A Fixed Point Semantics for the ATMS

Michel Cayrol
Olivier Palmade
Thomas Schiex
Institut de Recherche en Informatique de Toulouse
Université Paul Sabatier, Toulouse, France

February 18, 1993

Abstract

In [CTW90], the ATMS resolution mechanisms have been formalized, using the concept of DK-tree. We propose in this paper a new and simple approach based on the fixed point theory and an original demonstration of the completeness of these mechanisms. We first introduce the basic definitions of terms, sets and functions which are used in the paper and build a lattice. Then, the basic formal tools to describe the ATMS mechanisms are introduced: The notion of DK-Clash (in fact, the inference rule), and a monotonic function, which represents the deduction process and whose least fixed point corresponds to the output of an ATMS. We prove the soundness and the completeness (for specific clauses) of these mechanisms. This demonstration brings to light an original and interesting property, whose applications are discussed at the end of the paper.

Keywords : Assumption based Truth Maintenance System, propositional logic, fixed point theory, automatic theorem proving.

Introduction

Our purpose in this paper is to give a semantics to the specific mechanisms of the ATMS (which allow to produce labels and nogoods). The declarative semantics given by Reiter and De Kleer [RK87] defined the label concept in the ATMS thanks to a natural and simple analogy with the prime implicants/implicates of the Boolean theory, but not the mechanisms which were to be used to produce the prime implicants/implicates.

The operational semantics given by [FH90] is slightly different and deals only with Horn Clauses. Our intention is to propose an operational semantics (similar to the one given for Logical Programming by [EK76]) for the specific deduction process of the ATMS, which takes into account the assumption distinction and the forward chaining strategy in the case of general clauses. Furthermore, our concept of DK-clash provides a new way to compute labels, nogoods and so on...

1 An informal introduction to the ATMS

The specific mechanisms of an assumption-based TMS are based upon the distinction between two different sets of data (propositional variables), that are supposed to be defined at the beginning : assumptions and non-assumptions. The basic idea is that the assumptions are the primitive data from which all other data can be derived. They are the parameters which characterize the different situations that are to be considered for a given problem, which is described by a set of clauses which are called justifications.

We may, as an example, consider the following justification set, using upper-case symbols (A, B, C and D) to denote assumptions :

$$\mathcal{J} = \{Ab \rightarrow c, B \rightarrow b, C \rightarrow b, bD \rightarrow\}$$

Two main questions appear:

1. May a given conjunction of assumptions be true, if we want to satisfy the set of justifications ? This corresponds to the "nogood" concept in an ATMS : A nogood is a minimal assumption set such that the assumptions that it contains, cannot be true together, with respect to the set of justifications. Minimal means that if we withdraw an assumption from the set, it is no more inconsistent with the justifications. (NB : An assumption set which includes a nogood is inconsistent and each inconsistent assumption set includes a nogood);
2. On which assumption sets is some assertion to be believed, if we want to satisfy a given set of justifications ? This corresponds to the "label" concept in an ATMS : The label of a proposition is a "complete", and "minimal" collection of consistent environments (i.e assumption sets) such that if the justifications are satisfied and if each assumption of the environment is true, then the proposition is true. Minimal means that there may not be two different environments in the label such that one is included in the other. Complete means that each environment that makes the proposition true includes at least an element of the label. The nogood set is often improperly[1] considered as the label of the specific data $\perp$, which denotes inconsistency.

The main purpose of an ATMS is to answer these two questions by providing the user with the nogood set and the label of every data. In the case of our previous justification set, an ATMS will produce :

- the nogood set : $\{\{B, D\}, \{C, D\}\}$;
- the label of b : $\{\{B\}, \{C\}\}$;
- the label of c : $\{\{A, B\}, \{A, C\}\}$;

To get closer to an homogeneous representation and treatment of labels and nogood set, we may note that the fact that a set of justification $\mathcal{J}$ leads to an inconsistent environment $E = \{A_1, \ldots, A_n\}$, could be simply and equivalently represented by the following associated clause :

$$A_1 \ldots A_n \rightarrow$$

In the same way, the fact that an environment $F = \{B_1, \ldots, B_n\}$ supports a data "b" may be simply and equivalently represented by the following associated clause :

$$B_1 \ldots B_n \rightarrow b$$

One should note that these clauses have only assumptions in their left part (negated litterals). From now on, such clauses will be called Cats[2] [CTW90, CT89, Tay90a, Tay90b]. In our example, the clauses that may be associated to a nogood or to an element of a label are :

- nogoods : $\{BD \rightarrow, CD \rightarrow\}$
- "b" label : $\{B \rightarrow b, C \rightarrow b\}$

[1] This is in contradiction with the consistency property of every environment in a label.
[2] From the french "Clause à Antécédent Typé".

- "c" label : $\{AB \rightarrow c, AC \rightarrow c\}$

The minimal set (for subsumption) of all the Cats which are logical consequences of the set of justification clearly contains the associated clause of each nogood and each label element[3], and therefore contains the ATMS usual output: nogoods and labels. The mathematical object we will focus on in this paper is therefore the set of Cats (noted $\mathcal{CL}(\mathcal{J})_{min}$) which respects the following properties :

- **soundness** : every element of $\mathcal{CL}(\mathcal{J})_{min}$ should be a logical consequence of $\mathcal{J}$;
- **completeness** : every Cat that is a logical consequence of $\mathcal{J}$ should be subsumed by an element of $\mathcal{CL}(\mathcal{J})_{min}$;
- **minimality** : there should be no element of $\mathcal{CL}(\mathcal{J})_{min}$ that is subsumed by another element of $\mathcal{CL}(\mathcal{J})_{min}$.

In our example, this set is equal to :

$$\mathcal{CL}(\mathcal{J})_{min} = \{BD \rightarrow, CD \rightarrow, B \rightarrow b, C \rightarrow b, AB \rightarrow c, AC \rightarrow c\}$$

and leads immediately to nogoods and labels.

2 Definitions, notations and terminology

Definition 2.1 *We define two distinct vocabularies Assum and NAssum, to be finite disjoint sets of symbols. The elements of Assum, called assumptions, will be denoted by upper-case letters, the elements of NAssum, called non-assumptions, by lower-case letters.*

Definition 2.2 *A clause is defined as an element of:*

$$Clause = P(Assum \cup NAssum) \times P(Assum \cup NAssum)^4$$

We may use implication as an alternate representation for clauses : the clause $(\{A,\ b,\ C\},\ \{A,\ e\})$ may be represented by $A\ b\ C\ \rightarrow\ A\ e$.

Definition 2.3 *A clause (π, δ) is a Cat iff π is a subset of Assum.*

Definition 2.4 *We introduce the operator $\Cup$ that we will call* fusion*, defined as an application from Clause×Clause to Clause as :*

$$(\pi, \delta) \Cup (\pi', \delta') = (\pi \cup \pi', \delta \cup \delta')$$

For example, $(Ab \rightarrow Cd) \Cup (E \rightarrow aC) = AbE \rightarrow aCd$.

Definition 2.5 *A clause (π, δ) is* fundamental *iff $\pi \cap \delta = \varnothing$ otherwise it is called a* tautology*. We shall distinguish two types of tautologies :*

- trivial *: of the form $A \rightarrow A$, where $A \in Assum$,*

[3] In fact it produces more, for we do not limit ourself to the production of Horn Cats. Therefore, the minimal set of all the Cats which are logical consequences of $\mathcal{J}$ may hold clauses of the form "$AB \rightarrow cd$" which is the associated clause of an environment $\{A, B\}$ that supports the disjunction $(c \vee d)$.

[4] Where $P(E)$ = set of subsets of E.

- non-trivial *: otherwise e.g.* $Ab \to A, xb \to x$.

FClause denotes the set of all the fundamental clauses.

Definition 2.6 *The set FClause is ordered by the relation*[5] $\Subset$ *defined by :*

$$(\pi, \delta) \ \Subset \ (\pi', \delta') \Leftrightarrow (\pi \ \subseteq \ \pi', \ \delta \ \subseteq \ \delta')$$

Definition 2.7 *Let E be a subset of FClause, we shall say that $c \in E$ is E-*minimal *if there is no $c' \in E$ with $c' \neq c$ and $c' \Subset c$. E_{min} shall be the set of E-minimal elements of E. A set of clauses E shall be said* minimal *iff $E_{min} = E$.*

Note that :

1. E_{min} is included in E,
2. When the fusion of two clauses c and c' is fundamental, it is the least upper bound in *FClause* of the set $\{c, c'\}$ for the partial ordering $\Subset$.

Definition 2.8 *Let us now consider the set MParts of minimal parts of Fclause. An element of Mparts is a minimal set of fundamental clauses.*

Example 2.1 For *Assum*$= \{A\}$, *NAssum*$= \{a\}$.

$$\begin{aligned}
Clause &= \{\to, \to A, \to a, \to A\ a, A \to, A \to A, A \to a, A \to A\ a, a \to, a \to A, \\
&\quad\ a \to a, a \to A\ a, A\ a \to, A\ a \to A, A\ a \to a, A\ a \to A\ a\} \\
Fclause &= \{\to, \to A, \to a, \to A\ a, A \to, A \to a, a \to, a \to A, A\ a \to\} \\
Mparts &= \{\{\to\}, \{A \to, a \to A\}, \ldots\}
\end{aligned}$$

Definition 2.9 *Mparts is ordered by the relation $\preccurlyeq$ defined by:*

$$\forall P \in Mparts, \forall P' \in Mparts, P \preccurlyeq P' \Leftrightarrow \forall c \in P, \exists c' \in P' \textit{ such that } c' \Subset c$$

Reflexivity and transitivity are obvious. As to anti-symmetry, it results from the minimality of the elements of *Mparts* : $P \preccurlyeq P' \preccurlyeq P$ means that $\forall c \in P, \exists c' \in P'$ and $c'' \in P$ such that $c'' \Subset c' \Subset c$. Since P is minimal and $\Subset$ is transitive, that implies that $c = c''$. So, $c \Subset c'$ and $c' \Subset c$ so $c = c'$.Therefore $P \subseteq P'$.

In the same way, (from $P' \preccurlyeq P \preccurlyeq P'$) it is easily shown that P' is included in P; so $P' = P$.

Note: We shall use $P' \preccurlyeq P$ or $P \succcurlyeq P'$ indifferently.

3 $(Mparts, \preccurlyeq)$ is a complete lattice

Theorem 3.1 *$(Mparts, \preccurlyeq)$ is a complete lattice*[6]*, its least element is $\{\}$, its greatest is $\{\to\}$.*

Proof :*Mparts* being finite, it is enough to show that *Mparts* is a lattice. We shall show that $\forall P, P' \in Mparts, (P \cup P')_{min}$ is the least upper bound of $\{P, P'\}$; we shall note it $(P \ \cup_{min} P')$.

Every element of P is an element of $(P \cup P')$ which is subsumed by an element of $(P \cup P')_{min}$. In the same way, it is easily shown that $(P \cup P')_{min} \succcurlyeq P'$. So, $(P \cup P')_{min}$

[5] Obviously reflexive, anti-symmetric and transitive.

[6] For our purpose, it would be enough to demonstrate that $(Mparts, \preccurlyeq)$ is a pointed partially ordered set.

is an upper bound of $\{P, P'\}$. We have now to show that $(P \cup P')_{min}$ is the least upper bound of $\{P, P'\}$. Let us consider $F \in Mparts, F \succcurlyeq P$ and $F \succcurlyeq P'$. Every element of $(P \cup P')_{min}$ is an element of $(P \cup P')$, i.e. either an element of P which is subsumed by an element of F, or an element of P' which is also subsumed by an element of F. So $(P \cup P')_{min} \preccurlyeq F$.□

The greatest lower bound is somewhat more difficult to build. We shall note it $\cap_{min}$. It is defined by : $P \cap_{min} P' = (Fclause \cap (P \Cup P'))_{min}$, where $(P \Cup P')$ means the set of clauses $c \Cup c'$ with $c \in P$ and $c' \in P'$.

Proof :

- $P \cap_{min} P' \in MParts$ by construct.
- Every element c'' of $P \cap_{min} P'$ is the value of $c \Cup c'$ for some $c \in P$ and $c' \in P'$. So $c \Subset c''$ and $c' \Subset c''$. It comes $P \cap_{min} P' \preccurlyeq P$ and $P \cap_{min} P' \preccurlyeq P'$.
- Let us show now that $P \cap_{min} P'$ is the greatest lower bound in *MParts*. Let $P'' \in Mparts$ such that $P'' \preccurlyeq P$ and $P'' \preccurlyeq P'$. Then each element c'' of P'' is subsumed by an element c of P and by an element c' of P'. So, c'' is subsumed by $c \Cup c'$, which cannot be a tautology since c'' belongs to P'' which is a part of *FClause*. Therefore, $c \Cup c'$ belongs to $Fclause \cap (P \Cup P')$; it is subsumed by a clause of $P \cap_{min} P'$. Then, this clause subsumes c'' and $P'' \preccurlyeq (P \cap_{min} P')$.□

Example 3.1 $\{A \to b, b \to d\} \cap_{min} \{A \to e, \to beF\}$

$$\begin{aligned} &= (Fclause \cap \{A \to be, A \to beF, Ab \to de, b \to bdeF\})_{min} \\ &= \{A \to be, A \to beF, Ab \to de\}_{min} \\ &= \{A \to be, Ab \to de\} \end{aligned}$$

4 Formal basis for ATMS mechanisms

We shall define an inference rule that corresponds to label computing in De Kleer's ATMS [dK86a, dK86b, dK88]. The formalization detailed below has been described in [CTW90]. Let us use the same notations where $\alpha b \to \beta Cd$ represents the clause $(\alpha \cup \{b\}, \beta \cup \{C, d\})$.

We first define labels and nogoods with respect to a set of justifications $\mathcal{J}$:

The nogood set is the set of assumption sets α such that :

1. $\alpha \to$ is a logical consequence of $\mathcal{J}$.
2. If α and α' are in the nogood set and $\alpha \subseteq \alpha'$, then $\alpha = \alpha'$.
3. If $\alpha' \to$ is a logical consequence of $\mathcal{J}$, then there is an α'' in the nogood set such that $\alpha'' \subseteq \alpha'$.

A label of a proposition x is the set of consistent assumption sets α such that :

1. $\alpha \to x$ is a logical consequence of $\mathcal{J}$.
2. If α and α' are in the label and $\alpha \subseteq \alpha'$, then $\alpha = \alpha'$.
3. If $\alpha' \to x$ is a logical consequence of $\mathcal{J}$ with α' consistent, then there is an α'' in the label such that $\alpha'' \subseteq \alpha'$.

In both cases, we remind the reader of what we need in order to produce labels and nogoods is the minimal set of Horn Cats which are logical consequences of $\mathcal{J}$. For completeness reasons, the mechanisms that we define hereunder will not produce this set but a superset of it : The minimized set of Cats (not only Horn Cats) that are logical consequences of $\mathcal{J}$.

4.1 The notion of DK-clash

Let us first introduce informally the notion of DK-Clash. The clause $aBcD \rightarrow eF$ means that if a, B, c and D are true, then e or F is true. Our purpose is to determine the data sets[7] which support the disjunctions. In this case, the disjunction $(e \vee F)$ is supported by $\{a, B, c, D\}$. De Kleer's idea is to consider only the supporting data sets composed of assumptions.

Let us now consider, along with the previous clause $aBcD \rightarrow eF$, the following clauses (note that underlined literals appear in the left-hand side of the first clause) :

$$\begin{cases} AB \rightarrow \underline{a}e \\ B \rightarrow \underline{B} \\ AC \rightarrow \underline{c}f \\ \rightarrow \underline{D} \end{cases}$$

This "dummy" clause is introduced to handle all variables in the antecedent part of the initial clause $aBcD \rightarrow eF$ in a similar way.

We can then deduce $ABC \rightarrow eFf$. Note that the set $\{A, B, C\}$ is only composed of assumptions. This leads us to the following definition :

Definition 4.1 *A* DK-Clash *of order n ($n \geq 0$), shall be defined as a pair (h, L) where :*

- *h is a clause $a_1 \ldots a_n \rightarrow \gamma$, element of Clause;*
- *L is a tuple of n Cats of Clause :*
  - $l_1 = \alpha_1 \rightarrow a_1\beta_1$,
  - $l_2 = \alpha_2 \rightarrow a_2\beta_2$,
  - ...
  - $l_n = \alpha_n \rightarrow a_n\beta_n$,

where α_i, β_i and γ denote sets of propositional variables with $a_i \notin \beta_i$ and α_i included in Assum.

The clause h is called the head *of the DK-Clash, the elements of L are called* lateral *clauses or simply laterals, a_i is the* resolved symbol *of the lateral $\alpha_i \rightarrow a_i\beta_i$. The head and the laterals are the* leaves *of the DK-Clash.*

The clause $r = \cup_i\alpha_i \rightarrow \gamma \cup (\cup_i\beta_i)$ is called the root *of the DK-Clash.*

It is easily shown that r is a Cat and a logical consequence of the leaves of the DK-Clash. We shall write $r = \Re(h, L)$. See figure 1.

A specific property of the DK-Clash concept will now be presented; its utility will appear only in further sections. Informally, the question that we will consider is : If we look at a given DK-Clash (h, L), whose root is r, and if a "better" set L' of laterals is provided (i.e the clauses from L' subsume the clauses from L), what can we say about (h, L') ? Is it a DK-Clash ? If it is, what is its root ? We will first look at two examples :

Example 4.1 Taking back our DK-Clash of figure 1, we consider that, instead of the lateral $AC \rightarrow cf$, the clause $A \rightarrow f$ is available. c was the resolved symbol in $AC \rightarrow cf$ but it has disappeared in $A \rightarrow f$. That is why it is not possible to build a DK-Clash with this clause instead of $AC \rightarrow cf$. However, there is an interesting result : $A \rightarrow f$ directly subsumes the root $ABC \rightarrow eFf$. From a logical point of view, though the DK-Clash has become useless, we have got a "better" result (the new clause).

[7] A "data set" represents here a conjunction of propositional variables.

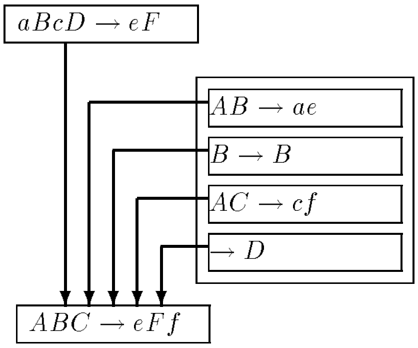

Figure 1: A DK-Clash

Example 4.2 Other example, still with the DK-Clash of figure 1 : This time, we provide four new laterals: $A \to ae, \to B, A \to c$ and $\to D$. In each of these clauses, there is still the appropriate resolved symbol. Therefore, we have a valid DK-Clash $(aBcD \to eF, \{A \to ae, \to B, A \to c, \to D\})$. Its root is $A \to eF$, which subsumes the old root $ABC \to eFf$. What is interesting is that we obtain as in the previous example a "better" result with "better" laterals.

These two examples are representative of the following theorem :

Theorem 4.1 *Let (h, L), a DK-Clash whose laterals are in a set $\mathcal{C}$. Let $\{\mathcal{C}_1, \mathcal{C}_2\}$ a partition of $\mathcal{C}$ and $\mathcal{C}_1'$ a set such that any element of $\mathcal{C}_1$ is subsumed by an element of $\mathcal{C}_1'$. Then, the root r of (h, L) is either subsumed by an element of $\mathcal{C}_1'$ or by the root of a DK-Clash (h, L') with the same head, whose laterals are in $\mathcal{C}_1' \cup \mathcal{C}_2$.*

Proof : Let (h, L) the DK-Clash whose laterals are in $\mathcal{C}_1 \cup \mathcal{C}_2$ and whose root is r. If each lateral l in $\mathcal{C}_1$ is subsumed by an element l' of $\mathcal{C}_1'$ that contains the resolved literal of l then the result of the substitution of every l by the corresponding l' in (h, L) is also a DK-Clash whose root subsumes r (obvious) and whose laterals are in $\mathcal{C}_1' \cup \mathcal{C}_2$. Otherwise, $\exists l \in \mathcal{C}_1, \exists l' \in \mathcal{C}_1'$ such that l' subsumes l and l' does not contain the resolved literal of l. Then l' subsumes r. $\square$

Example 4.3 $AD \to ce$ is the root of $(Ab \to c, \{D \to A, A \to be\})$ and is subsumed by the root of $(Ab \to c, \{D \to A, \to b\})$ which is $D \to c$ ($\to b$ subsumes $A \to be$). If we consider the set $\mathcal{L}' = \{D \to A, A \to e\}$, $A \to e$ directly subsumes the root $AD \to ce$.

4.2 DK-Clash and resolution principle

A short comparison with the resolution principle will provide an illustration of this new concept. Let us consider the example in figure 1. An equivalent deduction is presented in figure 2, as it could have been performed with the resolution principle.

As a first approximation, we could say that the root r of the DK-Clash (h, L) is the clause that we could derive if we applied the resolution principle n times : Between the head h and the lateral l_1, to produce a clause r_1, then between r_1 and l_2, to produce r_2,..., at the end between r_{n-1} and l_n to produce $r_n = r$ (In fact, it is not exactly so; the result of n such resolutions may strictly subsume the root of the DK-Clash. See below).

What is the advantage of such a concept ? Suppose that we want to perform the possible deductions using the rule $aBcD \to eF$. As it has been told, our purpose is to produce Cats and only Cats. That is the reason why we defined the DK-Clash. Two points are important :

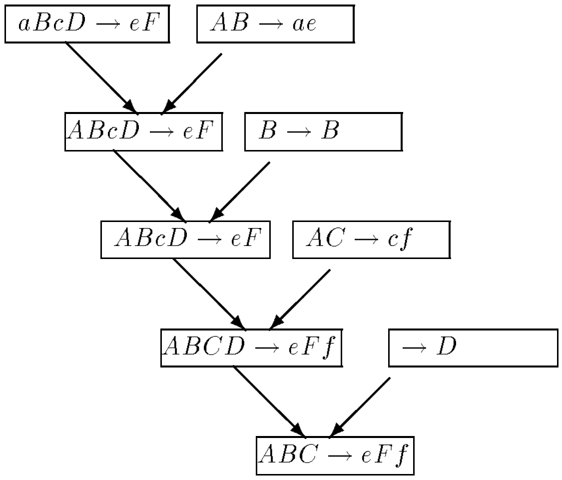

Figure 2: Resolution principle and DK-Clash

1. We do not produce any clause if there is not an available lateral for each literal in the antecedent of h. For example, if we do not know that the Cat $AB \rightarrow ae$ is available (or any Cat with an occurrence of a in the consequent), we will not do anything. Deduction about $aBcD \rightarrow eF$ will be delayed until such a Cat has been derived from other rules.

2. If there is a lateral for each literal in the antecedent of the head, there will be a DK-Clash and its root will be derived. Here, the DK-Clash provides a kind of heuristics to achieve this deduction : Given n laterals and the head, we perform in a single step the n operations which produce the root. An unrestricted application of the resolution principle could lead to much more work and to a lot of created clauses which are not Cats: n applications between h and l_i (to produce n clauses h_i), $n.(n-1)/2$ applications between h_i and l_j, $j \neq i$ (to produce $n.(n-1)/2$ clauses h_{ij}), $n.(n-1).(n-2)/6$ applications between h_{ij} and l_k, $j \neq i$, $j \neq k$, $i \neq k$, and so on...In our example, even if we look only at the two last laterals ($AC \rightarrow cf$ and $\rightarrow D$), the clauses $aABCD \rightarrow efF$, $aABC \rightarrow efF$, $aBc \rightarrow eF$ are not taken into account in the DK-Clash, though they are valid inferences.

On the other hand, the DK-Clash has a drawback : the computation of the root is the most simple one (this is a good point), but it is not the most efficient. For example, to compute the root of the DK-Clash ($aA \rightarrow, \{A \rightarrow a, \rightarrow A\}$), we perform a work which is equivalent to two applications of the resolution principle to get the root $A \rightarrow$, though a more interesting clause ($\rightarrow$) could be derived, also with two resolutions ($aA \rightarrow$ with $A \rightarrow a$, then $A \rightarrow$ with $\rightarrow A$). But in fact, this drawback is not specific to the DK-Clash inference rule : It is a problem of strategy; we have to perform a first DK-Clash ($aA \rightarrow, \{A \rightarrow a\}$) whose root is $A \rightarrow$ (one resolution), then the DK-Clash ($\rightarrow A, \{A \rightarrow\}$) whose root is $\rightarrow$ (one other resolution). This problem of strategy is exactly the same in pure resolution.

4.3 $\Gamma_{\mathcal{J}}$ function

The function $\Gamma_{\mathcal{J}}$ is defined from *Mparts* to *Mparts*. Its definition depends on a set $\mathcal{J}$, subset of *Clause*, called "set of justifications" in De Kleer terms. Note that $\mathcal{J}$ is not limited to Horn clauses and disjunctions of assumptions (De Kleer's choose).

Definition 4.2 $\Gamma_{\mathcal{J}}(\mathcal{C}) = \mathcal{C} \cup_{min} \{x \mid$ *x is the root of a DK-Clash whose head is an element of* $\mathcal{J}$ *and whose laterals are either elements of* $\mathcal{C}$, *or trivial tautologies*$\}$.

Example 4.4 Let $\mathcal{J} = \{abC \to c, dE \to e, abcde \to f\}$ and $\mathcal{C} = \{A \to a, B \to b, \to d\}$.
Two possible DK-Clashs with head in $\mathcal{J}$ and fundamental laterals in $\mathcal{C}$:

1. $(abC \to c, \{A \to a, B \to b, C \to C\})$ with root $ABC \to c$
2. $(dE \to e, \{\to d, E \to E\})$ with root $E \to e$.

So that : $\Gamma_{\mathcal{J}}(\mathcal{C}) = \{A \to a, B \to b, \to d, ABC \to c, E \to e\}$.
One step further: $\Gamma_{\mathcal{J}}(\Gamma_{\mathcal{J}}(\mathcal{C})) = \{A \to a, B \to b, \to d, ABC \to c, E \to e, ABCE \to f\}$
because of the DK-Clash $(abcde \to f, \{A \to a, B \to b, ABC \to c, \to d, E \to e\})$, which is now available with fundamental laterals in $\Gamma_{\mathcal{J}}(\mathcal{C})$.

N.B :

- If $\mathcal{C}$ is a set of Cats, $\Gamma_{\mathcal{J}}(\mathcal{C})$ will also be a set of Cats according to the definition of the root of a DK-Clash. In particular, $\Gamma_{\mathcal{J}}(\varnothing) = \{j \in \mathcal{J} \mid j \text{ is a Cat}\}_{min}$.
- $\forall \mathcal{C} \in Mparts, \mathcal{C} \preccurlyeq \Gamma_{\mathcal{J}}(\mathcal{C})$.

Theorem 4.2 *$\Gamma_{\mathcal{J}}$ is monotonic.*

Proof : Let us consider $\mathcal{C}$ and $\mathcal{C}'$ two elements of *Mparts*, $\mathcal{C} \preccurlyeq \mathcal{C}'$ and r an element of $\Gamma_{\mathcal{J}}(\mathcal{C})$. Let us show that $\exists r' \in \Gamma_{\mathcal{J}}(\mathcal{C}')$ such that $r' \Subset r$.
$\forall r \in \Gamma_{\mathcal{J}}(\mathcal{C}), r$ is :

- either an element of $\mathcal{C}$, which is subsumed by an element r' of $\mathcal{C}'$ which is subsumed by an element r'' of $\Gamma_{\mathcal{J}}(\mathcal{C}')$. So $\exists r'' \in \Gamma_{\mathcal{J}}(\mathcal{C}')$ such that $r'' \Subset r$.
- or the root of a DK-Clash whose laterals are elements of $\mathcal{C}$ or trivial tautologies. So, r is the root of a DK-Clash whose fundamental laterals are subsumed by elements of $\mathcal{C}'$. According to Theorem 4.1, r is either subsumed by an element r' of $\mathcal{C}'$, which is subsumed by an element r'' of $\Gamma_{\mathcal{J}}(\mathcal{C}')$, or subsumed by the root r' of a DK-Clash whose fundamental laterals are in $\mathcal{C}'$ and which is subsumed by an element r'' of $\Gamma_{\mathcal{J}}(\mathcal{C}')$. In any case, $\exists r'' \in \Gamma_{\mathcal{J}}(\mathcal{C}')$ such that $r'' \Subset r$.

So, $\Gamma_{\mathcal{J}}(\mathcal{C}) \preccurlyeq \Gamma_{\mathcal{J}}(\mathcal{C}')$.□

N.B :

- If $\mathcal{J} \succcurlyeq \mathcal{I}$, then $\Gamma_{\mathcal{J}}(\mathcal{C}) \succcurlyeq \Gamma_{\mathcal{I}}(\mathcal{C})$.
So, $\Gamma_{\mathcal{J}}(\varnothing) \succcurlyeq \Gamma_{\mathcal{I}}(\varnothing)$; So $\Gamma_{\mathcal{J}}(\Gamma_{\mathcal{J}}(\varnothing)) \succcurlyeq \Gamma_{\mathcal{J}}(\Gamma_{\mathcal{I}}(\varnothing)) \succcurlyeq \Gamma_{\mathcal{I}}(\Gamma_{\mathcal{I}}(\varnothing))$.
- By induction, it is easily proved that if $\mathcal{J} \succcurlyeq \mathcal{I}, \Gamma^n_{\mathcal{J}}(\varnothing) \succcurlyeq \Gamma^n_{\mathcal{I}}(\varnothing)$.

According to a simple version of Tarski's theorem in finite lattices [Bar81, Llo87, Tar55], we know that $\Gamma_{\mathcal{J}}$ has a least fixed point and that this fixed point is $lfp(\Gamma_{\mathcal{J}}) = \Gamma^{\infty}_{\mathcal{J}}(\varnothing) = \cup_{min}\{\Gamma^i_{\mathcal{J}}(\varnothing)\}$. Our purpose is to show that this fixed point is the minimal set of fundamental Cats which are logical consequences of $\mathcal{J}$.

5 Logical aspects

We give hereafter some important results.

Theorem 5.1 *Every element of $lfp(\Gamma_{\mathcal{J}})$ is a Cat.*

Easily shown by induction. See N.B. in § 4.3.

Theorem 5.2 $\forall \mathcal{C} \in Mparts, \mathcal{J} \cup \mathcal{C} \models \Gamma_{\mathcal{J}}(\mathcal{C})$

Proof : Every element of $\Gamma_{\mathcal{J}}(\mathcal{C})$ is either an element of $\mathcal{C}$ or the root of a DK-Clash whose leaves are either in $\mathcal{J}$ (head), in $\mathcal{C}$ (laterals) or trivial tautologies (laterals). The root of a DK-Clash is a logical consequence of its leaves. Therefore, every element of $\Gamma_{\mathcal{J}}(\mathcal{C})$ is a logical consequence of $\mathcal{J} \cup \mathcal{C}$.□

Corollary 5.1 $\mathcal{J} \models \Gamma_{\mathcal{J}}(\emptyset)$

Theorem 5.3 $\mathcal{J} \models lfp(\Gamma_{\mathcal{J}})$ *(soundness)*

Proof : For every element r of $lfp(\Gamma_{\mathcal{J}}) = \Gamma^{\infty}_{\mathcal{J}}(\emptyset)$, $\exists i$ such that $r \in \Gamma^{i}_{\mathcal{J}}(\emptyset)$. We will use induction on i.

Every element of $\emptyset$ is a logical consequence of $\mathcal{J}$ ($i = 0$). Let us suppose that the property holds for every element of $\Gamma^{i}_{\mathcal{J}}(\emptyset)$; $\forall r \in \Gamma^{i+1}_{\mathcal{J}}(\emptyset)$, r is either an element of $\Gamma^{i}_{\mathcal{J}}(\emptyset)$ (so $\mathcal{J} \models r$) or the root of a DK-Clash whose head is in $\mathcal{J}$ and fundamental laterals are in $\Gamma^{i}_{\mathcal{J}}(\emptyset)$. Every model of $\mathcal{J}$ is (induction hypothesis) a model of the laterals. The root of a DK-Clash is a logical consequence of its leaves, therefore r will be true in every model of $\mathcal{J}$. □

Let $\mathcal{CL}(\mathcal{J})$ be the set of all the fundamental Cats which are logical consequences of $\mathcal{J}$. We have proved that $lfp(\Gamma_{\mathcal{J}}) \preccurlyeq \mathcal{CL}(\mathcal{J})_{min}$. Let us prove now the completeness of the process.

Theorem 5.4 *(completeness)*

$$lfp(\Gamma_{\mathcal{J}}) = \mathcal{CL}(\mathcal{J})_{min} = \Gamma^{card(\mathcal{J})}_{\mathcal{J}}(\emptyset)$$

The proof lies on the following lemmas :

Lemma 5.1 *Every clause in $\mathcal{CL}(\mathcal{J})$ is*

- *either the root of a DK-Clash, whose head is in $\mathcal{J}$ and whose laterals are tautologies (possibly non-trivial) or elements of $\mathcal{CL}(\mathcal{J})$,*
- *or subsumed by an element of $\mathcal{J}$.*

Proof : Let $\alpha \rightarrow \beta$ be a clause of $\mathcal{CL}(\mathcal{J})$ which is not subsumed by an element of $\mathcal{J}$. Let us suppose that there is no DK-Clash $(\gamma \rightarrow \delta, \{\alpha' \rightarrow \beta' g, \ldots\})$ such that the root is $\alpha \rightarrow \beta$. This may be for three reasons :

1. There is no possible head; formally, there is no $\gamma \rightarrow \delta$ in $\mathcal{J}$ such that $\delta \subseteq \beta$ (according to the definition of the DK-Clash, the consequent of the head must be included in the consequent of the root). In this case, the set $((\cup_{\gamma \rightarrow \delta \in \mathcal{J}} \delta) - \beta) \cup \alpha$ is a model of $\mathcal{J}$ and is not a model of $\alpha \rightarrow \beta$ (since $\alpha \cap \beta = \emptyset$). This is contradictory with our hypothesis $\alpha \rightarrow \beta \in \mathcal{CL}(\mathcal{J})$.
2. There is a clause $\gamma \rightarrow \delta$ in $\mathcal{J}$ with $\gamma = \emptyset$ and $\delta \subseteq \beta$. This is contradictory with our hypothesis that $\alpha \rightarrow \beta$ is not subsumed by an element of $\mathcal{J}$.
3. Otherwise, there is a clause for the head but there is a missing lateral : let $\gamma \rightarrow \delta$ be the clause in $\mathcal{J}$ such that $\delta \subseteq \beta$ and $\gamma \neq \emptyset$; there is one $g \in \gamma$ such that there is no lateral $\alpha' \rightarrow \beta' g$ with $\alpha' \subseteq \alpha$ and $\beta' \subseteq \beta$ in $\mathcal{CL}(\mathcal{J})$. In particular, $\alpha \rightarrow \beta g$ is not in $\mathcal{CL}(\mathcal{J})$. This is contradictory with our hypothesis $\alpha \rightarrow \beta \in \mathcal{CL}(\mathcal{J})$.

□

Lemma 5.2 *Let* $\alpha' \rightarrow \beta' \ g$ *in* $\mathcal{CL}(\mathcal{J})$ *a lateral of a DK-Clash whose head is* $\gamma \rightarrow \delta$*, with* $g \in \gamma$*. Then* $\alpha' \rightarrow \beta' \ g \in \mathcal{CL}(\mathcal{J} - \{\gamma \rightarrow \delta\})$*.*

> Proof : Let us suppose that $\alpha' \rightarrow \beta' \ g \notin \mathcal{CL}(\mathcal{J} - \{\gamma \rightarrow \delta\})$. So, there is a model M of $\mathcal{J} - \{\gamma \rightarrow \delta\}$ where $\alpha' \rightarrow \beta' \ g$ is false i.e. $\alpha' \subseteq M$, $M \cap \beta' = \emptyset$ and $g \notin M$; $g \notin M$ implies that $\gamma \rightarrow \delta$ is true in M. Therefore M is a model of $\mathcal{J}$. Since $\alpha' \rightarrow \beta' \ g$ is in $\mathcal{CL}(\mathcal{J})$, it is true in M. This is contradictory with our hypothesis.□

In fact, we can show this way that if a clause $\rho \rightarrow \sigma$ is a logical consequence of a set of clauses $\mathcal{J}$, then $\rho \rightarrow \sigma$ is a logical consequence of $\{\gamma \rightarrow \delta \in \mathcal{J} | \gamma \cap \sigma = \emptyset \text{ and } \delta \cap \rho = \emptyset\}$.

We can now give the proof of the completeness theorem :

> Let us show by induction on the cardinality of $\mathcal{J}$ that $\mathcal{CL}(\mathcal{J})_{min} \preccurlyeq \Gamma_{\mathcal{J}}^{card(\mathcal{J})}(\emptyset)$. More precisely, we will show that every element of $\mathcal{CL}(\mathcal{J})$ is subsumed by an element of $\Gamma_{\mathcal{J}}^{card(\mathcal{J})}(\emptyset)$. Since $\mathcal{CL}(\mathcal{J})_{min}$ is a subset of $\mathcal{CL}(\mathcal{J})$, that will also show that $\mathcal{CL}(\mathcal{J})_{min} \preccurlyeq \Gamma_{\mathcal{J}}^{card(\mathcal{J})}(\emptyset)$.
>
> When card$(\mathcal{J}) = 0$, $\mathcal{CL}(\emptyset) = \mathcal{CL}(\emptyset)_{min} = \emptyset$ and $\Gamma_{\emptyset}^{0}(\emptyset) = \emptyset$, so :
>
> $$\mathcal{CL}(\emptyset) = \mathcal{CL}(\emptyset)_{min} \preccurlyeq \Gamma_{\emptyset}^{0}(\emptyset)$$
>
> Let us now consider $\mathcal{J}$, card$(\mathcal{J}) = n \geq 0$ and let us suppose that $\forall \mathcal{K} \in$ *Mparts*, with card$(\mathcal{K}) \leq n$ then $\forall c \in \mathcal{CL}(\mathcal{K})$, c is subsumed by an element of $\Gamma_{\mathcal{K}}^{card(\mathcal{K})}(\emptyset)$. Let $\mathcal{J}$ be an element of *Mparts* with cardinality $(n+1)$. $\forall c \in \mathcal{CL}(\mathcal{J})$, c is either :
>
> 1. subsumed by an element j of $\mathcal{J}$ or
> 2. the root of a DK-Clash whose head h is in $\mathcal{J}$ and whose laterals are either tautologies (which are subsumed by trivial tautologies) or elements of $\mathcal{CL}(\mathcal{J} - \{h\})$.
>
> Let us consider each case :
>
> 1. j is a Cat (since c is a Cat) and is in $\Gamma_{\mathcal{J}}^{i}(\emptyset), \forall i > 0$. Therefore c is subsumed by an element of $\Gamma_{\mathcal{J}}^{n+1}(\emptyset)$.
> 2. $\mathcal{CL}(\mathcal{J} - \{h\}) \preccurlyeq \Gamma_{\mathcal{J}-\{h\}}^{n}(\emptyset)$ (induction hypothesis). Since $\Gamma_{\mathcal{J}-\{h\}}^{n}(\emptyset) \preccurlyeq \Gamma_{\mathcal{J}}^{n}(\emptyset)$, every element of $\mathcal{CL}(\mathcal{J} - \{h\})$ is subsumed by an element of $\Gamma_{\mathcal{J}}^{n}(\emptyset)$ and every tautology is subsumed by a trivial tautology. According to theorem 4.1, c is either subsumed by an element of $\Gamma_{\mathcal{J}}^{n}(\emptyset) \preccurlyeq \Gamma_{\mathcal{J}}^{n+1}(\emptyset)$ or by the root r of a DK-Clash whose laterals are in $\Gamma_{\mathcal{J}}^{n}(\emptyset)$ or trivial tautologies (i.e. an element of $\Gamma_{\mathcal{J}}^{n+1}(\emptyset)$). Therefore c is subsumed by an element of $\Gamma_{\mathcal{J}}^{n+1}(\emptyset)$.
>
> Since :
>
> $$\mathcal{CL}(\mathcal{J})_{min} \preccurlyeq \Gamma_{\mathcal{J}}^{card(\mathcal{J})}(\emptyset) \preccurlyeq \ldots \preccurlyeq \Gamma_{\mathcal{J}}^{card(\mathcal{J})+i}(\emptyset) \preccurlyeq \ldots \preccurlyeq \Gamma_{\mathcal{J}}^{\infty}(\emptyset) \preccurlyeq \mathcal{CL}(\mathcal{J})_{min}$$
>
> We conclude that :
>
> $$\Gamma_{\mathcal{J}}^{card(\mathcal{J})}(\emptyset) = \ldots = \Gamma_{\mathcal{J}}^{card(\mathcal{J})+i}(\emptyset) = \ldots = \Gamma_{\mathcal{J}}^{\infty}(\emptyset) = \mathcal{CL}(\mathcal{J})_{min}$$
>
> □

That means that for any Cat logical consequence of $\mathcal{J}$, there is a proof, whose depth is less or equal than $card(\mathcal{J})$.

6 Application : "Deterministic" proof of any Cat

6.1 Proof algorithm

Thanks to our previous theorem and to the associated lemmas, we are able to propose a "deterministic" algorithm for the proof of any fundamental Cat, logical consequence of $\mathcal{J}$. This algorithm does not use backtracking and finds a solution which is not necessarily optimal. The main points of this algorithm come directly from the proofs of the lemmas 5.1 and 5.2 :

1. Let us consider a fundamental Cat: $\alpha \rightarrow\ \beta$. According to Lemma 5.1, $\alpha \rightarrow\ \beta$ is a logical consequence of $\mathcal{J}$ if and only if it is either :

   - subsumed by an element of $\mathcal{J}$ (trivial case) ;
   - or the root of a DK-Clash whose head is in $\mathcal{J}$ and whose laterals are tautologies or elements of $\mathcal{CL}(\mathcal{J})$. This case is more tricky. Having found an arbitrary justification $\gamma \rightarrow\ \delta$ such that $\delta \subseteq \beta$, (it will be the head of our DK-Clash), we have to find laterals :
     - If $\alpha \rightarrow\ \beta$ is effectively a logical consequence of $\mathcal{J}$, then, for every literal g in γ, each $\alpha \rightarrow\ \beta\ g$ is necessarily a logical consequence of $\mathcal{J}$, and is a possible lateral ;
     - Conversely, if each $\alpha \rightarrow\ \beta\ g$ is a logical consequence of $\mathcal{J}$, as it is a possible lateral, then $\alpha \rightarrow\ \beta$ should be a logical consequence of $\mathcal{J}$ (Lemma 5.1).

   Therefore, the laterals will simply be the $\alpha \rightarrow\ \beta\ g$, for every literal g in γ. The problem is then reduced to the proof that every $\alpha \rightarrow\ \beta\ g$ is a logical consequence of $\mathcal{J}$ (problem which may be tackled recursively) or a tautology (trivial).

2. The Lemma 5.2 allows a simple (but fundamental) optimization : for the recursive proof of every lateral $\alpha \rightarrow\ \beta\ g$, the head of the DK-Clash may be (and will be) withdrawn from the set of justifications.

These considerations lead to the following algorithm for the proof that a fundamental Cat $\alpha \rightarrow\ \beta$ is a logical consequence of a set of justification $\mathcal{J}$:

```
procedure PROOF(α → β:Cat, J:set of clauses)

if α → β is a tautology or subsumed by an element of J
   then return ''True''
   else
       find a clause γ → δ in J such that δ ⊆ β
       if failure
          then return ''False''
          else
              if for every g in γ, PROOF(α → β g, J − {γ → δ}) is ''True''
                 then return ''True''
                 else return ''False''
              end if
       end if
endif
end.
```

Example 6.1 Let us consider $\mathcal{J} = \{A \rightarrow\ a\ d, d \rightarrow, B \rightarrow b\ d, A\ D \rightarrow, a\ b \rightarrow\ c\}$ and let us prove that $ABC \rightarrow\ c \in \mathcal{CL}(\mathcal{J})$. See figure 3 for a possible proof.

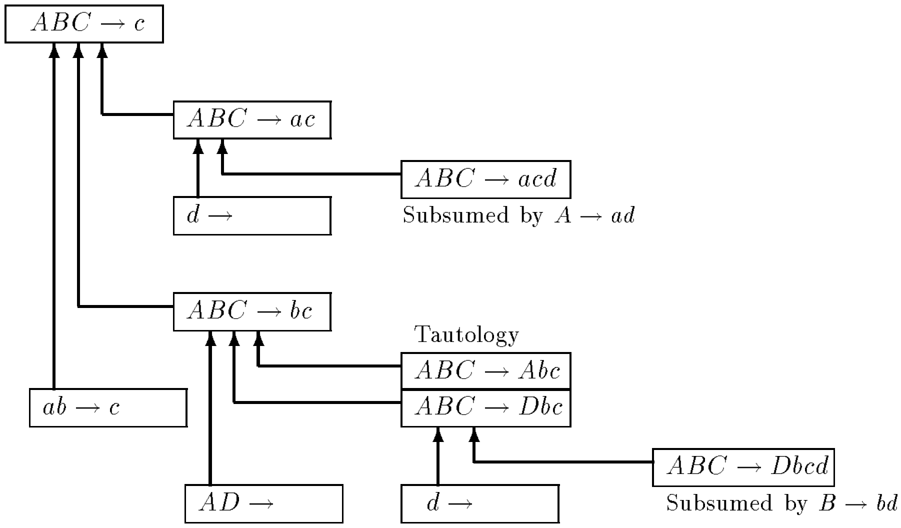

Figure 3: First Proof using DK-Clash

6.2 Reduction

We may reduce this tree, using theorem 4.1, to obtain only trivial tautologies, and fundamental $\mathcal{J}$-minimal leaves and roots. See figure 4.

We have simplified from the leaves to the root :

- $ABC \to\ acd$ is replaced by the justification $A \to\ ad$. The new root is $A \to\ a$.
- $ABC \to\ Dbcd$ is replaced by the justification $B \to\ b\ d$. The new root is $B \to\ b$.
- This new lateral does not contain the resolved literal D and therefore subsumes the root of the DK-Clash (dotted box) which disappears.
- The DK-Clash whose head is $a\ b \to c$ has new laterals : $A \to\ a$ and $B \to\ b$.
- The new root is $A\ B \to\ c$. This $\mathcal{CL}(\mathcal{J})$-minimal clause subsumes directly $ABC \to\ c$.

In fact the Lemma 5.1 makes possible some improvements :

- Before the beginning of the proof of $\alpha \to\ \beta$ (the initial clause), we shall withdraw every justification whose consequent contains an element of α, or whose left part contains an element of β.
- When we want to prove the lateral $\alpha' \to\ \beta'\ g$, we shall withdraw every justification whose left part contains g.

The reduction mechanism is also very interesting to cut subtrees in the deduction process : In the above example, the lateral $A\ B\ C \to\ A\ b\ c$ of figure 3 (which is here a tautology but which could be also the root of a big subtree) disappears in figure 4 and is not to be developed, once we have developed the other lateral.

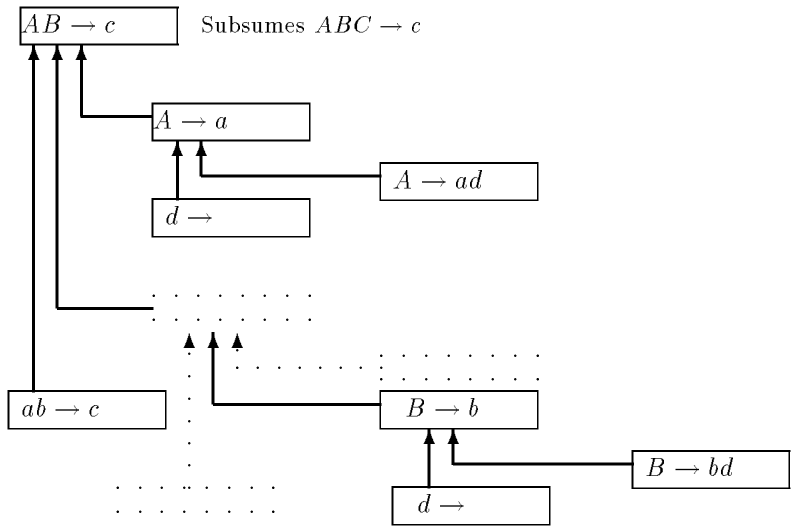

Figure 4: First Proof reduced

6.3 Label computation

Another application of this algorithm brings the discussion back to the ATMS. Let us consider the example of figure 5. This tree corresponds to the (failed) proof of the clause $\to$ with the set of justifications $\{a\ b \to, A \to\ a, A\ B \to\ b, \to\ A\ b, D \to\ B\ b\}$. Two leaves in the complete tree show this failure : $\to\ A\ a$ and $\to\ D\ B\ b$. They are laterals that are not subsumed by any justification and that cannot be the root of a DK-Clash with a head in the remaining justifications at this node. We have called them failure leaves for that reason and we will show that they are in fact quite interesting.

Let us ask the question : What conjunction of assumptions should be true so that the proof succeeds ? In other words, what set of clauses of the form $\to\ X$ (where X is an assumption) could be added to the set of justifications so that the proof succeeds ?

The answer is simple : The proof succeeds if failure leaves are no more failure leaves, that is to say if we are able to build new subtrees starting from these nodes, without failure leaves thanks to new justifications, or directly if they are subsumed by one of these new justifications. A clause of the form $\to\ X$ cannot be the head of a DK-Clash with a failure leaf as a root. If the consequent of $\to\ X$ is included in the consequent of a failure leaf, that means that $\to\ X$ subsumes the failure leaf. Conclusion : The proof succeeds if for each failure leaf, we add a clause $\to\ X$ where X is an assumption which appears in the consequent of the failure leaf. Thanks to this remark, we discover the nogoods of the set of justifications of our example :

- $\to\ A\ a$ may only be subsumed by $\to\ A$;
- $\to\ D\ B\ b$ may be subsumed by $\to\ D$ or $\to\ B$;

Therefore, the proof succeeds if we add $\{\to\ A, \to\ B\}$ or $\{\to\ A, \to\ D\}$. The nogoods are $\{A, B\}$ and $\{A, D\}$.

We have developped this way a complete set of techniques to compute labels, interpretations, extensions,..., in backward chaining.

An implementation of these algorithms with these improvements and some other ones has been performed and proved to be very efficient.

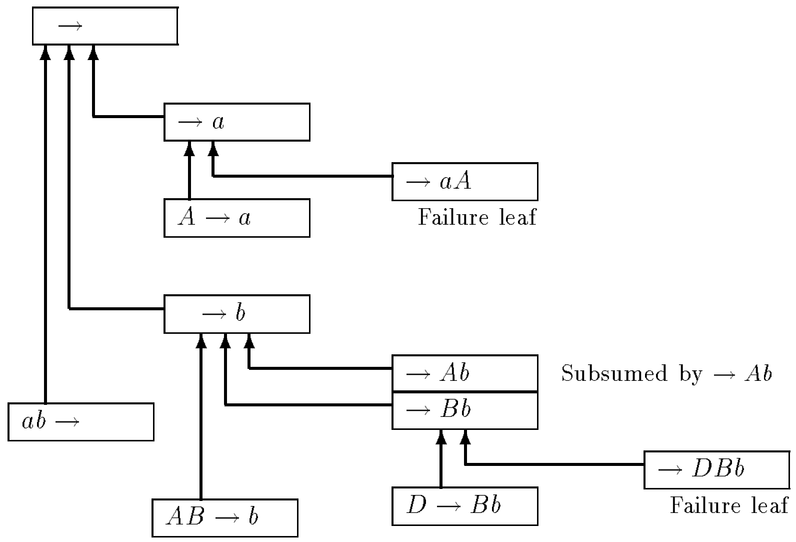

Figure 5: Label computing using failure leaves

Furthermore, we have developed some other models of "backward-clash" : A general clash summarizes their common theoretic properties, while specific models provide either better efficiency in pure theorem proving or ATMS related functionalities. For example, the constraint "the consequent of the head is included in the consequent of the root" may be replaced by a similar constraint upon the antecedent part. This is only a simple change but it modifies the structure of deduction trees and allows to compute other useful concepts than labels and nogoods. But it was not our purpose to develop this subject here. We just intended to give an application of the formal principles that were proposed in the previous sections.

Conclusion

In this paper, we have developed an operational semantics for a generalized ATMS whose set of justification is not limited (including tautologies, non-Horn clauses...). The task of our ATMS is to build the least fixed point of our function $\Gamma_{\mathcal{J}}$. We have shown that this process is sound and complete. In fact, we have found something more : First, the completeness demonstration does not use the resolution principle completeness. Furthermore, thanks to the powerful notion of DK-Clash and to its properties, we have introduced an interesting ("deterministic") algorithm for a propositional prover, which would be able to prove whether a fundamental Cat is a logical consequence of the set of justifications or not. This algorithm is not specifically adapted to the ATMS and may be used without the notion of Cat or assumption ($NAssum = \emptyset$).

As a future work, we are considering the extension of this formalism to first order logic and an illustration of the theoretical properties in a concrete implementation.

References

[Bar81] H.P. Barendregt, *The λ-calculus, its syntax and semantics*, North Holland Publishing Company, Amsterdam, Netherlands, 1981.

[CT89] Michel Cayrol and Pierre Tayrac, *Les résolutions CAT-correcte et CCT-correcte, la résolution Cat-correcte dans l'ATMS*, Colloque international sur l'informatique cognitive des organisations (Québec, Canada), 1989.

[CTW90] Michel Cayrol, Pierre Tayrac, and Robert Wigetman, *Theoretical foundations of the de kleer ATMS*, Tech. report, Institut de Recherche en Informatique de Toulouse, Toulouse, France, 1990.

[dK86a] Johan de Kleer, *An assumption-based truth maintenance system*, Artificial Intelligence **28** (1986), 127–162.

[dK86b] Johan de Kleer, *Extending the ATMS*, Artificial Intelligence **28** (1986), 163–196.

[dK88] Johan de Kleer, *A general labeling algorithm for assumption-based truth maintenance*, Proceedings of the national Conference on Artificial Intelligence (St Paul, MN), August 1988, pp. 188–192.

[EK76] M.H. Van Emden and R.A. Kowalski, *The semantics of predicate logic as a programming language*, Journal ACM **23** (1976), no. 4, 733–742.

[FH90] Yasushi Fujiwara and Shinichi Honiden, *On logical foundations of the ATMS*, ECAI-90 Workshop on Truth Maintenance Systems (Stockholm), August 1990.

[Llo87] Lloyd, *Foundations of logic programming*, 2nd ed., Springer-Verlag, 1987.

[RK87] R. Reiter and J. De Kleer, *Foundations of assumption-based truth maintenance systems*, Proc. North. Am. Conf. on AI (Seattle, Washington), July 1987, pp. 183–188.

[Tar55] A. Tarski, *A lattice-theoretical fixpoint theorem and its applications*, Pacific J. Math., 1955, pp. 285–309.

[Tay90a] Pierre Tayrac, *ARC : An extended ATMS based on directed CAT-correct resolution*, ECAI-90 Workshop on Truth Maintenance Systems (Stockholm), August 1990.

[Tay90b] Pierre Tayrac, *Etude de nouvelles stratégies de résolution, application à l'ATMS*, Ph.D. thesis, Université Paul Sabatier, Toulouse, Juin 1990.