A Uniform Tableaux-Based Approach to Concept Abduction and Contraction in $\mathcal{ALN}$

S. Colucci[1], T. Di Noia[1], E. Di Sciascio[1],
F.M. Donini[2], M. Mongiello[1]
1: Politecnico di Bari, BARI, Italy
{s.colucci,t.dinoia,disciascio,mongiello}@poliba.it
2: Università della Tuscia, VITERBO, Italy donini@unitus.it

Abstract

We present algorithms based on truth-prefixed tableaux to solve both Concept Abduction and Contraction in $\mathcal{ALN}$ DL. We also analyze the computational complexity of the problems, showing that the upper bound of our approach meets the complexity lower bound. The work is motivated by the need to offer a uniform approach to reasoning services useful in semantic-based matchmaking scenarios.

1 Motivation

In recent papers [16, 15], Description Logics (DLs) have been proposed to model knowledge domains in Semantic Web scenarios. A challenging issue in such scenarios is the matchmaking problem which is finding an offered resource described by a formalism with an unambiguous semantics [21, 8, 17]. Using DLs to describe resources, it is possible to infer which of them satisfies the request either completely (*i.e.*, subsumes the request) or potentially (*i.e.*, the conjunction of the requested resource and the offered one is satisfiable) or partially(*i.e.*, the conjunction of the requested resource and the offered one is not satisfiable).

In [9, 7] Concept Abduction and Concept Contraction have been proposed as non-standard inference services in DL, to capture in a logical way the reasons why a resource S_1 should be preferred to another resource S_2 for a given request D, and vice versa. Although efficient reasoning methods based on tableaux have been successfully implemented for standard inference services in DL —satisfiability, subsumption, instance check, etc. [1, Ch.8-9] — non-standard reasoning services have been usually solved by different methods, such as automata[1, Ch.6], making a complete system built on heterogeneous technologies. Such an approach leads to incomparable optimization techniques and partial duplication of services — *e.g.*, a module computing *least common subsumer* computes also subsumption. This motivates our research in tableaux-based methods for Concept Abduction and Concept Contraction.

Related work on abduction using tableaux is in [6], where tableaux are used for the multimodal logic **K**, corresponding to the DL $\mathcal{ALC}$. However, in that work the purpose was not to find efficient methods, and contraction was not considered. Here we devise more efficient methods, for a $\mathcal{ALN}$ logic, which would correspond to a syntactically-restricted modal logic with graded modalities.

2 Abduction and Contraction in $\mathcal{ALN}$

We start with a few definitions of the problem and then move on to discuss its computational complexity. We assume the reader is familiar with DLs, and refer to [1] for a thourough introduction to $\mathcal{ALN}$, TBoxes, satisfiability, subsumption (denoted as $\sqsubseteq$) and subsumption w.r.t. a TBox $\mathcal{T}$ (denoed as $\sqsubseteq_\mathcal{T}$).

Here we deal only with a simple form of axioms in the TBox, where in the left hand side of inclusions only concept names can appear and with each concept name as at most one left-hand side of an axiom. Moreover, we admit only *acyclic* TBoxes, in the following sense.

Definition 1 (Dependency Graph of a TBox) Let $\mathcal{T}$ be a TBox. The dependency graph of $\mathcal{T}$ is a graph $\mathcal{G}_\mathcal{T} = (N, V)$ whose nodes are concept names, and whose arcs are defined as follows: if $A \sqsubseteq C \in \mathcal{T}$, and concept name B appears in C, then there is an arc from node A to node B.

A TBox $\mathcal{T}$ is said to be *acyclic* if $\mathcal{G}_\mathcal{T}$ contains no cycles. Even for this simple form of acyclic TBox, it is known that subsumption is coNP-hard [18] and also satisfiability is coNP-hard [5, 4]. However, all hardness reductions rely on "deep" TBoxes — TBoxes in which the length of the longest path in $\mathcal{G}_\mathcal{T}$ is allowed to grow as large as $O(|\mathcal{T}|)$. For$\mathcal{ALN}$ TBoxes that, in Nebel's words [18], are "bushy but not deep", satisfiability and subsumption can be solved in polynomial time [3].

Definition 2 (Bushy TBox) A sequence of acyclic TBoxes $\mathcal{T}_1, \ldots, \mathcal{T}_n, \ldots$ are *bushy* if the size of the longest path in $\mathcal{G}_{\mathcal{T}_i}$ is bounded by $O(\log |\mathcal{T}_i|)$.

In the rest of the paper, we limit our attention to bushy TBoxes in $\mathcal{ALN}$.

2.1 Concept Abduction in $\mathcal{ALN}$

We follow the notation in [9, 7], excluding the choice of the DL which in our case is always $\mathcal{ALN}$.

Definition 3 Let C, D, be two concepts in $\mathcal{ALN}$, and $\mathcal{T}$ be a set of axioms in $\mathcal{ALN}$, where both C and D are satisfiable in $\mathcal{T}$. A *Concept Abduction Problem* (CAP), denoted as $\langle C, D, \mathcal{T}\rangle$, is finding a concept $H \in \mathcal{ALN}$ such that $\mathcal{T} \not\models C \sqcap H \equiv \bot$, and $\mathcal{T} \models C \sqcap H \sqsubseteq D$.

We use $\mathcal{P}$ as a symbol for a CAP, and we denote with $SOLCAP(\mathcal{P})$ the set of all solutions to a CAP $\mathcal{P}$. For $SOLCAP(\mathcal{P})$ the three following minimality criteria have been proposed.

Definition 4 Let $\mathcal{P}$ =$\langle C, D, \mathcal{T}\rangle$ be a CAP. The set $SOLCAP_\sqsubseteq(\mathcal{P})$ is the subset of $SOLCAP(\mathcal{P})$ whose concepts are maximal under $\sqsubseteq_\mathcal{T}$. The set $SOLCAP_\leq(\mathcal{P})$ is the subset of $SOLCAP(\mathcal{P})$ whose concepts have minimum length. The set $SOLCAP_\sqcap(\mathcal{P})$ is the subset of $SOLCAP(\mathcal{P})$ whose concepts are minimal conjunctions, *i.e.*, if $C \in SOLCAP_\sqcap(\mathcal{P})$ then no sub-conjunction of C is in $SOLCAP(\mathcal{P})$. We call such solutions *irreducible abductions*.

The three forms of minimality are related by: both $SOLCAP_\sqsubseteq(\mathcal{P})$ and $SOLCAP_\leq(\mathcal{P})$ are included in $SOLCAP_\sqcap(\mathcal{P})$ [9, Prop.2].

2.2 Concept Contraction in $\mathcal{ALN}$

As defined by Gärdenfors' [12], who formalized the revision of a knowledge base $\mathcal{K}$ with a new piece of knowledge A, is made up of *(i)* a *contraction* operation, which results in a new knowledge base $\mathcal{K}_A^-$ such that $\mathcal{K}_A^- \not\models \neg A$, *(ii)* the conjunction of A to $\mathcal{K}_A^-$.

Definition 5 Let C, D, be two concepts in $\mathcal{ALN}$, and $\mathcal{T}$ be a set of axioms in $\mathcal{ALN}$, where both C and D are satisfiable in $\mathcal{T}$. A *Concept Contraction Problem* (CCP), denoted as $\langle C, D, \mathcal{T}\rangle$, is finding a pair of concepts $\langle G, K\rangle$ (both in $\mathcal{ALN}$) such that $\mathcal{T} \models C \equiv G \sqcap K$, and $K \sqcap D$ is satisfiable in $\mathcal{T}$. We call K a *contraction* of C according to D and $\mathcal{T}$.

Also for Concept Contraction, one is interested in a minimal contraction, according to some form of minimality.

Definition 6 Let $\mathcal{Q}$ =$\langle C, D, \mathcal{T}\rangle$ be a CCP. The set $SOLCCP_{\sqsubseteq}(\mathcal{Q})$ is the subset of solutions $\langle G, K\rangle$ in $SOLCCP(\mathcal{Q})$ such that G is maximal under $\sqsubseteq_{\mathcal{T}}$. The set $SOLCCP_{\leq}(\mathcal{Q})$ is the subset of $SOLCCP(\mathcal{Q})$ such that G has minimum length. The set $SOLCCP_{\sqcap}(\mathcal{Q})$ is the subset of $SOLCCP(\mathcal{Q})$ whose concepts are minimal conjunctions, *i.e.*, if $\langle G, K\rangle \in SOLCCP_{\sqcap}(\mathcal{Q})$ then no sub-conjunction G' of G is such that $\langle G', K'\rangle \in SOLCCP(\mathcal{Q})$ for any K'. We call such solutions *irreducible contractions*.

We now analyze the complexity of computing a minimum-length concept abduction in $\mathcal{ALN}$. Proposition 3 in [9] yields a trivial polynomial-time lower bound for Concept Abduction in $\mathcal{ALN}$ with a bushy TBox. Using a simple reduction, we show a tighter lower bound, using an elementary form of Tbox: the problem is NP-hard. It is sufficient to have a constant-depth concept hierarchy — *i.e.*, a set of inclusions between concept names where the longest path in $\mathcal{G}_{\mathcal{T}}$ has length 1 — to model the set-covering model for abduction [19].

Definition 7 (Set Covering) Let $U = \{a_1, \ldots, a_n,\}$ be a set, let $s_1, \ldots, s_m$, be a collection of subsets of U such that $\cup_i s_i = U$ and let $k \leq n$ be an integer. The *Set covering* problem is deciding whether there exists a subcollection of subsets $s_{i_1}, \ldots, s_{i_k}$ whose union covers U.

Theorem 1 Minimal-length Concept Abduction in $\mathcal{ALN}$ is NP-hard, even when $\mathcal{T}$ is a bushy concept hierarchy.

Given an instance of Set Covering, we construct a CAP $\mathcal{P}$ =$\langle C, D, \mathcal{T}\rangle$ as follows. Let $A_1, \ldots, A_n, B_1, \ldots, B_n$ be $2n$ concept names, where each A_i and B_i is one-one with a_i, and let $S_1, \ldots, S_m$, be also concept names, one-one with subsets of U. Let the Tbox $\mathcal{T}$ be defined as follows: $\{S_i \sqsubseteq A_j, S_i \sqsubseteq B_j | a_j \in s_i\}$. Now we prove that $s_{i_1}, \ldots, s_{i_k}$ is a minimal set covering iff $S_{i_1} \sqcap \cdots \sqcap S_{i_k} \in SOLCAP_{\leq}(\mathcal{P})$, where $C = \top$ and $D = A_1 \sqcap \cdots \sqcap A_n \sqcap B_1 \sqcap \ldots \sqcap B_n$. First of all, we prove a property of this construction.

Property 1 Every minimal-length abduction H of $\mathcal{P}$ contains neither A_i nor B_i, for every $i = 1, \ldots, n$.

Proof. Let $H \in SOLCAP(\mathcal{P})$ and suppose A_3 – say – is a conjunct of H. If there is a concept S in H, such that $S \sqsubseteq A_3 \in \mathcal{T}$, then H without A_3 is a shorter abduction. Otherwise, since$C \sqcap H \equiv H \sqsubseteq D$, also B_3 must be a conjunct of H. In this case, let S be a concept such that $S \sqsubseteq A_3, S \sqsubseteq B_3 \in \mathcal{T}$. Then the concept$H$ without A_3, B_3 and with S is a solution one conjunct shorter. The same line of reasoning could be repeated if a concept B is a conjunct of H. Therefore, every minimal-length abduction contains neither A_i nor B_i, for every $i = 1, \ldots, n$. □

(If) Suppose $s_{i_1}, \ldots, s_{i_k}$ is a set covering. Then, $H \doteq S_{i_1} \sqcap \cdots \sqcap S_{i_k}$ is such that $C \sqcap H$ is satisfiable (in fact, every conjunction is satisfiable in this CAP), and $C \sqcap H \sqsubseteq D$. Moreover, if H is not a minimal-length abduction, then let $H' \in SOLCAP_{\leq}(\mathcal{P})$. For the above property, H' does not contain A's and B's. Then it is straightforward to define a shorter set covering from H',contradicting the fact that $s_{i_1}, \ldots, s_{i_k}$ was a minimal set covering. (Only-if) On the other hand, suppose $H \in SOLCAP_{\leq}(\mathcal{P})$. Then H does not contain A's and B's, so it can be written as $S_{i_1} \sqcap \cdots \sqcap S_{i_k}$, which identifies a collection of subsets $\mathcal{S}_H = s_{i_1}, \ldots, s_{i_k}$. Since $H \sqsubseteq D$, also $\mathcal{S}_H$ covers U; moreover, if S_H was not minimal, it would define a shorter solution for $\mathcal{P}$, contradicting the hypothesis. □

We observe that a (more realistic) CAP allows one to put weights and probabilities attached to concepts in order to measure the importance that a user gives to a specified characteristic. Obviously, also this weighted version of CAP is NP-hard.

3 Calculus and Algorithms

In the following we assume the reader be familiar with tableaux (*e.g.*, [14]). In this section two algorithms working on tableaux for $\mathcal{ALN}$ concepts are presented. They both use the same set of rules: the first one (*contract*) computes a solution $\langle G, K\rangle$ for a CCP, the second one (*abduce*) solves a CAP computing H.
Tableaux for DLs use a labeling function $\mathcal{L}$ to map an individual x to a set of concepts $\mathcal{L}(x)$ such that for every concept C, $C \in \mathcal{L}(x)$ stands for the formula $C(x)$, and similarly for roles $R \in \mathcal{L}(x, y)$. Here we distinguish between formulas labeled "true" and formulas labeled "false" in the tableaux[20], hence we use two labeling functions **T**() and **F**(), both going from individuals to sets of concepts, and from pairs of individuals to sets of roles. A (usual) tableau branch is now represented by two functions **T**() and**F**(). Moreover, we write in the name of an individual x its history, *i.e.*, the string identifying x is made up of integers and role symbols, such as $x = 1R3Q7$, which means that individual x is used for concepts in a quantification involving role R, and inside, a quantification involving role Q. Integers in between roles make sure that such strings are unique, *i.e.*, there can be two individuals with the same role sequence, but not with the same integer sequence[11].

Given an individual x in a tableau, an interpretation $(\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ satisfies two tableau labels $\mathbf{T}(x)$ and $\mathbf{F}(x)$ if, for every concept $C \in \mathbf{T}(x)$ and every concept $D \in \mathbf{F}(x)$, it is $x^{\mathcal{I}} \in C^{\mathcal{I}}$ and $x^{\mathcal{I}} \notin D^{\mathcal{I}}$ respectively. Similarly, $(\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ satisfies two tableau labels $\mathbf{T}(x, y)$ and $\mathbf{F}(x, y)$ if for every role $R \in \mathbf{T}(x, y)$ and for every role $Q \in \mathbf{F}(x, y)$ it holds $(x^{\mathcal{I}}, y^{\mathcal{I}}) \in R^{\mathcal{I}}$ and $(x^{\mathcal{I}}, y^{\mathcal{I}}) \notin Q^{\mathcal{I}}$. We note that for $\mathcal{ALN}$ DL, every role Q appearing in a label $\mathbf{F}(x, y)$ is of the form $\neg R$, hence $Q \in \mathbf{F}(x, y)$ means, in fact, $(x^{\mathcal{I}}, y^{\mathcal{I}}) \in R^{\mathcal{I}}$ too. An interpretation satisfies a tableau branch if it satisfies $\mathbf{T}(x)$, $\mathbf{F}(x)$, $\mathbf{T}(x, y)$ and $\mathbf{F}(x, y)$ for every individual x, and for every pair of individuals x, y in the branch.

We assume that concepts are always simplified in Negation Normal Form (NNF, see [1, ch.2]), so that negations come only in front of concept names. Observe that for $C \in \mathcal{ALN}$, $\overline{C}$ may not belong to $\mathcal{ALN}$ since it is not closed under negation. In what follows, given a concept C, we denote with $\overline{C}$ the NNF of $\neg C$. Rules come in pairs, first the (usual) version with a construct in the **T**-constraints, then the dual construct in the **F**-constraints. However, groups 2 and 3 have only **F**-constraints because the correspondent formulae do not appear in our tableaux for $\mathcal{ALN}$.

1. conjunctions:

   T⊓) if $C \sqcap D \in \mathbf{T}(x)$, then add both C and D to $\mathbf{T}(x)$.

   F⊔) if $C \sqcup D \in \mathbf{F}(x)$, then add both C and D to $\mathbf{F}(x)$.

2. disjunctions (branching rules):

   F⊓) if $C \sqcap D \in \mathbf{F}(x)$, then add either C or D to $\mathbf{F}(x)$.

3. existential quantifications:

   F∀) if $\forall R.C \in \mathbf{F}(x)$, then pick up a new individual $y = x \circ R \circ m$ (where m is an integer such that y is unique), add $\neg R$ to $\mathbf{F}(x, y)$, and let $\mathbf{F}(y) := \{C\}$.

4. universal quantifications:

   T∀) if $\forall R.C \in \mathbf{T}(x)$ and there exists an individual y such that either $R \in \mathbf{T}(x, y)$, or $\neg R \in \mathbf{F}(x, y)$, then add C to $\mathbf{T}(y)$.

   F∃) if $\exists R.C \in \mathbf{F}(x)$, and there exists an individual y such that either $R \in \mathbf{T}(x, y)$, or $\neg R \in \mathbf{F}(x, y)$, then add C to $\mathbf{F}(y)$.

5. at-least number restrictions:

   T⩾) if $\geqslant n\, R \in \mathbf{T}(x)$, with $n > 0$, and for every individual y neither $R \in \mathbf{T}(x, y)$ nor $\neg R \in \mathbf{F}(x, y)$, then pick up a new individual $y = x \circ R \circ m$ (where m is an integer such that y is unique), add R to $\mathbf{T}(x, y)$, and let $\mathbf{T}(y) := \emptyset$.

   F⩽) if $\leqslant n\, R \in \mathbf{F}(x)$ and for every individual y neither $R \in \mathbf{T}(x, y)$ nor $\neg R \in \mathbf{F}(x, y)$, then pick up a new individual $y = x \circ R \circ m$ (where m is an integer such that y is unique), add $\neg R$ to $\mathbf{F}(x, y)$, and let $\mathbf{F}(y) := \emptyset$.

6. at-most number restrictions:

   T⩽) if $\leqslant 1\, R \in \mathbf{T}(x)$, and there are 2 individuals y_1, y_2 such that for $i \in 1, 2$ it is either $R \in \mathbf{T}(x, y_i)$ or $\neg R \in \mathbf{F}(x, y_i)$, then let $\mathbf{T}(y_1) := \mathbf{T}(y_1) \cup \mathbf{T}(y_2)$, let $\mathbf{F}(y_1) := \mathbf{F}(y_1) \cup \mathbf{F}(y_2)$, and eliminate y_2 in the branch.

   F⩾) if $\geqslant 2\, R \in \mathbf{F}(x)$ and there are 2 individuals y_1, y_2 such that for $i \in 1, 2$ it is either $R \in \mathbf{T}(x, y_i)$ or $\neg R \in \mathbf{F}(x, y_i)$, then let $\mathbf{T}(y_1) := \mathbf{T}(y_1) \cup \mathbf{T}(y_2)$, let $\mathbf{F}(y_1) := \mathbf{F}(y_1) \cup \mathbf{F}(y_2)$, and eliminate y_2 in the branch.

7. axioms in $\mathcal{T}$:

F$\sqsubseteq$) if x is an individual such that either $A \in \mathbf{T}(x)$ or $\neg A \in \mathbf{F}(x)$ in the branch, and $A \sqsubseteq C \in \mathcal{T}$, then add $A \sqcap \overline{C}$ to $\mathbf{F}(x)$.

F$\doteq_1$) if x is an individual such that either $A \in \mathbf{T}(x)$ or $\neg A \in \mathbf{F}(x)$ in the branch, and $A \doteq C \in \mathcal{T}$, then add $A \sqcap \overline{C}$.

F$\doteq_2$) if x is an individual such that either $\neg A \in \mathbf{T}(x)$ or $A \in \mathbf{F}(x)$ in the branch, and $A \doteq C \in \mathcal{T}$, then add $C \sqcap \neg A$ to $\mathbf{F}(x)$.

When more than one rule can be applied, we always give *lowest* precedence to Rules **T$\geqslant$**) and **F$\leqslant$**), while other rules can be applied in any order. In group 7 (axioms in $\mathcal{T}$) a *lazy unfolding* of the TBox is taken into account [2, 13]. Following this strategy, axioms in $\mathcal{T}$ are dealt in a deterministic manner avoiding the exponential increase in the search space due to the non-deterministic choices in a pure-tableau approach.

We now split the definition of clash (an explicit inconsistency) between clashes involving the same truth prefix (homogeneous clashes) and those involving both prefixes (heterogeneous clashes).

Definition 8 (Clash) *A branch contains a* homogeneous clash *if it contains one of the following:*

1. *either $\bot \in \boldsymbol{T}(x)$ or $\top \in \boldsymbol{F}(x)$, for some individual x;*
2. *either $A, \neg A \in \boldsymbol{T}(x)$ or $A, \neg A \in \boldsymbol{F}(x)$ for some individual x and some concept name A;*
3. *either $\geqslant n\, R, \leqslant m\, R \in \boldsymbol{T}(x)$ with $m < n$, or $\leqslant n\, R, \geqslant m\, R \in \boldsymbol{F}(x)$ with $m-1 < n+1$, for some individual x, and some role name R.*

A branch contains a heterogeneous clash *if it contains one of the following:*

1. *$\boldsymbol{T}(x) \cap \boldsymbol{F}(x)$ contains either A or $\neg A$ for some individual x and some concept name A;*
2. *either $\geqslant n\, R \in \boldsymbol{T}(x)$ and $\geqslant m\, R \in \boldsymbol{F}(x)$ with $m - 1 < n$, or $\leqslant n\, R \in \boldsymbol{T}(x)$ and $\leqslant m\, R \in \boldsymbol{F}(x)$ with $n < m + 1$, for some individual x, and some role R*

A branch is *complete* if no new rule application is possible to labels in the branch. A complete branch is *open* if it contains no clash, otherwise it is *closed*. A complete tableau is open if it contains at least one open branch, otherwise it is closed. We call a branch with a homogeneous clash *as good as complete*. Soundness and completeness of the calculus follow from the version without prefixes [10].

Theorem 2 Let C, D be two concepts in $\mathcal{ALN}$, and $\mathcal{T}$ an acyclic TBox in $\mathcal{ALN}$. Then $C \sqsubseteq D$ in $\mathcal{T}$ iff the tableau starting from $C \in \mathbf{T}(x), D \in \mathbf{F}(x)$ is closed.

Moreover, with prefixed tableaux we can distinguish between "real"subsumption, and subsumption stemming from inner contradiction in concepts.

Theorem 3 Let C, D be two concepts in $\mathcal{ALN}$, and $\mathcal{T}$ an acyclic TBox in $\mathcal{ALN}$. If every branch of the tableau starting from $C \in \mathbf{T}(1), D \in \mathbf{F}(1)$ contains a homogeneous clash, then either $C \equiv \bot$ or $D \equiv \top$ in $\mathcal{T}$.

We now present the two algorithms for Concept Contraction and Concept Abduction, that need some preliminary definitions.

Both algorithms use a function $roles(x)$, that given an individual x (as a sequence of integers and roles) returns the sequence of roles in x (without integers). For example, $roles(1R3Q7) = RQ$. We let $roles(k) = \varepsilon$, *i.e.*, when x is just one integer, $roles(x)$ returns the empty sequence. For a given concept C, and a sequence of roles σ, we define $\forall\sigma.C$ as $\forall R_1.(\cdots(\forall R_n.C)\cdots)$ if $\sigma = R_1 \cdots R_n$, and $\forall\sigma.C \doteq C$ in the special case in which $\sigma = \varepsilon$.

Moreover, we assume that atomic concepts (names and number restrictions) can be given a unique index, as in $A^1 \sqcap \forall R.((\leqslant 1\, Q)^2 \sqcap A^3)$. Hence the substitution of an *occurence* of a concept can be defined: we let $D[C \rightarrow \top]$ denote the substitution of an occurrence of an indexed atomic concept C with the concept $\top$, inside a concept D. For example, if D is the concept above, then $D[A^1 \rightarrow \top] = \top \sqcap \forall R.((\leqslant 1\, Q)^2 \sqcap A^3)$, while $D[A^3 \rightarrow \top] = A^1 \sqcap \forall R.((\leqslant 1\, Q)^2 \sqcap \top)$. Multiple substitutions are denoted by a set of concepts, *e.g.*, if $\mathcal{G} = \{A, B\}$ then $D[C \rightarrow \top]_{C\in\mathcal{G}}$ means $(D[A \rightarrow \top])[B \rightarrow \top]$. Observe that since we substitute only atomic concepts, the order of substitutions is ininfluent. For both algorithms, we assume that concepts are indexed, so that substitutions are unambiguous.

Algorithm $contract$
input: $\mathcal{ALN}$ concepts C, D, acyclic TBox $\mathcal{T}$
output: concepts K (keep), G (giveup)
begin
  compute a complete tableau τ for $\mathcal{T}$, $D \in \mathbf{T}(x), \overline{C} \in \mathbf{F}(x)$
  if τ is open **then**
    /* no contraction needed */
    return $G := \top, K := D$
  else if every branch in τ contains a homogeneous clash **then**
    /* either C or D is unsatisfiable in $\mathcal{T}$ */
    return fail
  else
    choose(*) a branch β containing only heterogeneous clashes;
    let $\mathcal{G} := \{\langle C_i, x_i\rangle | C_i \in \mathbf{T}(x_i), \overline{C_i} \in \mathbf{F}(x_i)$ is a clash in $\beta\}$
    let $G := \sqcap_{\langle C_i,x_i\rangle\in\mathcal{G}} \forall roles(x_i).C_i$
    let $K := D[C_i \rightarrow \top]_{\langle C_i,x_i\rangle\in\mathcal{G}}$
    return G, K
end

Observe that the algorithm *contract* contains a choice in step (*). This choice is needed to select the contraction according to some minimality criterion. Only branches without homogeneous clashes need to be completely expanded, even after the first clash has been found. Observe also that substituting an occurrence of a concept C with $\top$ corresponds, in $\mathcal{ALN}$, to eliminating the occurrence. We preferred this notation instead of eliminating occurrences, since it appears more concise.

Theorem 4 The concepts G, K returned by the Algorithm $contract$ are a Contraction of D w.r.t. C and $\mathcal{T}$.

Proof. First, note that $K \sqcap C$ is satisfiable by definition of K; in fact, the tableau for $K \sqcap C$ is the same as the tableau for $D \sqcap C$, but it has now at least one open branch β, in which all clashes have been removed. Secondly, $D \equiv G \sqcap K$ by construction. □

Note that Algorithm $contract$ proves that Concept Contraction in $\mathcal{ALN}$ with bushy TBoxes is solvable in polynomial time.

We now present the algorithm for Concept Abduction, which also uses the tableaux rules previously defined.

Algorithm $abduce$
input: $\mathcal{ALN}$ concepts C, D, acyclic TBox $\mathcal{T}$
output: concept H (hypotheses)
begin
  compute a complete tableau τ for $\mathcal{T}$, $C \in \mathbf{T}(x), D \in \mathbf{F}(x)$
  if τ is closed **then**
    /* no abduction needed */
    return $H := \top$
  else
    choose(*) a set of pairs $\mathcal{H} := \{\langle C_i, x_i \rangle\}$ and
    let $H := \sqcap_{\langle C_i, x_i \rangle \in \mathcal{H}} \forall roles(x_i).C_i$
    such that (1) every open branch in τ contains at least
        one constraint $C_i \in \mathbf{F}(x_i)$ from $\mathcal{H}$
        (2) $C \sqcap H$ is satisfiable in $\mathcal{T}$
    return H
end

Theorem 5 The concept H returned by the Algorithm $abduce$ is a solution of the CAP $\langle C, D, \mathcal{T} \rangle$.

Proof. Let τ be the tableau built by *abduce*. The tableau starting from $C \sqcap H \in \mathbf{T}(1), D \in \mathbf{F}(1)$ is τ, plus the constraints signed **T** from H. Hence, it is closed. Hence, $\mathcal{T} \models C \sqcap H \sqsubseteq D$. Regarding the condition $C \sqcap H$ satisfiable in $\mathcal{T}$, it is enforced by Condition (2) in the choice of $\mathcal{H}$. □

Condition (2) is necessary in $abduce$, since heterogeneous clashes could be formed also by contradicting an axiom in $\mathcal{T}$. In that case, although it still holds $C \sqcap H \sqsubseteq D$ in $\mathcal{T}$, the subsumption trivially holds since $C \sqcap H \equiv \bot$. We conclude the section by showing that our Algorithm $abduce$ puts an upper bound to Concept Abduction that meets the lower bound proved in the previous section.

Theorem 6 Let $\mathcal{P} = \langle C, D, \mathcal{T} \rangle$ a Concept Abduction Problem, where C, D are concepts in $\mathcal{ALN}$, $\mathcal{T}$ is a bushy TBox in $\mathcal{ALN}$ and k is an integer. Deciding whether there exists a solution of length k in $SOLCAP_{\leq}(\mathcal{P})$ is NP-complete.

Proof. Hardness was shown in Thm. 1. Membership in NP is proved by the correctness of Algorithm $abduce$, since it is sufficient to run the algorithm, and guessing in the nondeterministic step (*) a set $\mathcal{H}$ that defines a concept H of length k. □

4 Conclusion

We have shown how Concept Abduction and Concept Contraction for DL $\mathcal{ALN}$ can be performed using prefixed-tableaux. For such DL, we proved optimality of the methods by showing that they meet lower bounds obtained by a complexity analysis. Although devised for a simple DL, we believe that the proposed approach could be easily extended to more expressive DLs.

Acknowledgements

We wish to thank A. Calí and D. Calvanese for fruitful discussions and the anonymous reviewers for helping in improving paper quality. This work was carried out in the framework of projects PON CNOSSO and MS3DI.

References

[1] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel Schneider (eds.). *The Description Logic Handbook*. Cambridge University Press, 2003.

[2] F. Baader, B. Hollunder, B. Nebel, H. Profitlich, and E. Franconi. An empirical analysis of optimization techniques for terminological representation systems or "making KRIS get a move on". In *KR'92*, pages 270–281.

[3] A. Borgida and P. F. Patel-Schneider. A Semantics and Complete Algorithm for Subsumption in the CLASSIC Description Logic. *J. of Artificial Intelligence Research*, 1:277–308, 1994.

[4] Martin Buchheit, Francesco M. Donini, Werner Nutt, and Andrea Schaerf. A refined architecture for terminological systems: Terminology = schema + views. *Artif. Intell.*, 99(2):209–260, 1998.

[5] D. Calvanese. Reasoning with inclusion axioms in description logics. In *ECAI'96*, pages 303–307. John Wiley & Sons, 1996.

[6] M. Cialdea Mayer and F. Pirri. Modal propositional abduction. *Journal of the IGPL*, 3(6):99–117, 1995.

[7] S. Colucci, T. Di Noia, E. Di Sciascio, F.M. Donini, and M. Mongiello. Concept abduction and contraction in description logics. In *DL 2003*, 2003.

[8] T. Di Noia, E. Di Sciascio, F.M. Donini, and M. Mongiello. A system for principled matchmaking in an electronic marketplace. In *WWW'03*. ACM, 2003.

[9] T. Di Noia, E. Di Sciascio, F.M. Donini, and M. Mongiello. Abductive matchmaking in description logics. In *IJCAI 2003*, 2003.

[10] F. M. Donini, M. Lenzerini, D. Nardi, and W. Nutt. The complexity of concept languages. *Information and Computation*, 134:1–58, 1997.

[11] F.M. Donini and F. Massacci. Exptime tableaux for $\mathcal{ALC}$. *Artif. Intell.*, 124:87–138, 2000.

[12] P. Gardenfors. *Knowledge in Flux*. Mit Press, Bradford Book, 1988.

[13] I. Horrocks. Using an expressive description logic: FaCT or fiction? In *KR'98*, pages 636–645.

[14] I. Horrocks and U. Sattler. Ontology reasoning in the SHOQ(D) description logic. In *IJCAI 2001*, pages 199–204, 2001.

[15] I. Horroks, P. Patel-Schneider, and F. van Harmelen. From shiq to rdf to owl: The making of a web ontology language. *Journal of Web Semantics*, 1(1):7–26, 2003.

[16] M. Klein, J. Broekstra, D. Fensel, F. van Harmelen, and I. Horrocks. *Ontologies and schema languages on the web*. MIT Press, 2003.

[17] L. Li and I. Horrocks. A software framework for matchmaking based on semantic web technology. In *WWW'03*. ACM Press, 2003.

[18] B. Nebel. Terminological reasoning is inherently intractable. *Artif. Intell.*, 43:235–249, 1990.

[19] J.A. Reggia, D.S. Nau, and Y. Wang. Diagnostic expert systems based on a set covering model. *International Journal on Man Machine Studies*, 19:437–460, 1983.

[20] R. M. Smullyan. *First-Order Logic*. Springer-Verlag, New York inc., 1968.

[21] D. Trastour, C. Bartolini, and C. Priest. Semantic web support for the business-to-business e-commerce lifecycle. In *WWW'02*, pages 89–98. ACM, 2002.