Describing evolving systems by uncertain default transition rules

Florence Dupin de Saint-Cyr
LERIA,
Angers, France
Email: florence.bannay@univ-angers.fr

Henri Prade
IRIT
Toulouse, France
Email: prade@irit.fr

***Abstract*— Representing a dynamic system demands a formalism which can both handle uncertainty and default information. Uncertainty is present because of the insufficient knowledge about the evolving system: the state of the system and the transition rules to which it obeys are ill known. Moreover, even if the knowledge about the system was complete, it would be impossible to express it exhaustively. The use of default rules can help solving this problem. Default rules are also useful for reasoning with incomplete information. We propose a formalism in which it is possible to express both default and uncertain transition rules. Two methods for syntactically computing the new state of the system after a transition are presented. Their principle is based on the translation of a base of default uncertain transition rules into a non-defeasible uncertain transition rules base. The possibility theory is used for representing both uncertainty and defeasibility.**

I. Introduction

Representing a dynamic system in artificial intelligence research has been a challenging problem for a long time [13]. The classical action representation problems, namely the "frame", "qualification" and "ramification" problems are well known now. Indeed, describing an action only by its preconditions and its effects is not enough to allow to deduce the state of the world after the action. An automatic system able of reasoning about change should know which fluents are persisting and which fluents are changing when the action takes place. The impossibility to enumerate every fluent which is not changed by an action is called "the frame problem". The difficulty to define exactly all the preconditions of an action is called "the qualification problem". "Ramification", the third action representation problem, is related to the difficulty of describing every direct or indirect effect of an action, and can be handled by representing static dependencies.

Many proposals have been made to solve the qualification and frame problems: see for instance [10]. A common idea is to use default comportment descriptions for expressing persistence. The expression of default transitions may be also useful for coping with the qualification problem. Default reasoning allows to use general rules in which exceptions are not mentioned. The benefits of a simplified presentation has a counterpart: exceptions can contradict general rules. A way to avoid this contradiction is to consider that more specific rules have priority upon less specific ones.

Besides, the available knowledge about the way a real system under study can evolve may be incomplete. This is why uncertainty should also be represented, at least in a qualitative way. Notice that the idea of uncertainty level should be distinguished from the notion of specificity level: a rule may have many possible kinds of exceptions but be quite sure.

In this paper we propose a representation which can manage both default transition rules and transitions with an uncertain conclusion, taking advantage of the possibility theory framework.

II. Background on possibility theory

We assume a representation language $\mathscr{L}$ made of a set of propositional variables $\mathscr{V}$. Note that the variables set may contain occurrences of action. More formally, let $\mathscr{A}$ be the set of action symbols. We consider that the variables set $\mathscr{V}$ contains in addition to the symbol representing facts all the symbols $do(a)$ where $a \in \mathscr{A}$, representing action occurrences. When there is ambiguity, variables may be indexed by a number representing the time point in which it is considered. We denote by $\varphi[t]$ the formula φ in which all variables are indexed by time point t. The set of interpretations associated to this language is denoted by Ω. An interpretation $\omega \in \Omega$ represents a state of the system under study. A possibility distribution is a mapping from Ω to a numerical scale such as the real interval $[0, 1]$. However a finite linearly ordered scale could be used as well. This possibility distribution rank-orders the interpretations according to their plausibility level. In particular, $\pi(\omega) = 0$ means that the interpretation is impossible; π is said normalized if $\exists\omega$ such as $\pi(\omega) = 1$; there may exist several distinct interpretations with a possibility degree equal to 1; they correspond to the most plausible interpretations. Given a possibility distribution π and a formula φ, two measures are defined over the set of models of φ:

- $\Pi(\varphi) = \max_{\omega \models \varphi} \pi(\omega)$ called the possibility of φ. It measures how unsurprising the formula φ is for the agent. $\Pi(\varphi) = 0$ means that φ is bound to be false.
- $N(\varphi) = 1 - \Pi(\neg\varphi) = 1 - \max_{\omega \not\models \varphi} \pi(\omega)$ called the necessity of φ. It corresponds to the impossibility of "not φ". $N(\varphi) = 1$ means that φ is bound to be true.

Definition 1 (certainty level): A possibilistic knowledge base is a set $K = \{(\varphi_i, a_i), i = 1 \ldots n\}$. Where φ_i is a propositional formula of $\mathscr{L}$ and its *certainty* level a_i is such that $N(\varphi_i) \geq a_i$, N being a necessity measure.

In possibilistic logic, the models of the knowledge base K are rank-ordered by a possibility distribution π_K. This π_K is the greatest solution of the set of constraints $N_K(\varphi_i) \geq a_i, i = 1 \ldots n$. It has been shown in [7] that $\pi_K(\omega)$ the possibility of a state of the world ω is all the smaller as the model ω falsifies a formula in K with a high certainty level.

The semantic entailment of a weighted formula (φ, a) from a possibilistic knowledge base K is defined by:

$K \models (\varphi, a) \Leftrightarrow \forall\omega \not\models \varphi, \pi_K(\omega) \leq 1 - a$ or, equivalently, $K \models (\varphi, a) \Leftrightarrow N_K(\varphi) \geq a$, where N_K is associated with π_K.

Given a possibilistic knowledge base K, K is consistent if and only if its associated possibility distribution π_K is normalized (since there is at least one interpretation in agreement with Σ which is completely possible). More generally, the degree of inconsistency of Σ [6] is defined by:

$$Inc(K) = 1 - max_{\omega \in \Omega} \pi_K(\omega)$$

When K is partially inconsistent ($Inc(K) > 0$) the semantic entailment can be refined (into $\models_\pi$) by considering as legitimately entailed only those propositions to which a degree of certainty, strictly higher than the level of inconsistency, can be attached, namely,

$$\exists a, \quad K \models_\pi (\varphi, a) \Leftrightarrow N_K(\varphi) > Inc(K)$$

If the weighted formulas of K are put under the form of weighted clauses (which is always feasible, since $N(\wedge_i \varphi_i) > a \Leftrightarrow \forall i, N(\varphi_i) > a$). The following resolution rule [8] is valid in possibilistic logic:

$$(\alpha \vee \beta, a); (\neg\alpha \vee \delta, b) \vdash (\beta \vee \delta, \min(a, b)) \quad (R)$$

The resolution rule can be used in order to compute the maximal certainty degree which can be attached to a formula according to the constraints expressed by the knowledge base K. This can be done by adding to K the clauses obtained by refuting the proposition to evaluate, with a necessity degree equal to 1. Then it can be shown that any lower bound obtained on $\perp$, by resolution, is a lower bound of the necessity of the proposition to evaluate. In case of partial inconsistency of K, a refutation carried out in a situation where $Inc(K \cup \{(\neg\varphi, 1)\}) = a > Inc(K)$ yields the nontrivial conclusion (φ, a), only using formulas whose degrees of certainty is strictly greater than the level of inconsistency of the base. This is the syntactic counterpart denoted $\vdash_\pi$ of the semantic entailment $\models_\pi$.

Example 1: We are going to consider the language built on the three variables $do(hm)_0$, W_0,W_1. These variables respectively represent the occurrence of the action "hit the coffee machine" at time 0, the fact that the coffee machine is working at time point 0, and at time point 1. Let us consider the following possibilistic knowledge base K_0 :

φ_1	$do(hm)_0 \wedge W_0 \to \neg W_1$	0.8
φ_2	$W_0 \to W_1$	0.6
φ_3	$do(hm)_0$	1
φ_4	W_0	0.7

The first formula means that if you hit the machine when it is working you may expect that it won't work in the next state with a necessity value of 0.8. The numbers here are just encoding a certainty ordering, and their nature in itself is not really meaningful. The second formula is a kind of persistence law, saying that if the machine is working at time point 0 then it will be working at time point 1 with a necessity value of 0.6 (sometimes it can become out of order for an unknown reason). The third formula expresses that the user has hit the machine (not necessarily on purpose, but he is sure that he unfortunately has done this action). The fourth one expresses that the machine is supposed to be working but the user is not absolutely sure about it, it has a necessity of 0.7.

This knowledge base K_0 induces the constraints:

$\forall\omega \models do(hm)_0 \wedge W_0 \wedge W_1,$	$\pi_{K_0}(\omega) \leq 0.2$
$\forall\omega \models W_0 \wedge \neg W_1$	$\pi_{K_0}(\omega) \leq 0.4$
$\forall\omega \models \neg do(hm)_0$	$\pi_{K_0}(\omega) \leq 0$
$\forall\omega \models \neg W_0$	$\pi_{K_0}(\omega) \leq 0.3$

The eight possible interpretations are $\omega_0 = (do(hm)_0 \ . \ W_0 \ . \ W_1)$, $\omega_1 = (do(hm)_0 \ . \ W_0$

$. \neg W_1)$, $\omega_2 = (do(hm)_0 \ . \ \neg W_0 \ . \ W_1)$, $\omega_3 = (do(hm)_0 \ . \ \neg W_0 \ . \ \neg W_1)$, $\omega_4 = (\neg do(hm)_0 \ . \ W_0 \ . \ W_1)$, $\omega_5 = (\neg do(hm)_0 \ . \ W_0 \ . \ \neg W_1)$, $\omega_6(\neg do(hm)_0 \ . \ \neg W_0 \ . \ W_1)$, $\omega_7 = (\neg do(hm)_0 \ . \ \neg W_0 \ . \ \neg W_1)$.

The greatest possibility distribution π_{K_0} satisfying the constraints is:

$$\begin{array}{ll} \pi_{K_0}(\omega_0) = 0.2 & \pi_{K_0}(\omega_4) = 0 \\ \pi_{K_0}(\omega_1) = 0.4 & \pi_{K_0}(\omega_5) = 0 \\ \pi_{K_0}(\omega_2) = 0.3 & \pi_{K_0}(\omega_6) = 0 \\ \pi_{K_0}(\omega_3) = 0.3 & \pi_{K_0}(\omega_7) = 0 \end{array}$$

There is no interpretation which is totally possible, so K_0 is partially inconsistent. $Inc(K_0) = 1 - 0.4 = 0.6$. $N_{K_0}(\neg W_1) = 1 - \max_{\omega \models W_1} \pi_{K_0}(\omega) = 0.7 > Inc(K_0)$. It means that $K_0 \models_\pi \neg W_1$, i.e., the coffee machine is likely to be out of order at time point 1.

$Inc(K_0)$ and this conclusion can also be computed syntactically using the possibilistic resolution rule on K_0 and on $K_0 \cup \{(W_1, 1)\}$ respectively. Moreover it would be enough to consider $K_0 \setminus \{\varphi_2\}$ instead of K_0 (since $N(\varphi_2) \leq Inc(K_0)$).

III. Handling specificity in default rules

The specificity level is a degree which is associated to a default rule. Let us recall the definition of default rules [15] encoded in the possibility theory framework [2] . A default rule is a formula $H \rightsquigarrow C$ where H and C are propositional formulas of $\mathscr{L}$ and $\rightsquigarrow$ is a new symbol. $\rightsquigarrow$ is translated by $\Pi(H \wedge C) > \Pi(H \wedge \neg C)$, a constraint which expresses that having C true is strictly more possible than having it false when H is true.

In [14] an algorithm based on a "minimal specificity" principle is given which allows to stratify the set of default rules in a way that reflects the specificity of the rules. Roughly speaking, the first stratum contains the most specific rules i.e, which do not admit exceptions (at least, expressible in the considered language), the second stratum have exceptions only in the first stratum and so on.

Definition 2 (System Z stratification [14]): A default transition rule $\alpha \rightsquigarrow \beta$ is tolerated by a set of default rules Δ if it exists an interpretation ω such that $\omega \models \alpha \wedge \beta$ (the rule $\alpha \rightsquigarrow \beta$ is said verified by ω) and $\forall di \in \Delta$, $\omega \models \neg\alpha_i \vee \beta_i$ (di is said satisfied by ω). This definition allows to stratify Δ into $(\Delta_0, \Delta_2, \ldots, \Delta_n)$ such that Δ_0 contains the set of rules tolerated by Δ, Δ_1 contains the set of rules tolerated by $\Delta \setminus \Delta_0$ and so on. The number $Z(r)$ corresponds to the number of the stratum in which the rule r is.

Definition 3 (possibilistic specificity degree): For each default rule $r = \alpha \rightsquigarrow \beta$ of a default base, it can be associated [2] a possibilistic formula $(\alpha \rightarrow \beta, a)$, where a is a number representing its specificity level $a = \frac{Z(r)+1}{n+2}$.

Example 2: The classical example given for instance in [2] illustrates these definitions:

d_1	$b \rightsquigarrow f$	(a bird generally flies),
d_2	$p \rightsquigarrow \neg f$	(a penguin generally does not fly),
d_3	$p \rightsquigarrow b$	(a penguin is generally a bird).

It can be easily verified that the rule d_1 is tolerated by d_2 and d_3, since the interpretation $(b.f.\neg p)$ verifies d_1 and satisfies both d_2 and d_3. This contrasts with the other rules for which it is not possible to find an interpretation which verifies d_2 (resp. d_3) and satisfies both d_1 and d_3 (resp. d_1 and d_2). So, we have $Z(d_1) = 0$ and $Z(d_2) = Z(d_3) = 1$. The possibilistic knowledge base K constructed from the initial default base is:

$$\begin{array}{llll} \{ & \varphi_1 & = & (b \rightarrow f, \frac{1}{3}), \\ & \varphi_2 & = & (p \rightarrow b, \frac{2}{3}), \\ & \varphi_3 & = & (p \rightarrow \neg v, \frac{2}{3})\} \end{array} .$$

What can we deduce from the fact b? This can be computed by adding $(b, 1)$ to K. We can draw the following derivations:

1. $(\neg p \vee b, \frac{2}{3}); (\neg b \vee f, \frac{1}{3}) \vdash (\neg p \vee f, \frac{1}{3})$ $(R\varphi_1, \varphi_2)$
2. $(\neg p \vee f, \frac{1}{3}); (\neg p \vee \neg f, \frac{2}{3}) \vdash (\neg p, \frac{1}{3})$ $(R1., \varphi_3)$
3. $(\neg p, \frac{1}{3}); (b, 1) \vdash (\bot, \frac{1}{3})$ $(R2., (b, 1))$

where R stands for a resolution application to the formulas explicitly indicated or to the result of the given line. Hence $Inc(K \cup \{(b, 1)\}) = \frac{1}{3}$. Now, we can prove by refutation that $\neg f$ is a logical consequence of $K \cup \{(p, 1)\}$. Since adding the piece of information $(f, 1)$ we can draw the following derivations:

4. $(\neg p \vee \neg f, \frac{2}{3}); (p, 1) \vdash (\neg f, \frac{2}{3})$ $(R\varphi_3, (b, 1))$
5. $(\neg f, \frac{2}{3}); (f, 1) \vdash (\bot, \frac{2}{3})$ $(R4., (f, 1))$

So $Inc(K \cup \{(b, 1), (f, 1)\}) = \frac{2}{3} > Inc(K \cup \{(b, 1)\})$, Which means that $K \cup \{(b, 1)\} \vdash_\pi (\neg f, 1)$.

Note that default rules may concern fluents at different time points. The knowledge base $M_0 \rightsquigarrow G_1, F_0 \rightsquigarrow M_0, F_0 \rightsquigarrow \neg G_1$ is another instance of this example. The first rule could be read "usually if there is money in the coffee machine at time point 0 then a goblet will be delivered on the tap at time point 1". The second rule could be "a faked coin is usually considered as money" and the third rule be "usually if there is a faked coin inserted in the coffee machine then at time point 1 a goblet will not be delivered." By translation of the previous example we can obtain that if we have a faked coin at time point 0 then the goblet will not be delivered at time point 1.

IV. Uncertain transition rules

If we consider an evolving system, we can represent its behavior like in [11] by the mean of:

- a possibilistic knowledge base $K = \{(\varphi_i, a_i), i = 1, n\}$, $(\forall i, \varphi_i \in \mathscr{L}, a_i \in [0, 1])$

representing what is known about the initial state of the system.

- a set $T = \{(H, C, a)\}$ $(H, C \in \mathscr{L}, a \in [0, 1])$ of uncertain transition rules.

Example 3: For instance, if we imagine a simple coffee machine which can be working (W) or be broken, have got enough money from the user (M), have a goblet under the tap (G) or be delivering coffee (C). The occurrence of the action "hit machine" can be described by this uncertain transition rules: $(do(hm) \wedge W, \neg W, 0.8)$ and $(do(hm) \wedge \neg W, W, 0.1)$. This rule means that if you hit the machine when it is working you may expect that it will not work in the next state with a necessity value of 0.8. This action is a conditional effect action (its effects depend on the initial state) [16] with uncertain result (it does not necessarily work).

From the set of uncertain transition rules it is possible to define a fuzzy relation Γ representing transition from a state ω to another state ω' of the system:

$$\mu_\Gamma(\omega, \omega') = 1 - \max_{(H,C,a)\in T | \omega \models H; \omega' \not\models C} a$$

with the convention that "max" taken over an empty set yields 0. The fuzzy upper image of a possibility distribution π by Γ gives a possibility distribution π' defined as follows:

$$\pi'(\omega') = \max_{\omega\in\Omega} \min\{\pi(\omega), \mu_\Gamma(\omega, \omega')\}$$

It has been shown in [11] that $\forall (H, C, b) \in T$, $N_{K'}(C) \geq \min\{N_K(H), b\}$. It gives a syntactic way to directly build the possibilistic knowledge base K' representing the next state:

$$K' = \{(C, a) | \exists H \text{s.t. } (H, C, b) \in T \wedge a = \min(b, N_K(H))\} \quad \text{(UTR)}$$

and it shows that the syntactic computation is consistent, i.e., each formula of K' is in the belief set associated with the distribution π' (made of the formulas φ such that $\exists a > 0, N_{\pi'}(\varphi) \geq a$).

Hence updating the base can be directly performed at the syntactic level from the transitions pairs and the possibilistic logic base. However as it is noticed in [11], this computation is not complete since there can be a loss of information in the process. For instance, if $T = \{(A, B, a); (C, D, b)\}$ and $K = \{(A \vee C, 1)\}$ then the syntactic computation gives an empty knowledge base (which corresponds to a constant distribution equal to 1 everywhere), whereas the upper image computation validates the formula $(B \vee D, \min(a, b, 1))$. This problem is solved by using the closure of T denoted T^* given by the following formula:

$$T^* = \{(A, B, a) | a = 1 - \max_{\omega\models A; \omega' \not\models B} \mu_{\Gamma(\omega,\omega')}$$

Since the computation of the closure T^* of T is not reasonable in practice, [11] proposes to eliminate the information that is not useful from T^* by defining the concept of informativeness. A transition rule $(H1, C1)$ is more informative than $(H2, C2)$ iff $H2 \models H1$ and $C1 \models C2$. This concept leads to define a transition set $T3$ which is a subset of the closure T^* of T and which is sufficient to keep all information of T^*.

$T3 = \{(\vee_I(\wedge_J H_i), \vee_I(\wedge_J C_i)) | \; \forall (H_i, C_i, a_i) \in T\}$ where I and J are any independent sets of indices of rules in T.

The new possibilistic knowledge base K' obtained from K after transition T is given by the equation (UTR) in which T is replaced by $T3$:

$$K' = \{(C, a) | \exists H \text{s.t. } (H, C, b) \in T3 \wedge a = \min(b, N_K(H))\}$$

From this result, an algorithm which allows to compute directly the possibilistic knowledge base corresponding to the following state of the system is given in [11]. An advantage of this algorithm is that it remains at the syntactical level.

V. Uncertain default transition rules

In order to have a very expressive representation formalism, we choose to represent the system under study by means of transition rules which can be defeasible and uncertain. The idea of describing pieces of evolution under the form of pairs possibly associated with complementary terms related to uncertainty or non-monotonicity can be found in several authors, e.g., [5].

In the present paper, the idea is to extend the results in [11] restated in the previous section to the case where transition rules are expressed as default rules, maybe pervaded with uncertainty.

Definition 4 (uncertain default transition rule): An uncertain default transition rule is denoted by $(H \rightsquigarrow C, \quad a)$ where H and C are propositional formulas of $\mathscr{L}$ and a is the certainty level of the rule. It means "by default" if H is true at time point t then C has a necessity degree of at least a to be true at time $t + 1$. (if a is not given it is considered to be equal to 1).

For instance, the description of the action "give money" $(do(gm) \rightsquigarrow M, 0.9)$ is an uncertain default transition rules since, for an unknown reason, the machine may not accept the money given. It is also a default transition since it admits known exceptions: for instance, $(do(gm) \wedge F \rightsquigarrow \neg M, 0.7)$ when the coin is faked (denoted by F).

In order to handle the frame problem, we need to define a frame axiom (this kind of axiom is defined by many authors see for instance in [10]).

Among all kinds of fluents we can distinguish persistent fluents (for which a change of value is surprising) from non persistent ones (which are also called dynamic [16]). Alternative fluents represent another type of fluents (which should change their value at each time point); alternative fluents are non persistent fluents, their behavior can easily be described by transition rules of the form $f \rightsquigarrow \neg f$ and $\neg f \rightsquigarrow f$. Here, we consider that a set of non persistent literals NP is defined. Note that occurrences of actions are clearly non persistent fluents: $\{do(a) | a \in \mathscr{A}\} \subseteq NP$.

Definition 5 (frame axiom): $\forall f \in \mathscr{V}$, if $f \notin NP$ then $f \rightsquigarrow f$ and if $\neg f \notin NP$ then $\neg f \rightsquigarrow \neg f$.

More generally it is possible to assign to each fluent its own persistence degree. For instance, in the Deaf Turkey Problem [5] $asleep$ is persistent but it is less persistent than $deaf$. It means that instead to be considered equal to 0 or 1 as in our example, our formalism allows to adapt the certainty of the default persistence rule to each fluent. We will see in section VI that our formalism can also encode decreasing persistence.

Given the description of an evolving system composed of a set of uncertain default transition rules T describing its behavior (T contains pure dynamic laws and default persistence rules (coming from the frame axiom)) and a possibilistic knowledge base K which describes the initial state of the world, the problem is to predict the next state K' of the world.

- A first idea is to used possibility theory in order to select which rules in T can be fired when the initial state is K. Then remove the rules which are not selected and then apply the algorithm in [11].
- Another idea is to translate the set of uncertain default transition rules T into a set of transition rules which are only uncertain and then also apply the algorithm in [11]. The rewriting of uncertain default transition rules into uncertain propositional transition rules can be done by explicitly naming the exceptions of each rule. In order to find exceptions, the stratification based on the specificity of the rules will be used.

In both cases, the process has two steps: first, forget the certainty degrees and reason on specificity in order to obtain a set of non-defeasible transition formulas. Second, the certainty degrees of the transitions are considered and the algorithm given in [11] can be used.

The following example inspired from [11] shows how to describe a coffee machine behavior with uncertain default transition rules.

Example 4: Let us consider again the coffee machine which can be working (W) or be broken, have got enough money from the user (M), have a goblet under the tap (G) or be delivering coffee (C). Then we can roughly describe its normal behavior:

$\varphi_1 : M \rightsquigarrow G \wedge \neg M$

$\varphi_2 : G \rightsquigarrow C$

The first rule means that if the machine has money in it then in the next step a goblet is under the tap and the money is spent. The second means that when there is a goblet under the tap then in the next step coffee is delivered. These first two rules describe the intended coffee machine comportment supposing it is working correctly. But they admit exceptions:

$\varphi_3 : M \wedge \neg W \rightsquigarrow \neg G$

$\varphi_4 : G \wedge \neg W \rightsquigarrow \neg C$

We can suppose that an agent is able to perform three actions on this machine : “give money” (gm), “take goblet” (tg), “hit machine” (hm). Here are the descriptions of the effects of the 3 actions.

$\varphi_5 : do(tg) \wedge G$	$\rightsquigarrow$	$\neg G$	
$\varphi_6 : do(gm)$	$\rightsquigarrow$	M	0.9
$\varphi_7 : do(gm) \wedge F$	$\rightsquigarrow$	$\neg M$	0.7
$\varphi_8 : do(hm) \wedge W$	$\rightsquigarrow$	$\neg W$	0.8
$\varphi_9 : do(hm) \wedge \neg W$	$\rightsquigarrow$	W	0.1

The first law means that taking a goblet is a *deterministic action* which leads to make disappear the goblet from the tap. The second law has an uncertain effect since giving money can failed if the coin is faked money. The third action has already been described in section IV.

We consider M as the only non persistent fluent (as soon as it is true, it becomes false because of the rule φ_1): $NP = \{M\}$. It means that we have the following default transition rules for representing persistence:

$\varphi_{10} : G \rightsquigarrow G$

$\varphi_{11} : C \rightsquigarrow C$

$\varphi_{12} : W \rightsquigarrow W$

$\varphi_{13} : F \rightsquigarrow F$

$\varphi_{14} : \neg M \rightsquigarrow \neg M$

$\varphi_{15} : \neg G \rightsquigarrow \neg G$

$\varphi_{16} : \neg C \rightsquigarrow \neg C$

$\varphi_{17} : \neg W \rightsquigarrow \neg W$

$\varphi_{18} : \neg F \rightsquigarrow \neg F$

In the initial state the agent is not absolutely sure that the coffee machine is working but he puts money in it. $K = (do(gm)_0, 1)$, $(\neg F_0, 0.99)$, $(\neg M_0, 0.9)$, $(W_0, 0.8)$, $(\neg C_0, 1)$, $(\neg G_0, 1)$, $(\neg do(x)_0, 1)$ if $x \neq gm$. A priori the user thinks that his coin is not faked, there is no money in the machine, no goblet, so no coffee in it and the user has not committed any other action.

A. Reasoning on the specificity degree

In this section, two levels are taken into account: a level linked to specificity and a level of uncertainty. Our proposal is to reason separately on the two scales, first use the specificity level to determine which rules are going to be fired and then deal with their uncertain results. We propose to use the system Z stratification principle [14] on the set of uncertain default rules (ignoring their certainty degrees). If no stratification is possible then it means that two rules of same specificity level are in contradiction. In the following, we consider that this is not the case, and thus Z stratification always succeed.

Definition 6 (specificity degrees principle): Let T be the set of default transition rules and K the possibilistic knowledge base representing what is known about the current state of the system.

- Compute the stratification of the default rules in T forgetting their necessity degrees, i.e., stratify $\Delta = \{\alpha[t] \rightsquigarrow \beta[t+1] | \exists a \text{ s.t. } (\alpha \rightsquigarrow \beta, a) \in T\}$. It gives a set $\Delta_0, \ldots, \Delta_n$.
- Associate to each default rule $r = \alpha \rightsquigarrow \beta \in \Delta$ its specificity degree $d(r) = \frac{Z(r)+1}{n+2}$. Let E be the possibilistic knowledge base s.t. $E = \{(r : \alpha[t] \rightarrow \beta[t+1], d(r)) | r : \alpha \rightsquigarrow \beta \in \Delta\}$.
- Remove each formula (φ_i, a_i) of E such that $a_i \leq Inc(E \cup K)$, the remaining transition rule base forgetting the specificity levels is: $T1 = \{(\alpha_i, \beta_i) | \exists a_i | (\alpha_i[t] \rightarrow \beta_i[t+1], a_i) \in E \text{ and } a_i > Inc(E \cup K)\}$

Example 5: System Z gives the following stratification (necessity degrees are reminded (in italics)):

$\varphi_1 : M \rightsquigarrow G \wedge \neg M$	*1*
$\varphi_2 : G \rightsquigarrow C$	*1*
$\varphi_{10} : G \rightsquigarrow G$	*1*
$\varphi_{11} : C \rightsquigarrow C$	*1*
$\varphi_{12} : W \rightsquigarrow W$	*1*
$\varphi_{13} : F \rightsquigarrow F$	*1*
$\varphi_{14} : \neg M \rightsquigarrow \neg M$	*1*
$\varphi_{15} : \neg G \rightsquigarrow \neg G$	*1*
$\varphi_{16} : \neg C \rightsquigarrow \neg C$	*1*
$\varphi_{17} : \neg W \rightsquigarrow \neg W$	*1*
$\varphi_{18} : \neg F \rightsquigarrow \neg F$	*1*
$\varphi_3 : M \wedge \neg W \rightsquigarrow \neg G$	*1*
$\varphi_4 : G \wedge \neg W \rightsquigarrow \neg C$	*1*
$\varphi_5 : do(tg) \wedge G \rightsquigarrow \neg G$	*1*
$\varphi_6 : do(gm) \rightsquigarrow M$	*0.9*
$\varphi_8 : do(hm) \wedge W \rightsquigarrow \neg W$	*0.8*
$\varphi_9 : do(hm) \wedge \neg W \rightsquigarrow W$	*0.1*
$\varphi_7 : do(gm) \wedge F \rightsquigarrow \neg M$	*0.7*

The third stratum is the more specific one, the rule in it does not admit exceptions. The second stratum contains rules which admit exceptions only because of rules in the third stratum. The first stratum contains rules which admit exceptions in the second stratum.

Here is the possibilistic knowledge base E associated to the transitions given in our example (the ignored necessity degrees are reminded (in italics)):

$\varphi_7 : do(gm)_0 \wedge F_0 \rightarrow \neg M_1$	0.75	*0.7*
$\varphi_3 : M_0 \wedge \neg W_0 \rightarrow \neg G_1$	0.5	*1*
$\varphi_4 : G_0 \wedge \neg W_0 \rightarrow \neg C_1$	0.5	*1*
$\varphi_5 : do(tg)_0 \wedge G_0 \rightarrow \neg G_1$	0.5	*1*
$\varphi_6 : do(gm)_0 \rightarrow M_1$	0.5	*0.9*
$\varphi_8 : do(hm)_0 \wedge W_0 \rightarrow \neg W_1$	0.5	*0.8*
$\varphi_9 : do(hm)_0 \wedge \neg W_0 \rightarrow W_1$	0.5	*0.1*
$\varphi_1 : M_0 \rightarrow G_1 \wedge \neg M_1$	0.25	*1*
$\varphi_2 : G_0 \rightarrow C_1$	0.25	*1*
$\varphi_{10} : G_0 \rightarrow G_1$	0.25	*1*
$\varphi_{11} : C_0 \rightarrow C_1$	0.25	*1*
$\varphi_{12} : W_0 \rightarrow W_1$	0.25	*1*
$\varphi_{13} : F_0 \rightarrow F_1$	0.25	*1*
$\varphi_{14} : \neg M_0 \rightarrow \neg M_1$	0.25	*1*
$\varphi_{15} : \neg G_0 \rightarrow \neg G_1$	0.25	*1*
$\varphi_{16} : \neg C_0 \rightarrow \neg C_1$	0.25	*1*
$\varphi_{17} : \neg W_0 \rightarrow \neg W_1$	0.25	*1*
$\varphi_{18} : \neg F_0 \rightarrow \neg F_1$	0.25	*1*

The user can check that $Inc(E \cup K) = 0.25$ since $E \cup K \vdash_\pi (M_1, 0.5)$ from rule φ_6 and from the fact $do(gm)_0$. And we have also $E \cup K \vdash_\pi (\neg M_1, 0.25)$ from rule φ_{14} and the fact $(\neg M_0, 0.9)$.

Hence the final uncertain transition base $T1$ is:
$\varphi_7 : (do(gm) \wedge F, \neg M, 0.7)$
$\varphi_3 : (M \wedge \neg W, \neg G, 1)$
$\varphi_4 : (G \wedge \neg W, \neg C, 1)$
$\varphi_5 : (do(tg) \wedge G, \neg G, 1)$
$\varphi_6 : (do(gm), M, 0.9)$
$\varphi_8 : (do(hm) \wedge W, \neg W, 0.8)$
$\varphi_9 : (do(hm) \wedge \neg W, W, 0.1)$

The above example shows a drawback of this method: all the persistence rules are drowned. Hence in the second step, we will not be able to determine the value of the fluents which are not concerned by transitions. A way to avoid this problem is for instance the use of the lexico ordering [1]. This ordering associates to each interpretation a tuple which values are the number of violated formulas in each stratum. The problem is that the lexico ordering is defined on interpretations, it means that we can not stay at the syntactical level. Another way to avoid the drowning effect could be to stratify separately the set of persistence rules. In [4], a local stratification is proposed in order to avoid the drowning effect. In the following section, we present another way to obtain a set of non-defeasible uncertain transition rules from the initial set of default uncertain transition rule.

B. Rewriting the rules by giving explicitly their exceptions

The idea, close to circumscription [12], is to generate automatically a set of rules in which exceptions are mentioned explicitly. Note that the rules of the last stratum being the most specific they do not admit exceptions. So the algorithm begin from the rules of the stratum $n-1$.

Definition 7 (rewriting principle):
Let T be the set of default transition rules.

- Compute the stratification of T forgetting necessity degrees, i.e., stratify $\Delta = \{\alpha[t] \rightsquigarrow \beta[t+1] | \exists a$ s.t. $(\alpha \rightsquigarrow \beta, a) \in T\}$. It gives a set $\Delta_0, \ldots, \Delta_n$.

{Let s be the number of the current stratum and D (resp. D_s) be the set of all transition rules of Δ (resp. of Δ_s) already rewritten}
Set $s = n-1$ and $D = \{(\alpha,\beta) | \alpha \rightsquigarrow \beta \in \Delta_n\}$ and $D_s = \varnothing$.
while Δ_s is not empty **do repeat**
  for each rule $r = \alpha \rightsquigarrow \beta \in \Delta_s$ **do**:
    remove r from Δ_s {r is being examined }
    if it exists a transition rule $r0 = (\alpha', \beta')$
      in D such that $\beta' \wedge \beta \vdash \bot$
    then add to Δ_s the new default rule
      $\alpha \wedge \neg\alpha' \rightsquigarrow \beta$
    else add the non-defeasible transition
      rule (α, β) to D_s
if $s \neq 0$ **then** add D_s to D; $s := s-1$ {examine the previous stratum }
else end.

- $T1$ is the uncertain transition rules base corresponding to D in which necessity degrees appear. T to the corresponding formulas of D

Proposition 1: If the Z stratification is possible then this algorithm terminates.

Proof: The algorithm examines each rule of each stratum. For a rule of a stratum Δ_s, the algorithm executes at most two consistency tests with each rule of the stratum $s+1$. Since each stratum is finite, the algorithm terminates. ■

Definition 8: A transition rules base $\{(\alpha_i, \beta_i)\}$ is said consistent if the propositional formula $\wedge(\alpha_i[0] \rightarrow \beta_i[1])$ is consistent.

Proposition 2: This algorithm gives a consistent transition rule base.

Proof: At the beginning $D = \{(\alpha_i, \beta_i)\}$ is consistent since it is built on the set Δ_n of rules tolerated by the set $\Delta \setminus (\Delta_0 \cup \ldots \Delta_{n-1}) = \Delta_n$. Hence, it exists ω_0 verifying the first rule of Δ_n and satisfying every other rules of Δ_n. It means that $\omega_0 \models \alpha_{n1} \wedge \beta_{n1} \wedge_{\alpha_{ni} \rightsquigarrow \beta_{ni} \in \Delta_n} (\neg\alpha_{ni} \vee \beta_{ni})$. So at the beginning D is consistent. At each step, a rule is added to the result D only if its conclusion is consistent with every conclusion of a rule of D. For a rule $r = \alpha \rightsquigarrow \beta$ from a stratum Δ_s. if it exists a rule $r' = (\alpha', \beta')$ in D such that $\beta' \wedge \beta \vdash \bot$, then r is replaced by $\alpha \wedge \neg\alpha' \rightsquigarrow \beta$. Note that $\alpha \wedge \neg\alpha'$ is consistent since, by construction, every rule of Δ_{s+1} is tolerated by r, it means that it exists $\omega \models \alpha \wedge \beta \wedge (\neg\alpha' \vee \beta')$, i.e, $\omega \models \alpha \wedge \neg\alpha' \wedge \beta$. r modified by specifying all its exception is added to D only when there is no more rule in Δ_{s+1} which conclusion is inconsistent with β.
So D remains consistent. ■

Note that each rule of the initial default transition base is present, modified or not, in the resulting transition rules base. So, there is no loss of information as with the previous method. However, for a one-step reasoning some rules may not be useful, because not applicable in the initial step. But it is interesting to keep them for a several-steps reasoning.

Example 6: Now we can rewrite the rule by describing explicitly their exceptions starting from the last stratum. It gives the following uncertain transition set $T1$:

$\varphi_7 : (do(gm) \wedge F, \neg M, 0.7)$
$\varphi_3 : (M \wedge \neg W, \neg G, 1)$
$\varphi_4 : (G \wedge \neg W, \neg C, 1)$
$\varphi_5 : (do(tg) \wedge G, \neg G, 1)$
$\varphi_6 : (do(gm) \wedge \neg\mathbf{F}, M, 0.9)$
$\varphi_8 : (do(hm) \wedge W, \neg W, 0.8)$
$\varphi_9 : (do(hm) \wedge \neg W, W, 0.1)$
$\varphi_1 : (M \wedge \mathbf{W} \wedge \neg\mathbf{do(tg)} \wedge \neg\mathbf{do(gm)}, G \wedge \neg M, 1)$
$\varphi_2 : (G \wedge \mathbf{W}, C, 1)$
$\varphi_{10} : (G \wedge (\neg\mathbf{M} \vee \mathbf{W}) \wedge (\neg\mathbf{do(tg)}), G, 1)$
$\varphi_{11} : (C \wedge (\neg\mathbf{G} \vee \mathbf{W}), C, 1)$
$\varphi_{12} : (W \wedge \neg\mathbf{do(hm)}, W, 1)$
$\varphi_{13} : (F, F, 1)$
$\varphi_{14} : (\neg M \wedge (\neg\mathbf{do(gm)} \vee \mathbf{F}), \neg M, 1)$
$\varphi_{15} : (\neg G, \neg G, 1)$
$\varphi_{16} : (\neg C, \neg C, 1)$
$\varphi_{17} : (\neg W \wedge \neg\mathbf{do(hm)}, \neg W, 1)$
$\varphi_{18} : (\neg F, \neg F, 1)$

Notice that persistence laws exceptions are actions occurrences which seems very natural.

C. Second step: reasoning with an uncertain transition base

In this step we start from the uncertain transition base $T1$ obtained after the first step. We then compute
$K' = \{(C,a) | \exists H$s.t. $(H,C,b) \in T3 \wedge a = \min(b, N_K(H))\}$ where $T3 = \{(\vee_I(\wedge_J H_i), \vee_I(\wedge_J C_i)) |$ for all $(H_i, C_i, a_i) \in$

$T1\}$ where I and J are any independent sets of indices of rules in $T1$.

Example 7: In our example, $N_K(do(gm) \wedge \neg F) = 0.99$ so K' contains $(M, 0.9)$ where 0.9 is the min of $N_K(do(gm) \wedge \neg F) = 0.99$ and of the degree of the transition $\varphi_6 : (do(gm) \wedge \neg F, M, 0.9)$ of $T1$. $(W, 0.8)$ can be deduced by rule φ_{17} : $W \wedge \neg do(hm), W, 1)$ and $N_K(W_0 \wedge \neg do(hm)) = 0.8$. The user can check that $\neg C$ and $\neg G$ persist: $N(\neg C_1) = 1$ and $N(\neg G_1) = 1$. All this computation can be done again for time point 2, if no action is done at step 1, it will give a possibilistic knowledge base K'' in which the goblet is on the tap with a necessity of 0.8.

VI. Application to fuzzy default rules

In the preceding section, the necessity degrees associated to rules are levels which are fixed by the user. An interesting application is to use variable weights. Let us first remind that a possibilistic formula of the form $(\alpha \wedge \beta \rightarrow \gamma, a)$ is semantically equivalent to $(\alpha \rightarrow \gamma, min(a, v(\beta))$ where $v(\beta) = 0$ if β is false and $v(\beta) = 1$ if β is true [7]. It expresses that $\alpha \rightarrow \gamma$ is a-certain given the proviso that β is true. More generally, β can be a vague predicate and $v(\beta)$ can take its value in the $[0, 1]$ interval, in this case $v(\beta)$ corresponds to the membership function of β. This result allows us to handle fuzzy default rules [3] of the form "the more β, the more it is certain that α implies γ is true". For instance, "the smaller a bird is, the more certain it flies". This kind of rule can be encoded by $(b \rightsquigarrow f, \mu_s)$ where μ_s is the membership function of small. Note that this certainty degree μ_s should not be confused with the specificity level of the rule. In our example, we can imagine a rule of this kind, for instance: "the more strongly you hit the coffee machine, the more it is certain that it will not work in the next state" encoded by $(do(hm) \wedge W \rightsquigarrow \neg W, \mu_{\text{strongly}})$.

The possibility to affect variable degrees to a rule may be also very useful in representing dynamic systems. Since they can express decreasing persistence [9]: the more the time has passed the less it is certain that a fluent has kept its value. A decreasing persistence rule is generally of the form $(M_t \rightsquigarrow M_{t+n}, f(M, n))$ where the degree attached to the rule is function of the fluent quality (highly persistent or dynamic) and of the length of the time interval.

VII. Conclusion

We propose a representation language which allows to handle transition rules which are both uncertain and by default. This tool is useful in the context of dynamic systems since it helps solving the "frame" and "qualification" problems, thanks to default rules.

References

[1] S. Benferhat, C. Cayrol, D. Dubois, J. Lang, and H. Prade. Inconsistency management and prioritized syntax- based entailment. In Ruzena Bajcsy, editor, *Proc. of the 13th IJCAI*, pages 640–645, Chambery, France, 1993. Morgan-Kaufmann.

[2] S. Benferhat, D. Dubois, and H. Prade. Representing default rules in possibilistic logic. In W. Swartout B. Nebel, C. Rich, editor, *Proc. of the 3rd KR*, pages 673–684, Cambridge, MA, October 1992.

[3] S. Benferhat, D. Dubois, and H. Prade. Towards fuzzy default reasoning. In T. Sudkamp R. Davé, editor, *18th International Conference of the North American Fuzzy Information Processing Society (NAFIPS)*, pages 23–27, New York, USA, June 1999.

[4] S. Benferhat and L. Garcia. Handling locally stratified inconsistent knowledge bases. *Studia Logica Journal*, 70:77–104, 2002.

[5] M.O. Cordier and P. Siegel. Prioritized transitions for updates. In *Lecture Notes in Artificial Intelligence 946 (Proc. of ECSQARU-95)*, pages 142–150, Fribourg, Switzerland, 1995.

[6] D. Dubois, J. Lang, and H. Prade. Handling uncertainty, context, vague predicates, and partial inconsistency in possibilistic logic. In *Fuzzy logic and fuzzy control*, pages 45–55. Springer-Verlag, 1991.

[7] D. Dubois, J. Lang, and H. Prade. Possibilistic logic. In D.M. Gabbay, C.J. Hogger, and J.A. Robinson, editors, *Handbook of logic in Artificial Intelligence and logic programming*, volume 3, pages 439–513. Clarendon Press - Oxford, 1994.

[8] D. Dubois and H. Prade. Necessity measures and the resolution principle. In *IEEE Trans. on Systems, man and cybernetics*, volume 17, pages 474–478, 1987.

[9] F. Dupin de Saint-Cyr and J. Lang. Reasoning about unpredicted change and explicit time. In *Lecture Notes in Artificial Intelligence (Proc. of ECSQARU-97)*, 1997.

[10] F. Lin J. Lang and P. Marquis. Causal theories of action: A computational core. In *Proceedings of 18th International Joint Conference on Artificial Intelligence (IJCAI'03)*, pages 1073–1078, Acapulco, 2003.

[11] B. Mailhé and H. Prade. Updating of a possibilistic knowledge base by crisp or fuzzy transition rules. In *Proc. of the 9th KR*, pages 338–346, Whistler, BC (Canada), 2004.

[12] J. McCarthy. Applications of circumscription to formalizing common-sense knowledge. *Artificial Intelligence*, 28(1):1038–1044, 1986.

[13] J. McCarthy and P.J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. In B. Meltzer and D. Mitchie, editors, *Machine Intelligence*, volume 4, pages 463–502. Edinburgh University Press, 1969.

[14] J. Pearl. System Z : a natural ordering of defaults with tractable applications for default reasoning. In M. Vardi, editor, *Proceedings of theoretical aspects of reasoning about knowledge*, pages 121–135, San Mateo, 1990. Morgan Kaufman.

[15] R. Reiter. A logic for default reasoning. *Artificial Intelligence*, 13(1-2):81–132, 1980.

[16] E. Sandewall. *Features and Fluents*. Oxford University Press, 1995.