



**HAL**  
open science

## Hypothetical reasoning in possibilistic logic: basic notions and implementation issues

Salem Benferhat, Didier Dubois, Jérôme Lang, Henri Prade

► **To cite this version:**

Salem Benferhat, Didier Dubois, Jérôme Lang, Henri Prade. Hypothetical reasoning in possibilistic logic: basic notions and implementation issues. Wang, P.Z.; Loe, K.F. Between Mind and Computer, Fuzzy Science and Engineering, 1, World Scientific, pp.1–29, 1994, Advances in Fuzzy Systems - Applications and Theory, 10.1142/9789814503679\_0001 . hal-04053436

**HAL Id: hal-04053436**

**<https://hal.science/hal-04053436>**

Submitted on 31 Mar 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# **HYPOTHETICAL REASONING IN POSSIBILISTIC LOGIC: BASIC NOTIONS, APPLICATIONS AND IMPLEMENTATION ISSUES**

**SALEM BENFERHAT, DIDIER DUBOIS, JÉRÔME LANG  
and HENRI PRADE<sup>(\*)</sup>**

*Institut de Recherche en Informatique de Toulouse – CNRS  
Université Paul Sabatier, 118 route de Narbonne, 31062 Toulouse Cedex, France*

Possibilistic ATMS are truth maintenance systems oriented towards hypothetical reasoning where *both* assumptions and justifications can bear an uncertainty weight. Uncertainty is represented in the framework of possibility theory. In possibilistic logic uncertain clauses are handled as such and then in possibilistic ATMS the management of uncertainty is not separated from the other classical capabilities of the ATMS. The main interest of a possibilistic ATMS is to take advantage of the uncertainty pervading the available knowledge so as to rank-order environments in which a given statement is true. The approach is illustrated on a fault diagnosis problem. Basic algorithms associated with possibilistic ATMS are given.

## **1. Introduction**

Assumption-based truth-maintenance systems (ATMS) ([9, 10, 11]; see also [30]) are automated reasoning systems oriented towards hypothetical reasoning since they are able to determine under which set of assumption(s) a given proposition is true. This set is called the "label" of the proposition. In this paper we present an extension of the ATMS, called "possibilistic ATMS" (or  $\Pi$ -ATMS for short), where the management of uncertainty is integrated inside the basic capabilities of the ATMS. Uncertainty pervading justifications or grading assumptions is represented in the framework of possibility and necessity measures ([32], [16]); these measures agree with the ordinal nature of what we wish to represent (it enables us to distinguish between what is plausible and what is less plausible). The certainty of each granule in the knowledge base (represented by a clause in possibilistic logic [15]) is evaluated under the form of a lower bound of a necessity measure. This uncertainty in the deduction process is propagated by means of an extended resolution principle. Uncertainty degrees are then naturally attached to the configurations of assumptions in which a given proposition is true; one can also evaluate to what degree a given configuration of assumptions is inconsistent or compute the more or less certain consequences of a configuration of assumptions. This approach enables us to handle

---

(\*) A preliminary and short version of this paper was presented at the 1st Maghreb Symposium Programming and Systems, Algiers, Oct. 21-23, 1991. See the Proceedings pp. 153-173 (available from Institute of Computer Science, UST HB, Algiers).

disjunctions and negations of assumptions without particular problem. Moreover, by rank-ordering configurations according to the degrees attached to them,  $\Pi$ -ATMS provides a way of limiting combinatorial explosion when using ATMS in practice.

We present the basic definitions and results of possibilistic logic first. In Section 4 we give the basic definitions and functionalities of the  $\Pi$ -ATMS, illustrated by a fault diagnosis problem previously introduced in Section 3. Section 5 presents an algorithm for the computation of labels and contradictory environments ("nogoods") based on an extension of the so-called CAT-correct resolution, initially developed by Cayrol and Tayrac [6, 7]. In Section 6, we will show how the  $\Pi$ -ATMS can be used to revising inconsistent knowledge bases. An example of a data fusion problem will be described in this section. Lastly in Section 7 the proposed approach is briefly compared with existing uncertainty-handling ATMS's.

## 2. Possibilistic Logic

### 2.1. Modeling Uncertainty with Possibility and Necessity Measures

Possibilistic logic [18, 21, 24] is an extension of classical logic where one manipulates propositional or first-order calculus closed formulas weighted by lower bounds of possibility or necessity degrees which belong to  $[0,1]$ . In this paper we restrict ourselves to a fragment of possibilistic logic, the "clausal possibilistic propositional logic", where the considered formulas are exclusively conjunctions of possibilistic propositional clauses weighted by necessity degrees [15, 17].

A possibility measure  $\Pi$  satisfies the following axioms [32, 16]:

- (i)  $\Pi(\perp) = 0; \Pi(\top) = 1$
- (ii)  $\forall p, \forall q, \Pi(p \vee q) = \max(\Pi(p), \Pi(q))$

where  $\perp$  and  $\top$  denote respectively contradiction and tautology. We emphasize that we only have  $\Pi(p \wedge q) \leq \min(\Pi(p), \Pi(q))$  in the general case. A necessity measure is associated by duality with a possibility measure by  $\forall p, N(p) = 1 - \Pi(\neg p)$ . Axiom (ii) is then equivalent to  $\forall p, \forall q, N(p \wedge q) = \min(N(p), N(q))$  and, as a consequence,  $N(p) > 0 \Rightarrow \Pi(p) = 1$ . We adopt the following conventions:

- $N(p) = 1$  means that, given the available knowledge,  $p$  is certainly true; conversely, if  $p$  is said to be true we can consider  $p$  as certain.
- $1 > N(p) > 0$  that  $p$  is somewhat certain and  $\neg p$  not certain at all (since the axioms imply that  $\forall p, \min(N(p), N(\neg p)) = 0$ ).
- $N(p) = N(\neg p) = 0$  (equivalent to  $\Pi(p) = \Pi(\neg p) = 1$ ) corresponds to the case of total ignorance; it expresses that, from the available knowledge, nothing enables us to say if  $p$  is rather true or rather false.

- $0 < \prod(p) < 1$  (equivalent to  $1 > N(\neg p) > 0$ ) that  $\neg p$  is somewhat certain and  $p$  not certain at all.
- $\prod(p) = 0$ , that  $p$  is certainly false.

Possibilistic logic is well-adapted to the representation of states of incomplete knowledge, since we can distinguish between the complete lack of certainty in the falsity of a proposition  $p$  ( $N(\neg p) = 0$ ) and the total certainty that  $p$  is true ( $N(p) = 1$ ).  $N(p) = 1$  entails  $N(\neg p) = 0$  but the converse is false. It contrasts with probability measures where  $\text{Prob}(p) = 1$  is equivalent to  $\text{Prob}(\neg p) = 0$ . Possibilistic logic contrasts also with usual multiple-valued logics which are fully truth-functional and deal with fuzzy propositions, while possibility degrees apply to standard propositions and are truth-functional only with respect to disjunction (see [15] on this point) .

## 2.2. Possibilistic Clauses and Possibilistic Resolution

An uncertain clause will be a first-order logic clause  $c$  to which a valuation is attached, taken as a lower bound of its necessity measure. Thus, in the following we shall write  $(c \alpha)$  as soon as the inequality  $N(c) \geq \alpha$  is taken for granted.

Resolution has been extended to possibilistic logic [15, 18]. The classical rule for propositional clauses are generalized by

$$\begin{array}{c} (c \alpha) \\ (c' \beta) \\ \hline (c'' \min(\alpha, \beta)) \end{array}$$

where  $c''$  is a classical resolvent of the classical clauses  $c$  and  $c'$ .

The refutation method can be generalized to possibilistic logic [14, 15]. Indeed if we are interested in proving that a formula  $f$  is true, necessarily to some degree, we add to the knowledge base  $\sum$  the assumption  $N(\neg f) = 1$ , i.e. that  $f$  is false (with total certainty). Let  $\mathcal{H}'$  be the new knowledge base obtained by adding to  $\sum$  the clauses issued from  $\neg f$ , with the weight 1. Then it can be proved that any valuation attached to the empty clause produced by the extended resolution pattern from  $\mathcal{H}'$  is a lower bound  $\alpha$  of the necessity measure of the conclusion  $f$ . It entails the existence of "optimal refutations", i.e. derivations of an empty clause with a maximal valuation.

## 2.3. Semantics and Partial Inconsistencies

A semantics has been defined for clausal possibilistic logic [17, 21, 24]. Here we only consider the case of necessity-valued propositional clauses. If  $c$  is a clause,  $M(c)$  the set of the models of  $c$ , then the models of  $(c \alpha)$  will be defined by a fuzzy set  $M(c \alpha)$  with a membership function

$$\begin{aligned}\mu_{M(c \ \alpha)}(I) &= 1 \text{ if } I \in M(c) \\ &= 1 - \alpha \text{ if } I \in M(\neg c),\end{aligned}$$

where  $I$  is a (classical) interpretation.

Then the fuzzy set of models of a set of weighted clauses  $\Sigma = \{C_1, C_2, \dots, C_n\}$ , where  $C_i$  stands for  $(c_i \ \alpha_i)$ , is the intersection of the fuzzy sets  $M(C_i)$ , i.e.

$$\mu_{M(\Sigma)}(I) = \min_{i=1, \dots, n} \mu_{M(C_i)}(I).$$

The consistency degree of  $\Sigma$  will be defined by  $\text{cons}(\Sigma) = \max_I \mu_{M(\Sigma)}(I)$ ; it estimates the degree to which the set of models of  $\Sigma$  is not empty. The quantity  $\text{Inc}(\Sigma) = 1 - \text{cons}(\Sigma)$  will be called degree of inconsistency of  $\Sigma$ .

Finally we say that  $\mathcal{F}$  is a logical consequence of  $\Sigma$  if and only if  $\forall I, \mu_{M(\mathcal{F})}(I) \geq \mu_{M(\Sigma)}(I)$ , which will be written  $\Sigma \models \mathcal{F}$ . Let us note that all these definitions recover those of classical logic and are in accordance with Zadeh's theory of approximate reasoning [33]. We take the notation  $\Sigma \vdash (c \ \beta)$  if and only if there exists a  $\beta'$ -refutation with  $\beta' \geq \beta$ , i.e. a deduction of  $(\perp \ \beta')$ , from the set of weighted clauses  $\Sigma'$  obtained by adding  $N(\neg c) = 1$  to  $\Sigma$ . Then we have the following completeness theorem ([17, 21, 24]): let  $\Sigma$  be a set of weighted clauses; then  $\Sigma \vdash (c \ \beta)$  if and only if  $\Sigma \models (c \ \beta)$ .

#### 2.4. Hypothetical Reasoning

As it was pointed out in [19], the weighted clause  $(\neg p \vee q \ \alpha)$  is semantically equivalent to the weighted clause  $(q \ \min(\alpha, v(p)))$  where  $v(p)$  is the truth value of  $p$ , i.e.  $v(p) = 1$  if  $p$  is true and  $v(p) = 0$  if  $p$  is false. Indeed, for any uncertain proposition  $(p \ \alpha)$  we can write  $\mu_{M(p \ \alpha)}(I)$  under the form  $v(p \ \alpha) = \max(v(p), 1 - \alpha)$ , where  $v(p)$  is assigned by interpretation  $I$ . Then obviously:

$$\begin{aligned}v(\neg p \vee q \ \alpha) &= \max(v(\neg p \vee q), 1 - \alpha) = \max(1 - v(p), v(q), 1 - \alpha) \\ &= \max(v(q), 1 - \min(v(p), \alpha)) = v(q \ \min(v(p), \alpha))\end{aligned}$$

The equivalence between the weighted clauses  $(\neg p \vee \neg s \vee q \ \alpha)$  and  $(\neg p \vee q \ \min(\alpha, v(s)))$  expresses that the rule "if  $p$  and  $s$  are true then  $q$  is certain to the degree  $\alpha$ " means that "in an environment where  $s$  is true, if  $p$  is true then  $q$  is certain to the degree  $\alpha$ ", when we decide to consider  $s$  as an assumption. This equivalence leads to the following modified presentation of the resolution rule

$$\frac{(\neg p \vee q \text{ min}(\alpha, v(s)) \quad (p \vee r \text{ min}(\beta, v(t))))}{(q \vee r \text{ min}(\alpha, v(s), \beta, v(t)))}$$

This enables us to express that if the clause  $\neg p \vee q$  is certain to the degree  $\alpha$  in an environment where  $s$  is true and if the clause  $p \vee r$  is certain to the degree  $\beta$  in an environment where  $t$  is true, then the resolvent clause  $q \vee r$  is certain to the degree  $\min(\alpha, \beta)$  in an environment where  $s$  and  $t$  are true (since  $v(s \wedge t) = \min(v(s), v(t))$ ). It turns out that when  $\alpha = \beta = 1$ , the above resolution rule is very close to the CAT-correct resolution rule [5, 6, 7], which separates the assumptions from the other literals by sorting them, and where resolution is restricted so as to get rid of the non-assumption literals. Hence possibilistic logic can capture not only the propagation of uncertainty, but at the same time the propagation of the reasons for uncertainty.

It has been shown that if, adding to the knowledge base the clauses corresponding to  $N(\neg p) = 1$ , we obtain by resolution the empty clause with a strictly positive weight, then this weight is a lower bound of the necessity degree of  $p$ . The exploration of all proof paths leading to the empty clause will thus help determine the set of environments in which  $p$  is somewhat certain if all assumptions are carried over to the weight side, before running the refutation. This enables us to define a possibilistic ATMS in Section 4 and to develop the associated basic procedures.

### 3. An Example

This example, inspired from Siegel [29], relates to the functioning of an engine. This example may appear somewhat fancy, for it is not easy to satisfy the triple requirement of a short, realistic and pedagogical example in order to show the different capabilities of possibilistic ATMS. The clauses describing how the system works contain three kinds of atoms:

- the assumptions, associated with each component which may be faulty: by convention, assumptions will be understood as "absence of fault", assumption  $A$  being true if component  $A$  works, false if it is faulty. These atoms are
  - OIL (there is still some oil)
  - BELT (the engine belt is not broken)
  - RECTIFIER (the rectifier is not cut off)
  - BATTERY (the battery is not faulty)
  - FUSE (the fuse is not melt)
- the "sensors" atoms, which describe observable facts whose truth value may be known by the user, but with a possibly non total certainty:
  - indicator (the temperature indicator is red)
  - ammeter (the charge ammeter is positive)
  - engine (the engine is running)

headlights (the headlights are working)  
horn (the horn is working)

- two intermediary atoms
  - coil (the coil is supplied)
  - secondary-circuits (the secondary circuits are supplied)

Uncertainty is involved at three different levels:

- on clauses describing how the system works: for instance, "if the secondary circuits are supplied then it is moderately certain (0.7) that headlights are working";
- on "sensors" atoms (we might be unsure of a result given by a sensor)
- on assumptions: by default, components are not faulty (but some faults are more plausible than others); it leads to the clauses ( $A_i \alpha_i$ ) expressing that we are certain to the degree  $\alpha_i$  that  $A_i$  will not be faulty.

The system is described by the following statements:

- (1) if the temperature indicator is red then there is no oil left or the belt is broken (certainty: 0.9)
- (2) if the temperature indicator is red then secondary circuits are supplied (certainty: 1)
- (3) if there is no oil left and if secondary circuits are supplied then the temperature indicator is red (certainty: 0.7)
- (4) if secondary circuits are supplied then the headlights are working (certainty: 0.7)
- (5) if the battery is not faulty then the headlights are working (certainty: 0.9)
- (6) if the engine is running then the coil is supplied (certainty: 1)
- (7) if the battery is not faulty then secondary circuits are supplied (certainty: 0.7)
- (8) if the engine is running, the belt is not broken and the rectifier is not cut off then the charge ammeter is positive (certainty: 0.8)
- (9) if the engine is running and the rectifier is not cut off then the charge ammeter is positive (certainty: 0.3)
- (10) if the coil is supplied and the fuse is not blown then the charge ammeter is positive (certainty: 0.6)
- (11) if the fuse is not blown and the battery is working then the horn is working (certainty: 0.9)

Then we write that by default, the components are not faulty:

- (12) the fuse is not melt (certainty: 0.2)
- (13) the belt is not broken (certainty: 0.4)
- (14) the battery is not faulty (certainty: 0.3)
- (15) the rectifier is not cut off (certainty: 0.1)
- (16) there is some oil left (certainty: 0.3)

It leads to the following clauses:

- C1  $\neg$ indicator  $\vee$   $\neg$ OIL  $\vee$   $\neg$ BELT (0.9)
- C2  $\neg$ indicator  $\vee$  secondary-circuits (1)
- C3 OIL  $\vee$   $\neg$ secondary-circuits  $\vee$  indicator (0.8)
- C4  $\neg$ secondary-circuits  $\vee$  headlights (0.7)
- C5  $\neg$ BATTERY  $\vee$  headlights (0.9)
- C6  $\neg$ engine  $\vee$  coil (1)
- C7  $\neg$ BATTERY  $\vee$  secondary-circuits (0.7)
- C8  $\neg$ engine  $\vee$   $\neg$ RECTIFIER  $\vee$   $\neg$ BELT  $\vee$  ammeter (0.8)
- C9  $\neg$ engine  $\vee$   $\neg$ RECTIFIER  $\vee$  ammeter (0.3)
- C10  $\neg$ coil  $\vee$  ammeter  $\vee$   $\neg$ FUSE (0.6)
- C11  $\neg$ FUSE  $\vee$   $\neg$ BATTERY  $\vee$  horn (0.9)
- C12 FUSE (0.2)
- C13 BELT (0.4)
- C14 BATTERY (0.3)
- C15 RECTIFIER (0.1)
- C16 OIL (0.3)

Let us point that clause C3 is not a Horn clause and cannot be represented by a production rule.

#### 4. Basic Principles and Definitions of a Possibilistic A.T.M.S.

Classical ATMS require that the clauses contained inside the knowledge base (justifications and disjunctions of assumptions) are certain; we may wish to handle more or less uncertain information without losing the capacities of the ATMS. The basic principle of the  $\Pi$ -ATMS introduced in [19] is to associate to each clause a weight  $\alpha$  which is a lower bound of its necessity degree<sup>1</sup>. Assumptions may also be weighted, i.e. the user or the inference engine may decide at any time to believe an assumption with a given certainty degree. A  $\Pi$ -ATMS is able to answer the following questions:

- (i) Under what configuration of the assumptions is a fact  $d$  certain to some degree? (i.e., what assumptions shall we consider as true, and with what certainty degrees in order to have  $d$  certain to degree  $\alpha$ ?)
- (ii) What is the inconsistency degree of a given configuration of assumptions ?
- (iii) In a given configuration of assumptions, to what degree is each observed fact certain ?

---

<sup>1</sup> It is not difficult to define possibilistic ATMS with both lower bounds of necessity degrees and possibility degrees; however, for the sake of clarity, we restrict ourselves in this paper to necessity-valued possibilistic ATMS.

The kind of classical ATMS extended here is Cayrol and Tayrac's [5, 6, 7] generalized ATMS, where each piece of information is represented by a (general) propositional clause. It make it possible to have:

- a uniform representation for all pieces of knowledge (no differentiated storage and treatment between justifications and disjunctions of assumptions).
- the capability of handling negated assumptions as assumptions, i.e. environments and nogoods may contain negations of assumptions (they are called generalized environments and nogoods by Cayrol and Tayrac); this approach differs from DeKleer "NATMS" [12] where negated assumptions do not appear inside the environments.
- a simple and uniform algorithm for the computation of labels, based on a restricted form of resolution (see Section 5).

### *Environments*

First, the basic notions attached to the classical ATMS can be generalized. Let  $\Sigma$  be a set of necessity-valued clauses. Let  $E$  be a set of valued assumptions; the following definitions are useful:

- )  $[E \alpha]$  is an environment of the fact  $d$  if and only if  $N(d) \geq \alpha$  is a logical consequence of  $E \cup \Sigma$ , where the assumptions of  $E$  are considered as certainly true (the certainty degree of the associated clauses is 1);
- )  $[E \alpha]$  is an  $\alpha$ -environment of  $d$  if and only if  $[E \alpha]$  is an environment of  $d$  and if  $\forall \alpha' > \alpha, [E \alpha']$  is not an environment of  $d$  ( $\alpha$  is maximal);
- )  $[E \alpha]$  is an  $\alpha$ -contradictory environment, or  $\alpha$ -nogood if and only if  $E \cup \Sigma$  is  $\alpha$ -inconsistent (i.e.  $E \cup \Sigma \models (\perp \alpha)$ ), with  $\alpha$  maximal. The notation  $nogood_{\alpha} E$  is used. The  $\alpha$ -nogood  $[E \alpha]$  is said to be minimal if there is no  $\beta$ -nogood  $[E' \beta]$  such that  $E \supset E'$  and  $\alpha \leq \beta$ .

### *Labels*

In order to define the label of a fact  $d$ , we consider only non-weighted assumptions (i.e. they will have the implicit weight 1). It can be shown that it is useless to weight the assumptions inside the labels (this remark holds also for the base of no-goods). The label of the fact  $d$ ,  $L(d) = \{[E_i \alpha_i], i \in I\}$  is the unique fuzzy subset of the set of environments for which the four following properties hold:

- (*weak*) *consistency*:  $\forall [E_i \alpha_i] \in L(d), E_i \cup \Sigma$  is  $\beta$ -inconsistent, with  $\beta < \alpha_i$  (i.e.  $\text{Inc}(E_i \cup \Sigma) < \alpha_i$  in the sense of Section 2, the certainty degree associated to the  $E_i$ 's being 1); it guarantees that either  $E_i$  is consistent (i.e.  $\beta = 0$ ), or its inconsistency degree is anyway strictly less than the certainty with which  $d$  can be deduced from  $E_i \cup \Sigma$  (i.e. we are sure to use a consistent sub-base of  $E_i \cup \Sigma$  to deduce  $d$ , see [17, 21]).

- *soundness*:  $L(d)$  is sound if and only if  $\forall [E_i \alpha_i] \in L$  we have  $E_i \cup \Sigma \models (d \alpha_i)$  where  $\models$  has been defined in Section 2; i.e.,  $L(d)$  contains only environments of  $d$ .
- *completeness*:  $L(d)$  is complete if and only if for every environment  $E'$  such that  $E' \cup \Sigma \models (d \alpha')$  then  $\exists i \in I$  such that  $E_i \subset E'$  and  $\alpha_i \geq \alpha'$ . I.e., all minimal  $\alpha$ -environments of  $d$  are present in  $L(d)$ .
- *minimality*:  $L(d)$  is minimal if and only if it does not contain two environments  $(E_1 \alpha_1)$  and  $(E_2 \alpha_2)$  such that  $E_1 \subset E_2$  and  $\alpha_1 \geq \alpha_2$ . It means that  $L(d)$  only contains the most specific  $\alpha$ -environments of  $d$  (i.e. all their assumptions are useful).

Ranking environments according to their weight in the label of each fact provides a way for limiting the consequences of combinatorial explosion (the problem was already pointed out by Provan [27] and Raiman [28]): indeed when a label contains too many environments, the  $\prod$ -ATMS can help the user by giving the environments with the greatest weight(s) only.

These notions are now illustrated on the example introduced in Section 3. The only nogoods existing in the system are negations of all "non-fault" assumptions, i.e.  $\text{nogood}_{0,4}\{\neg\text{BELT}\}$ ,  $\text{nogood}_{0,3}\{\neg\text{BATTERY}\}$ , etc. A nogood containing only a positive "non-fault" atom would mean that it is somewhat certain that a given component has to be faulty whatever the results given by the sensors, which would not be too realistic; a nogood containing several "non-fault" atoms (positive or negative) would mean that some faults are not independent (it is not the case in our example). Once the system has been described, let us introduce information given by the sensors. Let us first introduce "the engine is running (certainty: 1)".

C17 engine (1)

The label of the statement "the charge ammeter is positive" is:

$\{\{\}_{0,2}, \{\text{RECTIFIER}\}_{0,3}, \{\text{RECTIFIER}, \text{BELT}\}_{0,8}, \{\text{FUSE}\}_{0,6}\}$ , meaning that the charge ammeter is positive with a high certainty degree (0.8) if the rectifier is not cut off and the belt is not broken, with a moderate certainty degree (0.6) if the fuse is not blown, and with a low certainty degree (0.3) if the rectifier is not cut off (and no extra assumption). Moreover, the presence in the label of the empty environment weighted by 0.2 means that if no extra information is given by the sensors, it is already certain to the degree 0.2 that the charge ammeter is positive.

$\{\{\}_{0,2}, \{\text{RECTIFIER}\}_{0,3}, \{\text{RECTIFIER}, \text{BELT}\}_{0,9}, \{\text{FUSE}\}_{0,6}\}$  would not be a correct label for the same fact since adding to the clauses already present in the system the two clauses (RECTIFIER 1) and (BELT 1) does not enable to deduce (ammeter 0.9).

$\{\{\}_{0,2}, \{\text{RECTIFIER}\}_{0,3}, \{\text{RECTIFIER}, \text{BELT}\}_{0,8}, \{\text{FUSE}\}_{0,6}, \{\text{FUSE}, \text{BELT}\}_{0,5}\}$  would not be minimal since  $\{\text{FUSE}, \text{BELT}\}_{0,5}$  is a redundant environment, being subsumed by  $\{\text{FUSE}\}_{0,6}$ . Contrarily, having in the label the  $\alpha$ -environments

$\{\text{RECTIFIER}\}_{0.3}$  and  $\{\text{RECTIFIER, BELT}\}_{0.8}$  keeps the label minimal, since none of these two  $\alpha$ -environments is redundant.

$\{\{\text{RECTIFIER}\}_{0.3}, \{\text{RECTIFIER, BELT}\}_{0.8}, \{\text{FUSE}\}_{0.5}\}$  would not be complete since (FUSE 1) enables to deduce that  $N(\text{ammeter}) \geq 0.6$ , which is not expressed in the above set of environments since the highest certainty degree with which the fact can be deduced is only 0.5.

$\{\}_{0.2}, \{\text{RECTIFIER}\}_{0.3}, \{\text{RECTIFIER, BELT}\}_{0.8}, \{\text{FUSE}\}_{0.6}, \{\neg\text{BATTERY}\}_{0.2}\}$  would not be weakly consistent since  $\{\neg\text{BATTERY}\}$  is a 0.3-nogood.

Besides, the label of "indicator" is  $\{\{\neg\text{OIL, BATTERY}\}_{0.7}\}$ , meaning that the temperature indicator is red with a high certainty degree if there is no oil left and if the battery is working.

### Contexts

To extend the ATMS notion of context, we now consider *weighted assumptions*. A weighted assumption is a couple  $(H \alpha)$  where  $H$  is an assumption and  $\alpha \in [0,1]$  is the a priori certainty degree assigned to  $H$ .

The *context* associated with the set of weighted assumptions  $\mathcal{C}$  is the set of all couples  $(d, \text{val}_{\mathcal{C}}(d))$ , where  $d$  is a fact or an assumption, and  $\text{val}_{\mathcal{C}}(d) = \sup\{\alpha, \mathcal{C} \cup \Sigma \models (d \alpha)\}$ . Let us now give the following theorem:

Let  $\mathcal{C}$  be a set of valued assumptions. Let  $d$  be a fact; it can be shown that  $\mathcal{C} \cup \Sigma \models (d \alpha)$  in possibilistic logic if and only if  $\exists [E_i \alpha_i] \in \text{Label}(d)$ ,  $E_i = \{H_{i,1}, H_{i,2}, \dots, H_{i,n}\}$  such that

- (i)  $\mathcal{C}^* \supset E_i$  where  $\mathcal{C}^*$  is the classical set of assumptions obtained from  $\mathcal{C}$  by ignoring the weights.
- (ii)  $\alpha \leq \min(\alpha_i, \beta_1, \beta_2, \dots, \beta_n)$  where  $\beta_1, \beta_2, \dots, \beta_n$  are the weights attached to  $H_{i,1}, H_{i,2}, \dots, H_{i,n}$  in  $\mathcal{C}$ .
- (iii)  $\alpha > \text{Inc}(E \cup \Sigma)$ .

The proof of the theorem is obvious using the results of classical ATMS. This theorem gives an immediate algorithm to compute contexts, given the label of every fact:

```

 $\alpha_{\text{max}} \leftarrow 0;$ 
  For every  $[E_i \alpha_i] \in \text{Label}(d)$  do
    if  $\mathcal{C}^* \supset E_i$ 
      then  $\alpha_{\text{max}} \leftarrow \max(\alpha_{\text{max}}, \min(\alpha_i, \beta_1, \beta_2, \dots, \beta_n))$  where
         $\beta_1, \dots, \beta_n$  are the weights attached to  $H_{i,1}, \dots, H_{i,n}$  in  $\mathcal{C}$ .
    end

```

Intuitively, for each environment of the label of  $d$  included in  $\mathcal{E}^*$ , the algorithm computes the degree of certainty with which this environment entails  $d$ ; this degree depends upon the justifications used in deriving  $d$  (via  $\alpha_i$ ) and the weights of the assumptions in this environment.

**Example** (continued):

Let us consider the system  $\{C1, \dots, C17\}$  and let  $E$  be the environment  $E = \{\text{RECTIFIER}, \text{BELT}, \text{BATTERY}\}$ . The context associated with  $E$  is:

$\{ \underline{\text{RECTIFIER}} (1), \underline{\text{BELT}} (1), \underline{\text{BATTERY}} (1), \text{FUSE} (0.4), \text{OIL} (0.1), \text{engine} (1), \text{coil} (1), \text{horn} (0.4), \text{secondary-circuits} (0.7), \text{headlights} (0.9), \text{ammeter} (0.8) \}$  (the non-underlined atoms were already deducible before adding the three assumptions), which means that in the configuration of assumptions where it is certain that the rectifier, the belt and the battery are working, then it is certain to the degree 0.7 that the secondary circuits are supplied, etc.

Let now the environment contain *weighted assumptions*  $\mathcal{E}^1 = \{(\text{RECTIFIER } 0.5), \text{BELT} (0.4), \text{BATTERY} (0.6)\}$ . The context associated with  $\mathcal{E}^1$  is  $\{ \underline{\text{RECTIFIER}} (0.5), \underline{\text{BELT}} (0.4), \underline{\text{BATTERY}} (0.6), \text{FUSE} (0.4), \text{OIL} (0.1), \text{engine} (1), \text{coil} (1), \underline{\text{horn}} (0.4), \underline{\text{secondary-circuits}} (0.6), \underline{\text{headlights}} (0.6), \underline{\text{ammeter}} (0.4) \}$ .

*Fault diagnosis*

Let us now add to the system, not only that the engine is running, but also the two pieces of information:

C18    indicator (1)  
C19     $\neg$  ammeter (0.9)

The system is now 0.3-inconsistent: indeed, from the clauses C1, C13, C16 et C18, a contradiction of degree 0.3 can be deduced. It was somewhat certain at the beginning that the belt was not broken (C13) and that there was some oil left (C16). The extra information (C18) enables us to deduce that these two "non-fault" assumptions are mutually exclusive, and with a greater certainty (0.9). The new minimal nogoods are  $\text{nogood}_{0.6}\{\text{FUSE}\}$ ,  $\text{nogood}_{0.8}\{\text{RECTIFIER}, \text{BELT}\}$  and  $\text{nogood}_{0.9}\{\text{OIL}, \text{BELT}\}$ ,  $\text{nogood}_{0.4}\{\text{RECTIFIER}\}$ ,  $\text{nogood}_{0.9}\{\text{OIL}\}$  expressing that:

- it is certain to the degree 0.6 that the fuse is blown;
- it is certain to the degree 0.8 that the rectifier is cut off or that the belt is broken;
- it is certain to the degree 0.9 that there is no oil left or that the belt is broken.

Moreover we have, like in the initial system,  $\text{nogood}_{0.4}\{\neg\text{BELT}\}$ ; contrarily,  $\{\neg\text{FUSE}\}$ ,  $\{\neg\text{RECTIFIER}\}$ ,  $\{\neg\text{BATTERY}\}$  and  $\{\neg\text{OIL}\}$  are not minimal nogoods anymore since

their membership degree to the nogood base is now less than the inconsistency degree of the knowledge base, which can be translated by  $\text{nogood}_{0.3}$ .

In classical logic, in order to provide the user with an answer under the form of a set of alternative fault configurations, the nogoods

$$\neg\text{FUSE} \wedge (\neg\text{RECTIFIER} \vee \neg\text{BELT}) \wedge (\neg\text{OIL} \vee \neg\text{BELT})$$

are expressed in the equivalent disjunctive form:

$$(\neg\text{FUSE} \wedge \neg\text{RECTIFIER} \wedge \neg\text{OIL}) \vee (\neg\text{FUSE} \wedge \neg\text{BELT}).$$

In possibilistic logic, the transformation is trickier since, if from  $(\neg\text{FUSE} 0.6) \wedge (\neg\text{RECTIFIER} \vee \neg\text{BELT} 0.8) \wedge (\neg\text{OIL} \vee \neg\text{BELT} 0.9)$  it can be deduced that  $N(\neg\text{FUSE} \wedge (\neg\text{RECTIFIER} \vee \neg\text{BELT}) \wedge (\neg\text{OIL} \vee \neg\text{BELT})) \geq 0.6$ , and then that  $N((\neg\text{FUSE} \wedge \neg\text{RECTIFIER} \wedge \neg\text{OIL}) \vee (\neg\text{FUSE} \wedge \neg\text{BELT})) \geq 0.6$ , however nothing can be immediately concluded on  $N(\neg\text{FUSE} \wedge \neg\text{RECTIFIER} \wedge \neg\text{OIL})$  and  $N(\neg\text{FUSE} \wedge \neg\text{BELT})$ , which does not enable us to rank the fault configurations. On the contrary, one can try to find a lower bound of the certainty of these configurations of faults, by refutation, by adding their negation (weighted by 1) to the system. Thus we obtain that  $N(\neg\text{FUSE} \wedge \neg\text{RECTIFIER} \wedge \neg\text{OIL}) \geq 0.4$  (but no non-trivial lower bound can be found for  $N(\neg\text{FUSE} \wedge \neg\text{BELT})$ ), which suggests to the user that the configuration (fuse blown, rectifier cut off, no oil left) is more plausible than (fuse blown, belt broken).

#### *Interpretations in possibilistic logic*

As said above an environment in the classical ATMS is a set of assumptions, and every set of assumptions characterizes a context. An environment E is said to be an interpretation (in the classical ATMS) if it is consistent and if adding any assumption to it make it inconsistent, namely E satisfies the two following conditions;

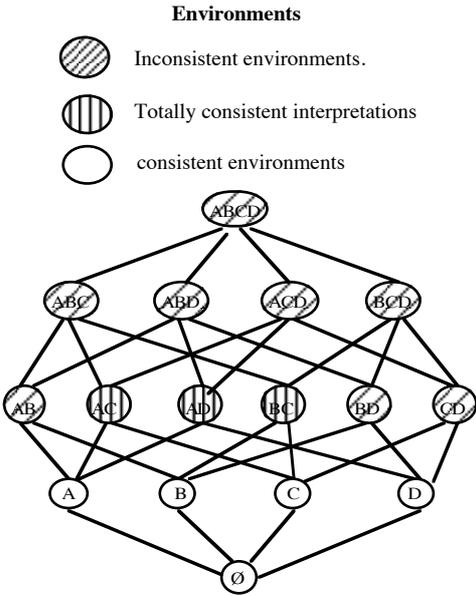
1. E is consistent (with  $\Sigma$ )
2.  $\forall H \notin E$  (H is an assumption),  $E \cup \{H\}$  is inconsistent.

Interpretations are complete descriptions of a given situation in terms of assumptions. One approach for computing interpretations consists in building a lattice of environments  $(2^T, \subseteq)$  where T is the set of all assumptions, and nodes in the lattice represent subsets of assumptions. This lattice is obtained by drawing an edge from each environment E to every subset of this environment. The top of the lattice is the set of all assumptions and the bottom is the empty set. Figure 1 gives an example of such a lattice for the set of assumptions  $T = \{A, B, C, D\}$ .

Notice that if a given environment E is a nogood then all of its supersets are contradictory environments as well. So, once nogoods are computed, we remove their elements (and hence their supersets) from the lattice, and therefore only the consistent environments remain and hence any maximal set of assumptions (relatively to the

inclusion relation  $\subseteq$ ) represents an interpretation. For example, in Figure 1, where the set of nogoods is  $\{\{A,B\}, \{C,D\}\}$ , the set of interpretations is  $\{\{A,C\}, \{A,D\}, \{B,C\}, \{B,D\}\}$ .

Figure 1



In possibilistic logic, an interpretation E may be partially inconsistent, namely it may contains a nogood which does not however violate weak consistency. For the need of the application, we say that an interpretation E is totally consistent if it satisfies the two following requirements:

1.  $Incons(\Sigma \cup E) = 0$
2.  $\forall H \notin E$  (H is an assumption),  $Incons(\Sigma \cup E \cup \{H\}) > 0$ .

where  $Incons(X)$  is the degree of the inconsistency of X.

We now give another approach for computing totally consistent interpretations using the notions of *candidates* which are environments containing at least one assumption in every nogood. A candidate C is said to be *minimal* if it does not exist a candidate C' such that  $C' \subset C$ . The following proposition holds:

**Proposition:** the complement of a minimal candidate is a totally consistent interpretation.

**Proof**

| Let C be a minimal candidate, and I its complement.

Assume that I is inconsistent (partially or totally), then there exists a nogood  $\mathcal{N}^\circ$  which is included in I. By definition, a candidate C contains at least one element of every nogood, and particularly it contains an element of  $\mathcal{N}^\circ$ , so it contains also an element of I which contradicts our assumption that I is the complement of C.

Let us show now that I is maximal; suppose that  $I \cup \{H\}$  is consistent, then it does not contain any nogood and therefore for every nogood  $\mathcal{N}^\circ$  there exist an element which belongs to  $C - \{H\}$ , which means that  $C - \{H\}$  is a candidate and contradict therefore our assumption that C is minimal.

The following algorithm computes minimal candidates from the set of nogoods:

Procedure Computing-minimal-candidates

begin

```

/* let C    : the set of minimal candidates initialized to {{}}
   Nog     : the set of nogoods. */
- loop while Nog  $\neq \emptyset$ :
  - let  $\mathcal{N}^\circ$  an element of Nog;
  - remove  $\mathcal{N}^\circ$  from Nog
  -  $C := \{c_i \cup H_j / c_i \in C \text{ and } H_j \in \mathcal{N}^\circ\}$ 
  - remove from C all not minimal candidates.

```

end

and therefore totally consistent Interpretations can be computed from the following relation:

$$I = \{T - c / T \text{ is the set of all assumptions, and } c \text{ is a minimal candidate}\}.$$

**Example** (continued)

Let us consider the system  $\{C1, \dots, C16\}$  where the nogoods are:

$$\text{Nogoods} = \{[\{\neg\text{FUSE}\} 0.2], [\{\neg\text{BELT}\} 0.4], [\{\neg\text{BATTERY}\} 0.3], [\{\neg\text{OIL}\} 0.3], [\{\neg\text{RECTIFIER}\} 0.1]\}$$

The algorithm described below computes the following minimal candidates:

$$\text{Candidates} = \{[\{\neg\text{FUSE}, \neg\text{BELT}, \neg\text{BATTERY}, \neg\text{OIL}, \neg\text{RECTIFIER}\}]\}$$

and therefore by complementation, we find only one totally consistent interpretation:

$$I = \{[\{\text{FUSE}, \text{BELT}, \text{BATTERY}, \text{OIL}, \text{RECTIFIER}\}]\}$$

Remark:

In this section we have seen that the degrees associated to nogoods are not used to build minimal candidates. We will see in Section 6. that the use of the valuations is very important for revising inconsistent possibilistic knowledge bases and reducing the complexity of the revision process by selecting a smaller number of maximal consistent sub-bases.

## 5. Basic Algorithms of a Possibilistic ATMS

The type of ATMS that we extend to possibilistic logic is the one proposed by Cayrol and Tayrac [5, 6, 7]. Contrary to De Kleer's [9, 10] classical ATMS, this system, called  $\pi$ -RCO (Oriented CAT-correct Possibilistic Resolution) (CAT: Clause whose Antecedent is Typed) uses possibilistic resolution in order to deduce a specific type of clauses of interest for the ATMS, called  $\pi$ -CATI (Possibilistic Clause whose Antecedent is Typed and Interesting).

As in the classical ATMS, the set of data (called also "literals" or "atomic propositions") is divided into two groups: T and NT containing respectively the assumption data and the non-assumption ones.

The computation of labels and nogoods requires some further definitions.

### 5.1. Definition of a $\pi$ -CATI

A clause is of the form  $A \rightarrow B$  where A is a conjunction of positive literals and B is a disjunction of positive literals. Each propositional formula has a unique clausal representation. A is called antecedent and B consequent. A weighted clause  $(C \alpha)$  is a  $\pi$ -CATI, if and only if:

1. every literal in the antecedent side of C is an assumption, and,
2. it contains at most one non-assumption literal in the consequent side of C.

If  $(C \alpha)$  satisfies only the first condition, call it a  $\pi$ -CAT (Possibilistic Clause with Typed Antecedent).

**Example:**

Let  $T = \{A,B\}$ , and  $NT = \{a,b,c,d\}$

The weighted clause  $(A \rightarrow aB \ \alpha_1)$  is a  $\pi$ -CATI, because A belongs to T and it contains exactly one non-assumption literal (a) in his consequent side, in contrast with the following clauses  $(a \rightarrow b \ \alpha_2)$  and  $(A \rightarrow cd \ \alpha_3)$  which are not  $\pi$ -CATI because the first one contains a literal in his antecedent side which is not an assumption (a), and the latter has two non-assumption data in its consequent side (c and d).

## 5.2. $\pi$ -RCO Possibilistic Resolution

The basic idea of the  $\pi$ -RCO (Oriented CAT-correct Possibilistic Resolution) is to use the possibilistic resolution rule in order to deduce every  $\pi$ -CATI which is logically entailed from the initial set of clauses.

In order to minimize the number of applications of possibilistic resolution, an arbitrary but complete ordering  $O$  is introduced on the set of non-assumption data  $NT$ .

The  $\pi$ -RCO consists in applying possibilistic resolution between two weighted clauses  $C1$  and  $C2$  of the form:

$$\begin{array}{rcll}
 C1: & A1 & \rightarrow \{x\} \cup B1 & \alpha1 \\
 C2: & A2 \cup \{x\} & \rightarrow B2 & \alpha2 \\
 \hline
 \mathfrak{R}(C1,C2): & A1 \cup A2 & \rightarrow B1 \cup B2 & \min(\alpha1,\alpha2)
 \end{array}$$

where  $\mathfrak{R}(C1,C2)$  denotes the resolvent of  $C1$  and  $C2$ , under one of the two following conditions:

1.  $C1$  and  $C2$  are  $\pi$ -CATI such that the consequent side of  $\mathfrak{R}(C1,C2)$  contains at most one non-assumption literal.
2. the three following conditions are satisfied:
  - 2.1.  $C1$  is a  $\pi$ -CAT,
  - 2.2.  $x$  is the smallest (relatively to  $O$ ) non-assumption datum in the antecedent side of  $C2$ .
  - 2.3.  $x$  is one of the two smallest non-assumption data in the consequent side of  $C1$ .

### Example:

$$\text{Let } T = \{A,B\}, NT = \{a,b,c\}, O: a < b < c$$

and the initial set of clauses is:

$$\begin{array}{rcl}
 (A \rightarrow c & \alpha1) \\
 (B \rightarrow abc & \alpha2) \\
 (c \rightarrow a & \alpha3)
 \end{array}$$

The resolution between the clauses  $(B \rightarrow abc \ \alpha2)$  and  $(\underline{c} \rightarrow a \ \alpha3)$  is not a  $\pi$ -RCO resolution because  $c$  does not belong to the two smallest non-assumption data in the consequent side of the clause  $(B \rightarrow abc \ \alpha2)$ . Thus, only the resolution between the clauses  $(A \rightarrow \underline{c} \ \alpha1)$  and  $(\underline{c} \rightarrow a \ \alpha3)$  is a  $\pi$ -RCO resolution.

It can be proved that the use of  $\pi$ -RCO is complete for deduction of all possible  $\pi$ -CATI. A proof of  $\pi$ -RCO completeness in the classical case can be found in (Tayrac [7]).

### 5.3. The Algorithm

As in the classical ATMS, the truth maintenance system  $\pi$ -RCO is incremental, provided that the set of clauses  $\Sigma$  is saturated, which means that before the introduction or the deduction of a new clause, all possible possibilistic resolutions  $\pi$ -RCO are executed, thus only the possibilistic resolutions which are enabled by the newly introduced clause are processed.

On the other hand, after the introduction or the inference of the new clause, the set of clauses  $\Sigma$  must be minimized by removing every tautology and every subsumed clause.

We recall that a clause  $(A \rightarrow B \ \alpha_1)$  is a tautology, if and only if, the intersection of the sets of literals A and B is not empty. The clause  $(A_1 \rightarrow B_1 \ \alpha_1)$  subsumes the clause  $(A_2 \rightarrow B_2 \ \alpha_2)$ , if and only if,  $A_2 \supseteq A_1$ ,  $B_2 \supseteq B_1$  and  $\alpha_1 \geq \alpha_2$ .

Each time a new clause is introduced in the system, it is checked if it is a tautology or a subsumed clause; then, if none of these holds, all clauses which are subsumed by the newly introduced clause are removed, and lastly all possible  $\pi$ -RCO resolutions are triggered with this new clause.

This cycle is repeated for every deduced clause, and we give below an algorithm summarizing these operations:

```

procedure introduce (cl)
begin
    ; cl is the newly introduced clause,  $\Sigma$  represents the knowledge base.
    J:={cl}; J contains all deduced or introduced clauses.
    while (J $\neq\emptyset$ ) do
        .choose a weighted clause c of J.
        .remove c from J.
        .if c is not subsumed and c is not a tautology then
            .remove all clauses from  $\Sigma$  which are subsumed by c.
            .add c to  $\Sigma$ .
            .K:=all- $\pi$ -RCO(c)
            ; K contains all possible clauses which allow us to apply  $\pi$ -RCO
            ; resolution with c.
            while (K $\neq\emptyset$ ) do
                .remove a weighted clause c1 from K
                .add to J the  $\pi$ -RCO resolvent  $\mathfrak{R}(c1,c)$ 
    end introduce

```

### 5.4. Using $\pi$ -RCO in a Possibilistic ATMS

We show in this section how possibilistic resolution  $\pi$ -RCO can be used to compute every minimal nogood and the label of each datum.

#### 5.4.1. Computing labels

We recall that the label of a literal  $d$ ,  $L(d) = \{[E_i \alpha_i] / i \in I\}$ , is the only fuzzy subset of the set of environments which is weakly consistent, sound, complete and minimal, where  $E_i$  denotes a set of assumptions and negated assumptions.

By definition,  $[E \alpha]$  is an environment of a literal  $d$ , relatively to the set of clauses  $\Sigma$ , if and only if,  $E \cup \Sigma \models (d \alpha)$ ; this is equivalent to  $\Sigma \models (E \rightarrow d \alpha)$  [21, 24]. Thus an environment of a literal  $d$  is characterized by a clause containing exactly the literal  $d$  in his consequent side, and only assumptions or negated assumptions in his antecedent side. Obviously, this clause is a  $\pi$ -CATI. Thus, the label of a non-assumption literal  $d$  is:

$$L(d) = \{(P \cup \{\neg x / x \in Q\} \alpha) / (P \rightarrow \{d\} \cup Q \alpha) \in \Sigma \text{ is a } \pi\text{-CATI}\}.$$

#### 5.4.2. Computing nogoods

An environment  $[E \alpha]$  is a nogood if and only if  $(\perp \alpha)$  is a logical consequence of  $E \cup \Sigma$ , i.e  $\Sigma \models (E \rightarrow \perp \alpha)$ . Thus a nogood is denoted by a clause containing only assumptions in their antecedent part; these clauses are also  $\pi$ -CATI. Therefore, the nogoods are computed as below:

$$\text{Nogood} = \{(P \cup \{\neg x / x \in Q\} \alpha) / (P \rightarrow Q \alpha) \in \Sigma \text{ is a } \pi\text{-CATI and } T \supseteq Q\}.$$

### 6. Revising Knowledge Bases

In classical logic, when the inconsistency appears in a given knowledge base, then any formula is a logical consequence of this base, and therefore we can infer from this base that both  $p$  and its denial  $\neg p$  are simultaneously true. In possibilistic logic, the situation is less trivial since it is possible to use such bases considering as valid all formulas having valuations strictly higher than the degree of inconsistency of the knowledge base (the others being inhibited). This strategy verifies the revision rationality postulates of Alchourron, Gärdenfors and Makinson [23]. However, at the syntactic level it is not entirely satisfactory because some of the inhibited clauses have nothing to do with the inconsistency. For instance the knowledge base  $\Sigma = \{(a .5), (b .6), (c .4), (\neg a \vee \neg b \ 1)\}$  has a level of inconsistency equal to .5 due to the contradictory clauses (a. 5), (b. 6) and  $(\neg a \vee \neg b \ 1)$ . Hence (c. 4) will be inhibited. However it seems more satisfactory to delete (a. 5) from  $\Sigma$  in order to restore consistency rather than deleting both (a. 5) and (c. 4). Indeed a more parsimonious revision is obtained by deleting only the least certain formulas involved in a contradiction. It suggests to choose the revised knowledge base among the largest ones which preserve consistency.

### 6.1. Preferred consistent sub-bases

**Definition 6.1:** A sub-base  $\Sigma_i$  of  $\Sigma$  is said to be maximal consistent if it is consistent and if adding any formula  $\Phi$  of  $(\Sigma - \Sigma_i)$  to  $\Sigma_i$  produces an inconsistent knowledge base.

In practice, the number of maximal consistent sub-bases of an inconsistent knowledge base increases proportionally with the size of the base and in general, it is not possible to take into account all the maximal consistent sub-bases. The idea is then to take advantage of the weights to rank-order the maximal consistent sub-bases and keep only the best ones.

The wish to delete only the least certain formulas which cause an inconsistency leads to consider maximal consistent sub-bases that obey the following definition:

**Definition 6.2:** A sub-base  $\Sigma_i$  is said to be strongly maximal consistent sub-base if and only if:

- (i)  $\Sigma_i$  is totally consistent; i.e.  $\exists \alpha > 0, \Sigma \models (\perp \alpha)$
- (ii)  $\forall (\phi \alpha) \in \Sigma - \Sigma_i, \text{Incons}(\Sigma_i \cup \{(\phi \alpha)\}) = \alpha > 0.$

There is another way of considering a strongly maximal consistent sub-base. Namely,  $\Sigma$  can be viewed as a layered knowledge base, i.e.  $\Sigma = B_1 \cup B_2 \cup \dots \cup B_n$  where  $B_j$  contains only clauses of necessity weight  $\alpha_j$ , and  $\alpha_1 = 1 > \alpha_2 > \dots > \alpha_n$ . Then  $\Sigma_i$  is a strongly maximal consistent sub-base of  $\Sigma$  if and only if it can be constructed as follows

$$\Sigma_i = E_1 \cup E_2 \cup \dots \cup E_n \text{ where } \forall j = 1, n, E_1 \cup E_2 \cup \dots \cup E_j$$

is a maximal consistent sub-base of  $B_1 \cup B_2 \cup \dots \cup B_j$ . The proof of this equivalence is in the annex. Such maximal consistent sub-bases are exactly what Brewka [35] calls "preferred sub-theories" (see also Testemale [31]). An algorithm for computing the strongly maximal consistent sub-bases can be found in [24].

We can also introduce an ordering among the maximal consistent sub-bases of an incoherent knowledge base  $\Sigma$ . A maximal consistent sub-base  $\Sigma_i$  is said to be lexicographically preferred to  $\Sigma_j$ , denoted by  $\Sigma_i \leq \mathcal{L} \Sigma_j$  if and only if  $\mathcal{L}(\Sigma - \Sigma_i) \leq \mathcal{L}(\Sigma - \Sigma_j)$  where  $\mathcal{L}(X) = \{\alpha_i / (\Phi_i \alpha_i) \in X\}$  is a multiset of valuations of formulas and " $\leq$ " is defined as:

$$\begin{aligned} & \mathcal{L}(\Sigma - \Sigma_i) \leq \mathcal{L}(\Sigma - \Sigma_j) \\ \Leftrightarrow & \text{(i) } \mathcal{L}(\Sigma - \Sigma_i) = \emptyset \\ \text{or} & \text{(ii) } \text{Max}(\mathcal{L}(\Sigma - \Sigma_i)) < \text{Max}(\mathcal{L}(\Sigma - \Sigma_j)) \\ \text{or} & \text{(iii) } \text{Max}(\mathcal{L}(\Sigma - \Sigma_i)) = \text{Max}(\mathcal{L}(\Sigma - \Sigma_j)) \\ \text{and} & \mathcal{L}(\Sigma - \Sigma_i) - \text{Max}(\mathcal{L}(\Sigma - \Sigma_i)) \leq \mathcal{L}(\Sigma - \Sigma_j) - \text{Max}(\mathcal{L}(\Sigma - \Sigma_j)). \end{aligned}$$

It has been shown [20] that the minimal element of  $\Sigma$ 's sub-bases of an inconsistent knowledge base  $\Sigma$  relatively to the ordering " $\leq_{\mathcal{L}}$ " are strongly maximal consistent sub-bases.

It is important to distinguish between strongly maximal consistent sub-bases and lexicographically maximal consistent sub-bases. Although each lexicographically maximal consistent sub-base is a strongly maximal consistent sub-base, the contrary is not true. Indeed, consider an example where our knowledge base contains the following set of uncertain clauses:  $\Sigma = \{C_1: (a \ 0.8), C_2: (a \rightarrow b \ 0.7), C_3: (\neg a \ 0.8), C_4: (\neg a \rightarrow \neg b \ 0.6), C_5: (\neg b \ 0.6)\}$ . It is easy to check that  $\Sigma$  is partially inconsistent and admits two strongly maximal consistent sub-bases  $\Sigma_1 = \{C_1, C_2, C_4\}$  and  $\Sigma_2 = \{C_2, C_3, C_4, C_5\}$  but only  $\Sigma_2$  is lexicographically maximal consistent since  $\mathcal{L}(\Sigma - \Sigma_1) = \{0.8, 0.6\}$ ,  $\mathcal{L}(\Sigma - \Sigma_2) = \{0.8\}$  then  $\Sigma_2$  is preferred then  $\Sigma_1$ .

The next section shows how the  $\pi$ -ATMS can be used to compute the maximal consistent sub-bases of an inconsistent knowledge base.

## 6.2. Using the $\Pi$ -ATMS

The computation of maximal consistent sub-bases of an inconsistent knowledge base can easily be done using the functionalities of the  $\Pi$ -ATMS. We first point out the close relationship between totally consistent interpretations and maximal consistent sub-bases.

Let  $\Sigma = \{(C_i \ \alpha_i)\}$  be an inconsistent uncertain knowledge base to be revised. Let  $\Sigma'$  be a new knowledge base built from  $\Sigma$  by adding to the antecedent side of every clause  $C_i$  an assumption  $H_i$ , we obtain then:  $\Sigma' = \{(C'_i \ \alpha_i)\}$ , where  $C'_i = H_i \rightarrow C_i$ , we will say that the assumption  $H_i$  controls the clause  $C_i$ . We assume that only the set of data  $\{H_i\}$  are assumptions in  $\Sigma'$ . The  $\Pi$ -ATMS will work on  $\Sigma'$ , and we can show besides what the different notions known in the  $\Pi$ -ATMS represent for  $\Sigma$ . Notice first that an environment  $E = \{H_1, \dots, H_n\}$  represents the set of clauses  $\mathcal{C} = \{C_1, \dots, C_n\}$  in  $\Sigma$ . The following property is easy to check:  $\forall \mathcal{C} = \{(C_1 \ \alpha_1), \dots, (C_n \ \alpha_n)\} \subseteq \Sigma$ ,  $\mathcal{C}$  is inconsistent iff  $\exists \mathbb{H} \subseteq \mathcal{H} = \{H_1, \dots, H_n\}$  such that  $\mathbb{H}$  belongs to the nogood set of  $\Sigma'$ .

The nogoods of  $\Sigma'$  then represents the set of clauses of  $\Sigma$  which allows us to deduce the contradiction when we substitute every assumption  $H_i$  by the clause  $C_i$  it controls. Since the nogood has the completeness property, then all possible sets of clauses which are responsible for an inconsistency of  $\Sigma$  are computed by the  $\Pi$ -ATMS. An immediate application is to check the consistency of the knowledge base  $\Sigma$ . Indeed if the nogood base of  $\Sigma'$  is empty then  $\Sigma$  is consistent .

It is possible to relate the notion of totally consistent interpretations with the notion of the maximal consistent sub-bases, since, as it can be easily seen,  $\forall \mathcal{C} \subseteq \Sigma$ ,  $\mathcal{C}$  is a maximal consistent sub-base of  $\Sigma$  iff  $\mathcal{H} = \{H_1, \dots, H_n\}$  is a totally consistent interpretation of  $\Sigma'$ .

An immediate result of this fact is that the candidates in  $\Sigma'$  represent the smallest set of clauses such that if we remove their elements from  $\Sigma$ , then the knowledge base  $\Sigma$  becomes inconsistent.

The algorithm presented in Section 4 computes the set of totally consistent interpretations. And we have seen that each totally consistent interpretation represents a maximal consistent sub-base. Then the problem of selecting the lexicographically maximal consistent sub-bases require only to sort the set  $\mathcal{S} = \{\mathcal{L}(\Sigma - \Sigma_i)\}$  using the ordering " $<\mathcal{L}$ " defined above where  $\Sigma_i$  is a maximal consistent sub-base and  $\mathcal{L}(\Sigma - \Sigma_i)$  is a set of the weights of clauses which are in  $\Sigma - \Sigma_i$ .

### Example

Assume that our knowledge base contains the following set of uncertain clauses:

$$\Sigma = \{C_1: (a \ 0.8), C_2: (a \rightarrow b \ 0.7), C_3: (\neg a \ 0.8), C_4: (\neg a \rightarrow \neg b \ 0.6), \\ C_5: (\neg b \ 0.6)\}.$$

Let  $\Sigma'$  the possibilistic knowledge base be constructed from  $\Sigma$  by associating to each formula  $C_i$  in  $\Sigma$  a specific assumption  $H_i$ , then we obtain :

$$\Sigma' = \{(H_1 \rightarrow a \ 0.8), ((H_2 \wedge a) \rightarrow b \ 0.7), (H_3 \rightarrow \neg a \ 0.8), (H_4 \wedge \neg a \rightarrow \neg b \ 0.6), \\ (H_5 \rightarrow \neg b \ 0.6)\}.$$

The  $\pi$ -ATMS computes the base of nogoods which is equal to:

$$\text{Nogood}(\Sigma') = \{(H_1 \ H_2 \ H_5), \{H_1 \ H_3\}\}.$$

Then applying the algorithm presented in Section 4 we find three totally consistent interpretations  $I_1 = \{H_2 \ H_3 \ H_4 \ H_5\}$ ,  $I_2 = \{H_1 \ H_4 \ H_5\}$  and  $I_3 = \{H_1 \ H_2 \ H_4\}$  which correspond to the maximal consistent sub-bases  $\Sigma_1 = \{C_2 \ C_3 \ C_4 \ C_5\}$ ,  $\Sigma_2 = \{C_1 \ C_4 \ C_5\}$  and  $\Sigma_3 = \{C_1 \ C_2 \ C_4\}$ .

Deciding which sub-base is lexicographically maximal consistent requires to sort the set  $\mathcal{S} = \{\mathcal{L}(\Sigma - \Sigma_i)\} = \{\{0.8\}, \{0.7 \ 0.8\}, \{0.8 \ 0.6\}\}$  using the ordering " $<\mathcal{L}$ ". Then we only find  $\Sigma_1$  as a lexicographically consistent maximal sub-base.

### 6.3. Application to data fusion

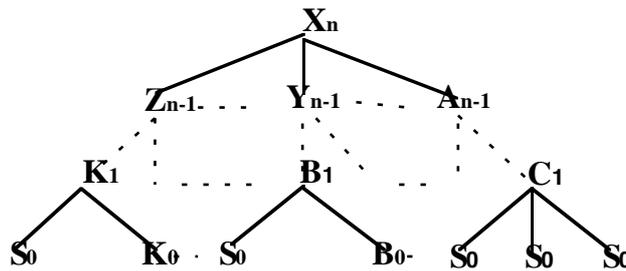
An application to data fusion problems has been carried out (Monai et al. [34]). It consists in finding a structure for a particular object from observations of elementary objects by aggregating the partial observations in a coherent manner.

Objects handled in the system are layered in different levels. Each level contains different kinds of objects. Every object of level  $i$  is either observed (then is said to be elementary) or constructed only from objects of level  $i - 1$  (it is then said to be composite). We assume that objects of level 0 are all elementary objects. The construction of a complex object must satisfy the following constraint:

"Two objects of level  $i$  containing a common object of level  $(i - 1)$  must not be aggregated together"

The knowledge base of the system is organised into  $n$  independent parts. Each part of the knowledge base gives the conditions for aggregating objects of level  $i$  to objects of higher levels.

Figure 2



The hierarchy shown in Figure 2 summarizes the way objects are aggregated. Here is an example of a rule of level 0:

$$\forall x_1, \forall x_2, S_0(x_1) \wedge B_0(x_2) \rightarrow B_1(t) \alpha(t).$$

This rule means that if we observed two objects  $x_1$  and  $x_2$  of level 0 belonging respectively to classes  $S_0$  and  $B_0$ , then these two objects can be aggregated to give an object of level 1, which belongs to the class  $B_1$ .  $\alpha(t)$  is a real number between 0 and 1 which estimates the compatibility of aggregating the objects  $S_0(x_1)$  and  $S_0(x_2)$ . For instance  $\alpha(t)$  may be all the greater as the locations of  $x_1$  and  $x_2$  are close to each other. If  $\alpha(t) = 0$  then  $S_0(x_1)$  and  $S_0(x_2)$  cannot be aggregated together (and in this case the rule will not be instantiated).

In a similar manner, a base of facts is organised into  $n$  independent parts. Each part  $i$  brings together all observed objects of level  $i$ .

#### Processing data

Processing data consists in aggregating observed objects, starting from the lower level in order to deduce the structure of objects of higher levels. The system executes, in a recursive manner (for every level  $i$ ), the two following steps:

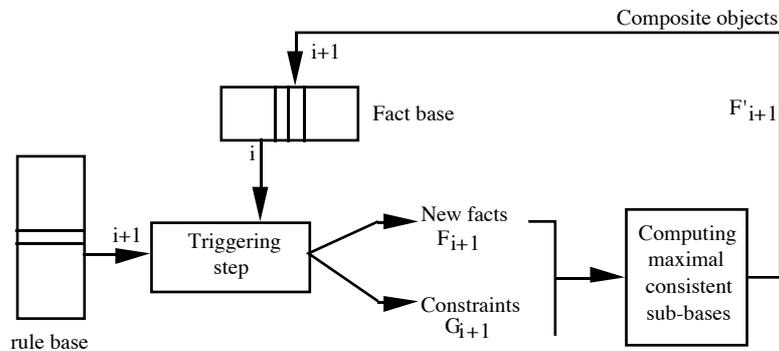
- 1 – Triggering rules (the rules produce potential objects)
- 2 – Computing maximal consistent sub-bases (the best objects are selected).

The first step consists in triggering rules relating facts of level  $(i - 1)$  to objects of level  $i$ . Two objects  $A_i(t_1)$  and  $A_i(t_2)$  are said to be contradictory if they contain at least one common sub-object (of level  $i - 1$ ). This constraint will be expressed by the following clause:

$$A_i(t_1) \wedge A_i(t_2) \rightarrow \perp \quad 1$$

A clause of this type is added each time two rules having a common instantiated variable have been triggered to form composite objects  $t_1$  and  $t_2$ . The knowledge base resulting from step 1 is composed of facts  $F_{i+1}$  (objects newly constructed from observed or constructed objects of level  $i$ ) and constraints  $G_{i+1}$  on these objects. This knowledge base is generally inconsistent (except if the set of constraints is empty).

Figure 3



The second step is added in order to limit the computational complexity due to the number of possible combinations of aggregating objects of level  $i$ . Its main role is then to only select the best sub-bases  $F'_{i+1}$  of the base  $F_{i+1}$  of potential composite objects. The elements of  $F'_{i+1}$  will be added to the base of facts of level  $i+1$ . The system will start again by applying the two steps described above to part  $i+1$  of the knowledge base. The schema of Figure 3 sums up the two steps and their iteration.

### Example

Assume our knowledge base contains only rules for aggregating objects of level 0 to level 1 (for the sake of simplicity, we restrict ourselves to the lowest level of hierarchy shown in Figure 2).

Let suppose that initially the set of facts  $F_0$  contains only the following elementary (observed) objects  $x_1, x_2, x_3, x_4$  and  $x_5$  which belong respectively to classes  $S_0, S_0, S_0, S_0$  and  $B_0$ . The set of constraints  $G_0$  is empty since all the objects in  $F_0$  are elementary and they do not share the same sub-object. Let the numbers  $(0.9, 0.8, 0.8, 0.7, 0, 0.8, 0.7, 0)$  be the estimates of the compatibility between objects respectively in the aggregations  $(x_1 \ x_2 \ x_3), (x_1 \ x_2 \ x_4), (x_1 \ x_3 \ x_4), (x_2 \ x_3 \ x_4), (x_1 \ x_5), (x_2 \ x_5), (x_3 \ x_5), (x_4 \ x_5)$ .

Let us show now how the algorithm presented above proceeds in aggregating the elementary objects. The first step of the algorithm (triggering step) allows us to deduce six complex objects  $C_1(t_1)$ ,  $C_1(t_2)$ ,  $C_1(t_3)$ ,  $C_1(t_4)$ ,  $B_1(t_1)$  and  $B_1(t_2)$  of level 1, constructed respectively from  $(x_1 x_2 x_3)$ ,  $(x_1 x_2 x_4)$ ,  $(x_1 x_3 x_4)$ ,  $(x_2 x_3 x_4)$ ,  $(x_2 x_5)$ ,  $(x_3 x_5)$ . In the six complex objects constructed in the first step, there are objects which share the same elementary object(s); therefore the system generates the following set of constraints  $G_1$ :

$$\begin{aligned} C_1(t_1) C_1(t_2) &\rightarrow \perp 1; C_1(t_1) C_1(t_3) \rightarrow \perp 1; C_1(t_1) C_1(t_4) \rightarrow \perp 1; \\ C_1(t_2) C_1(t_3) &\rightarrow \perp 1; C_1(t_2) C_1(t_4) \rightarrow \perp 1; C_1(t_3) C_1(t_4) \rightarrow \perp 1; \\ B_1(t_1) C_1(t_1) &\rightarrow \perp 1; B_1(t_1) C_1(t_2) \rightarrow \perp 1; B_1(t_1) C_1(t_4) \rightarrow \perp 1; \\ B_1(t_1) B_1(t_2) &\rightarrow \perp 1; B_1(t_2) C_1(t_1) \rightarrow \perp 1; B_1(t_2) C_1(t_3) \rightarrow \perp 1; \\ B_1(t_2) C_1(t_4) &\rightarrow \perp 1. \end{aligned}$$

Note that the knowledge base  $\Sigma$  composed of the set of facts  $F_1 = \{(C_1(t_1) 0.9), (C_1(t_2) 0.8), (C_1(t_3) 0.8), (C_1(t_4) 0.7), (B_1(t_1) 0.8), (B_1(t_2) 0.7)\}$  and the previous set of constraints  $G_1$  is partially inconsistent. The inconsistency of the knowledge means that not all of the aggregated objects can be formed.

As it was shown in the last section, the aim of the second step of the algorithm (computing maximal consistent sub-bases) is to select the best set of objects which will be added to the base of facts of level 1. This task corresponds to compute the best maximal sub-base of the inconsistent knowledge base  $\Sigma$ . Using the  $\pi$ -ATMS, we can obtain four possibilities

$$\Sigma_1 = \{C_1(t_1)\}; \Sigma_2 = \{C_1(t_2), B_1(t_2)\}; \Sigma_3 = \{C_1(t_3), B_1(t_1)\}; \Sigma_4 = \{C_1(t_4)\}.$$

Each sub-base represents a maximal set of complex objects which are made of distinct sub-objects.

Finally from the ordering " $\prec_{\mathcal{L}}$ " defined above, only the object  $C_1(t_1)$  is remained, and will be added to the base of facts of level 1. See [2] and Monai et al. [34] for a complete example.

This abstract example suggests the application of this technique to the reconstruction of scenarii from partial information.

## 7. Possibilistic A.T.M.S. vs. Other Existing "Uncertainty-Handling" A.T.M.S.

De Kleer and Williams [11], in a multiple faults diagnosis problem, compute the probability of a candidate (i.e. a configuration of faults) given the outcomes of the system and assuming that faults are mutually independent. Falkenhainer [22] has incorporated Dempster-Shafer theory into Doyle's T.M.S. [13]. Provan [27], D'Ambrosio [8], Laskey

and Lehner [25] have independently incorporated belief functions into the ATMS; their approaches are somewhat different but share the same basic features: first, a mass function is associated with each assumption; then, for each datum  $x$  we compute the belief  $Bel(x)$  of  $x$  by first computing the label of  $x$  symbolically (which can be done in the classical way), then we compute  $Bel(x)$  from  $Label(x)$ ; Provan computes  $Bel(x)$  by expressing logically  $Label(x)$  under the form of a formula containing only independent sub-formulas, assuming mutual independence of assumptions. D'Ambrosio's approach is based on Baldwin's Support Logic Programming method [1] and does not use the full Dempster-Shafer theory; besides, his way of propagating uncertainty is closer to a rule-based system than to a pure logical system (as he uses conditional beliefs and not beliefs of logical implications). Laskey-Lehner's and Provan's approaches both incorporate the full Dempster-Shafer theory into the ATMS, and differ essentially by their algorithm for the computation of  $Bel(x)$ . See also Pearl [26] whose proposal is close to Provan's (without normalizing Dempster rule). Bigham [4] incorporates Dempster-Shafer theory into the ATMS in order to compute belief functions. Lastly, Bernasconi et al. [3] have outlined the definition of an ATMS based on fuzzy logic in Lee and Mukaidono's style.

Our approach differs from the preceding ones on several points:

First, the ATMS with embedded belief functions enable us to rank interpretations (i.e. maximal consistent environments), as they intend to do, but they cannot handle the uncertainty pervading the knowledge base if any, i.e. the justifications. They hold the justifications as classical knowledge (i.e. certain) and, putting weights on the assumptions, they deduce the weights attached to the other data. Our approach does not attach any a priori weight to each datum (assumptions or non-assumptions) but assigns a certainty degree to the clauses contained in the knowledge base (not only justifications, but also disjunctions of assumptions and nogoods specified by the user); then, taking into account this uncertain knowledge, we establish a mapping from certainty degrees assigned to assumptions to the certainty degrees of the data (i.e. the consequences of the assignment).

Secondly, in the other approaches, truth maintenance and uncertainty management are completely separated, i.e. the label of a datum  $x$  is first computed, then  $Bel(x)$  is computed (of course, taking into account  $Label(x)$ ). Our approach completely integrates the uncertainty management into the truth maintenance system, by assigning a weight to each environment of a datum. Besides, the handling of weights will be done during the computation of labels, since certainty weights are present inside the label.

Thirdly, the main purpose of introducing uncertainty into A.T.M.S. is the capability of ranking solutions (i.e. interpretations) and eliminating solutions which are too uncertain. Hence the precise values of certainty degrees are not as important as their ordering. Possibility theory offers a framework where the ordering of the uncertainty degrees is more important than their precise values, and which requires easier computations (only min and max operations are used) than the general Dempster-Shafer approach. Furthermore, the weights assigned to nogoods are always available and no extra computations are required in order to renormalize results.

Fourthly, in Provan's model, the belief of the datum  $x$  is first computed formally by rewriting the label of  $x$  into an "independent form" and then computed numerically; if we decide to change the weight of an assumption, the formal expression of the label of  $x$  remains unchanged and the re-computation of  $\text{Bel}(x)$  is immediate; however this is not the case if the label of  $x$  changes (which may happen each time we add a new justification): then we have to transform again the expression of the label of  $x$ ; in the possibilistic model, both changes of weights or labels lead to almost no more computations than in De Kleer's original model.

Lastly, the other models cannot handle disjunctions of assumptions; besides, Provan's model needs to consider the assumptions as independent as long as they are not mutually exclusive (i.e. not containing any nogood). Possibilistic ATMS do not require this assumption, and can handle disjunctions of assumptions, as well as negated assumptions.

## **8. Conclusion**

Possibilistic ATMS enable a joint handling of assumptions and uncertainty relative to a knowledge base, in the framework of possibilistic logic. The fact that possibilistic logic remains on many points close to classical logic facilitates the extension of efficient procedures, such as those based on Oriented CAT-correct resolution for the computation of labels, nogoods and contexts, by ordering the environments of a datum according to the certainty with which it can be deduced from each of them.

The truth maintenance system based on the  $\pi$ -RCO resolution seems to have a number of advantages over the classical ATMS. Firstly, it allows the use of general clauses without requiring the creation of new data, justifications and disjunctions of assumptions as in [10]. The second advantage is that all justifications of the knowledge base are stored under the form of general weighted clauses, while in the classical ATMS, the justifications are stored in four different areas depending on the clause type: Horn clauses, disjunctions of assumptions, nogoods and environments. Finally, we have seen that negated assumptions are naturally taken in account, which is not possible in the classical ATMS, and they may appear in environments, nogoods and labels.

Possibilistic ATMS can be applied to truth maintenance problems in presence of uncertainty, to default reasoning, to fault diagnosis, or to the generation of (the most plausible) explanations.

## **Acknowledgements**

This work has been supported by the European ESPRIT Basic Research Action number 3085 entitled Defeasible Reasoning and Uncertainty Management Systems (DRUMS), and by a grant from Thomson CSF, Colombes, France.

## Annex

Let  $\Sigma = B_1 \cup B_2 \cup \dots \cup B_n$  be a possibilistic knowledge base where  $B_j$  contains clauses of the form  $(a_{ij}, \alpha_j)$ , and  $\alpha_1 > \alpha_2 > \dots > \alpha_n$ . Then  $\Sigma' \subseteq \Sigma$  is a strongly maximal consistent sub-base of  $\Sigma$  if and only if  $\Sigma' = E_1 \cup E_2 \cup \dots \cup E_n$  where  $\forall j = 1, n$ ,  $E_1 \cup E_2 \cup \dots \cup E_j$  is a maximal consistent sub-base of  $B_1 \cup B_2 \cup \dots \cup B_j$ .

### Proof

Assume  $\Sigma'$  is not strongly maximal consistent. Then there is an  $\alpha_j$  and a clause  $(a_{ij}, \alpha_j)$  such that the level of inconsistency of  $\Sigma' \cup \{(a_{ij}, \alpha_j)\}$  is smaller than  $\alpha_j$ , say  $\text{Inc}(\Sigma' \cup \{(a_{ij}, \alpha_j)\}) = \alpha_k < \alpha_j$ . This implies that  $E_1 \cup E_2 \cup \dots \cup E_j \cup \{(a_{ij}, \alpha_j)\}$  is consistent. Indeed the inconsistency of  $\Sigma' \cup \{(a_{ij}, \alpha_j)\}$  can disappear by taking away clauses in  $E_k \cup E_{k+1} \cup \dots \cup E_n$ . Hence  $E_1 \cup E_2 \cup \dots \cup E_j$  is not maximal consistent. Conversely if  $\Sigma'$  is maximal consistent but  $E_1 \cup \dots \cup E_j$  is not maximal consistent in  $B_1 \cup \dots \cup B_j$ , then there is  $(a_{ij}, \alpha_j)$  such that  $E_1 \cup E_2 \cup \dots \cup E_j \cup \{(a_{ij}, \alpha_j)\}$  is consistent; then  $\Sigma' \cup \{(a_{ij}, \alpha_j)\}$  is inconsistent with a level of inconsistency that is smaller than  $\alpha_j$ , since it involves some clause in  $E_{j+1} \cup \dots \cup E_n$ . Q.E.D.

## References

1. J.F. Baldwin, Evidential support logic programming, *Fuzzy Sets and Systems* **24** (1987) 1-26.
2. S. Benferhat, Raisonement hypothétique en logique possibiliste. D.E.A report, University of Paul-Sabatier, Toulouse (France), July 1991.
3. C. Bernasconi, S. Rivoira and S. Termini, On the notion of uncertain belief revision systems, (Extended Abstracts) *Proc. of the 3rd Inter. Conf. on Information Processing and Management of Uncertainty in Knowledge-based Systems*, Paris, 1990, 458-460.
4. J. Bigham, Computing numerical beliefs using propositional inference as a basis, (Extended Abstracts) *Proc. of the 3rd Inter. Conf. on Information Processing and Management of Uncertainty in Knowledge-based Systems*, Paris, 1990, 59-61
5. M. Cayrol and P. Tayrac, Exploitation de la méthode du consensus dans les ATMS: la résolution CAH-correcte. *Actes des 8èmes Journées Inter. sur les Systèmes Experts et leurs Applications*, Avignon, 1988, published by EC2, Nanterre, France, 85-99.
6. M. Cayrol and P. Tayrac, Les résolutions CAT-correcte et CCT-correcte, la résolution CAT-correcte dans l'ATMS. *Colloque Inter. sur l'Informatique Cognitive des Organisations (ICO'89)*, Québec, Canada, 1989.
7. P. Tayrac, ARC: an extended ATMS based on directed CAT-correct resolution. in *Truth Maintenance Systems* (Proc. of the ECAI-90 Workshop, Stockholm, Sweden, Aug. 1990), eds. J.P. Martins and M. Reinfrank (Springer Verlag, Berlin, 1991) 107-124.
8. B.D. D'Ambrosio, A hybrid approach to reasoning under uncertainty, in *Uncertainty in Artificial Intelligence 3*, eds. L.N. Kanal, T.S. Levitt and J.F. Lemmer (North-Holland, Amsterdam, 1989) 267-283.
9. J. De Kleer, An assumption-based TMS, *Artificial Intelligence* **28** (1986) 127-162.
10. J. De Kleer, Extending the ATMS, *Artificial Intelligence* **28** (1986) 163-196.
11. J. De Kleer and B. Williams, Diagnosing multiple faults, *Artificial Intelligence* **32** (1987) 97-130.

12. J. De Kleer, A general labeling algorithm for assumption-based truth maintenance, *Proc. of the Nat. Conf. on Artificial Intelligence*, Saint Paul, Minnesota, Aug. 21-26, 1988, 188-192.
13. J. Doyle, A truth maintenance system, *Artificial Intelligence* **12** (1979) 231-272.
14. D. Dubois, J. Lang and H. Prade, Theorem proving under uncertainty - A possibility theory-based approach, *Proc. of the 10th Inter. Joint Conf. on Artificial Intelligence*, Milano, August, 1987, 984-986.
15. D. Dubois and H. Prade, Necessity measures and the resolution principle, *IEEE Trans. Systems, Man and Cybernetics* **17** (1987) 474-478.
16. D. Dubois and H. Prade (with the collaboration of H. Farreny, R. Martin- Clouaire and C. Testemale) *Possibility Theory – An Approach to Computerized Processing of Uncertainty* (Plenum Press, New York, 1988).
17. D. Dubois, J. Lang and H. Prade, Advances in automated reasoning using possibilistic logic, in *Fuzzy Expert Systems*, eds. A. Kandel (CRC Press, Boca Raton, Fl., 1992) 125-134.
18. D. Dubois and H. Prade, Resolution principles in possibilistic logic, *Int. J. of Approximate Reasoning* **4(1)** (1990) 1-21
19. D. Dubois, J. Lang and H. Prade, A possibilistic assumption-based truth maintenance system with uncertain justifications, and its application to belief revision, in *Truth Maintenance Systems* (Proc. of the ECAI-90 Workshop, Stockholm, Sweden, Aug. 1990), eds. J.P. Martins and M. Reinfrank (Springer Verlag, Berlin, 1991) 87-106.
20. D. Dubois, J. Lang and H. Prade, Inconsistency in possibilistic knowledge bases –to live or not live with it–, in *Fuzzy Logic for the Management of Uncertainty*, eds. L.A.Zadeh and J. Kasprczyk (Wiley & Sons, 1991) to appear.
21. D. Dubois, J. Lang and H. Prade, Possibilistic logic, in *Handbook of Logic in Artificial Intelligence and Logic Programming, Vol. 2*, ed. D.M. Gabbay (Oxford University Press, 1992) to appear.
22. B. Falkenhainer, Towards a general-purpose belief maintenance system, in *Uncertainty in Artificial Intelligence 2*, eds. J.F. Lemmer and L.N. Kanal (North-Holland, Amsterdam, 1988) 125-132.
23. P. Gärdenfors, *Knowledge in Flux- Modeling the Dynamic of Epistemic States* (The MIT press, Cambridge, Mass., 1988).
24. J. Lang, Logique possibiliste: aspects formels, déduction automatique, et applications, PhD thesis, University of Toulouse, January 1991.
25. K.B. Laskey and P.E. Lehner, Belief maintenance; an integrated approach to uncertainty management, *Proc. of the 7th Nat. Conf. of Amer. for Artificial Intelligence*, Saint Paul, Minnesota, Aug. 21-26, 1988, 210-214.
26. J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference* (Morgan Kaufmann Publ. Inc., San Mateo, Ca., 1988).
27. G.M. Provan, An analysis of ATMS-based techniques for computing Dempster-Shafer belief functions, *Proc of the 9th Joint Conf. on Artificial Intelligence*, Detroit, Aug. 20-25, 1989, 1115-1120.
28. O. Raiman, Two heuristic integrating probabilities and logic: a preliminary report on parsimonious search, *Proc. of the Inter. Conf. on Computational Intelligence*, Milano, Italy, Sept 25-29, 1989.
29. P. Siegel, Représentation et utilisation de la connaissance en calcul propositionnel, Thèse d'Etat, July 1987.
30. B. Smith and G. Kelleher (Eds), *Reason Maintenance Systems and Their Applications* (Ellis Horwood Ltd, Chichester, 1988).
31. C. Testemale, Un modèle logique général pour le raisonnement révisable, Tech. Report n° IRIT/91-69/R, IRIT, Univ. P. Sabatier, Toulouse, France, 1991.
32. L.A. Zadeh, Fuzzy sets as a basis for a theory of possibility, *Fuzzy Sets and Systems* **1** (1978) 3-28.
33. L.A. Zadeh, A theory of approximate reasoning, in *Machine Intelligence*, Vol. 9, eds. J.E. Hayes, D. Michie and L.I. Mikulich (Elsevier, New York, 1979) 149-194.

34. F.F. Monai, B. Pochez and T. Chéhiré, Logique possibiliste et gestion d'hypothèses multiples: application à un problème de fusion de données, *Proc. of the 12th Inter. Conf. on Artificial Intelligence, Expert Systems and Natural Language (Avignon'92)*, Avignon, France, June 1-6, 1992, EC2, Nanterre, France, Vol. 1 : Conférence Scientifique, 209-220.
35. G. Brewka, Preferred subtheories: an extended logical framework for default reasoning, *Proc. of the Inter. Joint Conf. on Artificial Intelligence (IJCAI'89)*, Detroit, 1989, 1043-1048.