Tableau Calculi for Description Logics Revision*

Martín O. Moguillansky   Marcelo A. Falappa

Consejo de Investigaciones Científicas y Técnicas (CONICET)
Laboratorio de Investigación y Desarrollo en Inteligencia Artificial (LIDIA)
Departamento de Ciencias e Ingeniería de la Computación (DCIC)
Universidad Nacional del Sur (UNS)
Av. Alem 1253 - (B8000CPB) Bahía Blanca - Argentina
PHONE/FAX: (+54)(291)459-5136
E-MAIL: mom@cs.uns.edu.ar   mfalappa@cs.uns.edu.ar

Abstract

Focusing on the Ontology Change problem, we consider an environment where *Description Logics* (DLs) are the logical formalization to express knowledge bases, and the integration of distributed ontologies is developed under new extensions and modifications of the *Belief Revision* theories yielded originally in [2]. When using tableaux algorithms to reason about DLs, new information is yielded from the models considered in order to achieve knowledge satisfiability. Here a whole new theory have to be reinforced in order to adapt belief revision definitions and postulates to properly react over beliefs on extensions generated from these DL's reasoning services. In this text we give a brief background of these formalisms and comment the research lines to be taken in our way to this goal.

1 Introduction

Our main research interest relays in topics like *Ontology Integration* and *Ontology Merging* [3], for what we propose to use theory change formalizations in order to join consistently two terminologies, redefining or reinforcing sub-concepts. But following the reasoning methods exposed for DLs, like satisfiability, solved by tableaux algorithms originally defined in [4], a new area of interest arises. A new set of extensions is obtained from the models considered during the execution of the DL reasoning service. Here, is imperative to redefine the formalizations of the theory change exposed in [5] in order to revise beliefs on each extension and transitively in the knowledge base itself. Those research lines here exposed are a consequence of our previous research works cited in [6], [7] and [5].

The remainder of this paper is disposed as follows. Section 2 gives a brief description of tableau-based algorithms behavior by achieving satisfiability of two very simple DL examples. Section 3 briefly summarizes the kernel contractions of the theory change formalism. Section 4 explains some of the preliminary results obtained and the intended research lines to be followed in order to achieve formal and final results on our investigation. Finally, section 5 concludes making an analysis of this brief description and raises some new lines of investigation to be followed.

*This article assumes some extra knowledge on description logics and reasoning services from the reader. For a more exhaustive reading he may refer to [1].

Partially financed by CONICET (PIP 5050), Universidad Nacional del Sur (PGI 24/ZN11) and Agencia Nacional de Promoción Científica y Tecnológica (PICT 2002 Nro 13096).

2 DLs Reasoning Algorithms

Relevant inference problems usually are reduced to the consistency problem for ABoxes, provided that the DL at hand allows for conjunction and negation. However, for those description languages of DL systems that do not allow for negation, subsumption of concepts can be computed by so-called *structural subsumption algorithms*, *i.e.*, algorithms that compare the syntactic structure of (possibly normalized) concept descriptions.

While they are usually very efficient, they are only complete for rather simple languages with little expressivity. In particular, DLs with (full) negation and disjunction cannot be handled by structural subsumption algorithms. For such languages, so-called *tableau-based algorithms* have turned out to be very useful.

2.1 Basics for Tableau Algorithms

Instead of directly testing subsumption of concept descriptions, these algorithms use negation to reduce subsumption to (un)satisfiability of concept descriptions: $C \sqsubseteq D$ ***iff*** $C \sqcap \neg D$ ***is unsatisfiable.***

We illustrate the underlying ideas by two simple examples taken from [8]. Let A, B be concept names, and let R be a role name. As a first example, assume that we want to know whether $(\exists R.A) \sqcap (\exists R.B)$ is subsumed by $\exists R.(A \sqcap B)$. This means that we must check whether the concept description $C = (\exists R.A) \sqcap (\exists R.B) \sqcap \neg(\exists R.(A \sqcap B))$ is unsatisfiable.

Pushing all negation signs as far as possible into the description yields $C_0 = (\exists R.A) \sqcap (\exists R.B) \sqcap \forall R.(\neg A \sqcup \neg B)$, which is in negation normal form, *i.e.*, negation occurs only in front of concept names.

Then, we try to construct a finite interpretation $\mathfrak{I}$ such that $C_0^{\mathfrak{I}} \neq \emptyset$. This means that there must exist an individual in $\Delta^{\mathfrak{I}}$ that is an element of $C_0^{\mathfrak{I}}$. The algorithm just generates such an individual, say b, and imposes the constraint $b \in C_0^{\mathfrak{I}}$ on it, this means that b must satisfy all the three interpreted conjunctions that composes C_0.

From $b \in (\exists R.A)^{\mathfrak{I}}$ we can deduce that there must exist an individual c such that $(b, c) \in R^{\mathfrak{I}}$ and $c \in A^{\mathfrak{I}}$. Analogously, $b \in (\exists R.B)^{\mathfrak{I}}$ implies the existence of an individual d with $(b, d) \in R^{\mathfrak{I}}$ and $d \in B^{\mathfrak{I}}$. In this situation, one should not assume that $c = d$. Thus:

- *For any existential restriction the algorithm introduces a new individual as role filler, and this individual must satisfy the constraints expressed by the restriction.*

Since b must also satisfy the value restriction $\forall R.(\neg A \sqcup \neg B)$, and c, d were introduced as R-fillers of b, we obtain the additional constraints $c \in (\neg A \sqcup \neg B)^{\mathfrak{I}}$ and $d \in (\neg A \sqcup \neg B)^{\mathfrak{I}}$. Thus:

- *The algorithm uses value restrictions in interaction with already defined role relationships to impose new constraints on individuals.*

Now c might be such that $c \in (\neg A)^{\mathfrak{I}}$ or $c \in (\neg B)^{\mathfrak{I}}$. Assume the first possibility leads to an obvious contradiction, so we must choose the second one $c \in (\neg B)^{\mathfrak{I}}$. Analogously, we must choose $d \in (\neg A)^{\mathfrak{I}}$ in order to satisfy the constraint $d \in (\neg A \sqcup \neg B)^{\mathfrak{I}}$ without creating a contradiction to $d \in B^{\mathfrak{I}}$. Thus:

- *For disjunctive constraints, the algorithm tries both possibilities in successive attempts. It must backtrack if it reaches an obvious contradiction, i.e., if the same individual must satisfy constraints that are obviously conflicting.*

In the example, we have now satisfied all the constraints without encountering an obvious contradiction. This shows that C_0 is satisfiable, and thus $(\exists R.A) \sqcap (\exists R.B)$ is not subsumed by $\exists R.(A \sqcap B)$. The interpretation generated by the algorithm is $\Delta^{\mathfrak{I}} = \{b, c, d\};\ R^{\mathfrak{I}} = \{(b, c), (b, d)\};\ A^{\mathfrak{I}} = \{c\}$ and $B^{\mathfrak{I}} = \{d\}$.

In our second example, we now want to know whether $(\exists R.A) \sqcap (\exists R.B) \sqcap \leqslant 1R$ is subsumed by $\exists R.(A \sqcap B)$. The tableau-based satisfiability algorithm first proceeds as above, with the only difference that there is the additional constraint $b \in (\leqslant 1R)^{\mathcal{I}}$. In order to satisfy this constraint, the two R-fillers c, d of b must be identified with each other. Thus:

- *If an at-most number restriction is violated then the algorithm must identify different role fillers.*

The individual $c = d$ must belong to both $A^{\mathcal{I}}$ and $B^{\mathcal{I}}$, which together with $c = d \in (\neg A \sqcup \neg B)^{\mathcal{I}}$ always leads to a clash. Thus, the search for a counterexample to the subsumption relationship fails, and the algorithm concludes that $(\exists R.A) \sqcap (\exists R.B) \sqcap \leqslant 1R \sqsubseteq \exists R.(A \sqcap B)$.

3 Kernel Contractions

The **Kernel Contraction** operator is applicable to belief bases and belief sets. It consist of a contraction operator capable of the selection and elimination of those beliefs in K that contribute to infer α.

Definition 3.1 - [9]: Let K be a set of sentences and α a sentence. The set $K^{\perp\!\!\perp}\alpha$, called *set of kernels* is the set of sets K' such that (1) $K' \subseteq K$, (2) $K' \vdash \alpha$, and (3) if $K'' \subset K'$ then $K'' \nvdash \alpha$. The set $K^{\perp\!\!\perp}\alpha$ is also called *set of* α*-kernels* and each one of its elements are called α*-kernel*.

For the success of a contraction operation, we need to eliminate, at least, an element of each α-kernel. The elements to be eliminated are selected by an **Incision Function**.

Definition 3.2 - [9]: Let K be a set of sentences and "σ" be an *incision function* for it such that for any sentence α it verifies, (1) $\sigma(K^{\perp\!\!\perp}\alpha) \subseteq \bigcup(K^{\perp\!\!\perp}\alpha)$ and (2) If $K' \in K^{\perp\!\!\perp}\alpha$ and $K' \neq \emptyset$ then $K' \cap \sigma(K^{\perp\!\!\perp}\alpha) \neq \emptyset$.

Once the incision function was applied, we must eliminate from K those sentences that the incision function selects, *i.e.*, the new belief base would consist of all those sentences that were not selected by σ.

Definition 3.3 - [9]: Let K be a set of sentences, α a sentence, and $K^{\perp\!\!\perp}\alpha$ the set of α-kernels of K. Let "σ" be an incision function for K. The operator " $-_\sigma$ ", called *kernel contraction determined by* "σ", is defined as, $K -_\sigma \alpha = K \backslash \sigma(K^{\perp\!\!\perp}\alpha)$.

Finally, an operator " $-$ " is a kernel contraction operator for K if and only if there exists an incision function "σ" such that $K - \alpha = K -_\sigma \alpha$ for all sentence α.

4 First Advances and Future Research Lines

As seen in section 2.1, new beliefs may be generated from a satisfiability process applied following a tableau-based algorithm. It would be interesting so, to check out what may happen to the theory change definitions cited in section 3 applied to these extensions.

Let first give an example borrowed from [10] in order to understand more precisely the extensions obtained during a reasoning process. Let Σ be a knowledge base composed by $FRIEND(john, susan)$, $FRIEND(john, andrea)$, $LOVES(susan, andrea)$, $LOVES(andrea, bill)$, $Female(susan)$, $\neg Female(bill)$.

Now we want to know if is there some not $Female$ loving a $Female$ who is $FRIEND$ of $john$. This is a query $\Sigma \models^? \alpha$ such that α is $\exists FRIEND.(Female \sqcap (\exists LOVES.\neg Female))(john)$. Following the given tableau specifications, note that we have two different possibilities in order to achieve satisfiability of α, this is, two interpretations (models) named $\mathcal{M}_1$ and $\mathcal{M}_2$ satisfying the query $\Sigma \models \alpha$, where $\neg Female(andrea) \subseteq \mathcal{M}_1$ and $Female(andrea) \subseteq \mathcal{M}_2$.

Let analyze the yielding situation with respect to the α-kernels for each model.

1. Considering $\mathcal{M}_1$ there is only one proof set K' for α, this is $\Sigma^{\perp\!\!\perp}\alpha = \{FRIEND(john, susan), Female(susan), LOVES(susan, andrea), \neg Female(andrea)\}$
2. Considering $\mathcal{M}_2$ there is also only one proof set K' for α, $\Sigma^{\perp\!\!\perp}\alpha = \{FRIEND(john, susan), Female(andrea), LOVES(andrea, bill), \neg Female(bill)\}$

Now if we verify the restrictions given in definition 3.1 for α-kernels, we realize that the first restriction $K' \subseteq \Sigma$ is not verified due to the assumption taken during the reasoning service, *i.e.*, the beliefs adopted from each model $\mathcal{M}_1$ and $\mathcal{M}_2$ that are not part of the explicit knowledge base Σ. In order to facilitate further reference to this beliefs, let call $K'_{\mathcal{M}_1} = \{\neg Female(andrea)\}$ and $K'_{\mathcal{M}_2} = \{Female(andrea)\}$.

Note that we now have redefined the K' proof set as $K' = K'_\Sigma \cup K'_\mathcal{M}$, where K'_Σ is the proof subset that is part of the explicit KB Σ, and $K'_\mathcal{M}$ that extends it consistently, is the set of beliefs assumed in $\mathcal{M}$, *i.e.*, that are outside the KB Σ. Now we propose to formally redefine the definition of the α-kernels in the following way:

Definition 4.1 - Extended Set of α-kernels : Let Σ be a knowledge base and α a sentence. The set $\Sigma^{\perp\!\!\perp}\alpha$, called *set of kernels* is the set of sets K' such that (1) $K' = K'_\Sigma \cup K'_\mathcal{M}$, where $K'_\Sigma \subseteq \Sigma$, $K'_\mathcal{M} \nsubseteq \Sigma$, and $K' \not\vdash \perp$, (2) $K' \vdash \alpha$, and (3) if $K'' \subset K'$ then $K'' \nvdash \alpha$. The set $\Sigma^{\perp\!\!\perp}\alpha$ is also called *set of α-kernels* and each one of its elements are called *α-kernel*.

Following with the belief revision connection, let think about an incision function selecting beliefs from $\Sigma^{\perp\!\!\perp}\alpha$ in order to achieve a contraction of Σ by α, *i.e.*, we want to get $\Sigma -_\sigma \alpha$. In this case and following the original definitions given in definition 3.2, an incision function σ may select beliefs from each K' in $\Sigma^{\perp\!\!\perp}\alpha$, this means that we may have beliefs taken from K'_Σ and also from $K'_\mathcal{M}$. But now, special care we must have with those beliefs coming from outside the KB, *i.e.*, taken from $K'_\mathcal{M}$, because they are *"assumed"* beliefs and are not explicitly specified in the KB. In this sense we propose an extended incision function as,

Extended Incision Function: Let Σ be a knowledge base and "σ" be an *incision function* for it such that for any sentence α it verifies, (1) $\sigma(\Sigma^{\perp\!\!\perp}\alpha) = \sigma(K_\Sigma) \cup \sigma(K_\mathcal{M})$, where $\sigma(K_\Sigma) \subseteq \bigcup(K'_\Sigma)$ selects beliefs belonging to the KB, and $\sigma(K_\mathcal{M}) \subseteq \bigcup(K'_\mathcal{M})$ is an *"assumed"* subset from outside the KB, and (2) If $K' \in \Sigma^{\perp\!\!\perp}\alpha$ and $K' \neq \emptyset$ then $K' \cap \sigma(\Sigma^{\perp\!\!\perp}\alpha) \neq \emptyset$.

A contraction as defined in definition 3.3 only retracts the selected beliefs in the incision function from the KB, but now we have a different situation due to possible selected beliefs not belonging to the KB, *i.e.*, beliefs inside $\sigma(K_\mathcal{M})$. This type of beliefs need a special care in order to be "contracted" from the KB, because they are not inside the KB, we really need not to contract them from the KB, but revise their opposites. In this sense, we propose a new hybrid operation as follows,

Hybrid Contraction Determined by σ: Let Σ be a knowledge base, α a sentence, and $\Sigma^{\perp\!\!\perp}\alpha$ the extended set of α-kernels of Σ. Let "σ" be an extended incision function for Σ. The operator " $\circledast_\sigma$ ", referred as *hybrid contraction determined by* "σ", is defined as, $\Sigma \circledast_\sigma \alpha = (\Sigma \backslash \sigma(K_\Sigma)) * \neg\sigma(K_\mathcal{M})$.

Finally, an operator " $\circledast$ " is an hybrid contraction operator for Σ if and only if there exists an extended incision function "σ" such that $\Sigma \circledast \alpha = \Sigma \circledast_\sigma \alpha$ for all sentence α.

5 Conclusions and Future Work

Part of the formalization of the relying theory change definitions into a more general flavor in order to match extra-classic logics like DLs, have being done in [3], where the generalized postulates have being defined and their pertinent analysis developed.

Considering DLs reasoning services like tableau-based algorithms to solve satisfiability, not only sets us up in a more direct theory formalization, but also permits us to work purely description languages without need to translate beliefs to fragments of classic first order logic like we have done before in [5].

Tableaux algorithms are nowadays probably the most important reasoning algorithms used in the area. A distinctive feature of this reasoning service is the way it reasons over incomplete information, inferring new beliefs from knowledge in order to prove clauses' satisfiability. By this, we have a totally different way to reason about knowledge, due to a multiple generation of extensions.

When a tableaux algorithm proves satisfiability, it generates models for the knowledge base adding beliefs to it. But on different models, we have different extensions potentially inconsistent one from the other. Here we have given a first glimpse to what should be the extended revision of beliefs on extensions besides the knowledge base revisions we already have. A totally different reasoning dimension is generated by thinking on extensions if one thinks this is a way of predicting future information.

Another interesting research line to be studied is the incision function and its reaction over beliefs outside the KB, or what we have called "assumed" beliefs.

References

[1] M. Falappa M. Moguillansky. Tableau calculi for revision of description logics knowledge bases. Computing science technical report, Universidad Nacional del Sur (UNS), Departamento de Cs. e Ing. de Computación (DCIC), Laboratorio de Investigación y Desarrollo en Inteligencia Artificial (LIDIA), Bahía Blanca, Argentina, 2007. To be posted.

[2] Peter Gardenfors Carlos Alchourrón and David Makinson. On the logic of theory change: Partial meet contraction and revision functions.

[3] Giorgos Flouris. On belief change and ontology evolution. *Doctoral Dissertation, Department of Computer Science, University of Crete*, February 2006.

[4] Manfred Schmidt-Schauß and Gert Smolka. Attributive concept descriptions with complements. *Artificial Intelligence*, pages 48(1):1–26, 1991.

[5] M. Falappa M. Moguillansky. Towards a non monotonic description logics model. *XII Congreso Argentino de Ciencias de la Computación, CACIC'2006, Universidad Nacional de San Luis*, pages 1354–1365, ISBN 950–609–050–5, Octubre de 2006.

[6] M. Falappa M. Moguillansky. Aplicación de operaciones de cambio en sistemas basados en conocimiento. *IX Congreso Argentino de Ciencias de la Computación, CACIC'2003, Universidad Nacional de La Plata*, pages 1490–1501, Octubre de 2003.

[7] M. Falappa M. Moguillansky. On the use of belief revision to merge description logic terminologies. *WICC'2006, Universidad de Moron*, pages 57–62, Junio de 2006.

[8] W. Nutt F. Baader. Basic description logics. *In the Description Logic Handbook, Cambridge University Press*, pages 47–100, 2002.

[9] S. O. Hansson. Kernel contraction. *The Journal of Symbolic Logic*, pages 59:845–859, 1994.

[10] Enrico Franconi. Propositional description logics. *From his course Description Logics at http://www.inf.unibz.it/~franconi/dl/course/2006/slides/prop-DL/propositional-dl.pdf, dictated during argentinean springtime at Universidad Nacional del Sur, Bahía Blanca*, 2006.