

Un Mecanisme de prediction deductive pour le raisonnement temporel

Nicolas Chleq

► **To cite this version:**

Nicolas Chleq. Un Mecanisme de prediction deductive pour le raisonnement temporel. [Rapport de recherche] RR-1812, INRIA. 1992. <inria-00074860>

HAL Id: inria-00074860

<https://hal.inria.fr/inria-00074860>

Submitted on 24 May 2006

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.



UNITÉ DE RECHERCHE
INRIA-SOPHIA ANTIPOLIS

Institut National
de Recherche
en Informatique
et en Automatique

2004 route des Lucioles
B.P. 93
06902 Sophia-Antipolis
France

Rapports de Recherche

N°1812

Programme 2

*Calcul symbolique, Programmation
et Génie logiciel*

UN MÉCANISME DE PRÉDICTION DÉDUCTIVE POUR LE RAISONNEMENT TEMPOREL

Nicolas CHLEQ

Janvier 1993

Un mécanisme de prédiction déductive pour le raisonnement temporel

A mechanism for deductive predictions in temporal reasoning

Nicolas Chleq
chleq@sophia.inria.fr

Résumé

Ce document décrit un mécanisme de prédiction pour le raisonnement temporel. L'intérêt de ce mécanisme est sa simplicité, car il fonctionne en chaînage avant. On commence par décrire formellement la logique temporelle utilisée, qui est une logique temporelle réifiée trivaluée. Ce mécanisme fonctionne sur une classe particulière de théories de cette logique, que nous décrivons avant de donner le principe du mécanisme. Le mécanisme génère automatiquement les hypothèses qui lui sont nécessaires pour effectuer certaines déductions : pour l'implantation de celui-ci, nous utilisons un système de maintien de cohérence de type ATMS pour garder la trace des inférences effectuées sur la base de ces hypothèses, ainsi qu'un mécanisme de compilation des règles causales en un graphe à la RETE. On décrit ensuite quelques extensions possibles de ce mécanisme, en particulier pour traiter une logique temporelle plus complexe.

Abstract

This report describes a simple forward chaining mechanism to make predictions in temporal reasoning. We first describe a three-valued reified temporal logic. This logic serves as the formal basis of the mechanism. We then establish a restriction of this logic on which the mechanism is applicable. We outline the principles of the mechanism : it uses forward chaining on a set of facts, and can generate hypotheses to achieve some deductions. An implementation is then described. It uses ATMS and a compilation of causal rules in a RETE-like network. We finally give some possible extensions to the logic, which can be implemented in the same framework.

1 Introduction

Le travail décrit dans ce document propose une solution au problème de prédiction [12, 3] en raisonnement temporel. Le cadre choisi est celui d’une logique temporelle réifiée trivaluée pour laquelle nous avons construit un mécanisme de prédiction effectuant des inférences par modus-ponens et chaînage avant. Ce mécanisme génère automatiquement les hypothèses qui lui sont nécessaires pour effectuer certaines déductions. Dans ce cadre du problème de prédiction, un aspect que nous avons abordé est celui de la *persistance* (au sens temporel) des informations connues.

Un autre aspect développé dans ce travail est celui de l’expressivité de la logique temporelle utilisée. Nous utilisons plusieurs prédicats réalisant l’association entre entités temporelles et informations a-temporelles. Cette qualification dépend aussi de la nature des informations qualifiées : nous avons défini des catégories différentes d’informations suivant le type d’entités temporelles auxquelles elles sont associées : soit des instants qui sont des points de temps et qui n’ont pas de durée, soit des moments qui sont des périodes de temps possédant une durée non nulle.

1.1 La prédiction

La prédiction a été abondamment étudiée dans la littérature [12, 3, 16]. Pour notre travail, nous retiendrons qu’il s’agit de l’utilisation des liens de cause à effet (ou liens causaux) pour raisonner “en avant” dans le temps, et construire ainsi la ou les histoires correspondant à l’évolution possible de l’univers modélisé. L’idée retenue est de représenter ces liens par des implications logiques et d’utiliser le modus-ponens et le chaînage avant sur des instances de telles formules pour déduire de nouvelles informations. Un tel mode de raisonnement trouverait son utilité comme composant d’un système plus général de raisonnement temporel [10].

2 Motivations

Telles qu’elles sont décrites dans la littérature, les logiques temporelles réifiées sont très contraignantes en ne donnant qu’une forme de qualification temporelle. Il n’existe en général qu’un prédicat réalisant cette qualification, et celle-ci se fait toujours par rapport soit à des instants, soit à des intervalles de temps, sans offrir conjointement les deux possibilités. De plus, dans le cas d’une information liée à un intervalle de temps, on peut vouloir (ou pouvoir) ne donner qu’une des bornes de cet intervalle : on aimerait écrire $begin(t, \phi)$ ou $end(t, \phi)$, comme le Calcul d’Événements [8] permet de le faire en liant le début et la fin d’une période de temps à un événement. Par rapport à cette approche, nous ne souhaitons pas définir une ontologie complexe comme les logiques de McDermott [12] ou d’Allen [1]. Nous reprenons les critiques de Shoham [17] récusant l’utilité de ces nombreux concepts, et nous basons notre travail sur l’instant (sans durée), la période de temps représentée par un intervalle entre deux instants, et des informations “a-temporelles” à propos desquelles nous ne ferons, pour le moment, pas de suppositions particulières.

La signification intuitive des qualifications temporelles que nous avons choisies est la suivante :

begin(*point*, *fait*) signifie que le fait *fait* a une période de vérité qui commence à l’instant désigné par *point*. En particulier, on désire qu’il existe une période, si petite soit elle, finissant juste avant cet instant et pendant laquelle le fait ne soit pas vrai;

end(*point, fait*) signifie que le fait a une période de vérité qui finit exactement à l’instant *point*. Comme pour l’expression précédente, on souhaite qu’il existe une période commençant juste après cet instant final et où le fait ne soit pas vrai;

hold(*intervalle, fait*) signifie que le fait est vrai sur la période de temps dénotée par l’intervalle. En particulier, cette expression implique que le fait commence à la borne inférieure de l’intervalle et finit à la borne supérieure;

persist(*point-1, point-2, fait*) signifie que le fait est vrai sur une période de temps dans laquelle est inclus l’intervalle [*point-1, point-2*]. La signification de cette expression est inspirée de ce que Dean et McDermott [4] appellent “persistance” : l’association d’une période et d’un fait avec la même signification que celle que nous avons adoptée.

Toutes les informations ne sont pas qualifiables par le même type d’entités temporelles [13], on peut en particulier parler d’informations qui sont par nature *éphémères*, et qui ne seront associées qu’à des instants. On trouve souvent la dénomination d’*événements* pour ce genre d’information [1]. D’autres informations, au contraire, sont inséparables de l’idée de durée : on parlera d’informations *durables*, comme peut l’être le déplacement d’un objet entre deux endroits distincts. Enfin, et dernière caractérisation qui sera utilisée dans la suite, il existe des informations de nature durable, mais dont on peut aussi envisager la vérité par rapport à un instant : c’est le cas de ce qu’Allen identifie comme des *propriétés* [1]. Le fait pour un objet d’être posé à un endroit entre dans cette catégorie, on dira que “A est en B entre t_1 et t_2 ”, mais on peut aussi dire “qu’à un instant t compris entre t_1 et t_2 , A est en B”. Pour cette dernière catégorie d’informations, on parlera d’informations *stables*. Cette catégorie se retrouve dans la logique ETL de Sandewall [14] sous l’appellation d’*homogènes*, ou sous le nom de *liquides* dans la classification établie par Shoham [17].

Dans ce cadre général, le but est de construire un mécanisme déductif construisant des prédictions en utilisant des techniques simples comme le chaînage avant et le modus-ponens. Ce choix se fait au détriment de la complétude, dans le but de compenser cette perte par une efficacité accrue par rapport à des méthodes de preuve complètes mais complexes.

3 Une logique temporelle

Le but de cette partie est donc de décrire et de définir rigoureusement une logique temporelle réifiée permettant de qualifier temporellement les informations contenues dans une base de connaissances. La définition de cette logique se fait de manière analogue à ce qu’a proposé Shoham [17]. Notre logique étant trivaluée, on donne sa sémantique à l’aide d’interprétations partielles.

Dans un premier temps, les informations qualifiables (ou “atemporelles”) seront assimilées à des objets atomiques propositionnels.

Les qualifications temporelles seront réalisées au moyen de quatre prédicats : *begin*, *end*, *hold* et *persist*. On commence par la définition syntaxique des “formules bien formées” puis on enchaîne sur les structures d’interprétation et la définition sémantique.

3.1 Les catégories de propositions

Conformément aux idées énoncées dans le paragraphe 2, on établit une distinction de nature entre les propositions que l’on va qualifier. On se donnera donc trois ensembles de propositions \mathcal{P}_e , \mathcal{P}_d

et \mathcal{P}_s supposés disjoints, et correspondant respectivement aux propositions *éphémères*, *durables* et *stables*. L'union des ces trois ensembles forme l'ensemble \mathcal{P} .

3.2 Définition syntaxique

Préalablement à la définition des formules bien formées, on définit la forme des termes censés dénoter des entités temporelles. On se donne deux ensembles \mathcal{VT} et \mathcal{CT} . Le premier est un ensemble de symboles que l'on appellera "variables temporelles", et le deuxième est un ensemble de constantes temporelles dénotant des dates.

Avec les définitions précédentes, on définit une *expression d'instant* comme une des deux formes suivantes :

- t si $t \in \mathcal{VT} \cup \mathcal{CT}$, c'est à dire une variable temporelle ou une constante temporelle;
- $t + t'$, $t - t'$, $t * t'$ ou t/t' où t et t' sont deux expressions d'instant.

Une *expression d'intervalle* est de la forme $[t, t']$ avec t et t' étant deux expressions d'instant. Une *expression temporelle* est soit une expression d'instant, soit une expression d'intervalle.

Les formules bien formées sont les suivantes :

- $t < t'$ et $t \leq t'$ où t et t' sont deux expressions d'instant;
- $hold(t, p)$ où t est une expression temporelle d'instant et $p \in \mathcal{P}_e \cup \mathcal{P}_s$;
- $hold(e, p)$ où le premier terme argument e est une expression d'intervalle, et $p \in \mathcal{P}_d \cup \mathcal{P}_s$;
- $begin(t, p)$ où t est une expression d'instant, et p est une proposition appartenant à $\mathcal{P}_d \cup \mathcal{P}_s$;
- $end(t, p)$ où t est une expression d'instant, et p est une proposition appartenant à $\mathcal{P}_d \cup \mathcal{P}_s$;
- $persist(t, t', p)$ où t et t' sont des expressions d'instant, et p est une proposition appartenant à $\mathcal{P}_d \cup \mathcal{P}_s$;
- $\neg f$ où f est une formule bien formée;
- $f_1 \wedge f_2$, $f_1 \vee f_2$ où f_1 et f_2 sont des formules bien formées;
- $\forall t f$ où $t \in \mathcal{VT}$ et f est une formule bien formée.

Le connecteur \supset est défini classiquement par : $f_1 \supset f_2 \equiv \neg f_1 \vee f_2$, de même pour le quantificateur \exists par $\neg \forall \neg$. Pour simplifier l'écriture, les formes $hold(e, p)$ peuvent être écrites sous la forme $\{e, p\}$.

3.3 Définition sémantique

On commence par définir l'interprétation des expressions temporelles. On suppose un support pour le temps qui soit linéaire, dense et totalement ordonné. On prendra \mathbb{Q} , l'ensemble des nombres rationnels. Cet ensemble est supposé représenter un ensemble de *dates* ou *points* au sens habituel du terme, et il est muni d'une relation d'ordre total notée $<$.

3.3.1 L'interprétation des termes temporels

On se donne une fonction V_t de \mathcal{VT} dans \mathbf{Q} , que l'on appelle *affectation de variables*, et qui associe un nombre (c'est à dire une date) à chaque variable temporelle. Chaque symbole de fonction $+$, $-$, $*$, $/$ a son équivalent dans \mathbf{Q} comme étant la fonction arithmétique de même symbole. L'interprétation I_t d'une expression temporelle se fait ainsi :

- si $t \in \mathcal{VT}$ alors $I_t(t) = V_t(t)$;
- si $t \in \mathcal{CT}$ alors $I_t(t)$ est le nombre dont t est la représentation externe, c'est à dire qu'à la constante temporelle notée "1.23" correspond le nombre rationnel 1.23;
- pour les expressions composées avec les symboles fonctionnels $+$, $-$, $*$ et $/$ on étend V_t en posant : $I_t(e_1 f e_2) = I_t(e_1) f' I_t(e_2)$ où f' est l'opérateur équivalent dans \mathbf{Q} du symbole f .

Compte tenu de ceci, on définit l'interprétation par I_t d'une expression d'intervalle en lui faisant correspondre un intervalle de \mathbf{Q} suivant la signification habituelle de cette notion. On aura donc :

$$I_t([t, t']) = [I_t(t), I_t(t')]$$

On définit de manière classique l'appartenance d'un point à un intervalle ($t \in [t_1, t_2]$), l'inclusion stricte d'un intervalle dans un autre ($[t_1, t_2] \subset [t'_1, t'_2]$), et l'inclusion non-strictes (n'excluant pas l'égalité) est notée par \subseteq .

Une affectation de variables V_t est dite *cohérente* relativement à une expression d'intervalle $[t, t']$ si et seulement si $I_t(t) < I_t(t')$. Une affectation de variables est cohérente par rapport à un ensemble d'expressions d'intervalles, si elle est cohérente par rapport à toutes les expressions de cet ensemble.

Dans la suite, on ne parlera que d'affectations de variables cohérentes avec l'ensemble des expressions d'intervalles apparaissant dans la ou les formules considérées.

3.3.2 Les structures d'interprétation

L'interprétation des formules bien formées se fera par rapport à des structures (I, \mathbf{Q}) , où I est une fonction qui associe à chaque proposition de \mathcal{P} un ensemble de points de \mathbf{Q} si celle-ci appartient à \mathcal{P}_e et d'intervalles de \mathbf{Q} sinon. Dans le cas d'une proposition p appartenant à $\mathcal{P}_d \cup \mathcal{P}_s$, une contrainte que l'on s'impose est que les intervalles de $I(p)$ soient disjoints deux à deux :

$$\forall p \in \mathcal{P}_d \cup \mathcal{P}_s \quad \forall i, j \in I(p) \quad \text{on a } (i \cap j \neq \emptyset) \Rightarrow (i = j) \quad (1)$$

Cette contrainte peut sembler arbitraire, néanmoins elle est liée à la signification désirée pour les prédicats *begin* et *end*. La signification recherchée pour $begin(t, p)$ est que l'instant t soit le début d'une période où p est vrai, ce qui sous-entend en particulier que juste avant t la proposition p n'est pas vraie. Or la sémantique de cette formule $begin(t, p)$ est qu'elle sera vraie quand l'instant t sera la borne inférieure d'un intervalle de $I(p)$. Si la contrainte de disjonction (1) n'était pas imposée, quelle signification faudrait il donner à deux périodes de vérité de p qui se chevauchent ? Au milieu d'une des périodes, on trouverait un instant qui est aussi le début de l'autre période. Avec la signification de *begin* il y aurait probablement une contradiction, ou au minimum un cas particulier difficile à expliquer.

On écrira qu'une formule f est vraie dans une interprétation (I, \mathbf{Q}) , compte tenu de l'affectation V_t des variables temporelles, en écrivant : $I, V_t \models f$. Dans ce cas, la valeur de vérité de f , notée $I(f)$, est V (Vrai). La valeur de vérité de f est F (Faux) si $I, V_t \models \neg f$. Dans le cas où $I, V_t \not\models f$ et $I, V_t \not\models \neg f$, on dit que la valeur de vérité est indéfinie, notée \perp .

3.3.3 L'interprétation des formules bien formées

Dans ce qui suit, on donne les règles de satisfaction par une interprétation I , pour toutes les formules atomiques et leur négation. Pour les formules atomiques, on pose :

- $I, V_t \models (t < t')$ si et seulement si $I_t(t) < I_t(t')$;
- $I, V_t \models hold(t, p)$ pour $p \in \mathcal{P}_e$ si et seulement si $I_t(t) \in I(p)$;
- $I, V_t \models hold(t, p)$ pour $p \in \mathcal{P}_s$ si et seulement si $\exists i \in I(p) : I_t(t) \in i$;
- $I, V_t \models hold([t, t'], p)$ pour $p \in \mathcal{P}_d \cup \mathcal{P}_s$ si et seulement si $I_t([t, t']) \in I(p)$;
- $I, V_t \models begin(t, p)$ si et seulement si $\exists i \in I(p)$ dont $I(t)$ est la borne inférieure;
- $I, V_t \models end(t, p)$ si et seulement si $\exists i \in I(p)$ dont $I(t)$ est la borne supérieure.
- $I, V_t \models persist(t, t', p)$ si et seulement si $\exists i \in I(p)$ tel que $I_t([t, t']) \subseteq i$;

Pour les négations de formules atomiques, la satisfaction de certaines de ces expressions est définie de telle manière qu'elles ne puissent pas avoir de valeur de vérité indéfinie. L'explication en sera donnée dans le paragraphe 3.4 :

- $I, V_t \models \neg hold(t, p)$ pour $p \in \mathcal{P}_e$ si et seulement si $I_t(t) \notin I(p)$, c'est à dire si et seulement si $I, V_t \not\models hold(t, p)$;
- $I, V_t \models \neg hold(t, p)$ pour $p \in \mathcal{P}_s$ si et seulement si $\forall i \in I(p) \quad I_t(t) \notin i$, c'est à dire si et seulement si $I, V_t \not\models hold(t, p)$;
- $I, V_t \models \neg hold([t, t'], p)$ pour $p \in \mathcal{P}_d \cup \mathcal{P}_s$ si et seulement si

$$\exists i \in I(p) \quad (i \neq I_t([t, t'])) \text{ et } (i \cap I_t([t, t']) \neq \emptyset)$$

- $I, V_t \models \neg begin(t, p)$ si et seulement si $\exists [t_1, t_2] \in I(p) \quad t_1 < I_t(t) \leq t_2$;
- $I, V_t \models \neg end(t, p)$ si et seulement si $\exists [t_1, t_2] \in I(p) \quad t_1 \leq I_t(t) < t_2$.
- $I, V_t \models \neg persist(t, t', p)$ si et seulement si

$$\exists i \in I(p) \quad (I_t([t, t']) \not\subseteq i) \text{ et } (i \cap I_t([t, t']) \neq \emptyset)$$

Enfin, pour les formules composées à l'aide des connecteurs \wedge et \vee , on obtient leur valeur de vérité de la manière suivante :

- $I(f_1 \wedge f_2) = \min_{\leq}(I(f_1), I(f_2))$

- $I(f_1 \vee f_2) = \max_{\preceq}(I(f_1), I(f_2))$

où l'ordre \preceq est défini par : $F \preceq \perp \preceq V$.

Dans le cas des formules avec quantificateurs, on écrit ceci :

- $I, V_t \models \forall v f$ si et seulement si $I, V'_t \models f$ pour tout V'_t coïncidant avec V_t sauf peut-être pour v .

Un *modèle* d'une formule (ou d'un ensemble de formules) est défini classiquement comme une interprétation qui satisfait la formule (ou toutes les formules de l'ensemble considéré), et ceci pour toutes les affectations de variables possibles.

Intuitivement, une proposition persiste sur tout sous-intervalle d'un intervalle où elle est vraie. Compte tenu de la condition (1) imposée aux intervalles de l'ensemble $I(p)$ on retrouve l'idée de la non-interruption d'une période de vérité. La persistance apparaît aussi comme une propriété d'une information durable analogue à la vérité sur un intervalle, mais moins forte car on n'impose rien sur "avant" et "après" la période considérée. Par comparaison, *hold* est plus contraignant car $hold([t, t'], p)$ signifie que p est vrai "exactement" sur l'intervalle $[t, t']$. En particulier, on a :

$$holds([t, t'], p) \supset begin(t, p)$$

et

$$holds([t, t'], p) \supset end(t', p)$$

3.4 Pourquoi une logique trivaluée ?

Une caractéristique souhaitable pour une hypothèse est qu'il s'agisse d'une information non déductible des connaissances que l'on possède déjà, et non-contradictoire avec celles-ci. De plus, une hypothèse doit être utile à la poursuite du raisonnement, dans notre cas pour effectuer une ou des prédictions (déductions) supplémentaires. Le premier point se formalise bien dans le cas où l'on considère une interprétation partielle I pour notre logique. Une condition nécessaire pour qu'une formule h soit une hypothèse est qu'elle vérifie $I \not\models h$ et $I \not\models \neg h$, c'est à dire qu'elle ne soit ni vraie, ni fausse. La définition adoptée pour notre logique limite ainsi la forme syntaxique des hypothèses aux formules de la forme $\{[t, t'], p\}$, $begin(t, p)$, $end(t, p)$ ou $persist(t, t', p)$ car seuls ces littéraux peuvent avoir une valeur de vérité indéfinie. En particulier, un littéral de la forme $\{t, p\}$ avec p stable ne pourra pas être une hypothèse. Nous préférons qu'une formule de cette forme ne puisse pas être supposée vraie "ex-nihilo" : elle devra être la conséquence d'une formule $\{[t_1, t_2], p\}$ ou $persist(t_1, t_2, p)$ telle que t soit dans l'intervalle $[t_1, t_2]$, cette deuxième formule pouvant être une hypothèse si nécessaire. De cette manière, on essaie de satisfaire un critère de "simplicité" dans les modèles d'une théorie, c'est à dire limiter le nombre de périodes de vérité d'une proposition.

Pour les quatre formes de littéraux susceptibles d'avoir une valeur de vérité indéfinie, la condition correspondante s'explique ainsi :

$$\left. \begin{array}{l} I, V_t \not\models \{[t, t'], p\} \\ I, V_t \not\models \neg\{[t, t'], p\} \end{array} \right\} \Leftrightarrow \forall i \in I(p) \quad i \cap I_t([t, t']) = \emptyset$$

$$\left. \begin{array}{l} I, V_t \not\models \text{begin}(t, p) \\ I, V_t \not\models \neg \text{begin}(t, p) \end{array} \right\} \Leftrightarrow \forall i \in I(p) \ I_t(t) \notin i$$

$$\left. \begin{array}{l} I, V_t \not\models \text{end}(t, p) \\ I, V_t \not\models \neg \text{end}(t, p) \end{array} \right\} \Leftrightarrow \forall i \in I(p) \ I_t(t) \notin i$$

$$\left. \begin{array}{l} I, V_t \not\models \text{persist}(t, t', p) \\ I, V_t \not\models \neg \text{persist}(t, t', p) \end{array} \right\} \Leftrightarrow \forall i \in I(p) \ i \cap I_t([t, t']) = \emptyset$$

La signification désirée est qu’une hypothèse, ayant donc une valeur de vérité indéfinie avant d’être supposée vraie, qualifie temporellement une proposition par un intervalle qui soit au mieux disjoint de ceux déjà connus pour cette proposition, et au minimum qui ne cause pas de violation de la contrainte (1) sur les intervalles des ensembles $I(p)$.

Il y a d’autres possibilités pour obtenir une logique trivaluée. En particulier, on peut envisager comme dans ETL [15] un ensemble de dates comprenant aussi bien des nombres que des “time token” (non liés à une date connue) et ordonné partiellement au moins en ce qui concerne les “time token”. Les expressions de comparaison entre dates peuvent donc avoir une valeur de vérité indéfinie. Ceci n’a pas été retenu dans le système qui va être décrit, car une des motivations était d’envisager de construire un modèle temporel sous la forme d’une base de faits, c’est à dire d’un ensemble de littéraux clos limité aux prédicats décrivant une qualification temporelle, ce que ne fait pas le prédicat $<$. Une hypothèse ou un fait de la forme $t < t'$ n’aurait pas trouvé sa place dans ce cadre.

4 Un mécanisme de déduction

Cette partie décrit un mécanisme, utilisable pour la prédiction, et fonctionnant en chaînage avant avec le modus-ponens comme règle d’inférence. L’idée est de considérer des théories comprenant des formules de notre logique temporelle de la forme $\phi_1 \wedge \dots \wedge \phi_n \supset f_1 \wedge \dots \wedge f_m$ comme décrivant des liens de causes à effets et d’utiliser le modus ponens pour déduire les f_i lorsque les ϕ_i sont vérifiés. Un tel mécanisme n’est évidemment pas une procédure de décision ou de preuve comme celle de la logique ETL [15], il s’agit plutôt d’un mécanisme comparable à un système à règles de production.

En assimilant une “histoire” du système modélisé à un modèle de la théorie, la prédiction se ramène au problème de construire ces modèles. Parmi tous les modèles d’une théorie, on peut, comme [9], en préférer certains comme correspondant mieux à l’intention qui a dicté l’écriture de la théorie. En particulier, une exigence intéressante est de préférer les modèles où les causes sont “justifiées” : c’est à dire qu’on s’intéresse aux modèles qui satisfont une formule, soit parce que celle-ci appartient à la théorie initiale, soit parcequ’il existe une instance de règle causale dont la formule considérée est en partie droite de la règle (les *conséquents*) et dont la partie gauche (les *antécédents*) est vérifiée par ce même modèle. Le mode de raisonnement choisi satisfait cette exigence.

4.1 Les théories acceptées

Pour le mécanisme décrit ici, on se donne dans un premier temps quelques contraintes sur la forme des problèmes que l’on pourra traiter. Les théories acceptées comprennent deux types de formules :

- soit des formules réduites à un littéral clos utilisant un des quatre prédicats *hold*, *begin*, *end* ou *persist*. Ces formules seront appelées *faits* dans la suite;
- soit des formules de la forme $\phi_1 \wedge \dots \wedge \phi_n \supset f_1 \wedge \dots \wedge f_m$, où les ϕ_i (les antécédents) sont des formules atomiques (ou des négations de formule atomique) utilisant n'importe quel prédicat de notre logique, et les f_i (les conséquents) sont des formules atomiques limitées aux prédicats *hold*, *begin*, *end* et *persist*. Ces formules seront appelées *règles causales*. Une contrainte supplémentaire est que toutes les variables temporelles ayant une occurrence dans la partie droite d'une règle causale aient aussi au moins une occurrence dans la partie gauche de celle-ci. Dans une règle causale, toutes les variables temporelles qui y apparaissent sont supposées quantifiées universellement.

Cette restriction se comprend mieux dans la mesure où l'on souhaite faire des déductions par modus-ponens sur des instances de règles causales. On obtient ainsi en conclusion des littéraux clos qui deviennent de nouveaux faits.

4.2 Principe

Le mode de déduction choisi pose deux problèmes qu'il faut résoudre. En premier lieu, nous étudions la possibilité de construire une "base de faits" pour représenter un modèle, et quelles sont les conséquences de ce choix. Le deuxième problème est de savoir construire les "bonnes" instances de règles causales, c'est à dire celles dont toutes les formules de la partie gauche sont vérifiées, ceci afin de les utiliser pour produire de nouveaux faits.

4.2.1 Peut on construire une "base de faits" ?

L'objectif de ce paragraphe est d'étudier la construction d'une base de faits pour notre logique. En effet, il est a priori possible de désigner intentionnellement une interprétation I par un ensemble F_I de faits satisfaits par cette interprétation. Comparée à la construction explicite des instants et intervalles de vérité de chaque proposition, cette méthode serait même souhaitable lorsqu'on considère les prédicats *begin* ou *end* : pour un littéral clos utilisant un de ces prédicats, il existe potentiellement une infinité de modèles qui satisfont cette formule. Cette discussion est limitée à une base de faits qui ne contiendrait que des littéraux clos utilisant un des quatre prédicats *hold*, *begin*, *end*, ou *persist*.

Le problème principal qui se pose si l'on désire construire une base de faits, est qu'on ne peut pas avoir simplement équivalence entre $l \in F_I$, et $I \models l$ où I est une interprétation, et F une base de faits. On peut au mieux avoir $l \in F_I$ implique $I \models l$. En effet, si $F_I = \{persist(1, 4, P)\}$ alors on a $I \models persist(1, 4, P)$ mais aussi, par définition de *persist*, $I \models persist(2, 3, P)$ mais cette formule n'appartient pas à F_I .

Dans la suite, on associera une base de faits F_I et une interprétation I par la relation suivante :

$$l \in F_I \Rightarrow I \models l \tag{2}$$

car exiger une équivalence dans cette relation n'est pas réalisable pratiquement. Le problème principal qui se pose alors est d'obtenir une condition nécessaire et suffisante pour que, connaissant F_I , un fait quelconque (positif ou négatif) soit satisfait par I .

Une première nécessité est d'assurer la cohérence de la base de faits. Comme l'on a imposé une contrainte assez forte sur les modèles, à savoir que pour une proposition durable ou stable tous ses intervalles de vérité doivent être disjoints deux à deux (relation (1)), il faut donc se donner le moyen de respecter cette contrainte avec la représentation des modèles qu'on se propose d'utiliser. Ceci se fait par le biais d'une première relation de contradiction entre littéraux clos, que l'on définit ainsi :

Définition : la *contradiction de type 1* entre littéraux clos est une relation symétrique, qui s'explique ainsi :

- $\{[t, t'], p\}$ pour $p \in \mathcal{P}_d \cup \mathcal{P}_s$ est contradictoire avec :
 - $\{[t_1, t_2], p\}$ tel que $t < t_1 < t'$ ou $t < t_2 < t'$, c'est à dire $[t, t'] \neq [t_1, t_2]$ et $[t, t'] \cap [t_1, t_2] \neq \emptyset$;
 - $begin(t_1, p)$ tel que $t < t_1 \leq t'$;
 - $end(t_1, p)$ tel que $t \leq t_1 < t'$;
 - $\neg\{[t_1, t_2], p\}$ tel que $t \leq t_1 < t_2 \leq t'$;
- $persist(t, t', p)$ est contradictoire avec :
 - $\{[t_1, t_2], p\}$ tel que $t < t_1 \leq t'$ ou $t \leq t_2 < t'$, c'est à dire $[t, t'] \not\subseteq [t_1, t_2]$ et $[t, t'] \cap [t_1, t_2] \neq \emptyset$;
 - $begin(t_1, p)$ tel que $t < t_1 \leq t'$;
 - $end(t_1, p)$ tel que $t \leq t_1 < t'$;
 - $\neg\{[t_1, t_2], p\}$ tel que $t \leq t_1 < t_2 \leq t'$;

Une base de faits sera dite *cohérente-1* si elle ne contient pas simultanément deux littéraux clos liés par la contradiction de type 1, ni un littéral clos et sa négation. En considérant une interprétation I et un ensemble F_I de faits, liés par la condition (2), on vérifie que la contrainte de disjonction (1) pour l'interprétation I implique la cohérence-1 de F_I . La réciproque de cette propriété n'est pas vraie car la relation (2) n'est pas une équivalence.

Il existe une deuxième forme de cohérence que doit respecter un ensemble de faits. Cette cohérence est induite par la nécessité de respecter certains théorèmes de la logique temporelle utilisée. Elle s'exprime aussi sous la forme d'une relation de contradiction, que nous appellerons *contradiction de type 2*. Cette relation est définie par :

Définition : la *contradiction de type 2* entre littéraux clos est une relation symétrique, qui s'explique ainsi :

- $\{[t, t'], p\}$ pour $p \in \mathcal{P}_d \cup \mathcal{P}_s$ est contradictoire avec :
 - $\neg begin(t, p)$;
 - $\neg persist(t_1, t_2, p)$ tel que $[t_1, t_2] \subseteq [t, t']$;
 - $\neg end(t', p)$;
- $\neg\{t, p\}$ pour $p \in \mathcal{P}_s$ est contradictoire avec :
 - $\{[t_1, t_2], p\}$ tel que $t_1 \leq t \leq t_2$;
 - $persist(t_1, t_2, p)$ tel que $t_1 \leq t \leq t_2$;

- $begin(t, p)$;
- $end(t, p)$.

De la même manière que pour la première forme de contradiction, on dira qu'une base de faits est *cohérente-2* si elle ne comprend pas simultanément deux faits liés par la contradiction de type 2. Une base de faits *cohérente* satisfait aux deux critères de cohérence précédents. Les théorèmes correspondant à la deuxième forme de contradiction sont :

$$\forall t, t' \{ [t, t'], p \} \Leftrightarrow begin(t, p) \wedge persist(t, t', p) \wedge end(t', p)$$

$$\forall t, t' \{ persist(t, t', p) \Rightarrow (\forall t_1, t_2 (t \leq t_1 < t_2 \leq t') \Rightarrow persist(t_1, t_2, p))$$

pour p appartenant à $\mathcal{P}_d \cup \mathcal{P}_s$, et

$$\forall t, t' \{ [t, t'], p \} \Rightarrow (\forall t'' (t \leq t'' \leq t') \Rightarrow \{ t'', p \})$$

pour p appartenant à \mathcal{P}_s .

Déterminer la satisfaction d'un fait positif : La satisfaction par I d'un fait positif l , connaissant une base de faits F_I cohérente, liée à I par la relation (2), peut se déterminer ainsi :

1. soit l appartient à F_I ; soit l peut être prouvé à partir des faits de F_I à l'aide d'une des quatre règles suivantes;
2. $\{ [t, t'], p \}$ peut être prouvé si on peut satisfaire $begin(t, p)$, ainsi que $persist(t, t', p)$ et aussi $end(t', p)$;
3. $persist(t, t', p)$ peut être prouvé si on peut satisfaire $\{ [t_1, t_2], p \}$ ou $persist(t_1, t_2, p)$ tel que $[t, t'] \subseteq [t_1, t_2]$;
4. $begin(t, p)$ peut être prouvé si on peut satisfaire $\{ [t, t'], p \}$ avec t' quelconque;
5. $end(t, p)$ peut être prouvé si on peut satisfaire $\{ [t', t], p \}$ avec t' quelconque;

Cette méthode est récursive, et il est impératif qu'elle soit complétée pour éviter des requêtes circulaires. Elle ne fournit une condition que pour la satisfaction d'un fait positif, mais pas pour la non-satisfaction de celui-ci.

Déterminer la satisfaction d'un fait négatif : La définition de la sémantique de notre logique permet de vérifier une propriété intéressante. Si I est une interprétation, et l un littéral clos positif de la forme $\{ [t, t'], p \}$, $begin(t, p)$, $end(t, p)$ ou $persist(t, t', p)$, on a l'équivalence entre les deux propositions suivantes :

- (i) $I \models \neg l$
- (ii) $I \models l'$ où l' est un littéral clos positif, et l et l' sont en contradiction de type 1.

Pour les autres formes de littéraux positifs, la satisfaction de $\neg l$ se ramène par définition à la non-satisfaction de l . La preuve de cette propriété se fait très simplement à partir de la définition sémantique et de la définition de la contradiction de type 1.

L'intérêt de cette propriété est de donner un moyen simple de vérifier la satisfaction ou d'un fait négatif. Malheureusement, la transposition de cette propriété à un ensemble F_I cohérent lié à une interprétation I par (2) donne :

$$(\exists l' \in F_I \ l' \text{ et } l \text{ sont en contradiction de type 1}) \Rightarrow I \models \neg l \quad (3)$$

toujours pour l et l' positifs. Ceci donne une condition suffisante pour la satisfaction d'un fait négatif, mais pas pour la non-satisfaction de celui-ci.

Cette "incomplétude" d'une base de faits peut être en partie contournée en munissant celle-ci du mécanisme d'interrogation décrit précédemment ainsi que d'un mécanisme qui complète celle-ci en ajoutant $holds([t, t'], p)$ dès que $begin(t, p)$, $persist(t, t', p)$ et $end(t', p)$ sont présents dans la base, et qui "fusionne" les persistances connues qui se recouvrent.

4.2.2 Le choix des inférences

Pour le choix des inférences possibles (l'ensemble de conflits), la première et la plus simple des solutions est de construire celles-ci à partir des faits déjà connus ou déduits auparavant : cet appariement entre un littéral clos et un antécédent d'une règle causale produit une ou plusieurs équations que doivent vérifier les variables temporelles. On collecte ces équations sur toute la partie gauche de la règle, on résout le système obtenu, et si il a une solution, on vérifie que celle-ci satisfait les antécédents utilisant le prédicat $<$ (comparaison entre dates). Par exemple, à partir de la règle causale

$$\{[t_1, t_2], P\} \wedge \{t_3, Q\} \wedge (t_3 < t_2 - 3) \supset \{[t_2, t_2 + 2], R\}$$

et des faits connus $\{[2, 6], P\}$ et $\{2.5, Q\}$ on peut instancier la règle avec $t_1 = 2$, $t_2 = 6$ et $t_3 = 2.5$ et en déduire $\{[6, 8], R\}$.

L'utilisation directe dans les règles et les faits des prédicats $begin$, end et $persist$, dont la définition est plus complexe que $hold$ donne de nouvelles possibilités d'appariements entre un fait et un littéral en partie gauche d'une règle. Dans l'exemple précédent, si on ne connaît que le fait $begin(2, P)$, on peut malgré tout envisager, sous certaines conditions, satisfaire $\{[t_1, t_2], P\}$. Pour cela, on a une équation $t_1 = 2$ et il faudra en plus pouvoir vérifier $t_1 < t_2$, $persist(2, t_2, P)$ et $end(t_2, P)$, avec une valeur pour t_2 éventuellement fournie par d'autres antécédents de la règle (ce qui ne pourra pas être le cas sur cette règle). L'idée est de généraliser l'appariement et que celui-ci retourne conjointement un ensemble d'équations, un ensemble d'inéquations, et une "preuve" décrivant comment l'antécédent apparié peut être satisfait à partir du fait qui a servi à l'appariement. Une méthode analogue a été proposée par Shanahan [16] dans le cadre d'un système de raisonnement abductif basé sur le Calcul d'Événements.

Les équations sont rassemblées sur la totalité de la partie gauche de la règle, puis le système obtenu est résolu. Si il a une solution on vérifie que celle-ci satisfait les inéquations ainsi que les antécédents utilisant le prédicat $<$ (comparaison entre instants). Si c'est le cas, l'instance correspondante de la règle peut être proposée au moteur d'inférences.

Les appariements possibles avec les littéraux positifs en antécédents sont répertoriés dans la table de la figure 4 de l'annexe A. Les formules marquées par * dans les preuves sont les formules qu'il faut vérifier en plus du fait pour satisfaire l'antécédent : ces formules, lorsqu'elles sont complètement instanciées, sont des hypothèses potentielles pour le moteur.

Les règles admettent des antécédents négatifs : le traitement de ceux-ci repose aussi sur la capacité à générer des hypothèses. Pour une règle de la forme $a \wedge \neg b \supset c$ avec a vrai et b non vrai, il subsiste deux cas, soit $\neg b$ est vrai ce qui ne pose pas de problèmes, soit $\neg b$ est indéfini (et par conséquent b aussi). Dans ce cas, $\neg b$ devient une hypothèse que l'on rajoute à ce que l'on sait déjà et on déduit c . De même que pour les antécédents positifs, on peut aussi appairier les antécédents négatifs avec des faits. Ces appariements sont répertoriés dans la table de la figure 5 de l'annexe A. Ces appariements reposent sur la définition sémantique de la logique donnée précédemment.

Les preuves retournées par l'appariement correspondent à différents théorèmes de la logique temporelle, et que l'on peut montrer à partir de la définition qui en a été donnée. Des exemples de ces théorèmes sont :

- $\forall t, t' \forall p \in \mathcal{P}_d \cup \mathcal{P}_s \{[t, t'], p\} \equiv (\text{begin}(t, p) \wedge \text{persist}(t, t', p) \wedge \text{end}(t', p))$
- $\forall t, t' \forall p \in \mathcal{P}_s \text{persist}(t, t', p) \Rightarrow (\forall t_1 t \leq t_1 \leq t' \Rightarrow \{t_1, p\})$

Parmi toutes les inférences possibles construites par ce mécanisme, on en choisit une particulière. Les critères de choix n'ont pas été approfondis dans ce mécanisme : on en donne quelques exemples au paragraphe 4.3.2. Une fois ce choix effectué, on dispose donc d'une règle causale, d'une substitution de ses variables temporelles, et d'un arbre de preuve dont l'étage inférieur correspond à l'instanciation de la règle avec la substitution. L'application de cette inférence se fait en appliquant toutes les déductions de l'arbre. Lorsqu'une formule feuille de la preuve d'un antécédent est à vérifier, il y'a trois cas possibles :

1. soit elle est vraie, c'est à dire qu'elle appartient à la base de faits, ou qu'elle peut en être déduite simplement à l'aide d'une des règles énoncées au paragraphe 4.2.1. Ce cas ne pose aucun problème particulier, et la déduction correspondante peut être effectuée.
2. Soit elle est fautive, c'est à dire que sa négation peut être satisfaite. Ce test peut se faire à l'aide de la condition (3) pour une formule positive. Dans ce cas, la déduction n'est donc pas faisable, et il faut choisir une autre inférence.
3. Si les deux tests précédents ont échoué, on n'a pourtant pas une condition nécessaire et suffisante pour affirmer que la formule n'est ni vraie, ni fautive. On peut cependant ajouter, en tant qu'hypothèses, cette formule à la base de faits sans que celle-ci ne devienne inconsistante, et ainsi effectuer la déduction concernée. Il est donc nécessaire de garder une trace de cette inférence car rien n'empêche qu'elle soit révisable ultérieurement, ni qu'elle ne conduise à une contradiction.

4.3 Implémentation

On utilise un moteur réalisant classiquement des inférences par modus-ponens et fonctionnant en chaînage avant à partir des instances des règles causales de la théorie. On a trois modules : le moteur, un module d'appariement, et un module de maintien de la cohérence RMS ("Reason Maintenance System").

- Le *moteur* est classiquement un moteur fonctionnant par modus-ponens. Il choisit une inférence dans un ensemble mis à jour par le module d'appariement, réalise l'inférence sélectionnée et produit des littéraux clos.
- Le module d'appariement comprend un compilateur des règles causales en un graphe à la RETE [5] et les mécanismes de propagation dans ce graphe. Une fois les règles compilées, ce module reçoit en entrée les littéraux clos produits par le moteur, et met à jour l'ensemble de conflits, qui regroupe les inférences possibles.
- Le RMS est le module de maintien de la cohérence, d'après la description qu'en a fait McDermott [11]. Dans l'utilisation qui en est faite ici, il est analogue à un ATMS qui gèrerait des clauses au lieu de justifications : chaque clause à n éléments du RMS correspond à n justifications pour un ATMS. Les littéraux de chaque clause sont étiquetés par un "label" qui est une expression booléenne d'hypothèses décrivant dans quel contexte (c'est à dire sous quelles hypothèses) chaque littéral ainsi étiqueté est vrai. Le RMS gère les hypothèses effectuées par le moteur et garde la trace de toutes les inférences effectuées, en particulier celles qui font intervenir des hypothèses, et s'assure aussi que deux littéraux contradictoires ne sont pas vrais dans le même contexte.

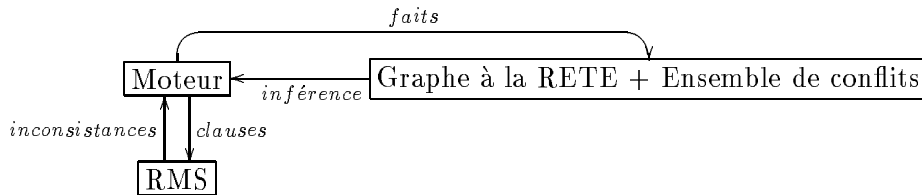


Figure 1 : Architecture du système

L'ensemble de conflits est partagé entre le module d'appariement et le moteur d'inférences. La base de fait reçoit les faits déduits par le moteur. Ces faits sont indexés par la proposition qu'ils qualifient. La base de faits possède deux mécanismes qui tentent de porter remède à son incomplétude (voir à ce sujet le paragraphe 4.2.1). Le premier mécanisme prend en charge les requêtes et détermine la satisfaction d'un fait suivant les règles du paragraphe 4.2.1. Le deuxième mécanisme travaille à partir des faits reçus par la base, et applique les deux règles d'inférence suivantes :

$$\frac{\textit{begin}(t,p) \quad \textit{persist}(t,t',p) \quad \textit{end}(t',p)}{\{[t,t'],p\}}$$

et

$$\frac{\textit{persist}(t_1,t_2,p) \quad \textit{persist}(t'_1,t'_2,p) \quad [t_1,t_2] \cap [t'_1,t'_2] \neq \emptyset}{\textit{persist}(\min(t_1,t'_1), \max(t_2,t'_2), p)}$$

pour compléter la base de faits. Les inférences réalisées par ces deux règles sont communiquées au RMS, et au module d'appariement en ce qui concerne les faits produits.

Le moteur structure l'espace de recherche sous forme d'ensembles d'hypothèses qui forment des contextes. Le RMS a en particulier en charge de détecter les ensembles inconsistants d'hypothèses ("nogood-sets") et d'invalider ceux-ci pour que le moteur ne les explore pas. La détection des contextes inconsistants se fait suivant la relation de contradiction entre les littéraux clos produits par le moteur.

4.3.1 Le module d'appariement

La compilation des règles causales Le graphe à la RETE est construit à partir des parties gauches des règles. Les différences avec un graphe de RETE classique sont les suivantes :

- les nœuds qui réalisent l'appariement entre un littéral clos et une formule temporelle quelconque utilisent les tables 4 et 5 et construisent conjointement un ensemble d'équations et d'inéquations entre les expressions temporelles et les constantes apparaissant dans le littéral clos, ainsi qu'une "preuve" permettant de vérifier la formule à partir du littéral clos reçu;
- il n'y a pas de nœuds "join" à deux entrées : ceux-ci sont remplacés par un nœud unique pour chaque règle causale. Ce nœud construit les systèmes d'équations et d'inéquations, les résout et, si il existe une solution, instancie les formules et les preuves avec les valeurs obtenues et crée une nouvelle inférence possible qui est ajoutée à l'ensemble de conflits.

A titre d'exemple, la figure 2 montre le graphe construit à partir des règles causales

$$R1 : \{t, pull-trigger\} \wedge \{t, loaded\} \supset end(t, loaded)$$

$$R2 : \{t, pull-trigger\} \wedge \{t, loaded\} \wedge \{t, alive\} \supset end(t, alive)$$

où *loaded* et *alive* sont des propositions stables, et *pull-trigger* est une proposition éphémère.

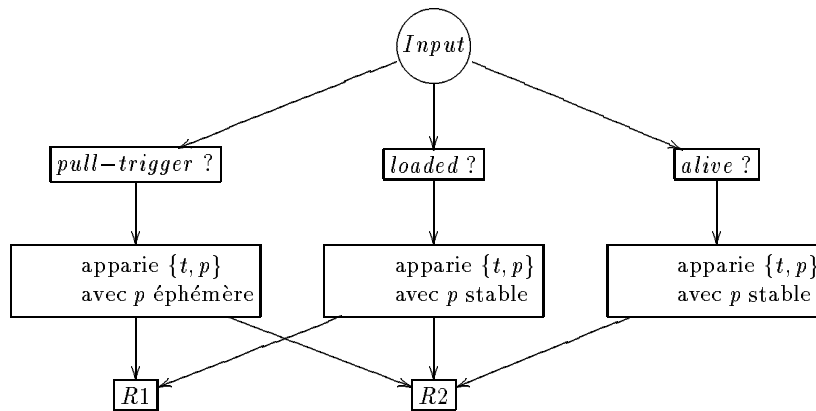


Figure 2 : Structure du graphe pour les règles causales R1 et R2

Fonctionnement Lorsque l'on passe la formule $\{4, pull-trigger\}$ à ce graphe, celui-ci donne au nœud R1 l'équation $t = 4$, aucune inéquation et aucune preuve. Lorsqu'ensuite on passe la formule $begin(1, loaded)$ au graphe, le nœud qui l'apparie avec $\{t, loaded\}$ va donner deux solutions possibles :

- la première donne l'équation $t = 1$, aucune inéquation, et la preuve :

$$\frac{begin(1, loaded)}{\{t, loaded\}}$$

- la seconde solution ne comprend aucune équation, l'inéquation $t > 1$ et la preuve suivante :

$$\frac{\text{begin}(1, \text{loaded}) \quad \text{persist}(1, t, \text{loaded})^*}{\{t, \text{loaded}\}}$$

dans lequel la formule $\text{persist}(1, t, \text{loaded})$ est marquée comme “hypothèse”.

Le nœud R1 rassemble ceci avec ce qu'il a précédemment reçu et construit d'abord le système d'équations $\{t = 1; t = 4\}$ qui n'a pas de solution, et ensuite le système d'équations $\{t = 4\}$ qui ne pose pas de problème à résoudre, le système d'inéquations $\{t > 1\}$ qui est vérifié par la valeur 4 pour t , et enfin l'arbre de preuve :

$$\frac{\frac{\text{begin}(1, \text{loaded}) \quad \text{persist}(1, 4, \text{loaded})^*}{\{4, \text{loaded}\}} \quad \{4, \text{pull-trigger}\}}{\text{end}(4, \text{loaded})}$$

obtenu en rassemblant et en instanciant les preuves qu'il possédait déjà.

Tout ceci forme une inférence possible, qui si elle est choisie par le moteur, provoquera la construction d'un nouveau contexte obtenu en ajoutant $\text{persist}(1, 4, \text{loaded})$ au contexte courant, et passera les clauses suivantes au RMS :

$$\neg \text{begin}(1, \text{loaded}) \vee \neg \text{persist}(1, 4, \text{loaded}) \vee \{4, \text{loaded}\}$$

$$\neg \{4, \text{pull-trigger}\} \vee \neg \{4, \text{loaded}\} \vee \text{end}(4, \text{loaded})$$

De manière analogue à [6], l'ensemble de conflits ainsi généré est indépendant du contexte de travail courant. Dans celui-ci on ne retient que les inférences dont les faits feuilles de la preuve sont vraies dans le contexte courant. L'inférence de l'exemple précédent ne sera visible par le moteur que dans les contextes où les formules $\text{begin}(1, \text{loaded})$ et $\{4, \text{pull-trigger}\}$ seront vraies.

4.3.2 Le moteur

Le moteur a en charge de choisir un contexte à explorer, qui est par défaut celui qui est en cours d'utilisation, de choisir une inférence dans l'ensemble de conflits et de réaliser celle-ci.

L'espace de recherche est structuré par les contextes qu'explore le moteur. Ces contextes sont organisés en treillis [2], dont seule la partie correspondant aux contextes déjà rencontrés est construite. Le moteur voit ce treillis comme un arbre, en privilégiant toujours un père particulier parmi ceux d'un contexte, celui par lequel on est arrivé sur ce contexte. Le retour arrière se fait sur ce contexte père, ce qui permet de simplifier l'implantation de ce retour arrière tout en gardant le plus d'information possible sur les contextes.

Un contexte est soit vide si il n'y a plus d'inférences possibles, soit inconsistant si le RMS l'a marqué comme tel, soit *ouvert* et donc encore utilisable pour faire des inférences.

Un contexte sert de filtre pour examiner à la fois la base et en extraire seulement les littéraux clos vrais dans le contexte, et aussi l'ensemble de conflits duquel on n'extrait que les inférences dont les feuilles de l'arbre de preuve non marquées comme hypothèse sont vraies dans le contexte. Une

inférence est ensuite considérée activable si pour toutes les feuilles hypothèses de son arbre, soit la formule correspondante est vraie, soit elle est inconnue et aucune des formules déjà connues et contradictoires avec celle-ci n'est vraie.

Diverses heuristiques sont possibles pour départager les inférences candidates restant en lice. On peut préférer celles dont le coût en hypothèse est le plus faible, celles dont la portée temporelle du conséquent est minimal ou maximal suivant la relation d'antériorité temporelle. On peut aussi choisir de réaliser toutes les inférences candidates. Pour le moment, ces considérations stratégiques n'ont pas été développées plus avant.

4.4 Un exemple

Un exemple classique du Raisonnement temporel est le “Yale Shooting Problem” (YSP) [7]. Celui-ci suppose un pistolet chargé dont on actionne la gachette, ce qui cause la mort de la personne (ou de l'animal, suivant les versions) supposée être dans la ligne de mire.

Par rapport au problème original, qui ne postule qu'une action sur la gachette, on en supposera une deuxième qu'on espère suivie d'aucun effet, sauf si l'on a rechargé le pistolet après le premier coup de feu.

La théorie correspondant à ce problème est la suivante :

$$\textit{begin}(1, \textit{loaded}) \tag{4}$$

$$\{4, \textit{pull-trigger}\} \tag{5}$$

$$\{7, \textit{pull-trigger}\} \tag{6}$$

$$\{2, \textit{alive}\} \tag{7}$$

$$\forall t \{t, \textit{pull-trigger}\} \wedge \{t, \textit{loaded}\} \supset \{[t + 0.1, t + 2], \textit{noise}\} \tag{8}$$

$$\forall t \{t, \textit{pull-trigger}\} \wedge \{t, \textit{loaded}\} \supset \textit{end}(t, \textit{loaded}) \tag{9}$$

$$\begin{aligned} \forall t \{t, \textit{alive}\} \wedge \{t, \textit{pull-trigger}\} \wedge \{t, \textit{loaded}\} \\ \supset \textit{end}(t, \textit{alive}) \end{aligned} \tag{10}$$

avec $\mathcal{P}_e = \{\textit{pull-trigger}\}$, $\mathcal{P}_d = \emptyset$ et $\mathcal{P}_s = \{\textit{loaded}, \textit{noise}, \textit{alive}\}$.

La proposition *pull-trigger* représente l'action sur la gachette du pistolet, *loaded* le fait que le pistolet soit chargé, *noise* désigne le bruit du coup de feu et *alive* le fait que la cible soit vivante.

Sur cet exemple, la figure 3 montre les contextes explorés par le moteur, ainsi que les formules closes qui sont vraies dans le contexte sélectionné, le seul qui n'ait pas de fils et ne soit pas inconsistant.

Le contexte dont l'ensemble d'hypothèses est $\{\textit{persist}(1, 7, \textit{loaded})\}$ est inconsistant, et par voie de conséquence tous ceux qui incluent cette hypothèse sont aussi inconsistants. En effet, du fait *persist*(1, 7, *loaded*) on peut déduire $\{4, \textit{loaded}\}$ et donc aussi *end*(4, *loaded*) ce qui est contradictoire avec l'hypothèse précédente.

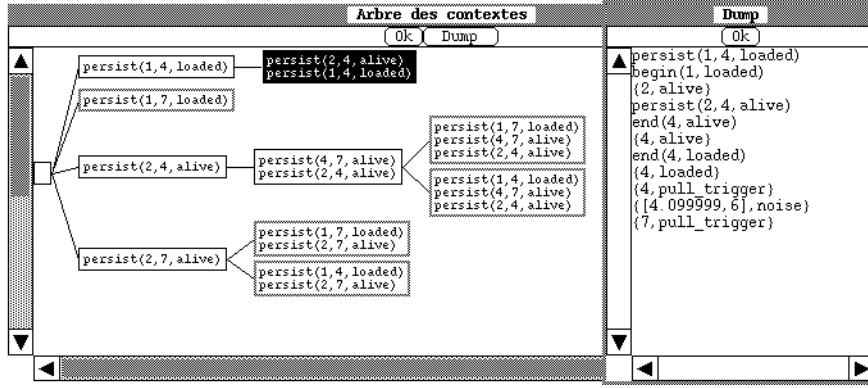


Figure 3 : Les contextes explorés pour le YSP

5 Extensions

5.1 Dédution avec horizon temporel limité

Une première extension possible permet de limiter la portée temporelle des déductions effectuées. Etant donnée une date limite, il suffit d'écartier les instances de règles causales dont un des conséquents est postérieur à cette date limite. Dans l'implémentation décrite ici, il suffit de ne pas transmettre au graphe de RETE les faits postérieurs à cette limite : l'ensemble de conflits ne grossit plus et s'épuise de lui même au fur et à mesure que le moteur travaille.

5.2 Extension du langage

Une extension souhaitable est d'enrichir le langage manipulé. Un premier souhait est de manipuler des connecteurs dans la partie "atemporelle" des formules : on qualifie temporellement un langage propositionnel avec les connecteurs \wedge et \vee . A titre d'exemple, une sémantique possible pour le connecteur \wedge pourrait être :

$$I, V_t \models \{[t, t'], p \wedge q\} \text{ ssi } \exists i \in I(p) \exists j \in I(q) \ i \cap j = I_t([t, t'])$$

dans le cas de deux propositions stables ou durables. Ceci peut être implanté aisément dans le cadre du mécanisme d'appariement : pour un antécédent de règle causale de la forme $\{[e, e'], p \wedge q\}$, on peut apparier de la même manière que si il s'agissait d'une proposition unique. Il faut fournir des faits qualifiant l'expression $p \wedge q$, soit si ils existent tel quels, soit en les fabriquant à partir des faits qualifiant p et des faits qualifiant q . Par exemple, de $\{[1, 4], P\}$ et $\{[2, 6], Q\}$, on peut déduire $\{[2, 4], P \wedge Q\}$. La génération d'hypothèses est toujours possible : de $\{[1, 4], P\}$ et $begin(2, Q)$ on peut déduire $begin(2, P \wedge Q)$, et aussi $\{[2, 4], P \wedge Q\}$ si l'on suppose que Q dure au moins jusqu'au temps 4, c'est à dire en supposant $persist(2, 4, Q)$. Ce mécanisme reste implantable dans le cadre d'un graphe à la RETE tel qu'il a été décrit.

Pour le connecteur \vee , une sémantique possible est :

$$I, V_t \models \{[t, t'], p \vee q\} \text{ ssi } \exists i \in I(p) \exists j \in I(q) \ i \cap j \neq \emptyset \text{ et } i \cup j = I_t([t, t'])$$

Néanmoins, il serait souhaitable d'étendre cette définition en considérant deux séquences d'intervalles respectivement de $I(p)$ et de $I(q)$, telles que l'union de tous les intervalles soit égale à $I_t([t, t'])$. Cette définition est plus difficilement mécanisable que la précédente, car la génération des fait qualifiant $p \vee q$ ne peut pas se faire uniquement à partir de deux faits portant respectivement sur p et sur q .

On peut ensuite vouloir introduire des quantificateurs et des variables non-temporelles [17]. Le problème qui survient est que l'ensemble des individus sur lequel on quantifie peut être variable en fonction du temps.

6 Conclusions

Le mécanisme décrit ici, bien qu'il n'admette pour le moment que des théories très restreintes, a l'avantage d'être simple et robuste. L'implémentation en est aisée et peut faire appel à des techniques améliorant l'efficacité comme la compilation des règles en un graphe à la RETE. L'aspect "stratégique" du raisonnement (choix de contexte, d'inférence) n'a pas été approfondi ici. Il reste beaucoup d'aspects qui n'ont pas été encore explorés, et qui seront utiles pour qu'un tel mécanisme devienne utilisable sur des problèmes réels.

Bibliographie

- [1] James F. Allen. Towards a general theory of action and time. *Artificial Intelligence*, 23(2):123–154, 1984.
- [2] J. de Kleer. An assumption-based tms. *Artificial Intelligence*, 28:127–162, 1986.
- [3] Thomas L. Dean and Mark Boddy. Reasoning about partially ordered events. *Artificial Intelligence*, 36:375–399, 1988.
- [4] Thomas L. Dean and Drew V. McDermott. Temporal data base management. *Artificial Intelligence*, 32(1):1–55, 1987.
- [5] Charles L. Forgy. Rete: A fast algorithm for the many pattern/many object pattern match problem. *Artificial Intelligence*, 19:17–37, 1982.
- [6] H. Freitag and M. Reinfrank. A non-monotonic deduction system based on (a)tms. In *Proc. of the 8th ECAI*, pages 601–606, Munich, 1988.
- [7] Steve Hanks and Drew V. McDermott. Nonmonotonic logic and temporal projection. *Artificial Intelligence*, 33:379–412, 1987.
- [8] R. Kowalski and M. Sergot. A logic-based calculus of events. *New Generation Computing*, 4:67–95, 1986.
- [9] S. Lorenz. Nonmonotonic temporal reasoning: Persistence, justified causation and event minimization. In Z. W. Ras and M. Zemankova, editors, *Methodologies for Intelligent Systems: Proc. of the 6th International Symposium ISMIS'91*, pages 540–549. Springer, Berlin, Heidelberg, 1991.

- [10] Jalal Maleki. Towards a system for temporal reasoning. In *Proc. of the Scandinavian Conference on AI (SCAI'89)*, pages 59–68, 1989.
- [11] D. McDermott. A general framework for reason maintenance. *Artificial Intelligence*, 50:289–329, 1991.
- [12] Drew V. McDermott. A temporal logic for reasoning about processes and plans. *Cognitive Science*, 6:101–155, 1982.
- [13] A. Nakhimovsky. The lexicon, grammatical categories and temporal reasoning. In *Proc. of the 1987 AISB Conference*, pages 35–48, University of Edinburgh, 1987.
- [14] Erik Sandewall. Non-monotonic entailment for reasoning about time and action Part I : Sequential actions. Research Report LiTH-IDA-R-88-27, Linköping University, September 1988.
- [15] Erik Sandewall. Non-monotonic entailment for reasoning about time and action Part III : Decision procedure. Research Report LiTH-IDA-R-88-29, Linköping University, September 1988.
- [16] Murray Shanahan. Prediction is deduction but explanation is abduction. In *Proc. of the 11th IJCAI*, pages 1055–1060, 1989.
- [17] Yoav Shoham. Temporal logics in AI: Semantical and ontological considerations. *Artificial Intelligence*, 33:89–104, 1987.

A Les appariements

Cette annexe regroupe les tables décrivant les appariements réalisés entre les antécédents des règles causales, et les faits de la base de faits. Les variables commençant par e désignent des expressions d'instant, qui apparaissent dans les antécédents, alors que les variables en t désignent des constantes temporelles apparaissant dans les faits.

antécédent	fait	équations	inéquations	preuve	
$\{e, p\}$ $p \in \mathcal{P}_e$	$\{t, p\}$	$e = t$			
$\{e, p\}$ $p \in \mathcal{P}_s$	$\{[t_1, t_2], p\}$		$t_1 \leq e \leq t_2$	$\frac{\{[t_1, t_2], p\}}{\{e, p\}}$	
	$\{t, p\}$	$e = t$			
			$e < t$	$\frac{\{t, p\} \text{ persist}(e, t, p)^*}{\{e, p\}}$	
			$e > t$	$\frac{\{t, p\} \text{ persist}(t, e, p)^*}{\{e, p\}}$	
	$\text{begin}(t, p)$	$e = t$		$\frac{\text{begin}(t, p)}{\{t, p\}}$	
			$e > t$	$\frac{\text{begin}(t, p) \text{ persist}(t, e, p)^*}{\{e, p\}}$	
	$\text{end}(t, p)$	$e = t$		$\frac{\text{end}(t, p)}{\{t, p\}}$	
			$e < t$	$\frac{\text{end}(t, p) \text{ persist}(e, t, p)^*}{\{e, p\}}$	
	$\text{persist}(t_1, t_2, p)$		$t_1 \leq e \leq t_2$	$\frac{\text{persist}(t_1, t_2, p)}{\{e, p\}}$	
	$\{[e_1, e_2], p\}$ $p \in \mathcal{P}_d \cup \mathcal{P}_s$	$\{[t_1, t_2], p\}$	$e_1 = t_1$ $e_2 = t_2$		
$\text{begin}(t, p)$		$e_1 = t$	$e_2 > t$	$\frac{\text{begin}(t, p) \text{ persist}(t, e_2, p)^* \text{end}(e_2, p)^*}{\{[t, e_2], p\}}$	
$\text{end}(t, p)$		$e_2 = t$	$e_1 < t$	$\frac{\text{begin}(e_1, p)^* \text{persist}(e_1, t, p)^* \text{end}(t, p)}{\{[e_1, t], p\}}$	
$\text{persist}(t_1, t_2, p)$		$e_1 = t_1$ $e_2 = t_2$		$\frac{\text{begin}(t_1, p)^* \text{persist}(t_1, t_2, p) \text{end}(t_2, p)^*}{\{[t_1, t_2], p\}}$	
$\text{begin}(e, p)$ $p \in \mathcal{P}_d \cup \mathcal{P}_s$	$\{[t_1, t_2], p\}$	$e = t_1$		$\frac{\{[t_1, t_2], p\}}{\text{begin}(t_1, p)}$	
	$\text{begin}(t, p)$	$e = t$			
$\text{end}(e, p)$ $p \in \mathcal{P}_d \cup \mathcal{P}_s$	$\{[t_1, t_2], p\}$	$e = t_2$		$\frac{\{[t_1, t_2], p\}}{\text{end}(t_2, p)}$	
	$\text{end}(t, p)$	$e = t$			
$\text{persist}(e_1, e_2, p)$ $p \in \mathcal{P}_d \cup \mathcal{P}_s$	$\{[t_1, t_2], p\}$	$e_1 = t_1$ $e_2 = t_2$		$\frac{\{[t_1, t_2], p\}}{\text{persist}(t_1, t_2, p)}$	
			$t_1 < e_1$ $e_1 < e_2$ $e_2 < t_2$	$\frac{\{[t_1, t_2], p\}}{\text{persist}(e_1, e_2, p)}$	
	$\text{persist}(t_1, t_2, p)$	$e_1 = t_1$ $e_2 = t_2$			
			$t_1 < e_1$ $e_1 < e_2$ $e_2 < t_2$	$\frac{\text{persist}(t_1, t_2, p)}{\text{persist}(e_1, e_2, p)}$	

Figure 4 : les appariements entre littéraux positifs.

antécédent	fait	équations	inéquations	preuve	
$\neg\{e, p\}$ $p \in \mathcal{P}_e$	$\neg\{t, p\}$	$e = t$			
$\neg\{[e_1, e_2], p\}$ $p \in \mathcal{P}_d \cup \mathcal{P}_s$	$\neg\{[t_1, t_2], p\}$	$e_1 = t_1$ $e_2 = t_2$			
	$begin(t, p)$		$e_1 < t \leq e_2$	$\frac{begin(t, p)}{\neg\{[e_1, e_2], p\}}$	
	$end(t, p)$		$e_1 \leq t < e_2$	$\frac{end(t, p)}{\neg\{[e_1, e_2], p\}}$	
	$\{[t_1, t_2], p\}$			$e_1 < t_1 < t_2 < e_2$	$\frac{\{[t_1, t_2], p\}}{\neg\{[e_1, e_2], p\}}$
				$e_1 < t_1 \leq e_2$	$\frac{\{[t_1, t_2], p\}}{\neg\{[e_1, e_2], p\}}$
				$e_1 \leq t_2 < e_2$	$\frac{\{[t_1, t_2], p\}}{\neg\{[e_1, e_2], p\}}$
	$\neg begin(t, p)$	$e_1 = t$	$e_2 \geq t$	$\frac{\neg begin(t, p)}{\neg\{[t, e_2], p\}}$	
	$\neg end(t, p)$	$e_2 = t$	$e_1 \leq t$	$\frac{\neg end(t, p)}{\neg\{[e_1, t], p\}}$	
$\neg persist(t_1, t_2, p)$		$e_1 \leq t_1 \leq t_2 \leq e_2$	$\frac{\neg persist(t_1, t_2, p)}{\neg\{[e_1, e_2], p\}}$		
$\neg\{e, p\}$ $p \in \mathcal{P}_s$	$\neg\{t, p\}$	$e = t$			
$\neg begin(e, p)$ $p \in \mathcal{P}_d \cup \mathcal{P}_s$	$\neg begin(t, p)$	$e = t$			
	$\{[t_1, t_2], p\}$		$t_1 < e \leq t_2$	$\frac{\{[t_1, t_2], p\}}{\neg begin(e, p)}$	
	$persist(t_1, t_2, p)$		$t_1 < e \leq t_2$	$\frac{persist(t_1, t_2, p)}{\neg begin(e, p)}$	
$\neg end(e, p)$ $p \in \mathcal{P}_d \cup \mathcal{P}_s$	$\neg end(t, p)$	$e = t$			
	$\{[t_1, t_2], p\}$		$t_1 \leq e < t_2$	$\frac{\{[t_1, t_2], p\}}{\neg end(e, p)}$	
	$persist(t_1, t_2, p)$		$t_1 \leq e < t_2$	$\frac{persist(t_1, t_2, p)}{\neg end(e, p)}$	
$\neg persist(e_1, e_2, p)$ $p \in \mathcal{P}_d \cup \mathcal{P}_s$	$begin(t, p)$		$e_1 < t \leq e_2$	$\frac{begin(t, p)}{\neg persist(e_1, e_2, p)}$	
	$end(t, p)$		$e_1 \leq t < e_2$	$\frac{end(t, p)}{\neg persist(e_1, e_2, p)}$	
	$\{[t_1, t_2], p\}$			$e_1 < t_1 \leq e_2$	$\frac{\{[t_1, t_2], p\}}{\neg persist(e_1, e_2, p)}$
				$e_1 \leq t_2 < e_2$	$\frac{\{[t_1, t_2], p\}}{\neg persist(e_1, e_2, p)}$
	$\neg persist(t_1, t_2, p)$		$e_1 \leq t_1 \leq t_2 \leq e_2$	$\frac{\neg persist(t_1, t_2, p)}{\neg persist(e_1, e_2, p)}$	

Figure 5 : les appariements avec les antécédents négatifs.