Analytic Tableaux for Default Logics

Vincent Risch

Laboratoire de Recherche en Informatique, U.R.A. C.N.R.S. 410,
Université de Paris-Sud, Bât. 490,
91405 Orsay Cédex, France
e-mail : risch@lri.fr

Abstract

Following the approach [Schwind, Risch, 91] to Reiter's default logic, this paper attempts to define a general tableaux-based framework for the computation of the extensions for several known default logics. The idea is to qualify the differences between the variants under consideration inside a single process that can switch easily from one variant to another. Indeed, the approach is related to the idea of the compilation of default theories.

1 Introduction

Since [Reiter, 80] introduced default logic (DL for short), many variants of this appealing theory have been proposed. [Lukaszewicz, 88] introduced a variant where, as opposed to Reiter's DL, an extension always exists. [Brewka, 91] introduced a cumulative version of DL which has been reinterpreted by [Schaub, 91]. The present paper is an attempt to define a general tableaux-based framework which allows us both to compute the extensions of a default theory, and to "jump" easily from one variant to another. Moreover, once the extensions of a default theory are computed, we should perform a minimum number of operations in order to verify whether a given formula belongs to an extension. Thus our approach can be considered as a kind of compiling method for default logics, where compiling means obtaining the extensions of a default theory in a tractable form. The plan is as follows: in part 2 below, we shall briefly recall some basic features of DL and give alternative characterizations of an extension for the variants under consideration. Part 3 defines analytic tableaux for first order logic and show some of their properties. In part 4, the computation of the extensions is described and some examples are given. Following [Besnard, Schaub, 92] we refer to Reiter's theory as classical default logic, and to Lukaszewics's theory as justified default theory. We are also concerned with constrained DL [Schaub, 91].

2 Default theories

As defined by [Reiter, 80], a closed default theory is a pair (W, D) where W is a set of closed first order sentences and D a set of default rules. A default rule has the form

$$\frac{\alpha : \beta}{\gamma}$$

where α, β and γ are closed first order sentences. α is called prerequisite, β justification and γ consequent of the default. $PREREQ(D)$, $JUST(D)$ and $CONS(D)$ are respectively the sets of all prerequisites, justifications and consequents that come from defaults in a set D. Whenever one of these sets is a singleton, we may identify it with the single element it contains. For instance, we prefer to consider $PREREQ(\{\frac{\alpha : \beta}{\gamma}\})$ as an element rather than a set. The following definition shows us how the use of a default is related to its prerequisite (cf. [Schwind, 90]):

Definition 1 (Schwind, 90) *A set D of defaults is* grounded *in W iff for all $d \in D$ there is a finite sequence $d_0, \ldots, d_k$ of elements of D such that*

- $PREREQ(\{d_0\}) \in Th(W)$,
- *for* $1 \leq i \leq k-1$, $PREREQ(\{d_{i+1}\}) \in Th(W \cup CONS(\{d_0, \ldots, d_i\})$,

and $d_k = d$.

An *extension* of a default theory is usually defined as a smallest fixed point of a set of formulas. It contains W, is deductively closed, and the defaults whose consequents belong to the extension verify a property which actually allows them to be used. The manner in which this property is considered is related to the variant of DL under consideration. In what follows, we directly give the characterizations obtained by [Schwind, Risch, 91], [Risch, 91], [Risch, 92] for the extensions in the sense of [Reiter, 80], [Lukaszewicz, 88], [Schaub, 91] respectively. The first are called *R-extension* (for Reiter's extensions), the second *j-extensions* (for justified extensions), and the third *c-extensions* (for constrained extensions).

Theorem 1 (Schwind, Risch, 91)(Risch, 91) (Risch, 92) *Let be $\Delta = (W, D)$, a default theory.*

- *E is an R-extension for Δ iff there is D' a grounded subset of D such that $E = Th(W \cup CONS(D'))$, and $\forall d \in D, d = \frac{\alpha : \beta}{\gamma}$:*
  - *(i) If $d \in D'$ then $\alpha \in E$ and $\neg\beta \notin E$,*
  - *(ii) If $d \notin D'$ then $\alpha \notin E$ or $\neg\beta \in E$.*
- *E is a j-extension with respect to F for Δ iff there is D' a grounded subset of D, maximal such that $E = Th(W \cup CONS(D'))$, $F = JUST(D')$, and $\forall d \in D'$, $d = \frac{\alpha : \beta}{\gamma}$:*
  - *(i) If $d \in D'$ then $\alpha \in E$ and $\neg\beta \notin E$.*
- *E is a c-extension with respect to C for Δ iff there is D' a grounded subset of D, maximal such that $E = Th(W \cup CONS(D'))$, $C = Th(W \cup JUST(D') \cup CONS(D'))$, and $\forall d \in D'$, $d = \frac{\alpha : \beta}{\gamma}$:*
  - *(i') If $d \in D'$ then $\alpha \in E$ and $E \cup JUST(D')$ consistent.*

Remark: Clearly, both conditions *(i)* and *(ii)* imply the maximality of D'. Hence, the only difference between Reiter's and Lukaszewicz's approach is in the behavior of the defaults that do not participate in the construction of an extension. These defaults have to verify condition *(ii)* in Reiter's DL. Our approach sheds light on the intuition which stands behind Lukaszewicz's variant. In the latter, we are never allowed to revise a justification used for deriving the consequent of a default. According to Lukaszewicz, it is only in this precise way that we may speak of "justification" in a correct sense. Indeed, note that the only difference between Lukaszewicz's and Schaub's approaches concerns the consistency of the set of justifications related to an extension. Note that default reasoning is decidable on condition that Th is defined on a decidable langage.

3 Analytic Tableaux

The theorem prover we have used, called TP, is based on the analytic tableaux method in the style of [Smullyan, 68] and is described in [Schwind, 90]. The tableau method is based on an iterative process that simultaneously parses a formula and looks for models to satisfy it. So when we show that a formula is a theorem, this is the same as verifying that its negation has no model. Let us consider the usual Smullyan's classes of formulas (cf.[Smullyan, 68], [Fitting, 90]):

α	α_1	α_2	β	β_1	β_2
$f \wedge g$	f	g	$f \vee g$	f	g
$\neg(f \vee g)$	$\neg f$	$\neg g$	$\neg(f \wedge g)$	$\neg f$	$\neg g$
$\neg(f \rightarrow g)$	f	$\neg g$	$f \rightarrow g$	$\neg f$	g

γ	$\gamma(p), p$ any parameter	δ	$\delta(p), p$ Skolem parameter
$(\forall x) f$	$f[x/p]$	$(\exists x) f$	$f[x/p]$
$\neg(\exists x) f$	$\neg f[x/p]$	$\neg(\forall x) f$	$\neg f[x/p]$

where:

- parameters are free occurrences of variables or functions whose arguments are parameters;
- Skolem parameters are in the form $\varphi(p_1, \ldots, p_n)$ where φ is any skolem function, and $p_1, \ldots, p_n$ are *already used* parameters.

TP can be defined recursively as a mapping between sets of formulas and sets of literals as follows:

$$\begin{aligned} TP(F) &= \{F\} \text{ if } F \text{ is a set of literals;} \\ TP(F) &= TP(F' \cup \{f\} \text{ if } \neg\neg f \in F \text{ and} \\ &\quad F' = F \setminus \{f\}; \end{aligned}$$

$$
\begin{aligned}
TP(F) &= TP(F' \cup \{\alpha_1\} \cup \{\alpha_2\}) \text{ if } \alpha \in F \text{ and} \\
&\quad F' = F \setminus \{\alpha\}; \\
TP(F) &= TP(F' \cup \{\beta_1\}) \cup TP(F' \cup \{\beta_2\}) \\
&\quad \text{if } \beta \in F \text{ and } F' = F \setminus \{\beta\}; \\
TP(F) &= TP(F' \cup \{\gamma(p)\} \text{ if } \gamma \in F \text{ and} \\
&\quad F' = F \setminus \{\gamma\}; \\
TP(F) &= TP(F' \cup \{\delta(p)\} \text{ if } \delta \in F \text{ and} \\
&\quad F' = F \setminus \{\delta\};
\end{aligned}
$$

The analytic tableau for a formula corresponds to its disjunctive normal form. Each set represents the conjunct of its elements and the tableau represents the disjunct of its elements e.g.

$$
\begin{aligned}
TP(\{(p \vee q) \wedge r\}) &= \{\{p, r\}, \{q, r\}\} \\
&= TP(\{(p \wedge r) \vee (q \wedge r)\}).
\end{aligned}
$$

Definition 2 *A set of literals is* closed[1] *if it contains two opposite literals (i.e. ℓ and $\neg\ell$). It is* open *otherwise. A set of sets of literals is closed if each of its elements is closed, and open otherwise.*

The fundamental property of TP for theorem proving is the following completeness theorem:

Theorem 2 (Smullyan, 68) *f is a theorem iff $TP(\{\neg f\})$ is closed.*

We consider TP not only as a theorem prover (or consistency checker) for first-order formulas but also as an application which has useful properties for formulas and formulas sets. We will frequently use the following properties of TP:

Lemma 1 *$TP(\{f \wedge g\}) = \{X \cup Y : X \in TP(\{f\})$ and $Y \in TP(\{g\})\}$.*

This will enable us to consider conjunction as an operation upon sets of sets of literals. Lemma 1 gives rise to the following notation:

$$f \otimes g = \{X \cup Y : X \in TP(f) \text{ and } Y \in TP(\{g\})\}.$$

Thus $TP(\{f \wedge g\}) = TP(\{f\}) \otimes TP(\{g\})$. TP will only have to compute the image of each formulas f and f *once*, and the image of $TP(\{f \wedge g\})$ will be obtained using the operation $\otimes$ whithout having to compute the whole conjunct.

As an immediate consequence of the completeness theorem, we have

[1] not to be confused with a *closed* first-order formula i.e. a formula with no free variables.

Corollary 1 (Schwind, 90) *Let Γ be a finite set of formulas and f a formula. $f \in Th(\Gamma)$ iff $TP(\Gamma) \otimes TP(\{\neg f\})$ is closed.*

Remark: TP frequently contains closed sets or sets which contain other sets in TP. In this case we are sometimes interested in a minimal form (especially when a TP is not closed), defined by

$$
\begin{aligned}
TS(M) &= TP(M) \setminus \{X : X \in TP(M) \text{ and} \\
&\quad X \text{ closed or } \exists Y \in TP(M) \\
&\quad \text{such that } Y \subset X \text{ and } X \neq Y\}.
\end{aligned}
$$

As stated above, the set yielded by TP for a formula f is the disjunctive normal form of that formula. Let us write φ to denote the mapping between $TP(f)$ and the corresponding disjunctive normal form $\varphi(TP(\{f\}))$. Clearly: $\varphi(TS(M)) \leftrightarrow \varphi(TP(M))$, because of $(A \wedge B) \vee A \leftrightarrow A$ and $(A \wedge \neg A \wedge B) \vee C \leftrightarrow C$. Very often this can considerably shorten a tableau.

Determining the class of formulas that a tableau prover can decide is a difficult question which is beyond the scope of this paper. In what follows, we will consider only decidable subclasses of a first-order langage e.g. propositional formulas or Horn clauses.

4 Computing extensions

We consider that the tableaux method is very interesting for computing extensions for two main reasons: it produces models in a natural way, and these models are close to minimal models. Another argument in favour of tableaux concerns the fundamental property: a tableau for f is closed iff $\neg f$ is a theorem, i.e. a tableau is open iff $\neg f$ is a theorem, which means that f is consistent. Nonmonotonic reasoning typically requires us to reason on consistency versus inconsistency. A tableau provides a straightforward means of transforming inconsistent sets into consistent ones by *removing* the literals responsible for contradiction within these sets. Similarly, the $\otimes$ operator appears to be very tractable, since using the fundamental property of TP, we obtain (cf. theorem 1):

(i) If $d \in D'$ then $TP(\{\neg\alpha\}) \otimes TS(W) \otimes TP(CONS(D')$ closed and $TP(\{\beta\} \otimes TS(W) \otimes TP(CONS(D')$ open;

(ii) If $d \notin D'$ then $TP(\{\neg\alpha\} \otimes TS(W) \otimes TP(CONS(D')$ open or $TP(\{\beta\} \otimes TS(W) \otimes TP(CONS(D')$ closed;

(i') if $d \in D'$ then $TP(\{\neg\alpha\} \otimes TS(W) \otimes TP(CONS(D')$ closed and $TP(JUST(D') \otimes TS(W) \otimes TP(CONS(D')$ open;

where D' is the set of defaults involved in the constitution of the extension $Th(W \cup CONS(D'))$. Note that *a priori* we are not allowed to use TS in preference to TP with $CONS(D')$, since otherwise some defaults could be forgotten by the process (e.g. the consequent of a given default is subsumed by the consequent of an other default which actually cannot be used).

The idea for our method is taken from what [Schwind, Risch, 91] have proposed for Reiter's default theories. Consider a j-extension for a default theory $\Delta = (W, D)$. It is, roughly speaking, the set of theorems over the union of W and a set of default consequences. It is a maximal grounded set of this kind in the sense that the addition of any other default consequence is not possible because the prerequisite or the condition justification does not hold. Thus, consider $W \cup CONS(D)$. There are two possible cases:

- $Th(W \cup CONS(D))$ is consistent i.e. $\Gamma = TS(W) \otimes TP(CONS(D))$ is open. Then D' is any maximal grounded subset of D such that every default verifies the prerequisite and justification condition of *(i)*. So every default for which such conditions are not verified has to be left out of D in order to obtain D'. For the justification condition, this means there are as many j-extensions as ways to remove consequences of defaults and "responsible" for a contradiction inside $Th(W \cup CONS(D) \cup \beta)$ (where $\beta \in JUST(D)$). In other words, there are as many j-extensions as ways to remove literals responsible for a closure inside $\Gamma \otimes TP(\{\beta\})$. In fact, every time we remove the consequence of a default among those incriminated, we generate a maximal subset D' of D.

- $Th(W \cup CONS(D))$ is inconsistent i.e. $\Gamma = TP(W \otimes CONS(D))$ is closed. It is possible to remove the incriminated consequences of defaults (in the other case W itself is inconsistent). Each removal defines a subset D' of D for which *(i)* has to be verified, as above, which means that literals from default consequences and responsible for a closure are removed from Γ. Indeed, note that for any grounded D', the prerequisite condition of *(i)* is verified.

The groundedness of each default implied in a j-extension $Th(W \cup CONS(D'))$ is verified in a recursive way: Choose $d \in D'$, $d = \dfrac{\alpha : \beta}{\gamma}$. If $TP(W \otimes CONS(D' \backslash \{d\})) \otimes \{\neg\alpha\})$ is closed, we have to find a finite sequence $d_0, \ldots, d_n$ of elements of D', $d_i = \dfrac{\alpha_i : \beta_i}{\gamma_i}$ such that $TP(\{\neg\alpha_0\}) \otimes TP(W)$ is closed, $TP(\{\neg\alpha_{i+1}\}) \otimes TP(W) \otimes TP(CONS(\{d_0, \ldots, d_i\})$ is closed for $0 \leq i \leq n-1$ and $d_n = d$. However, if there exist k, $0 \leq k \leq n$, such that $TP(\{\neg\alpha_k\}) \otimes TP(W) \otimes TP(CONS(D' \backslash \{d_k, \ldots, d_n\})$ is open i.e. $\alpha_k \notin Th(W \cup CONS(D' \setminus \{d_k, \ldots, d_n\}))$ the sequence $d_k, \ldots, d_n$ has to be withdrawn from D'. In this way all j-extensions are obtained.

Following theorem 1, R-extensions are j-extensions the corresponding set of defaults verifies *(ii)*. Hence, in order to obtain the R-extensions, for each j-extension we have to deal with the defaults that are not involved in it.

The c-extensions can be computed by transforming the justification-condition of *(i)*, which means requiring the consistency of set of justifications with $Th(W \cup CONS(D'))$, i.e. $TP(JUST(D') \otimes TP(W) \otimes TP(CONS(D')$ open.

In the following examples the formulas obtained from defaults are labelled in order to identify their origin. Thus we know what default we are concerned with at any moment.

Example 1 $\Delta = (W, D)$ where $W = \{A, D\}$, $D = \{\dfrac{A : B \wedge \neg C}{B}, \dfrac{D : C}{C}\}$. $TS(W) = \{\{A, D\}\}$, $TP(CONS(D)) = \{\{B_1, C_2\}\}$ where B_1 and B_2 are the consequents of the first and the second default respectively. Each default yielding a j-extension has to verify *(i)*. Consider D as a possible candidate in order to obtain an extension. However, the justification condition of *(i)* is not true for the first default: $TS(W) \otimes TP(D) \otimes TP(B_1 \wedge \neg C_1) = \{\{A, D, B_1, C_2, \neg C_1\}\}$ closed (where $\neg C_1$ comes from the justification of the first default). Hence, consider the following subsets of D: $D' = \{\dfrac{A : B \wedge \neg C}{B}\}$ and $D'' = \{\dfrac{D : C}{C}\}$. Now, both D' and D'' verify the justification condition of *(i)*: $TS(W) \otimes TP(D') \otimes TP(\{B_1 \wedge \neg C_1\}) = \{\{A, D, B_1, \neg C_1\}\}$ open and

$TS(W) \otimes TP(D') \otimes TP(\{C_2\}) = \{\{A, D, C_2\}\}$ open. Indeed, these two subsets are obviously grounded. Thus D has two j-extensions:

$$E_1 = Th(W_1 \cup CONS(\{\frac{A : B \wedge \neg C}{B}\})),$$
$$E_2 = Th(W_1 \cup CONS(\{\frac{D : C}{C}\})).$$

However, E_1 is not a Reiter's extension since *(ii)* does not hold for the second default: $TS(W) \otimes TP(\{B_1\}) \otimes TP(\{C_2\})$ is open instead of being closed (where C_2 is the justification of the second default). E_2 is a Reiter's extension since $TS(W) \otimes TP(\{C_2\}) \otimes TP(\{B_1 \wedge \neg C_1\})$ is closed, which is expected following *(ii)*.

Example 2 $\Delta = (W, D)$ where $W = \{P, S\}$, $D = \{\frac{: Q \wedge S}{P}, \frac{: Q \rightarrow \neg S}{R}\}$. $TS(W) = \{\{P, S\}\}$, $TP(CONS(D)) = \{\{P_1, \neg R_2\}\}$. First consider D as the possible unique candidate in order to get an extension. The justification condition of *(i)* holds for the both defaults:

- $TS(W) \otimes TP(CONS(D) \otimes TP(\{Q_1 \wedge S_1\}) = \{\{P, S, P_1, \neg R_2, Q_1, S_1\}\}$ is open.
- $TS(W) \otimes TP(CONS(D) \otimes TP(\{Q_2 \rightarrow \neg S_2\}) = \{\{P, S, P_1, R_2, \neg Q_2\}, \{P, S, P_1, R_2, \neg S_2\}\}$ is open;

Since D is obviously grounded, Δ has a single j-extension

$$E = Th(W \cup CONS(\{\frac{: Q \wedge \neg S}{P}, \frac{: Q \rightarrow \neg S}{\neg R}\})).$$

Now, E is not a c-extension because the set of justifications involved in the constitution of E is not consistent:

$$TS(W) \otimes TP(CONS(D) \otimes TP(JUST(D))$$
$$= \{\{P, S, P_1, R_2, Q_1, S_1, \neg Q_2\}, \{P, S, P_1, R_2, Q_1, S_1, \neg S_2\}\}$$

is closed. It can be opened either by removing the literals from the first default or by removing the literals from the second default. Hence we obtain two c-extensions:

$$E_1 = Th(W \cup CONS(\{\frac{: Q \wedge S}{P}\}),$$
$$E_2 = Th(W \cup CONS(\{\frac{: Q \rightarrow \neg S}{R}\}).$$

Example 3 $\Delta = (\emptyset, D)$ where $D = \{\frac{A : B}{B}, \frac{B : A}{A}\}$. The justification condition holds for both defaults of D. However, D is not grounded, neither is $\{\frac{A : B}{B}\}$ or $\{\frac{B : A}{A}\}$. Hence D has the only j-extension $E = \emptyset$.

In the algorithm below the formulas are indexed by the number of each default, as in the previous examples. The variable D_{just} represents the set of defaults that satisfy the justification condition, the variable D_{grnd} the set of grounded defaults, and the variable SD_{just} the set of the D_{just}. The part of the algorithm that tests the grounding of a set of defaults uses a variable S that represents a grounding path.

```
COMPUTE-EXTENSIONS(W, D):
   If W is not consistent
      then PRINT(W is not consistent)
      else For any maximal subset D' in D such
      that TP(W ⊗ CONS(D')) is open do
         D_just := ∅; D_grnd := ∅; SD_just := {D_just};
         JUSTIFICATIONS(D', D_just, SD_just);
         For any D_just in SD_just do
            GROUNDING(D_just, ∅, D_grnd);
            EXTENSIONS(D_just, D_grnd).

JUSTIFICATIONS(D', D_just, SD_just):
   Γ(D') := Th(W ⊗ CONS(D'));
   For any default d = α:β/γ ∈ D' do
      If TP({β}) ⊗ Γ(D') closed then
         For any maximal subset D'' in D' such
         that TP({β}) ⊗ Γ(D'') open, do
            JUSTIFICATIONS(D'', D_just, SD_just);
            If D_just ∉ SD_just
               then Add D_just to SD_just
               else Add d to D_just.

GROUNDING(D', S, D_grnd):
   For any default d = α:β/γ ∈ D' \ D_grnd do
      Γ := Th(W ⊗ CONS(D' \ (S ∪ {d})));
      If ¬α ⊗ Γ closed,
         then If α ∈ Th(W) or comes
         from a grounded set of defaults
            then Add d to D_grnd
            else Let be D_Γ the set of defaults
            of D' \ (S ∪ {d}) whose the consequent
            is used for deriving α, do
```

```
        GROUNDING(D_Γ, S ∪ {d});
        If D_Γ is grounded
            then Add d to D_grnd;
    else Remove the list of defaults S ∪ {d}
    from D'.

EXTENSIONS(D_just, D_grnd):
  If D_grnd is a maximal grounded subset of D then
      PRINT(Th(W ∪ CONS(D_grnd)) is a j-extension
      with respect to JUST(D_grnd)).
```

To switch from Lukaszewicz to Reiter it is sufficient to add the following procedure to EXTENSIONS(D_{just}, D_{grnd}) (within the scope of the "if"-condition):

```
(If D_grnd is a maximal grounded subset of D then)
    ...
      If there exists d  α:β/γ ∈ D \ D_grnd such that
      TP({β}) ⊗ TS(W) ⊗ TP(CONS(D_grnd)) open and
      TP({¬α}) ⊗ TS(W) ⊗ TP(CONS(D_grnd)) closed
          then PRINT(Th(W ∪ CONS(D_grnd)) is
              not an R-extension)
          else PRINT(Th(W ∪ CONS(D_grnd)) is
              an R-extension);
  If none of the j-extensions are R-extensions
      then PRINT((W, D) has no R-extension).
```

To switch from Lukaszewicz to Schaub it is sufficient to add the following procedure to EXTENSIONS(D_{just}, D_{grnd}) (within the scope of the "if"-condition):

```
(If D_grnd is a maximal grounded subset of D then)
    ...
      If TP(JUST(D_grnd) ⊗ TS(W) ⊗ TP(CONS(D_grnd))
      is open
      then PRINT(Th(W ∪ CONS(D_grnd)) is
          a c-extension) with respect to
          Th(W ∪ CONS(D_grnd) ∪ JUST(D_grnd))
      else For D'_grnd any maximal subset of D_grnd
          such that
          TP(JUST(D'_grnd)⊗TS(W)⊗TP(CONS(D'_grnd))
          is open do GROUNDING(D'_grnd, S, D''_grnd);
If D''_grnd is a maximal grounded subset of D then
    PRINT(Th(W ∪ CONS(D''_grnd)) is a c-extension
    with respect to
    Th(W ∪ CONS(D''_grnd) ∪ JUST(D''_grnd)).
```

5 Conclusion

Our approach provides a definition of a simple correspondence between each extension of a default theory and a computation process for all defaults that are involved in the constitution of this extension. Furthermore, the definition of this correspondence has just to be modified (i.e. in practice the opening and closing criteria referred to) to obtain one or other of the variants under consideration. Indeed, we think that the characterizations of theorem 1 offers a new insight into the differences between these variants. The problem of the derivation of a formula f from an extension $E = Th(W \cup CONS(D'))$ is directly resolved by a proof refutation of $W \cup D' \cup \{\neg f\}$ with the help of the previously computed tableau-representation of E.

Hence the main idea is as follows: we prepare a tractable tableau-representation of j-extensions in order to allow further work. In that sense, the process can be seen as a kind of compilation method for default theories.

The practical use of the general method depends on good implementation of a tableau-prover. Now, the question of the complexity of the method is still a crucial point for a further study. Some improvements can already be made by restricting the underlying language or by using symmetries within defaults (cf. example 3). It can be pointed out that the method has been adapted to the terminological knowledge representation formalisms by [Baader, Hollunder, 92].

References

[Baader, Hollunder, 92] Baader, Hollunder, *Embedding Defaults into Terminological Knowledge Representation Formalisms*. Proceedings of KR'92.

[Besnard, 89] Besnard P., *An Introduction to Default Logic*. Springer Verlag.

[Besnard, Schaub, 92] Besnard P., Schaub T., *Possible Worlds Semantics for Default Logics*. Working Notes of fourth International Workshop on Nonmonotonic Reasoning, Hawk Inn, Plymouth, Vermont, U.S.A., mai 1992, p. 34-40.

[Brewka, 91] Brewka G., *Cumulative Default Logic: in Defense of Nonmonotonic Inference Rule*. Artificial Intelligence, 50, p. 183-205.

[Fitting, 90] Fitting M., *First-Order Logic and Automated Theorem Proving*. Springer Verlag.

[Lukaszewicz, 88] Lukaszewicz W., *Considerations on Default Logic - An Alternative Approach* . Computational Intelligence 4, p. 1-16

[Reiter, 80] Reiter R., *A Logic for Default Reasoning*. Artificial Intelligence, 13, 1, p. 81-132.

[Risch, 91] Risch V., *Démonstration à Base de Tableaux Sémantiques pour la Logique des Défauts au sens de Lukaszewicz*. Actes du 8ème congrès AFCET "Reconnaissances des Formes et Intelligence Artificielle", Lyon, France, p. 355-361.

[Risch, 92] Risch V., *Un Cadre Général de Calcul pour les Logiques Propositionnelles de Défauts*. Actes des 4èmes journées du PRC-GDR en Intelligence Artificielle, octobre 1992, Marseille, France, Teknea, p. 321-344.

[Schaub, 91] Schaub T., *On Commitment and Cumulativity in Default Logics*. Proceedings of the European Conference on Symbolic and Quantitative Approaches for Uncertainty, Marseille, France, Lecture Notes in Computer Science 548, Springer Verlag, p. 305-309.

[Schwind, 90] Schwind C., *A Tableau-based Theorem Prover for a Decidable Subset of Default Logic*. Proceedings of the 10th International Conference on Automated Deduction, CADE 10, Springer Verlag, p. 541-546.

[Schwind, Risch, 91] Schwind C., Risch V., *A Tableau-Based Characterisation for Default Logic*. Proceedings of the European Conference on Symbolic and Quantitative Approaches for Uncertainty, Marseille, France, Lecture Notes in Computer Science 548, Springer Verlag, p. 310-317.

[Smullyan, 68] Smullyan R. M., *First-Order Logic*. Springer Verlag.