq 0$
then we add to $\widehat{\mathcal{G}}_C^{\#}$ *all clauses c verifying:*
$\left(c = \forall R_1 R_2 ... R_n.c', c' \in \widehat{\mathcal{G}}^{\#}_{Approx_{\uparrow}(\neg \forall R_{n+1} ... R_{n+m}.P)}\right)$ *and* $(A \not\sqsubseteq c)$

- ***General case:***

else if $c_B \notin \widehat{\mathcal{G}}_A^{\#}$, *then we add* c_B *to* $\widehat{\mathcal{G}}_C^{\#}$.

Thus, for each clause c_B of $\widehat{\mathcal{G}}_B^{\#}$, zero, one or many clauses of $\widehat{\mathcal{G}}_C^{\#}$ will be generated such that the conjunction of these generated clauses and some clauses in $\widehat{\mathcal{G}}_A^{\#}$ gives back c_B after normalization. The reason for having a set of descriptions in the result is due to some clauses in $\widehat{\mathcal{G}}_B^{\#}$ that may lead to different possibilities to generate clauses of $\widehat{\mathcal{G}}_C^{\#}$. More precisely, only clauses c_B of $\widehat{\mathcal{G}}_B^{\#}$ that have the form $\forall R_1 R_2 ... R_{n-1}.(\leq 0\ R_n)$, with $n \geq 0$[5] (i.e. $\forall R_1 R_2 ... R_n.\bot$) may lead to such situations (this is the so-called "Inconsistency case" in theorem 2). All other configurations for c_B are trivially solved (this is the so-called "General case" in theorem 2).

Elements of proof are now given. In the inconsistency case, the weak approximation is used to generate clauses that stay in $\mathcal{ALN}$ (because we can't use the full negation constructor in $\mathcal{ALN}$). The other interesting point concerning the weak approximation is that it ensures the property of maximality w.r.t. subsumption that is required. Last but not least, we use the characterization of inconsistency in $\mathcal{ALN}$ showed in lemmas 4.2.2 and 6.1.4 of [11] as the main argument of completeness for this theorem. The detailed proof of lemma 2 is given in [13].

Based on lemma 2, the **compute$\mathcal{ALN}$SemDiff** algorithm computes *all and only all* descriptions in the semantic difference by computing all possible combinations of multiple clauses cases. Theorem 2 proves its soundness and completeness and gives its complexity (see [13] for the proof).

Theorem 2 (compute$\mathcal{ALN}$SemDiff characteristics). *Let* B *and* A *be two* $\mathcal{ALN}$*-descriptions, given in their clausal forms, such that* $B \sqsubseteq A$. *Algorithm* ***compute$\mathcal{ALN}$SemDiff*** *(given in [13]) computes the clausal form of exactly all* $\mathcal{ALN}$*-descriptions that belong to the semantic difference of* B *and* A *as defined in definition 1. This computation is PTIME wrt the sizes of* B *and* A *(i.e. the numbers of clauses in their clausal forms and the maximal number of roles in any of their clauses).*

[5] If $n = 0$, then it is the clause $\bot$.

4 Semantic versus syntactic difference in $\mathcal{ALN}$

In this section, we focus on the comparison between the semantic difference and the syntactic difference in $\mathcal{ALN}$. We first show how to compute the syntactic difference in $\mathcal{ALN}$, and then we compare both operators.

4.1 Syntactic difference in $\mathcal{ALN}$

Syntactic difference has been defined in [11, 7]. The aim of the syntactic difference $B - A$ is to remove from B all its *subdescriptions* that are redundant with A (i.e. that are also in A). The consequence is that the result is minimal in size. This operator was initially defined to evaluate the loss of information when approximating an $\mathcal{ALC}$-description by an $\mathcal{ALE}$-description [7].

The syntactic difference relies on the notion of subdescription. Intuitively, D is a subdescription of E if D can be obtained from E by removing conjuncts or disjuncts that are in E, replacing parts of the description of E by $\bot$ or by subdescriptions of these parts. This notion of subdescription defines the partial order $\preceq_d : D \preceq_d E$ iff D is a subdescription of E [3, 11, 7]. This partial order is used to define the syntactic difference $B - A$. Thereafter, we recall the definition of this operation using $\mathcal{ALE}$ for both B and A. It is the only case for which the difference is uniquely determined, modulo associativity and commutativity of concept conjunction, and for which a sound and complete algorithm exists [11, 7][6].

Definition 2 (Syntactic difference in $\mathcal{ALE}$). *Let A and B be two $\mathcal{ALE}$ descriptions. The syntactic difference $B - A$ of B and A is defined as the $\mathcal{ALE}$ description C which is minimal w.r.t. $\preceq_d$ such that $C \sqcap A \equiv A \sqcap B$.*

Looking at the previous definition, it seems that, in $\mathcal{ALN}$, the only difference between semantic and syntactic difference is how $\bot$ is processed: in the semantic difference, non trivial conjunctions equivalent to $\bot$ are computed, while they are not in the syntactic difference. If we extend the definition of $\preceq_d$ to $\mathcal{ALN}$, as well as the definition of the syntactic difference to $\mathcal{ALN}$, we can prove the previous intuition by the following theorem which is the second contribution of this paper. This theorem shows that the syntactic difference is basically a set difference between clauses of $A \sqcap B$ and A (see [13] for the full proof).

Theorem 3 (Building the syntactic difference in $\mathcal{ALN}$). *Let A and B be two $\mathcal{ALN}$-descriptions given in their clausal form. The syntactic difference $B - A$ defined in definition 2 is a unique $\mathcal{ALN}$-description C such that if $A \sqsubseteq B$, then $C = \top$, else $\widehat{\mathcal{G}}^{\#}_{C} = \widehat{\mathcal{G}}^{\#}_{A \sqcap B} \setminus \widehat{\mathcal{G}}^{\#}_{A}$.*

Thus, computing the syntactic difference between two $\mathcal{ALN}$ descriptions is PTIME wrt the numbers of clauses in their clausal forms.

[6] The case where B is an $\mathcal{ALC}$ description and A an $\mathcal{ALE}$ has been studied in [7], but only a heuristic has been given to compute it.

4.2 Comparing semantic and syntactic operators

Theorems 2 and 3 show that, for $\mathcal{ALN}$-descriptions, *both difference compute the same result*, except for $\bot$ in the two special cases presented below.

- **Case 1:**
  Let $B \equiv \forall R.\bot$ and $A \equiv \forall R.(\neg P \sqcap P')$. There is:
  Semantic difference: $B - A = \{\forall R.P, \forall R.\neg P'\}$.
  Syntactic difference: $B - A = \forall R.\bot$.
  This case has already been studied in [11], and the conclusion is the following. It is true that the semantic difference does give the semantic gap between B and A, i.e. what has to be added to A to get B. But multiple results are less easy to figure out by a user (e.g. by a knowledge engineer) than the unique one of the syntactic difference. Moreover, in this case, the result of the syntactic difference is more intuitive since it doesn't refer to any decomposition of $\bot$. Nevertheless, we could add that getting B as the result of $B - A$ amounts to say that there is no common point between B and A, whereas $lcs(B, A) \equiv A$.
- **Case 2:**
  Let $B \equiv \forall R.P$ and $A \equiv \forall R.\neg P$. There is:
  Semantic difference: $B - lcs(B, A) = \{\forall R.P\}$.
  Syntactic difference: $B - A = \forall R.\bot$.
  In that case, the result of the semantic difference seems to be more intuitive. Indeed, getting B as the result of $B - A$ amounts to say that there is no common point between B and A, which is verified in that case since $lcs(B, A) \equiv \top$. By the contrary, the result of the syntactic difference is harder to interpret, since it cannot be interpreted neither as what remains of B once A has been removed, nor as what to add to A to get B.

So none of the two operators always produces more intuitive or understandable results. On the one hand, syntactic one generates a unique result which can be easier to manipulate (especially by a human). On the other hand, the semantic operator really computes the semantic gap between two description, by possibly generating many results and handling non intuitive decompositions of $\bot$. These multiple results may be more difficult to manipulate, but can allow a more exhaustive processing of some task. Hence the overall conclusion is that the choice of a difference operator in $\mathcal{ALN}$ will eventually depend more on the applicative context than on other technical criteria. None is a priori better. However, in our application context [4, 5], the difference operation is used to define the notion of best cover of a concept using a terminology. The aim there is to reformulate a query Q into a description that contain as much as possible of *common information* with Q and as less as possible of *extra information* with respect to Q. Such a description is called a best cover of Q. In [5], the extra information contained in a query Q and not in its best cover E, computed using the difference $Q - E$, is used to query remote sources in a peer-to-peer integration system. In such a context, using the semantic difference turns out to be more adequate than the syntactic one as it enables to query more relevant sources than what is enabled by the syntactic difference.

5 Conclusion

In this paper, we investigated the problem of computing the semantic and the syntactic difference operators in the context of the $\mathcal{ALN}$ language. We provide two polynomial-time algorithms to compute them. We compare both and argue that the semantic one is better suited to extend the notion of concept covers previously studied in [10, 4, 5].

References

1. F. Baader, R. Küsters, A. Borgida, and D. McGuinness. Matching in description logics. *Journal of Logic and Computation*, 1999.
2. F. Baader, R. Küsters, and R. Molitor. Computing Least Common Subsumer in Description Logics with Existential Restrictions. In *Proceedings of IJCAI'99*.
3. F. Baader, R. Küsters, and R. Molitor. Rewriting concepts using terminologies. In *Proc. of KR2000*.
4. B. Benatallah, M.-S. Hacid, A. Leger, C. Rey, and F. Toumani. On automating web services discovery. *VLDB J.*, 14(1):84–96, 2005.
5. B. Benatallah, M.-S. Hacid, H. Paik, C. Rey, and F. Toumani. Towards semantic-driven, flexible and scalable framework for peering and querying e-catalog communities. *Information Systems, Special issue on semantic web and web services*, 31(4-5), 2006.
6. A. Borgida and P.F. Patel-Schneider. A Semantics and Complete Algorithm for Subsumption in the CLASSIC Description Logic. *JAIR*, 1:277–308, 1994.
7. S. Brandt, R. Küsters, and A.-Y. Turhan. Approximation and Difference in Description Logics. In *Proc. of KR2002*.
8. F. Goasdoué. *Réécriture de requêtes en termes de vues dans CARIN et intégration d'informations*. PhD thesis, Université Paris XI Orsay, 2001.
9. F. Goasdoué and M.-C. Rousset. Compilation and approximation of conjunctive queries by concept descriptions. In *Proc. of DL2002*, 2002.
10. M.S. Hacid, A. Léger, C. Rey, and F. Toumani. Computing concept covers: a preliminary report. In *Proc. of DL02 Toulouse. France*, April 2002.
11. R. Küsters. *Non-Standard Inferences in Description Logics*, volume 2100 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, 2001. Ph.D. thesis.
12. R. Küsters and R. Molitor. Computing least common subsumers in alen. LTCS-Report 00-07, LTCS, RWTH Aachen, Germany, 2000.
13. A. Léger, C. Rey, and F. Toumani. Semantic difference in the aln description logic. Research report, LIMOS, 2007. see http://www.isima.fr/∼rey/RRSemDiff.pdf.
14. R. Molitor. Structural subsumption for $\mathcal{ALN}$. Technical Report LTCS-98-03, LuFG Theoretical Computer Science, RWTH Aachen, March 1998.
15. B. Nebel. Terminological Reasoning is Inherently Intractable. *ai*, 43:235–249, 1990.
16. G. Teege. Making the difference: A subtraction operation for description logics. In *Proc. of KR'94*, May 1994.