

DOMINIQUE MERY

**Méthode axiomatique sur les propriétés de fatalité
des programmes parallèles**

RAIRO. Informatique théorique et applications, tome 21, n° 3 (1987),
p. 287-322

http://www.numdam.org/item?id=ITA_1987__21_3_287_0

© AFCET, 1987, tous droits réservés.

L'accès aux archives de la revue « RAIRO. Informatique théorique et applications » implique l'accord avec les conditions générales d'utilisation (<http://www.numdam.org/conditions>). Toute utilisation commerciale ou impression systématique est constitutive d'une infraction pénale. Toute copie ou impression de ce fichier doit contenir la présente mention de copyright.

NUMDAM

Article numérisé dans le cadre du programme
Numérisation de documents anciens mathématiques

<http://www.numdam.org/>

MÉTHODE AXIOMATIQUE UR LES PROPRIÉTÉS DE FATALITÉ S PROGRAMMES PARALLÈLES (*)

par Dominique MERY ⁽¹⁾

Communiqué par J.-E. PIN

Résumé. – L'étude des propriétés de fatalité sous hypothèse d'exécution équitable nous a permis de construire un système axiomatique basé sur les opérateurs de la logique temporelle. Ce système prend en compte l'hypothèse d'équité et contient une règle d'induction sur le nombre de processus concurrents. Nous restreignons notre étude au cas des programmes parallèles communicants par variables partagées et synchronisés par des sémaphores.

Abstract. – Eventuality properties under fairness hypothesis have led us to design a temporal-based proof system. This proof system takes into account the fairness hypothesis and contains an induction rule relative to the number of concurrent processes. We limit our investigations to concurrent programs sharing common variables and using semaphores as synchronization primitives.

I. INTRODUCTION

Nous présentons dans cet article un système axiomatique basé sur la logique temporelle. Une des caractéristiques de ce système est de permettre la preuve de propriétés de fatalité (correction totale, accessibilité en section critique, . . .), en y incluant implicitement l'hypothèse d'équité. Notre travail est basé sur les propositions faites par Owicki et Lamport [1982], qui caractérisent les hypothèses d'équité faible (ou justice) et d'équité forte (ou équité) à l'aide de quelques formules temporelles très simples. Ces formules permettent d'éliminer les traces d'exécutions non justes et/ou non équitables. Nous

(*) Reçu avril 1986, révisé en février 1987.

(¹) Laboratoire de Recherche en Informatique de Metz, Université de Metz, Faculté des Sciences, Ile-du-Saulcy, 57045 Metz Cedex et Centre de Recherche en Informatique de Nancy, B.P. n° 239, 54506 Vandœuvre-les-Nancy.

exploitons le formalisme temporel développé par Pnueli [1977]. Manna et Pnueli [1982], [1983 *a*], [1983 *b*], [1984] et quelques opérateurs temporels nous ont paru indispensables pour notre étude. Ces auteurs présentent plusieurs principes d'induction exprimés dans le formalisme temporel mais la forme d'expression de ces principes est un mélange de formel et d'informel, ce qui peut paraître insatisfaisant, vu le choix du formalisme temporel. Notre travail s'inscrit dans le cadre de la sémantique axiomatique des langages de programmation et propose une méthode axiomatique offrant une alternative axiomatique à l'alternative sémantique de Lehman, Pnueli et Stavi [1981] et à l'alternative sémantico-axiomatique de Manna et Pnueli [1982], [1983 *a*], [1983 *b*], [1984] pour le même type de propriétés. Cette méthode s'apparente clairement à celle de Burstall [1974] axiomatisée par Apt et Delporte [1983] pour le cas des programmes séquentiels uniquement où aucune hypothèse d'équité n'est utile. Il faut préciser que la méthode due à Lehman, Pnueli et Stavi [1981] ne concernait que la preuve de terminaison d'une classe de programmes.

Rappelons que l'hypothèse d'équité est caractérisé par un ensemble de traces d'exécution et que Park [1981] propose deux plus faibles préconditions «à la Dijkstra» associées respectivement à la justice (ou équipé faible) et à l'équité (ou équité forte). Nous avons choisi de caractériser les propriétés de fatalité d'un point de vue sémantique, à l'aide d'une démarche similaire, qui consiste à construire la «plus grande assertion» conduisant à une assertion donnée sous hypothèse d'exécutions équitables. La dérivation des propriétés de fatalité (ou d'inévitabilité) est réalisée par l'emploi d'une fonction d'aide à la convergence dans le travail de Manna et Pnueli [1982] et cette technique généralise celle de Floyd [1967] à ce type de situation. Pour notre part, nous avons été contraints de dégager une notion «plus syntaxique» permettant les preuves de fatalité dans le cadre de l'équité car nous devons, par la suite, proposer des règles spécifiques. Cette notion sous-jacente «plus syntaxique» est celle de groupe critique d'actions atomiques (concurrentes). Cette notion est clairement mise en évidence sur les exemples standards. Par contre, elle est très difficile à caractériser, car nous avons du utiliser un plus grand point fixe comme Park [1981]. Une étude sémantique des propriétés de fatalité dans le cadre de l'équité nous guide dans la construction d'un système axiomatique correct et sémantiquement complet que nous appelons FEPS (pour Fair Eventuality Proof System). Notre système FEPS est exprimé dans le symbolisme de la logique temporelle, mais ce symbolisme doit être considéré comme un support syntaxique à la méthode et, de ce fait, nous n'envisageons aucune solution à l'expressivité de certaines assertions intermédiaires, si ce n'est d'un point de vue ensembliste. FEPS est quelque peu dirigé par la syntaxe, mais

restreint à un caractère opérationnel du à la sémantique de base et au parallélisme. Notre propos est illustré par des exemples spécifiques montrant l'intérêt des notions introduites. Jusqu'à présent, nous avons négligé de préciser la classe des programmes considérés. Nous avons voulu nous limiter, ici, à des programmes séquentiels synchronisés par des sémaphores implantés comme des attentes actives équitables. Dans ce cadre-là, notre approche est une définition de la sémantique des programmes synchronisés par des sémaphores équitables. Nous présentons le plan de notre travail.

La deuxième partie présente les éléments fondamentaux permettant de donner le cadre formel de cette étude, notamment la syntaxe et la sémantique des programmes ainsi que les opérateurs temporels nécessaires. Dans la troisième partie, nous étudions les propriétés de fatalité sous hypothèse d'exécution équitable et nous caractérisons les éléments déterminants pour la convergence dans le cadre de l'équité: les groupes critiques d'actions atomiques concurrentes. Le système axiomatique FEPS est construit progressivement dans la quatrième partie et on aborde simultanément sa correction et sa complétude sémantique. La partie 5 contient des exemples pour illustrer ce système. Les parties suivantes sont destinées à la conclusion et aux références.

II. SYNTAXE ET SÉMANTIQUE DES PROGRAMMES

Dans un premier temps, nous décrirons la classe des programmes considérés que nous dénommerons classe des programmes parallèles et que nous définirons comme étant la classe des programmes séquentiels non déterministes et contenant les programmes constitués d'un ensemble de programmes (ou processus) séquentiels non-déterministes:

COBEGIN $P_1 // \dots // P_n$ COEND.

Les programmes séquentiels communiquent par variables partagées et sont synchronisés à l'aide de sémaphores ne disposant pas d'une file d'attente explicite. Les sémaphores sont, en fait, des attentes actives équitables. Nous noterons classiquement $P(S)$ et $V(S)$ les opérations sur le sémaphore S : $P(S)$ est équivalent à l'instruction l : $\langle \text{if } S=0 \text{ then goto } l \text{ else } S := S-1 \text{ fi} \rangle$ et $V(S)$ est équivalent à $\langle S := S+1 \rangle$. Les variables de tout programme parallèle, y compris les sémaphores, sont initialisées juste avant le début du programme. Afin de repérer le flût du contrôle à l'exécution, nous utilisons des étiquettes placées devant chaque instruction du programme considéré, à la fin de chaque processus constituant le programme ainsi qu'au début et après le programme. Considérons l'exemple suivant

```

P: = true;
0:
COBEGIN
l1: WHILE P DO l2: SKIP l3: OD; l4:
//
m1: P = false; m2;
COEND
1:

```

On notera \mathbb{P} un programme quelconque de notre classe et $\mathbb{P}_1, \dots, \mathbb{P}_n$, processus constituant \mathbb{P} , S, S_1, S_2, S' désigneront des instructions de \mathbb{P} .

Nous modélisons l'exécution d'un programme \mathbb{P} par le biais d'une sémantique opérationnelle classique qui simule l'exécution des processus de \mathbb{P} par entrelacement non déterministe. Une sémantique opérationnelle est un 4-uplet (S, l, t, T) où

1. S est l'ensemble des états de \mathbb{P} : un état σ est un couple formé d'une partie contrôle l qui spécifie les instructions courantes et une partie mémoire m qui est une fonction des variables et des sémaphores vers la mémoire; on notera $\sigma = (l, m)$.

2. $l \subset S$ est l'ensemble des états initiaux.

3. t est une relation sur S qui simule l'exécution pas-à-pas et exclusive de chaque action atomique choisie parmi les actions atomiques courantes activables.

4. T est l'ensemble des traces induites par t sur S à partir de l .

La notion d'action atomique détruit la structuration utile du programme mais permet de manipuler plus facilement les unités constituant le programme. Les actions atomiques d'un programme \mathbb{P} sont :

1. $\langle V := E \rangle$ pour l'affectation.
2. $\langle V := ? \rangle$ pour l'affectation non déterministe.
3. $\langle \text{SKIP} \rangle$ pour le saut.
4. $\langle B \rangle$ pour le test de la conditionnelle et de l'itération.
5. $\langle \text{ENDIFT} \rangle$ et $\langle \text{ENDIFF} \rangle$ pour la fin du corps correspondant respectivement au test vrai et au test faux.
6. $\langle \text{START} \rangle$ pour le positionnement simultané du contrôle devant chaque processus.
7. $\langle \text{FINISH} \rangle$ pour le positionnement simultané du contrôle juste après le programme \mathbb{P} .

Afin de raisonner sur les propriétés d'un programme \mathbb{P} , nous allons définir des opérateurs temporels classiques mais nous devons préciser le langage d'assertions \mathbb{A} . Une assertion de contrôle est une des assertions suivantes :

at \mathbb{S} , in \mathbb{S} , after \mathbb{S} où \mathbb{S} est une assertion quelconque du programme \mathbb{P} . Intuitivement, at \mathbb{S} est vrai dans l'état courant, si l'étiquette devant \mathbb{S} est courante dans l'état courant. Ce qui revient à exprimer que le contrôle est juste au début de \mathbb{S} , after \mathbb{S} est vrai si le contrôle courant se trouve juste après \mathbb{S} , ce qui signifie que \mathbb{S} est terminée. Enfin, in \mathbb{S} signifie que \mathbb{S} n'est pas terminée et que le contrôle courant est à l'intérieur de \mathbb{S} . Pour le cas d'une action atomique a , ina et ata sont équivalents. Ce type d'assertions a été introduit par Lamport [1980 b] et permet de suivre l'exécution du programme. Le langage d'assertions \mathbb{A} contient les assertions classiques de contrôle précitées, les assertions qui portent sur l'état des variables et toutes les bonnes combinaisons de ces assertions et des connecteurs \wedge (et), \vee (ou), \forall (ou généralisé) et \sim (non). Afin d'écartier pour cet article les problèmes liés à l'expression de toute propriété Prop nécessaire pour notre étude dans \mathbb{A} , nous supposons que \mathbb{A} est suffisamment expressif. En fait, nous associons un ensemble d'états à toute assertion p qui est celui où p est vraie pour tout état de cet ensemble et réciproquement si nous avons besoin d'une assertion pour caractériser un ensemble d'états nous en disposons d'une. On note cond(a) la condition d'activation de l'instruction atomique a . Une interprétation \mathbb{J} pour un programme \mathbb{P} est un 4-uplet $(\mathbb{S}, \mathbb{I}, t, \mathbb{T}')$ où \mathbb{S} , \mathbb{I} , t sont définis comme précédemment et où \mathbb{T}' est une partie de \mathbb{T} . Nous utilisons 3 interprétations :

1. $\mathbb{S}\mathbb{J}$ où $\mathbb{T} = \mathbb{T}'$, c'est l'interprétation classique et Standard sans hypothèse sur l'exécution du programme \mathbb{P} .
2. $\mathbb{J}\mathbb{J}$ où \mathbb{T}' est l'ensemble des traces justes, c'est l'interprétation juste ("JUST").
3. $\mathbb{F}\mathbb{J}$ où \mathbb{T}' est l'ensemble des traces équitables ("FAIR"), c'est l'interprétation équitable.

On notera « $\mathbb{J}, \sigma \mid = \text{Prop}$ » l'interprétation vrai de Prop dans \mathbb{J} en l'état σ et lorsque \mathbb{J} sera implicite nous noterons Prop(σ). Nous allons tout d'abord donner une définition de l'opérateur O dans le cadre d'une interprétation \mathbb{J} . Il est clair que la relation t joue un rôle central dans le sens de O mais nous définirons la relation t de transition à l'aide de cet opérateur, c'est-à-dire que, nous caractérisons la relation t par les propriétés temporelles en terme de l'opérateur O . Cette démarche est comparable à la sémantique axiomatique à l'aide d'axiomes «à la Hoare». Une autre alternative pour définir t est d'utiliser les étiquettes et des équations sur le contexte comme cela est fait par Cousot R. [1985]. On rappelle qu'on se donne un programme \mathbb{P} de la classe choisie.

DÉFINITION II. 1 : Soient p, q deux assertions, σ un état, \mathbb{J} une interprétation.

1. $\mathbb{J}, \sigma \mid = O p$. Si $\forall \sigma' \in \mathbb{S}, t(\sigma, \sigma') \Rightarrow p(\sigma')$.

On dit que p est vrai dans tous les états successeurs de σ .

2. $\mathbb{J}, \sigma \models p \Rightarrow q$, si $\sim p(\sigma) \vee q(\sigma)$ est vraie. ■

Notre opérateur O ne présume pas de l'exécution d'une ou d'une autre action atomique, mais il va nous permettre de spécifier la relation de transition. De plus, il est clair que si au moins une action atomique est activable, au moins une est activée. Afin de donner une explication globale, nous précisons que ces assertions sont à assimiler à des axiomes «à la Hoare» et que le schéma général de raisonnement utilise la notion d'action atomique. En fait, la transition élémentaire de t simule l'exécution d'une action a et la description opérationnelle de t revient à une description de l'effet de a . Pour cela, nous utilisons les assertions de contrôle ata et $after a$ intuitivement définies, mais suffisamment décrites ci-dessus. Ainsi on écrira des propriétés du type :

$$\mathbb{J} \models at a \wedge p_a \Rightarrow O(after a \Rightarrow q_a) \gg$$

que l'on interprète comme suit :

pour tout état tel que a est une action courante et p_a est vraie, alors l'exécution de a à partir de cet état conduit à un état validant q_a , mais l'expression «after $a \Rightarrow q_a$ » est vraie, si a n'est pas exécutée juste après, car, dans ce cas, $at a$ est vraie et $after a$ est faux.

Nous avons choisi ce type de définition car elle est, en fait, une sémantique temporelle des programmes parallèles considérés et est plus formelle que l'expression utilisée par Manna et Pnueli [1982], [1983 a], [1983 b] et [1984]. Nous donnons dans la définition qui suit toutes les transitions élémentaires possibles par cas syntaxiques. On notera que l'entrelacement non déterministe des différents processus est implicite à cause de ces propriétés.

DÉFINITION II.2 : Soit \mathbb{J} l'une des trois interprétations $\mathbb{S}\mathbb{J}$, $\mathbb{J}\mathbb{J}$, $\mathbb{F}\mathbb{J}$. La relation de transition t de \mathbb{J} est la réunion des relations r_i associées à chaque propriété T_i ci-dessous et définies comme suit : si T_i s'écrit sous la forme $at\mathbb{S} \wedge p \Rightarrow O(after \mathbb{S} \Rightarrow q)$, alors

$$r_i = \{ (\sigma, \sigma') \in \mathbb{S} \times \mathbb{S} / p(\sigma) \text{ et } at \mathbb{S}(\sigma) \text{ et } (\mathbb{J}, \sigma \models at \mathbb{S} \wedge p \Rightarrow O(after \mathbb{S} \Rightarrow q)) \}$$

et r_i est la plus grande relation vérifiant cette propriété.

T1 : $at (V := E) \wedge p[e/v] \Rightarrow O(after (V := E) \Rightarrow p)$ où $p[e/v]$ désigne l'assertion obtenue en substituant les occurrences libres de v par l'expression e .

T2 : $\forall c \in \text{Domaine}(V). at (V := ?) \wedge p[c/v] \Rightarrow O(after (V := ?) \Rightarrow p)$.

T3: at $\text{SKIP} \wedge p = > O$ (after $\text{SKIP} = > p$).

T4: at $(\text{if } B \text{ then } S1 \text{ else } S2 \text{ fi}) \wedge p \wedge b = > O$ (at $S1 = > p \wedge b$).

T5: at $(\text{if } B \text{ then } S1 \text{ else } S2 \text{ fi}) \wedge p \wedge \sim b = > O$ (at $S2 = > p \wedge \sim b$).

T6: after $S1 \wedge p = > O$ (after $(\text{if } B \text{ then } S1 \text{ else } S2 \text{ fi}) = > p$ }.

T7: after $S2 \wedge p = > O$ (after $(\text{if } B \text{ then } S1 \text{ else } S2 \text{ fi}) = > p$ }.

T8: at $(\text{while } B \text{ do } S \text{ od}) \wedge p \wedge b = > O$ (at $S = > p \wedge b$).

T9: at $(\text{while } B \text{ do } S \text{ od}) \wedge p \wedge \sim b$

$\Rightarrow O$ (after $(\text{while } B \text{ do } S \text{ od}) = > p \wedge \sim b$).

(Dans T10 et T11, S est le corps de la boucle $\text{while } B \text{ do } S \text{ od}$)

T10: after $S \wedge p \wedge b = > O$ (at $S = > p \wedge b$).

T11: after $S \wedge p \wedge \sim b = > O$ (after $(\text{while } B \text{ do } S \text{ od}) = > p \wedge \sim b$).

T12: at $\text{COBEGIN } \mathbb{P}1 // \dots // \mathbb{P}n \text{ COEND} \wedge p$

$\Rightarrow O$ (at $\mathbb{P}1 \wedge \dots \wedge \text{at } \mathbb{P}n = > p$).

T13: after $\mathbb{P}1 \wedge \dots \wedge \text{after } \mathbb{P}n \wedge p$

$\Rightarrow O$ (after $\text{COBEGIN } \mathbb{P}1 // \dots // \mathbb{P}n \text{ COEND} = > p$).

Soit S une variable de sémaphore et s la valeur de S .

T14: at $V(S) \wedge p[s+1/s] = > O$ (after $V(S) = > p$).

T15: at $P(S) \wedge p[s-1/s] \wedge (s > 0) = > O$ (after $P(S) = > p$). ■

La définition de l'ensemble des traces induites par t sur \mathbb{S} , noté \mathbb{T} , n'a pas été précisément donnée. Nous donnons cette définition dans ce qui suit puis nous abordons celle d'opérateurs liés à la fatalité.

DÉFINITION II. 3 : Soit \mathbb{S} un ensemble, \mathbb{I} une partie de \mathbb{S} et t une relation sur \mathbb{S} . L'ensemble \mathbb{T} des traces induites par t sur \mathbb{S} relativement à \mathbb{I} est le plus petit ensemble vérifiant :

- $\mathbb{T} \subset \mathbb{S}^{00}$.
- $\mathbb{I} \subset \mathbb{T}$.
- $\forall w \in \mathbb{T}. (\forall i \in \{1, \dots, |w| - 1\}, t(w(i), w(i+1)))$
 $\wedge ((w(1) \in \mathbb{I}) \vee (\exists \sigma \in \mathbb{I}. t^*(\sigma, w(1))))$ ■

Nous notons t^* la clôture réflexive transitive de la relation t . \leq désignera l'ordre partiel de \mathbb{S}^{00} , ensemble des mots sur \mathbb{S} . On notera \mathbb{T}' , une partie de \mathbb{T} et $\mathbb{T}'(\sigma)$ est l'ensemble des mots de \mathbb{T}' commençant par σ et cet ensemble

est vide, si σ n'est accessible à partir de \perp . Afin de simplifier l'expression, nous utilisons la notation « $w[a, b]$ » pour désigner les états $w(a)$, $w(a+1)$, \dots , $w(b)$. On note $\text{dom}(w) = \{1, \dots, |w|\}$, le domaine de w .

DÉFINITION II.4 : 1. $\mathbb{J}, \sigma \models \square p$ ssi $\forall \sigma' \in \mathbb{S}, t^*(\sigma, \sigma') = \Rightarrow p(\sigma')$.

p est toujours vraie à partir de σ .

2. $\mathbb{J}, \sigma \models p \mathcal{U} q$ ssi $\forall w \in \mathbb{T}'(\sigma)$.

$$(\forall i \in \text{dom}(w). p(w(i))) \vee (\exists i \in \text{dom}(w). p(w[1, i-1]) \wedge q(w(i))).$$

p est toujours vraie à partir de σ à moins que q ne soit vraie.

3. $\mathbb{J}, \sigma \models p - [\] \rightarrow q$ ssi

$p(\sigma) = \Rightarrow \forall w \in \mathbb{T}'(\sigma). \exists w' \in \mathbb{T}'(\sigma)$.

$$(w \leq w') \wedge (\exists i \in \text{dom}(w'). p(w'[1, i-1]) \wedge q(w'(i))).$$

Si p est vraie en σ , alors p est vraie jusqu'à ce que q soit vraie.

4. $\mathbb{J}, \sigma \models p \rightarrow q$ ssi

$$p(\sigma) = \Rightarrow \forall w \in \mathbb{T}'(\sigma). \exists w' \in \mathbb{T}'(\sigma). (w \leq w') \wedge (\exists i \in \text{dom}(w'). q(w'(i))).$$

Si p est vraie en σ , alors q est fatalement vraie ou p conduit fatalement à q dans l'interprétation \mathbb{J} . ■

Au lieu d'écrire « $\forall \sigma \in \mathbb{S}. \mathbb{J}, \sigma \models \text{Prop}$ », on écrira « $\mathbb{J} \models \text{Prop}$ » ou encore « $\models \text{Prop}$ », si \mathbb{J} est implicite. Notre interprétation est arborescente puisque l'on quantifie sur les traces d'exécution, ce qui n'est pas le cas d'Owicki et Lamport [1982]. Nous pouvons exprimer des propriétés de fatalité sous certaines hypothèses :

(1) $\mathbb{F}\mathbb{J} \models p \rightarrow q$: p conduit à q sous hypothèse d'exécutions équitables.

(2) $\mathbb{J}\mathbb{J} \models p \rightarrow q$: p conduit à q sous hypothèse d'exécutions justes.

Avant de poursuivre, nous signalons que nous utilisons aussi bien le langage ensembliste que le langage de la logique, lorsque nous raisonnons sur les assertions. De ce fait, nous ne distinguons pas l'assertion de l'ensemble des états la validant.

III. PROPRIÉTÉS DE FATALITÉ

Dans la partie précédente, nous avons donné les éléments mathématiques pour étudier les propriétés de fatalité. Dans cette partie, nous voulons caractériser les propriétés de fatalité dans le cadre d'une exécution équitable. Une

exécution équitable est une exécution en rapport avec l'interprétation \mathbb{FJ} se distinguant de l'interprétation standard \mathbb{SJ} par la restriction faite sur l'ensemble des traces, qui est une des composantes de définition de \mathbb{SJ} et la seule permettant de distinguer \mathbb{FJ} et \mathbb{SJ} . L'ensemble des traces est indispensable pour définir la fatalité, contrairement à la notion de possibilité qui est une quantification existentielle sur les traces et sur les états. Cette différence est due au caractère arborescent de notre interprétation et cette différence ne peut être comblée à l'aide de la relation t uniquement. Nous n'avons toujours pas défini intuitivement et formellement les notions de traces justes et de traces équitables.

Une trace w de \mathbb{T} est une trace juste, si w est une trace finie, ou bien, si, dans le cas où w est infinie, toute action atomique continuellement activable à partir d'un état de w est activée en fin de compte. Cette notion est plus simple à définir par restriction, c'est-à-dire, que la définition d'une trace non juste est plus simple. Une trace non juste w de \mathbb{T} est une trace infinie telle qu'il existe une action atomique a et un état σ de w tels que a est continuellement activable à partir de σ et jamais activée.

Considérons le programme cité au début de la partie I. L'action atomique $\langle P := \text{false} \rangle$ peut être retardée indéfiniment et la trace construite est alors non juste et, dans ce cas, on souhaite que l'interprétation valide la formule :

$$\ll \text{at} \langle P := \text{false} \rangle \rightarrow \text{after} \langle P := \text{false} \rangle \gg$$

De la même façon, on peut définir la notion de trace équitable plus adaptée au traitement des sémaphores. Une trace w de \mathbb{T} est équitable, si w est finie, ou, si, dans le cas où w est infinie, toute action infiniment souvent exécutable sur w (i.e. la condition d'activation devenant infiniment souvent vraie) est activée. Owicki et Lamport [1982] ont proposé une caractérisation des traces justes et des traces équitables à l'aide de la relation « \rightarrow ». Ainsi, l'ensemble des traces justes est le plus grand ensemble de traces validant les formules du type « $\text{at } a \rightarrow \text{after } a$ » pour toute a . En effet, la formule « $\text{at} \langle P := \text{false} \rangle \rightarrow \text{after} \langle P := \text{false} \rangle$ » n'est pas vérifiée, si la trace est non juste. L'idée de caractériser la justice ou l'équité de cette façon est intéressante, mais ne permet pas de construire un système complet. En effet, nous verrons que nous avons besoin d'exprimer la première occurrence de l'activation de l'action a depuis son attente. Dans l'exemple choisi, $\langle P := \text{false} \rangle$ est exécutée une seule fois, mais en règle générale une expression du type « $\text{at } a \rightarrow \text{after } a$ » n'exprime pas que, dès que $\text{after } a$ est vraie, il s'agit alors de la première activation depuis $\text{at } a$. Pour cela, nous exprimons cela à

l'aide des opérateurs « $\mathbf{Z}/12\mathbf{Z}$ » comme suit :

$$\text{at } a\mathbf{Z}/12\mathbf{Z} \text{ after } a$$

ou

$$\text{at } P := \text{false} \wedge p\mathbf{Z}/12\mathbf{Z} \text{ after } P = \text{false} \wedge \sim p.$$

Nous donnons une définition formelle des traces justes et équitables.

DÉFINITION III. 1 : 1. Une trace w de \mathbb{T} est non-juste, si :

$$- /w/ = \infty;$$

$$- \exists a. \exists i \in \text{dom}(w). \forall k \geq i. \text{cond}(a) \wedge \text{at } a(w(k)).$$

2. Une trace w de \mathbb{T} est non-équitable, si

$$- /w/ = \infty;$$

$$- \exists a. \exists i \in \text{dom}(w). \exists (i_1, \dots, i_j, \dots) \in \text{dom}(w)^\infty.$$

$$(i_1 = i) \wedge \text{cond}(a)(w(i_1), \dots, w(i_j), \dots, w(i_j), \dots) \wedge (\forall k \geq i. (\text{at } a)(w(k))). \blacksquare$$

Une trace non juste est non équitable. \mathbb{J} est obtenue à partir de \mathbb{T} en ôtant les traces non justes et \mathbb{F} est obtenu à partir de \mathbb{J} en ôtant les traces non équitables. Les termes de *justice* et d'*équité* sont dus à Lehman, Pnueli et Stavi [1981] et sont équivalents respectivement à l'*équité faible* et à l'*équité forte*. Nous allons caractériser les propriétés de fatalité sous hypothèse d'exécution équitable mais nous le ferons en plusieurs étapes pour décomposer la démarche et rendre la compréhension plus aisée.

Soit σ un état conduisant fatalement à q une assertion donnée quelconque (sous hypothèse d'exécution équitable) et tel que $\sim q$ est vraie en σ .

$\forall w \in \mathbb{F}(\sigma). \exists w' \in \mathbb{F}(\sigma). (w \leq w') \wedge (\exists i \in \text{dom}(w'). q(w'(i)))$. On suppose que σ est un état accessible à partir de l'ensemble \mathbb{I} .

Nous utilisons les éléments suivants :

- $A(\sigma)$ est l'ensemble des actions atomiques courantes en σ .

Soit a une action atomique de $A(\sigma)$.

- $F(a, q, \sigma)$ est l'ensemble des traces équitables commençant par σ telles que l'action atomique a est courante sur ces traces (i. e. non-exécutée) et, soit q est fausse sur les états de ces traces, soit q est vraie pour le dernier état de ces traces. Cet ensemble permet de représenter l'arbre d'exécution, à partir de σ , pour a et relativement à q (les feuilles validant q). Dans ce cas, on remarque que la non-exécution de a peut engendrer des traces infinies ne validant aucune fois q . Formellement,

$$F(a, q, \sigma) = \{ w \in \mathbb{F}(\sigma) / (\forall i \in \text{dom}(w). (\text{at } a \wedge \sim q)(w(i))) \}$$

$$\vee (\forall i \in \text{dom}(w) - \{ /w \} . (\text{at } a \wedge \sim q)(w(i)) \wedge (\text{at } a \wedge q)(w(/w))) \}$$

— $S(a, q, \sigma)$ est l'ensemble des états associés à $F(a, q, \sigma)$ et ne validant pas q . Un état de $S(a, q, \sigma)$ appartient à une trace de $F(a, q, \sigma)$. Formellement,

$$S(a, q, \sigma) = \{ \sigma' \in \mathbb{S} / \exists w \in F'(a, q, \sigma) . \exists i \in \text{dom}(w) . (w(i) = \sigma') \wedge \sim q(w[1, i]) \}$$

— $R(a, q, \sigma)$ est l'ensemble des états à partir desquels q est accessible mais pour lesquels q est faux. Ces états peuvent être atteints à partir d'un état initial de \mathbb{I} .

$$R(a, q, \sigma) = \{ \sigma' \in \mathbb{S} / t^*(\sigma, \sigma') \wedge \sim q([\sigma, \sigma']) \}$$

— $Y(a, q, \sigma)$ est l'ensemble des états conduisant à q mais tels que les états validant $\text{at } a$ à partir de σ ont été supprimés. Cet ensemble représente ce qu'il reste à « faire » pour atteindre q :

$$Y(a) = R(a, q, \sigma) - S(a, q, \sigma)$$

PROPRIÉTÉ III.1 : Soit σ un état quelconque conduisant à q et tel que $\sim q(\sigma)$. Pour toute action atomique a de $A(\sigma)$,

1. $\mathbb{F} \Downarrow \mid = S(a, q, \sigma) \mathbb{Z} / 12 \mathbb{Z} (Y(a, q, \sigma) \vee q)$;
2. $\mathbb{F} \Downarrow \mid = R(a, q, \sigma) \mathbb{Z} / 12 \mathbb{Z} q$;
3. $\mathbb{F} \Downarrow \mid = Y(a, q, \sigma) \mathbb{Z} / 12 \mathbb{Z} q$;
4. $\mathbb{F} \Downarrow \mid = S(a, q, \sigma) \rightarrow q$. ■

Preuve : Soit σ conduisant à q et tel que $\sim q(\sigma)$.

1. Supposons que $\sigma' \in S(a, q, \sigma)$. Il existe $w \in F(a, q, \sigma)$ tel que $w(i) = \sigma'$, pour un i de $\text{dom}(w)$ et $\sim q(w[1, i])$. a n'a pas été activé entre σ et σ' .

Soit w' une trace quelconque commençant par σ' . Il existe w'' telle que : $\exists i \in \text{dom}(w') . q(w''(i))$. Dans ce cas, soit a est courant en $w''(i)$ et depuis σ' , soit a a été activé. Ce qui revient à écrire : $w''(i) \in q \vee Y(a, q, \sigma)$.

2. Soit $\sigma' \in R(a, q, \sigma)$ et $w' \in \mathbb{F}(\sigma')$.

Puisque $\sigma' \in R(a, q, \sigma)$, q est invalidé le long du chemin $\sigma - * \rightarrow \sigma'$ (et de tous les chemins de ce type). De plus, σ conduit à q , donc la trace $\sigma - * \rightarrow \sigma'$ peut être étendue pour valider q . On en déduit que toute trace commençant par σ' peut se prolonger en une trace validant q .

3. Supposons que $\sigma' \in Y(a, q, \sigma)$. Dans ce cas,

$$\sigma' \in R(a, q, \sigma) - S(a, q, \sigma).$$

D'après 2,

$$\mathbb{F}\mathbb{J} \mid = R(a, q, \sigma) - [] \rightarrow q.$$

4. En utilisant 1 et 3, on en déduit que $\mathbb{F}\mathbb{J} \mid = S(a, q, \sigma) \rightarrow q$. ■

$R(a, q, \sigma)$ est l'arbre d'exécution issu de σ et aboutissant à q et $Y(a, q, \sigma)$ est ce même ensemble mais amputé d'un ensemble d'états précédant l'activation de a . En fait, ceci nous permet d'associer un ordre à σ mais, avant de préciser cet ordre, nous définissons la notion de groupe déterminant d'actions atomiques à partir de σ .

DÉFINITION III.2 : Un groupe g d'actions atomiques de $A(\sigma)$ est déterminant pour q , si :

$$\forall w \in \mathbb{F}(\sigma), \forall i \in \text{dom}(w), \\ (\bigwedge_{a \in g} \text{at } a)(w[1, i]) \Rightarrow ((\bigwedge_{a \in g} \sim Y(a, q, \sigma)(w[1, i])) \wedge \sim q(w[1, i])). \quad \blacksquare$$

Un groupe g est déterminant, si, tant que toute action atomique de g reste inactivée à partir de σ , $Y(a, q, \sigma)$ et q ne sont pas valides pour tout a de g . Il est clair que σ conduisant à q implique qu'au moins une action de $A(\sigma)$ est activable. Cette notion est très importante pour caractériser le plus grand ensemble d'états accessibles conduisant à q . Clairement, le groupe $A(\sigma)$ est déterminant et, si nous considérons $q \equiv \text{after } a$, $\{a\}$ est déterminant. Par conséquent, il existe un ensemble de groupes déterminants pour σ fixé et nous allons montrer qu'il en existe un en particulier qui sera appelé *critique*.

PROPRIÉTÉ III.2 : Il existe au moins un groupe déterminant d'actions atomiques concurrentes de $A(\sigma)$ pour une assertion donnée q . ■

Preuve. — Le groupe $g = A(\sigma)$ est déterminant. ■

PROPRIÉTÉ III.3 : Soit g un groupe déterminant d'actions de $A(\sigma)$.

Pour tout σ' de $\bigcap_{a \in g} S(a, q, \sigma)$,

$$\mathbb{F}\mathbb{J}, \sigma' \mid = \bigwedge_{a \in g} (\text{at } a \wedge S(a, q, \sigma)) \\ - [] \rightarrow q \vee (\text{after } a \wedge (\bigwedge_{\substack{b \in g \\ b \neq a}} \text{at } b) \wedge (Y(a, q, \sigma) \vee q)) \quad \blacksquare$$

Preuve : Soit w une trace commençant par $\sigma' \in \bigcap_{a \in g} S(a, q, \sigma)$. g est un groupe déterminant. Nous supposons que $\forall i \in \text{dom}(w). \forall a \in g. \text{at } a(w(i))$

sinon une des actions a été exécutée et

$$\text{after } a \wedge \left(\bigwedge_{\substack{b \in g \\ b \neq a}} \text{at } b \right) \wedge (Y(a, q, \sigma) \vee q) \text{ est vraie en } w(k)$$

où $k \in \text{dom}(w)$.

Supposons que w est infini et qu'aucune action de g n'a été activée. Alors il existe une trace issue de σ telle que cette trace est infinie et que q est toujours fausse. Cette conclusion est absurde car q est fatale. Donc w est finie et on peut la prolonger en une trace où l'une des actions de g a été exécutée car g est déterminant. On notera que la condition d'activation des actions a , notée $\text{cond}(a)$ n'apparaît pas explicitement. ■

PROPRIÉTÉ III.4 : Soit $[i]$, l'ensemble des groupes de i actions atomiques concurrentes de $A(\sigma)$. Il existe deux indices i_0 et j_0 tels que $1 \leq j_0 \leq i_0 \leq n$ et tous les groupes de $[i]$ pour $i \in \{i_0, \dots, n\}$ sont déterminants. $[j_0]$ ne contient qu'un groupe déterminant et j_0 est le plus petit indice vérifiant cette propriété. ■

Preuve : Soit i_0 le plus petit entier qui vérifie cette condition. Soit k_0 le plus petit entier tel que au moins un groupe est déterminant. On prend le plus petit entier vérifiant cela, en sautant les groupes de $[k_0]$ vers $[n]$. Le cas trivial est $i_0 = j_0 = n$. ■

Le groupe de $[j_0]$ est appelé *groupe critique* et est noté $g(\sigma)$. On note :

$$\begin{aligned} - \text{cond}(g) &\equiv \left(\bigwedge_{a \in g} \text{cond}(g) \right) \text{ et } \sim\text{cond}(g) \equiv \left(\bigwedge_{a \in g} \sim\text{cond}(g) \right); \\ - \text{at } g &\equiv \left(\bigwedge_{a \in g} \text{at } a \right) \text{ et } \text{after } g \equiv \bigvee_{a \in g} \text{after } a \wedge \left(\bigwedge_{\substack{b \in g \\ b \neq a}} \text{at } b \right) \end{aligned}$$

Nous définissons un opérateur $E(g, Y, X)$ associé à tout groupe g d'actions atomiques concurrentes comme suit :

$E(g, Y, X)$ est le plus grand ensemble d'états accessibles et conduisant à X sous hypothèse d'exécution équitable tel que $E(g, Y, X) \cap Y = \emptyset$ et $E(g, Y, X) \cap X = \emptyset$. Pour le construire nous utilisons un opérateur de point fixe et nous commentons les différents éléments de cet opérateur : (nous ajoutons des commentaires dans notre opérateur) O est un opérateur défini sur $\mathcal{P}(\mathbb{S})$ par: soit U une partie de \mathbb{S} . $\sigma \in O(U)$ si, et seulement si, $\{\sigma$ est un état accessible à partie de \mathbb{I} ; les actions de g sont courantes; σ n'appartient ni à Y , ni à X ; σ n'est pas un état bloqué et possède un successeur }

$$(\exists \sigma' \in \mathbb{I}. t^*(\sigma', \sigma)) \wedge \left(\bigwedge_{a \in g} \text{at } a(\sigma) \right) \wedge (\sigma \notin Y) \wedge (\sigma \notin X)$$

$$\wedge (\exists \sigma' \in \mathbb{S}. t(\sigma', \sigma))$$

{ si une action a de g est activable en σ , son exécution conduirait à X dans l'état suivant }

$$\wedge (\forall a \in g. \text{cond}(a)(\sigma) \Rightarrow (\forall \sigma' \in \mathbb{S}. (t(\sigma', \sigma) \wedge \text{after } a(\sigma')) \Rightarrow (\sigma' \in X)))$$

{ si aucune action de g n'est activée, alors l'état suivant appartient à U }

$$\wedge (\forall \sigma' \in g. t(\sigma, \sigma') \wedge \text{at } g(\sigma') \Rightarrow (\sigma' \in U))$$

{ si aucune action de g n'est activable en σ , alors toute trace finie commençant par σ peut se prolonger en une trace telle qu'au moins une action de g devient activable et tous les éléments de la trace obtenue appartienne à U jusqu'à U jusqu'à l'état d'activabilité }

$$\begin{aligned} \wedge (\forall a \in g. \sim \text{cond}(a)(\sigma) &\Rightarrow (\forall w \in \mathbb{F}(\sigma). (\wedge_{a \in g} (\text{at } a \wedge \text{cond}(a))(w[1, /w/])) \\ &\Rightarrow (\exists w' \in \mathbb{F}(\sigma'). (w \leq w') \\ &\wedge (\exists i \in \text{dom}(w'). (\wedge_{a \in g} \text{at } a \wedge \sim \text{cond}(a))(w'[1, i-1])) \\ &\wedge (\exists a \in g. \text{cond}(a)(w'(i))) \\ &\wedge (w'[1, i] \in U)) \end{aligned}$$

Il est clair que l'ensemble \emptyset est la plus petite solution de cet opérateur monotone et nous considérons le plus grand point fixe de l'opérateur O , que nous noterons $E(g, Y, X)$. Donnons quelques exemples de telles expressions et considérons l'exemple du paragraphe I.

1. $E(\{\langle P : = \text{false} \rangle\}, \text{after } a, \text{after } a) = \text{at } a \wedge \text{Acc}$ où Acc est l'ensemble des états accessibles.

2. $E(\{\langle P : = \text{false} \rangle\}, \text{at } a, \text{after } a) = \emptyset$, ceci est vrai car nous n'avons pas de GOTO dans notre langage.

3. $E(\{\langle P : = \text{false} \rangle\}, \text{after } a \wedge \text{in } w, \text{after } a \wedge \text{in } w) = \text{at } a \wedge \text{in } w \wedge \text{Acc}$.

PROPRIÉTÉ III.5 : Pour toute partie X et Y de \mathbb{S} , pour tout groupe g d'actions atomiques concurrentes du programme considéré, $\mathbb{F}\mathbb{J} = E(g, Y, X)\mathbb{Z}/12\mathbb{Z}X$. ■

Preuve : Soit $\sigma \in E(g, Y, X)$ et w une trace commençant par σ . Nous supposons qu'aucun état de w n'est dans Y ou X . Supposons que w est

infinie. Alors :

– ou bien, toutes les actions de g (ou une partie) sont continuellement activables mais ne sont jamais activées et, dans ce cas, w n'est pas juste, ce qui est absurde;

– ou bien, les actions de g sont inactivables en σ ; alors il existe une trace w' prolongeant w (et $w=w'$) et telle qu'au moins une des actions de g est activable en $w'(i_1)$. A partir de cet état;

– ou bien au moins une action de g est continuellement activable, ce qui est absurde par hypothèse de justice, ou bien en réitérant le même raisonnement on met en évidence une suite i_1, i_2, \dots , infinie telle que au moins une des actions de g est activable pour les états indicés par la suite. Puisque le nombre d'éléments de g est fini, il existe au moins une action de g infiniment souvent activable sur cette trace. Cette trace n'est pas équitable, ce qui est absurde.

Supposons que donc que w est finie et qu'aucune action de g n'a été activée. w peut être prolongée en une trace où l'une des actions de g est activée car $w(/w/)$ est élément de $E(g, Y, X)$ et on ne peut la prolonger indéfiniment, sans qu'une action a de g ne soit activée. Dans tous les cas, X est validée fatalement. ■

PROPRIÉTÉ III.6 : Soit q une assertion quelconque. Il existe une assertion $\text{Pre}[q]$ telle que :

1. $\mathbb{F}\mathbb{J} \mid = \text{Pre}[q]\mathbf{Z}/12\mathbf{Z}q$.
2. $\mathbb{F}\mathbb{J} \mid = \text{Pre}[q] \Rightarrow \text{Acc}$ où $\text{Acc}(\sigma) \equiv (\exists \sigma' \in \mathbb{I}. t^*(\sigma', \sigma))$.
3. Pour toute assertion p telle que $\mathbb{F}\mathbb{J} \mid = p\mathbf{Z}/12\mathbf{Z}q$, $\mathbb{F}\mathbb{J} \mid = p \wedge \text{Acc} \Rightarrow \text{Pre}[q]$. ■

Preuve : Soit $E(Y, X)$ défini par $\bigcup_{g \in \mathbb{G}} E(g, Y, X)$ où \mathbb{G} est l'ensemble de

tous les groupes possibles d'actions atomiques concurrentes du programme considéré. Posons $F[q](X) = (q \wedge \text{Acc}) \cup X \cup E(X, X)$, pour toute partie X de \mathbb{S} . $F[q]$ est monotone mais la preuve de cette propriété n'est pas simple et nous utiliserons simplement le fait que par sa définition $F[q]$ est inductive (i.e. $\forall X \subset \mathbb{S}, X \subset F[q](X)$), donc le point fixe classiquement définie par :

- $F[q]^{(0)} = \emptyset$; $F[q]^{(\alpha)} = F[q](F[q]^{(\beta)})$, pour tout $\alpha = \beta + 1$;
- $F[q]^{(\alpha)} = F[q](\bigcup_{\beta < \alpha} F[q]^{(\beta)})$, pour α un ordinal limite,

existe. Posons alors $\text{Pre}[q] = \bigcup_{\alpha} F[q]^{(\alpha)}$ qui est l'inductif point fixe de $F[q]$ et

on pourrait affirmer que c'est le plus petit point fixe, si on avait montré que

cet opérateur est monotone. Cette définition nous suffit pour l'existence de $\text{Pre}[q]$. Par construction de $F[q]$, $\text{Pre}[q]$ vérifie la propriété 1.

2. $\text{Pre}[q]$ est formé uniquement d'états accessibles.

3. Soit σ un état qui conduit à q et qui est accessible. On suppose que q n'est pas vraie en σ . D'après la propriété III. 4, on déduit que σ appartient à l'ensemble

$$E(g(\sigma), (\bigvee_{a' \in g(0)} \text{after } a \wedge (Y(a, g(\sigma), \sigma) \vee q))) \\ (\bigvee_{a' \in g(0)} \text{after } a \wedge (Y(a, g(\sigma), \sigma) \vee q)))$$

On construit une suite d'états correspondant à l'activation d'une action de chaque groupe critique approchant q :

$\sigma \rightarrow \dots \sigma_1 \xrightarrow{a_0} \dots \sigma_2 \xrightarrow{a_1} \dots \sigma_{i-1} \xrightarrow{a_i} q$. Nécessairement, q est atteint. Par un raisonnement inverse, on en déduit que $\sigma \in \text{Pre}[q]$, en utilisant q et $E(\cdot, Y, X)$. ■

Cette dernière propriété caractérise les propriétés de fatalité de type « jusqu'à conditionnel » et permet d'identifier les assertions intermédiaires intervenant dans une propriété de fatalité. Nous déduisons un résultat sur les propriétés générales de fatalité. Nous notons $-([\] \rightarrow)$, au lieu de répéter \rightarrow et $-[\] \rightarrow$ dans les théorèmes.

PROPRIÉTÉ III. 7 : Soient p, q deux assertions.

1. $\mathbb{F}\mathbb{J} \mid = p\mathbb{Z}/12\mathbb{Z}q$ ssi il existe une assertion i et une suite $(p(\alpha))_\alpha$ telles que :

$$a-) \mathbb{F}\mathbb{J} \mid = \text{INIT} \Rightarrow \square i$$

$$b-) \mathbb{F}\mathbb{J} \mid = p \wedge i \Leftrightarrow (\bigvee_{\alpha} p(\alpha))$$

$$c-) \mathbb{F}\mathbb{J} \mid = p(\alpha) - [\] \rightarrow (\bigvee_{\beta < \alpha} p(\beta)), \alpha > 0$$

$$d-) \mathbb{F}\mathbb{J} \mid = p(0) \Rightarrow q.$$

2. $\mathbb{F}\mathbb{J} \mid = p \rightarrow q$ ssi il existe une assertion i et une suite $(p(\alpha))_\alpha$ telles que :

$$a-) \mathbb{F}\mathbb{J} \mid = \text{INIT} \Rightarrow \square i$$

$$b-) \mathbb{F}\mathbb{J} \mid = p \wedge i \Rightarrow (\bigvee_{\alpha} p(\alpha))$$

$$c-) \mathbb{F}\mathbb{J} \mid = p(\alpha) - ([\]) \rightarrow (\bigvee_{\beta < \alpha} p(\beta)), \alpha > 0$$

$$d-) \mathbb{F}\mathbb{J} \mid = p(0) \Rightarrow q. \blacksquare$$

On en déduit ces propriétés très facilement des autres propriétés. Elles mettent en évidence les règles d'inférence que doit contenir au moins un système axiomatique. Cependant, nous avons vu, dans cette partie, que la notion de groupe critique d'actions atomiques joue un rôle fondamental dans la preuve de propriétés de fatalité. En effet, un cas particulier de programmes (sous hypothèse d'exécution juste ou équitable) est le cas de ceux ne réclamant aucune hypothèse d'exécution pour une propriété de fatalité donnée; dans ce cas, l'activation de toute action courante approche strictement l'assertion souhaitée. Par contre, l'exemple simpliste du début montre qu'une action et une seule par sa non-activation peut conduire à la non-accessibilité de la propriété donnée qui est la terminaison du programme. En ce sens, le rôle des groupes critiques est fondamental dans la preuve et nous devons inclure une règle particulière (ou plusieurs) prenant en compte cette caractéristique. Nous en aurons besoin dans la preuve de complétude sémantique.

IV. AXIOMATIQUE TEMPORELLE DE L'HYPOTHÈSE D'ÉQUITÉ

Nous considérons un système axiomatique basé sur les opérateurs temporels O , \square , $-[\]\rightarrow$, \rightarrow . Nous décomposons notre système FEPS en plusieurs étapes :

1. Une partie permettant de prouver les propriétés du type « $p \Rightarrow \square q$ », « $p \Rightarrow Oq$ » et « $p \Rightarrow q \mathcal{U} r$ » qui sont nécessaires d'après les propriétés de la partie III et d'après notre sémantique de l'équité.
2. Une seconde partie contenant des règles d'inférence valides quelles que soient les hypothèses d'exécution et qui permettent de prouver des propriétés de fatalité du type « $p -[\]\rightarrow q$ ».
3. Une troisième partie contenant des règles d'inférence valides pour l'interprétation \mathbb{FJ} et \mathbb{JJ} suivant le choix d'exécution.
4. Une quatrième partie qui permet de déduire des propriétés de fatalité du type « $p \rightarrow q$ » à partir de propriétés du type « $p -[\]\rightarrow q$ ».

Nous dénoterons « $\mathbb{M} \models$ » la validité dans l'interprétation \mathbb{M} et « \vdash » la dérivation dans FEPS de la partie à droite de ce signe. Étant donné un ensemble de théorèmes \mathbb{TH} qui expriment des propriétés s'énonçant à l'aide des connecteurs logiques \Rightarrow ou \Leftrightarrow . Un théorème dans FEPS est le dernier élément d'une suite finie composée :

- a. d'axiomes qui sont des membres de \mathbb{TH} ou des occurrences de membres de \mathbb{TH} ;
- b. des axiomes de FEPS;

c. des propriétés déduites à l'aide de règles de FEPS à partir de propriétés précédemment déduites.

Une telle suite est une preuve. On la note $\{TH_0, \dots, TH_i, \dots, TH_k\}$ et on dit que TH_k est un théorème de preuve $\{TH_0, \dots, TH_i, \dots, TH_k\}$. En particulier, $\{TH_0, \dots, TH_i, \dots, TH_k\}$ est une preuve de TH_i pour tout $i \leq n$. Démontrer dans FEPS un théorème revient à construire une telle suite quand elle existe.

IV. A. Axiomatique pour les propriétés d'invariance, de transition et de type « à moins que »

Nous utilisons cette partie pour démontrer des propriétés d'invariance (i. e. $p \Rightarrow \Box q$), de transition élémentaire (i. e. $p \Rightarrow Oq$) et de type « à moins que » ou « jusqu'à faible » (i. e. $p \Rightarrow q \mathcal{U} r$). Il est clair que l'opérateur O spécifie la relation t qui simule l'exécution pas-à-pas du programme. Par conséquent, les propriétés qui définissent t sont des axiomes de notre système.

T1 : $\text{at } (V := E) \wedge p[e/v] \Rightarrow O (\text{after } (V := E) \Rightarrow p)$ où $p[e/v]$ désigne l'assertion obtenue en substituant les occurrences libres de v par l'expression e .

T2 : $\forall c \in \text{Domaine}(V). \text{at } (V := ?) \wedge p[c/v] \Rightarrow O (\text{after } (V := ?) \Rightarrow p)$.

T3 : $\text{at SKIP} \wedge p \Rightarrow O (\text{after SKIP} \Rightarrow p)$.

T4 : $\text{at } (\text{if } B \text{ then } S1 \text{ else } S2 \text{ fi}) \wedge p \wedge b \Rightarrow O (\text{at } S1 \Rightarrow p \wedge b)$.

T5 : $\text{at } (\text{if } B \text{ then } S1 \text{ else } S2 \text{ fi}) \wedge p \wedge \sim b \Rightarrow O (\text{at } S2 \Rightarrow p \wedge \sim b)$.

T6 : $\text{after } S1 \wedge p \Rightarrow O (\text{after } (\text{if } B \text{ then } S1 \text{ else } S2 \text{ fi}) \Rightarrow p)$.

T7 : $\text{after } S2 \wedge p \Rightarrow O (\text{after } (\text{if } B \text{ then } S1 \text{ else } S2 \text{ fi}) \Rightarrow p)$.

T8 : $\text{at } (\text{while } B \text{ do } S \text{ od}) \wedge p \wedge b \Rightarrow O (\text{at } S \Rightarrow p \wedge b)$.

T9 : $\text{at } (\text{while } B \text{ do } S \text{ od}) \wedge p \wedge \sim b$

$\Rightarrow O (\text{after } (\text{while } B \text{ do } S \text{ od}) \Rightarrow p \wedge \sim b)$

(Dans T10 et T11, S est le corps de la boucle $\text{while } B \text{ do } S \text{ od}$)

T10 : $\text{after } S \wedge p \wedge b \Rightarrow O (\text{at } S \Rightarrow p \wedge b)$.

T11 : $\text{after } S \wedge p \wedge \sim b \Rightarrow O (\text{after } (\text{while } B \text{ do } S \text{ od}) \Rightarrow p \wedge \sim b)$.

T12 : $\text{at COBEGIN } \mathbb{P}1 // \dots // \mathbb{P}n \text{ COEND} \wedge p$

$\Rightarrow O (\text{at } \mathbb{P}1 \wedge \dots \wedge \text{at } \mathbb{P}n \Rightarrow p)$.

T13 : $\text{after } \mathbb{P}1 \wedge \dots \wedge \text{after } \mathbb{P}n \wedge p$

$\Rightarrow O$ (after COBEGIN $\mathbb{P}1// \dots // \mathbb{P}n$ COEND $\Rightarrow p$).

Soit S une variable de sémaphore et s la valeur de S .

T14 : at $V(S) \wedge p [s-1/s] \Rightarrow O$ (after $V(S) \Rightarrow p$).

T15 : at $P(S) \wedge p [s-1/s] \wedge (s > 0) \Rightarrow O$ (after $P(S) \Rightarrow p$).

Nous ajoutons à ce système des règles d'inférence propres à développer des propriétés de transition.

Règle 1 : Si $p \Rightarrow q$, $q \Rightarrow Or$ et $r \Rightarrow s$, alors $p \Rightarrow Os$. ■

Règle 2 : Si at $a \wedge p \Rightarrow O$ (after $a \Rightarrow q$) où a est une action atomique, alors at $a \wedge p \Rightarrow O$ (after $a \Rightarrow$ after $a \wedge q$). ■

Cette règle est déduite de la règle 1 mais est très utile pour les preuves pratiques. Elle permet d'introduire des assertions de contrôle dans les assertions.

Règle 3 : Si at $a \wedge \text{cond}(a) \wedge p \Rightarrow O$ (after $a \Rightarrow q$), pour toute action atomique a , alors $p \Rightarrow Oq$. ■

$\text{cond}(a)$ est équivalent à *vraie*, si a est différent de $P(S)$ et équivalent à $s > 0$ sinon.

PROPRIÉTÉ IV.1 (Correction et Complétude relative à O) : Soient p, q deux assertions et \mathbb{J} une des trois interprétations $\{\mathbb{S}\mathbb{J}, \mathbb{J}\mathbb{J}, \mathbb{F}\mathbb{J}\}$. FEPS $\vdash p \Rightarrow Oq$ ssi $\mathbb{J} \mid = p \Rightarrow Oq$. ■

Preuve: 1. On suppose que FEPS $\vdash p \Rightarrow Oq$. Pour montrer que $\mathbb{J} \mid = p \Rightarrow Oq$, on montre classiquement que les axiomes sont valides dans \mathbb{J} et que les règles d'inférences sont correctes par rapport à \mathbb{J} . Ce qui est immédiat d'après le choix des axiomes et la définition de O .

2. On suppose que $\mathbb{J} \mid = p \Rightarrow Oq$.

$$\mathbb{J} \mid = p \Rightarrow Oq \text{ ssi } \forall \sigma \in \mathbb{S}. p(\sigma) \Rightarrow (\forall \sigma' \in \mathbb{S}. t(\sigma, \sigma') \Rightarrow q(\sigma')).$$

Ce qui peut s'écrire : $\forall \sigma, \sigma' \in \mathbb{S}. (p(\sigma) \wedge t(\sigma, \sigma')) \Rightarrow q(\sigma')$. On note $A(p) = \{a \mid \exists \sigma \in \mathbb{S}. (p \wedge \text{cond}(a) \wedge \text{at } a)(\sigma)\}$.

(1) $\forall \sigma \in \mathbb{S}. p(\sigma) \Leftrightarrow \bigvee_g (\text{at } g \wedge p)(\sigma)$ où g est un n -uplet d'actions atomiques concurrentes du programme considéré ou une des deux actions START ou FINISH. On suppose qu'il y a n processus concurrents formant le programme considéré. Notons $\text{Pre}(a)(q)$ l'assertion définie par l'ensemble suivant :

$$\{\sigma \in \mathbb{S} / (\text{at } a \wedge \text{cond}(a))(\sigma) \wedge (\forall \sigma' \in \mathbb{S}. t(\sigma, \sigma') \wedge \text{after } a(\sigma') \Rightarrow q(\sigma'))\}.$$

(2) Pour toute action atomique a et par définition de $\text{Pre}(a)(q)$,

$$\Downarrow \models \text{Pre}(a)(q) \wedge \text{at } a \wedge \text{cond}(a) \Rightarrow O(\text{after } a \Rightarrow q).$$

(3) $\text{FEPS} \vdash \text{Pre}(a)(q) \wedge \text{at } a \wedge \text{cond}(a) \Rightarrow O(\text{after } a \Rightarrow q)$, par application de l'axiome associé à a . Par exemple : $\text{Pre}(a)(q)$ est équivalent à $q[e/v]$, si a est l'action $V := E$.

$$(4) \forall \sigma \in \mathcal{S}. (p \wedge \text{at } a \wedge \text{cond}(a))(\sigma) \Rightarrow (\text{Pre}(a)(q) \wedge \text{at } a \wedge \text{cond}(a))(\sigma)$$

d'après l'hypothèse et la définition de $\text{Pre}(a)(q)$. On notera que l'assertion $\text{Pre}(a)(q)$ peut être *fausse*.

(5) $\text{FEPS} \vdash p \wedge \text{at } a \wedge \text{cond}(a) \Rightarrow \text{Pre}(a)(q) \wedge \text{at } a \wedge \text{cond}(a)$, puisque TH contient ce théorème.

(6) $\text{FEPS} \vdash p \wedge \text{at } a \wedge \text{cond}(a) \Rightarrow O(\text{after } a \Rightarrow q)$, à l'aide de la règle 1 et des théorèmes ci-dessus (3) et (5).

(7) $\text{FEPS} \vdash p \Rightarrow Oq$, à l'aide de la règle 3 et du théorème ci-dessus (7). ■

Nous avons donné une démonstration formelle de cette complétude et celle-ci peut être assimilée à un schéma général de preuve de telles propriétés. Nous aurions dû envisager tous les cas pour le théorème (2) mais celle-ci aurait été trop longue. Les axiomes T1, . . . , T15 peuvent être vus comme des axiomes « à la Hoare » mais sont plutôt des flashes instantanés des transitions faites par l'exécution du programme. Nous ajoutons maintenant une règle pour dériver des propriétés d'invariance du type « $p \Rightarrow \square q$ ».

Règle 4 (d'invariance) : Si $p \Rightarrow i$, $i \Rightarrow Oi$ et $i \Rightarrow q$, alors $p \Rightarrow \square q$. ■

La correction et la complétude sémantique de cette règle ont été déjà données (voir par exemple Manna et Pnueli [1984]). Rappelons que la complétude sémantique revient à montrer l'existence d'une assertion i intermédiaire et que celle-ci n'est pas toujours expressible dans un langage d'assertions donné. De ce fait, nous raisonnons sur des ensembles d'états en tant qu'assertions. Nous aurons besoin d'une telle règle dont la prémisse « $i \Rightarrow Oi$ » peut être déduite par les règles 1, 2, 3 et les axiomes T1, . . . , T15, d'après le résultat précédent. Nous introduisons une règle pour un certain type de propriétés du type « à moins que ».

Règle 5 (de propriétés à moins que) : Soit g un groupe d'actions atomiques concurrentes.

Si, pour tout a de g ,

$$(\bigwedge_{b \in g} (\text{at } b \wedge \text{cond}(b))) \wedge p \Rightarrow O(\text{after } a \Rightarrow q_a)$$

et, pour toute action c qui n'est pas dans g et qui n'est pas dans le même processus que les actions de g ,

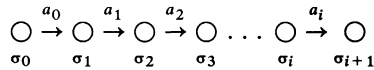
$$\begin{aligned} & (\bigwedge_{b \in g} (\text{at } b \wedge \text{cond}(b))) \wedge p \wedge \text{at } c \wedge \text{cond}(c) \\ & \Rightarrow O(\text{after } c \Rightarrow (\bigwedge_{b \in g} (\text{at } b \wedge \text{cond}(b))) \wedge p), \end{aligned}$$

alors

$$\begin{aligned} & (\bigwedge_{b \in g} (\text{at } b \wedge \text{cond}(b))) \wedge p \\ & \Rightarrow (\bigwedge_{b \in g} (\text{at } b \wedge \text{cond}(b))) \wedge p \mathcal{U} (\bigvee_{a \in g} (\text{after } a \wedge q_a)). \blacksquare \end{aligned}$$

Nous donnons maintenant une justification de cette règle mais elle n'est pas complète. Une version complète est donnée par Manna et Pnueli [1983 b].

Preuve : Nous supposons que les prémisses de la règle sont valides. Soit σ un état quelconque tel que $(\bigwedge_{b \in g} (\text{at } b \wedge \text{cond}(b))) \wedge p$ est vraie. Soit w une trace commençant par σ on note $\sigma_0 = \sigma$:



On distingue deux cas :

1. Aucun des a_i n'est élément de g .
2. Un des a_i est un élément de g et notons a_{i_0} celui-ci où i_0 est la première occurrence de cette action.
 1. D'après la seconde prémisses, chaque état de w valide l'assertion $(\bigwedge_{b \in g} (\text{at } b \wedge \text{cond}(b))) \wedge p$.
 2. D'après la seconde prémisses, chaque état de w de rang inférieur à i_0 valide l'assertion $(\bigwedge_{b \in g} (\text{at } b \wedge \text{cond}(b))) \wedge p$ et $w(i_0)$ valide $(\bigvee_{a \in g} (\text{after } a \wedge q_a))$ ■

Nous avons utilisé l'opérateur \mathcal{U} défini dans Lamport [1980 a] et Manna et Pnueli [1983 b]. Cet opérateur est plus faible que le « *Until* » classique et permet de construire les preuves de théorèmes de propriétés de fatalité de type « *jusqu'à conditionnel* » sous hypothèse d'exécution juste. Donnons un exemple de propriété exprimée à l'aide d'un tel opérateur, en utilisant

l'exemple simpliste :

$$\ll \text{at} \langle P := \text{false} \rangle \wedge p \Rightarrow \text{at} \langle P := \text{false} \rangle \wedge p \mathcal{U} \text{after} \langle P := \text{false} \rangle \wedge \sim p \gg$$

est valide dans toutes les interprétations \mathbb{J} utilisées.

IV. B. Axiomatique générale pour les propriétés de fatalité de la forme « $p - [] \rightarrow q$ »

Règle 6 (*d'implication*) : Si $p \Rightarrow q$, alors $p - [] \rightarrow q$. ■

Règle 7 (*d'extension*) : Si $p - [] \rightarrow q$ et $q \Rightarrow r$, alors $p - [] \rightarrow r$. ■

Règle 8 (*d'induction*) : Si, pour tout $\alpha > 0$, $p_\alpha - [] \rightarrow \bigvee_{\beta < \alpha} p_\beta$, alors
 $\bigvee_{\alpha} p_\alpha - [] \rightarrow p_0$, où α et β sont des ordinaux. ■

Règle 9 (*d'invariance fatale*) : Soit INIT l'assertion qui caractérise les états initiaux. Si $\text{INIT} \Rightarrow \Box i$, $p \wedge i \Leftrightarrow q$ et $q - [] \rightarrow r$, alors $p - [] \rightarrow r \wedge i$. ■

La correction de ces règles est valable pour les trois interprétations $\mathbb{S}\mathbb{J}$, $\mathbb{J}\mathbb{J}$ et $\mathbb{F}\mathbb{J}$. La règle 8 est importante et indispensable d'après les propriétés de la partie III. Ces règles permettent de manipuler les propriétés du type « $p - [] \rightarrow q$ » d'un point de vue formel. A partir de ces règles, on peut en déduire quelques autres règles très utiles en pratique dans les preuves. Nous n'abordons pas tout de suite le problème de la complétude sémantique car il nous faut ajouter des règles supplémentaires.

IV. C. Axiomatique des hypothèses de justice et d'équité

Owicki et Lamport [1982] ont proposé une expression temporelle et axiomatique de l'hypothèse de justice et de celle d'équité. Nous proposons une approche plus constructive et basée sur la correction de deux règles d'inférence dans le cadre des hypothèses d'équité et de justice. L'approche d'Owicki et Lamport [1982] consiste à spécifier temporellement l'ensemble des traces justes (resp. équitables) par les axiomes « $\text{at } a \rightarrow \text{after } a$ » (resp. $\text{at } a \wedge \Box \langle \rangle \text{cond}(a) \rightarrow \text{after } a$). Nous noterons que la spécification de la justice par les axiomes suppose implicitement que a est activable jusqu'à son activation ou du moins on ne sait si $\text{after } a$ est vrai pour la première fois ou la deuxième... depuis a . Cette notation ne l'exprime pas pourtant et nous

avons alors imaginé la règle suivante qui définit l'ensemble des traces justes :

Règle 10 (*de la justice*) : Si

$$(\bigwedge_{b \in g} (\text{at } b \wedge \text{cond}(b)) \wedge p) \Rightarrow (\bigwedge_{b \in g} (\text{at } b \wedge \text{cond}(b)) \wedge p) \mathcal{U} (\bigvee_{a \in g} (\text{after } a \wedge q_a)).$$

alors $(\bigwedge_{b \in g} (\text{at } b \wedge \text{cond}(b)) \wedge p) - [\] \rightarrow (\bigvee_{a \in g} (\text{after } a \wedge q_a)).$ ■

Cette règle est valide dans les interprétations \mathbb{J} et \mathbb{F} . Intuitivement, tant qu'aucune action de g n'est exécutée, p est vraie et, dès que l'une de celles-ci sera exécutée, alors $(\bigvee_{a \in g} (\text{after } a \wedge q_a))$ sera valide et, de plus, l'une sera exécutée par justice. La correction de la règle est immédiate. Nous introduisons une nouvelle règle uniquement valide pour \mathbb{F} . Nous la nommons règle d'équité car elle axiomatise cette notion et en donne une version constructive pour les preuves.

Soient g_a et g_b une partition de g , un groupe d'actions atomiques concurrentes tels que $g_a \cap g_b = \emptyset$ et $g_a \cup g_b = g$. Définissons les assertions suivantes :

- (1) $p \equiv \text{at } g \wedge \text{cond}(g_a) \wedge \sim \text{cond}(g_b) \wedge p'$;
- (2) $q_a \equiv \text{at}(g - \langle a \rangle) \wedge q'_a \wedge \text{after } a$, pour tout a de g_a ;

(3) $q \equiv \bigvee_{a \in g_a} q_a$;

(4) $r_b \equiv \text{at } g \wedge \text{cond}(g_a) \wedge \text{cond}(b) \wedge q'_b$, pour tout b de g_b ;

(5) $r \equiv \bigvee_{b \in g_b} r_b$.

Règle 11 (*d'équité*) : Si (1) $p - [\] \rightarrow (q \vee r)$.

(2) pour toute action c hors de g , et, pour tout b dans g_b , $r_b \wedge \text{at } c \wedge \text{cond}(c) \Rightarrow O$ ($\text{after } c \Rightarrow p \vee r$);

(3) pour toute action b de g_b , $r_b \Rightarrow O$ ($\text{after } b \Rightarrow q_b$);

(4) pour toute a de g_a , b de g_b , $r_b \Rightarrow O$ ($\text{after } a \Rightarrow q_a$);

(5) pour toute action b de g_b et b' distincte de b mais dans g_b ;

$$r_b \wedge \text{at } b' \wedge \text{cond}(b') \Rightarrow O \text{ (after } b' \Rightarrow q'_b).$$

alors $(p \vee r) - [\] \rightarrow (q \vee \bigvee_{b \in g_b} q_b).$ ■

Cette règle mérite une ample justification. Considérons que les prémisses de cette règle sont valides dans l'interprétation $\mathbb{F}\mathbb{J}$.

Soit σ un état tel que $(p \vee q)(\sigma)$ est vrai et w une trace commençant par σ . On suppose que cette trace w peut se prolonger en une trace infinie telle que $\sim q$ et, pour toute action b de g_b , $\sim q_b$ sont vrais sur cette trace. On la note w . On distingue deux cas :

1. Si $p(\sigma)$ est vrai, alors r devient vrai sur w car q ne peut être vrai. Ce raisonnement reste vrai pour tout état validant p sur w .

2. Si $r(\sigma)$ est vrai, alors l'une au moins des actions de g_b est activable et toutes les actions de g_a sont activables.

Par conséquent, en exécutant une action activable, on aboutit à :

(1) soit en l'état validant $p \vee q$ après exécution de l'une des actions, c , qui ne se trouvent pas dans g [d'après la prémise (2)];

(2) soit en l'état validant q_b d'après (3), ce qui est impossible par hypothèse;

(3) soit en l'état validant un q_a , d'après (4), ce qui est impossible par hypothèse;

(4) soit en l'état validant un q'_b , d'après la prémise (5) ce qui est impossible par hypothèse.

Par conséquent, le seul cas plausible est le cas 1. La suite w est infinie et, en appliquant le même raisonnement pour chaque état de w , on en déduit :

– soit, il existe un indice i_0 tel que, à partir de $w(i_0)$ une action de g est continuellement activable i. e. $g_a \neq \emptyset$;

– soit, il existe une suite infinie i_0, i_1, \dots d'états tels que une action de g_b est activable en i_0, i_1, \dots

Dans le premier cas, la suite w n'est pas juste et, dans le second cas, la suite w n'est pas équitable. Une telle suite ne peut exister dans \mathbb{J} et \mathbb{F} (pour le cas 1) et dans \mathbb{F} (pour le cas 2). ■

L'idée exprimée dans cette règle est que, dans ce cas, le plus extrême, où $g_b \neq \emptyset$, les actions de g_b sont infiniment souvent activables à partir d'un état validant p et, par équité, l'une d'entre elles sera activée. Cependant, la première prémise de cette règle ne peut être dérivée dans ce système et nous ajoutons la règle dite *de la dimension* qui permet d'intégrer un raisonnement récursif sur le nombre de processus concurrents constituant le programme. Notre méthode axiomatique est donc récursivement applicable comme celle de Lehman, Pnueli et Stavi [1981] à propos de la terminaison des programmes.

Nous notons « $n - k \vdash p$ » la dérivation de p dans le cadre d'un programme ayant $n - k$ processus concurrents.

Règle 12 (de la dimension) : Si

$$(1) \quad n - k \vdash (p' \wedge (\bigwedge_{a \in g} \sim \text{cond}(a))) - [\] \rightarrow (q' \vee (\bigvee_{a \in g} (\text{cond}(a) \wedge q'_a)))$$

où k est le cardinal de g , q' et q'_a sont des assertions sans références au contrôle dans les processus contenant les actions de

$$g: \quad g = \{ P_1(S_{i_1}), \dots, P_k(S_{i_k}) \},$$

$$(2) \quad p' \wedge (\bigwedge_{a \in g} \sim \text{cond}(a) \wedge \text{at } a) \Leftrightarrow p$$

$$(3) \quad q' \wedge (\bigwedge_{a \in g} \text{at } a) \Leftrightarrow q$$

$$(4) \quad q'_a \wedge \text{cond}(a) \wedge (\bigwedge_{a \in g} \text{at } a) \Leftrightarrow q_a'$$

alors $p - [\] \rightarrow (q \vee (\bigvee_{a \in g} q_a))$. ■

Intuitivement, si des processus sont bloqués devant des opérations $P(S)$ de sémaphores, le système se comporte comme ce même système sans les processus bloqués qui n'ont plus d'effets temporairement sur l'état du programme. La preuve se fait donc dans un système de dimension strictement plus faible. Cependant, nous noterons que la validité de cette règle, qu'intuitivement exacte, repose sur une définition de la sémantique d'un programme à n processus à partir d'une sémantique d'un programme à m processus où $m < n$. On utilise une restriction sur les traces d'exécution et sur les états.

IV. C. Axiomatique de la fatalité

Nous avons donné les règles et les axiomes pour construire des preuves de propriétés du type « $p - [\] \rightarrow q$ » qui sont des propriétés de fatalité élémentaires. A partir de ces propriétés élémentaires, on déduit les propriétés générales du type « $p \rightarrow q$ ».

Règle 13 : Si $p - [\] \rightarrow q$, alors $p \rightarrow q$. ■

Règle 14 : Soit INIT la spécification des états initiaux.

Si $\text{INIT} \Rightarrow \square i, p \wedge i \Rightarrow q, q \rightarrow r$, alors $p \rightarrow r$. ■

Dans la plupart des exemples pratiques, on doit montrer des propriétés du type « $\text{at } l / 12 \text{Zat } l'$ ». Cependant, l'information contenue par l'assertion « $\text{at } l$ » est en rapport uniquement avec le contrôle et elle ne spécifie pas complètement l'état courant des variables. Nous verrons ce genre de problème

dans le cas de l'algorithme d'exclusion mutuelle. Nous abordons maintenant la question de la complétude sémantique de FEPS pour \rightarrow et $\mathbf{Z}/12\mathbf{Z}$.

PROPRIÉTÉ IV.2 : Soient p, q deux assertions telles que $\mathbb{F}\mathbb{J} \mid = p - ([\]) \rightarrow q$. Alors $\vdash p - ([\]) \rightarrow q$. ■

Pour démontrer cette propriété, il faut donc construire une suite de théorèmes de FEPS.

Preuve : On ne précisera pas le signe \vdash pour la dérivation dans FEPS mais seulement le signe de validité $\mid =$.

Premier cas : $\mathbb{F}\mathbb{J} \mid = p\mathbf{Z}/12\mathbf{Z}q$: (1) D'après la propriété II.9, il existe une suite d'assertions $(p_\alpha)_\alpha$ et une assertion i telles que :

$$(a) \mathbb{F}\mathbb{J} \mid = \text{INIT} \Rightarrow \square i$$

$$(b) \mathbb{F}\mathbb{J} \mid = p_\alpha\mathbf{Z}/12\mathbf{Z} \bigvee_{\beta < \alpha} p_\beta \text{ où } \alpha \text{ est un ordinal non-nul.}$$

$$(c) \mathbb{F}\mathbb{J} \mid = p \wedge i \Leftrightarrow \bigvee_a p_\alpha$$

$$(d) \mathbb{F}\mathbb{J} \mid = p_0 \Rightarrow q$$

(2) $p_0 \Rightarrow q$, puisque c'est un théorème de $\mathbb{T}\mathbb{H}$ en utilisant la propriété (1) (d).

(3) $p \wedge i \Leftrightarrow \bigvee_a p_\alpha$, puisque c'est un théorème de $\mathbb{T}\mathbb{H}$ en utilisant la propriété (1) (c).

(4) $\text{INIT} \Rightarrow \square i$, en utilisant la propriété (1) (a) et la complétude par rapport à \square .

Analysons les éléments de la suite $(p_\alpha)_\alpha$. Pour tout état σ validant p_α où α est non nul, il existe un groupe critique noté $g(\sigma)$. Nous notons $G(\alpha)$, l'ensemble des groupes critiques associés à p_α :

$G(\alpha) = \{ g(\sigma)/(p_\alpha)(\sigma) \}$. Clairement, $p_\alpha \Leftrightarrow \bigvee_g \text{at } g \wedge p_\alpha$ où g parcourt l'ensemble des n -uples d'actions atomiques concurrentes du programme et de plus $\bigvee_g \text{at } g \wedge p_\alpha \Leftrightarrow \bigvee_{g \in G(\alpha)} \text{at } g \wedge p_\alpha$, ce qui revient à dire que p_α est partitionné suivant $G(\alpha)$. Nous reprenons la preuve après cette remarque.

(5) $p_\alpha \Leftrightarrow \bigvee_g \text{at } g \wedge p_\alpha$, où α est non nul, puisque c'est une assertion vraie, d'après la remarque ci-dessus, c'est un théorème de $\mathbb{T}\mathbb{H}$.

(6) $\text{at } g \wedge p_\alpha\mathbf{Z}/12\mathbf{Z} \text{after } g \wedge (\bigvee_{\beta < \alpha} p_\beta)$, pour tout g de $G(\alpha)$, nous démontrons cette propriété ultérieurement.

(7) $\text{after } g \wedge (\bigvee_{\beta < \alpha} p_\beta) \Rightarrow \bigvee_{\beta < \alpha} p_\beta$, puisque c'est une propriété valide du calcul propositionnel, c'est un théorème de \mathbb{TH} .

(8) $\text{at } g \wedge p_\alpha - [] \rightarrow (\bigvee_{\beta < \alpha} p_\beta)$, en appliquant la règle d'extension (de numéro 5) avec les théorèmes (6) et (7) ci-dessus comme prémisses.

(9) $\bigvee_{g \in G(\alpha)} \text{at } g \wedge p_\alpha - [] \rightarrow (\bigvee_{\beta < \alpha} p_\beta)$, par la règle d'induction (de numéro 6) appliquée pour le cas fini et en utilisant le théorème 8 comme prémisse.

(10) $p_\alpha \wedge i \Leftrightarrow \bigvee_{g \in G(\alpha)} \text{at } g \wedge p_\alpha$, à partir de la propriété (1) (c) et d'après la décomposition des assertions p_α suivant l'ensemble $G(\alpha)$; c'est un théorème de \mathbb{TH} .

(11) $p_\alpha - [] \rightarrow \bigvee_{\beta < \alpha} p_\beta$, en utilisant le règle 9 dite de l'invariance fatale avec les théorèmes (4), (9) et (10) comme prémisses.

(12) $\bigvee_a p_\alpha - [] \rightarrow p_0$, par la règle d'induction de numéro 8 avec le théorème (11).

(13) $\bigvee_a p_\alpha - [] \rightarrow q$, par la règle d'extension de numéro règle 7, à partir des théorèmes (2) et (12).

(14) $p - [] \rightarrow q$, par la règle d'invariance fatale à partir des théorèmes (3), (4) et (13).

Deuxième cas : $\mathbb{FJ} \mid = p \rightarrow q$: (1) D'après la propriété II.9, il existe une suite d'assertions $(p_\alpha)_\alpha$ et une assertion i telles que :

(a) $\mathbb{FJ} \mid = \text{INIT} \Rightarrow \square i$;

(b) $\mathbb{FJ} \mid = p_\alpha - [] \rightarrow \bigvee_{\beta < \alpha} p_\beta$ où α est un ordinal non-nul;

(c) $\mathbb{FJ} \mid = p \wedge i \Rightarrow \bigvee_a p_\alpha$;

(d) $\mathbb{FJ} \mid = p_0 \Rightarrow q$.

.2) $p_0 \Rightarrow q$, puisque c'est un théorème de \mathbb{TH} en utilisant la propriété (1) (d).

(3) $p \wedge i \Rightarrow \bigvee_a p_\alpha$, puisque c'est un théorème de \mathbb{TH} en utilisant la propriété (1) (c).

(4) $\text{INIT} \Rightarrow \square i$, en utilisant la propriété (1) (a) et la complétude par rapport à \square .

(5) $p_\alpha - [] \rightarrow \bigvee_{\beta < a} p_\beta$, en utilisant la règle 9 dite de l'invariance fatale avec les théorèmes (4), (9) et (10) du premier cas comme prémisses.

(6) $\bigvee_a p_\alpha - [] \rightarrow q$, d'après le premier cas

(7) $p \rightarrow q$, par la règle 14 à partir des théorèmes (4), (5) et (6).

Troisième cas : Il ne s'agit pas véritablement d'un cas mais d'une démonstration annexe au deux cas précédents. Démontrons que, pour tout groupe g de $G(\alpha)$,

$$\text{at } g \wedge p_\alpha - [] \rightarrow \text{after } g \wedge \left(\bigvee_{\beta < a} p_\beta \right),$$

Sous-cas 1 : Toutes les actions de g sont continuellement activables jusqu'à une activation d'une de ces actions.

Dans ce cas, pour tout b hors de g ,

$$\mathbb{FJ} \mid = \text{at } a \wedge p_a \wedge \text{at } b \wedge \text{cond}(b) \Rightarrow O(\text{after } b \Rightarrow \text{at } a \wedge p_a),$$

pour tout a de g ,

$$\mathbb{FJ} \mid = \text{at } a \wedge p_a \Rightarrow O(\text{after } a \Rightarrow \left(\bigvee_{\beta < a} p_\beta \right)),$$

d'après la construction des assertions p_α à l'aide de l'opérateur $F[q]$. On déduit les théorèmes suivants, d'après la propriété IV.1 relative à la complétude du système par rapport à l'opérateur O et aux propriétés de transition.

(1) pour tout b hors de g ,

$$\text{at } a \wedge p_\alpha \wedge \text{at } b \wedge \text{cond}(b) \Rightarrow O(\text{after } b \Rightarrow \text{at } a \wedge p_\alpha),$$

(2) pour tout a de g ,

$$\text{at } a \wedge p_\alpha \Rightarrow O(\text{after } a \Rightarrow \left(\bigvee_{\beta < a} p_\beta \right)),$$

(3) $\text{at } g \wedge p_\alpha \wedge \text{at } b \wedge \text{cond}(b) \Rightarrow \text{at } a \wedge p_\alpha \wedge \text{at } b \wedge \text{cond}(b)$, par déduction dans le calcul propositionnel à l'aide d'une propriété sur le « et » logique.

(4) $\text{at } g \wedge p_\alpha \wedge \text{at } b \wedge \text{cond}(b) \Rightarrow O(\text{after } b \Rightarrow \text{at } g \wedge p_\alpha)$, pour tout action atomique b hors de g , en utilisant la règle 1 avec les théorèmes (1) et (3) ci-dessus.

(5) $\text{at } g \wedge p_\alpha \wedge \text{at } b \wedge \text{cond}(b) \wedge \text{at } a \wedge \text{cond}(a) \Rightarrow O(\text{after } a \Rightarrow \left(\bigvee_{\beta < a} p_\beta \right))$.

d'après la complétude démontrée dans la propriété IV.1 à partir de la troisième hypothèse de ce cas.

(6) $at\ g \wedge p_\alpha \Rightarrow at\ g \wedge p_\alpha \mathcal{U} \text{ after } a \wedge (\bigvee_{\beta < \alpha} p_\beta)$, à l'aide de la règle 5, avec les théorèmes 4 et 5 ci-dessus.

(7) $at\ g \wedge p_\alpha - [] \rightarrow \text{after } a \wedge (\bigvee_{\beta < \alpha} p_\beta)$, en utilisant la règle de justice, à l'aide du théorème (6).

Sous-cas deux : Au moins une action de g est soit activable, soit non activable mais les autres restent continuellement activables. On note g_a (resp. g_b) l'ensemble des actions continuellement activables (l'ensemble des actions non activables); $g = g_a \cup g_b$. Notons

$$(1) p \equiv at\ g \wedge p_\alpha \wedge (\bigwedge_{a \in g_a} \text{cond}(a)) \wedge (\bigwedge_{b \in g_b} \sim \text{cond}(b))$$

$$(2) q \equiv \bigvee_{a \in g} \text{after } a \wedge (\bigvee_{\beta < \alpha} p_\beta)$$

$$(3) r \equiv \bigvee_{b \in g_b} (at\ b \wedge (\bigvee_{\beta < \alpha} p_\beta) \wedge \text{cond}(b) \wedge (\bigwedge_{a \in g_a} \text{cond}(a)))$$

Le point délicat est de démontrer que : $p - [] \rightarrow (q \vee r)$. Notons

$$p' \equiv (\bigwedge_{a \in g_a} at\ a) \wedge p_\alpha \wedge (\bigwedge_{a \in g_a} \text{cond}(a)) \wedge (\bigwedge_{b \in g_b} \sim \text{cond}(b)).$$

l'assertion obtenue à partir de p en supprimant les contrôles de g_b . Par le cas 3 que nous verrons plus tard et par induction sur le nombre des processus nous déduisons :

$$(1) p' - [] \rightarrow (q' \vee r') \text{ pour } n - |g_a|/\text{processus.}$$

$$(2) p' \wedge (\bigwedge_{b \in g_b} at\ b) \Leftrightarrow p$$

$$(3) q' \wedge (\bigwedge_{b \in g_b} at\ b) \Leftrightarrow q$$

$$(4) r' \wedge (\bigwedge_{b \in g_b} at\ b) \Leftrightarrow r$$

(5) $p - [] \rightarrow (q \vee r)$, en utilisant la règle 12 et les théorèmes (1), (2), (3) et (4).

(6) $r_b \wedge at\ c \wedge \text{cond}(c) \Rightarrow O(\text{after } c \Rightarrow p \vee r)$, pour tout c de g et pour tout b de g_b , par définition de r_b et par complétude par rapport du O .

(7) pour tout b de g_b , $r_b \Rightarrow O(\text{after } b \Rightarrow q_b)$

(8) pour tout a de g_a , pour tout b de g_b , $r_b \Rightarrow O(\text{after } a \Rightarrow q_a)$

(9) pour tout b de g_b , pour tout b' de g_b ,

$$r_b \wedge \text{at } b' \wedge \text{cond}(b') \Rightarrow O \text{ (after } b' \Rightarrow q_b)$$

(10) $(p \vee r) - [] \rightarrow (q \vee \bigvee_{b \in g_b} q_b)$, en utilisant la règle 11 et les théorèmes (5), (6), (7), (8) et (9).

(11) $(p \vee r) \wedge i \Leftrightarrow p_\alpha \wedge \text{at } g$, par définition de p , r et p_α .

$$(12) (q \vee \bigvee_{b \in g_b} q_b) \Leftrightarrow \bigvee_{\beta < \alpha} p_\beta$$

(13) $p_\alpha \wedge \text{at } g - [] \rightarrow \bigvee_{\beta < \alpha} p_\beta$, en utilisant la règle de l'invariance fatale et les théorèmes (1), (2) et (10).

Sous-cas trois : Le système comporte un seul processus. Pour terminer la preuve, il faut montrer que le système est complet pour le cas où il n'y a qu'un seul processus. On suppose donc :

$$\mathbb{F}\mathbb{J} \mid = \text{at } a \wedge p - [] \rightarrow \text{after } a \wedge q$$

où p et q sont des assertions ne portant pas sur le contrôle. Le programme ne comporte qu'une seule action a et nécessairement $\text{at } a \wedge p \Rightarrow \text{at } a \wedge \text{cond}(a) \wedge p$. Donc,

$$\text{at } a \wedge \text{cond}(a) \wedge p \Rightarrow \text{at } a \wedge \text{cond}(a) \wedge p \mathcal{U} \text{after } a \wedge q$$

et par justice simple on en déduit

$$\text{at } a \wedge p - [] \rightarrow \text{after } a \wedge q. \blacksquare$$

Notre système FEPS est donc sémantiquement complet. Cette preuve de complétude nous montre la méthode employée basée sur la notion de groupe critique. Cette méthode est équivalente à celle de Manna et Pnueli [1982] basée sur la notion de fonction d'aide à la convergence. Notre système est, par contre, axiomatique et permet de réaliser des preuves intéressantes que nous allons donner dans la suite.

V. EXEMPLES

Nous donnons deux exemples qui illustrent notre méthode. Pour ne pas allonger l'exposé, les exemples sont classiques.

V.1 EXEMPLE 1

Voici un exemple qui illustre la notion de justice et que nous avons conçu pour montrer l'intérêt de la notion de groupe critique. Il est inspiré d'un

exemple de calcul du pgcd de deux entiers proposé par Manna et Pnueli [1984]. x_0 et y_0 sont des valeurs entières arbitraires lues éventuellement.

```

X := x0; Y := y0;
COBEGIN
w1 : WHILE (X>0) ∨ (Y>0) DO a : X := X-1; OD
||
w2 : WHILE (X>0) ∨ (Y>0) DO b : Y := Y-1; OD
COEND
    
```

On va démontrer le théorème suivant :

$$\text{at } \mathbb{P} \wedge (x = x_0) \wedge (y = y_0) \rightarrow \text{after } \mathbb{P} \wedge ((x = 0) \vee (y = 0))$$

est vraie sous hypothèse d'exécution juste.

L'hypothèse de justice est indispensable car, si l'un des processus est retardé indéfiniment, alors le test $x > 0$ ou $y > 0$ risque de rester vrai indéfiniment, s'il l'était. En effet, le retard indéfini d'un processus conduit à la non-exécution de a ou de b qui sont les seules actions permettant de décroître x et y . Notons $\text{INIT} \equiv \text{at } \mathbb{P} \wedge (x = x_0) \wedge (y = y_0)$. Pour démontrer cette propriété, nous introduisons les assertions suivantes :

$$P(n) \equiv \text{in } w_1 \wedge \text{in } w_2 \wedge (\text{Sup}(x, 0) + \text{Sup}(y, 0) = n)$$

où $n \geq 0$ et $n \leq \text{Sup}(x_0, 0) + \text{Sup}(y_0, 0)$.

Nous donnons l'esquisse générale de la preuve.

(1) $\text{at } \mathbb{P} \wedge (x = x_0) \wedge (y = y_0) - [] \rightarrow \text{at } w_1 \wedge \text{at } w_2 \wedge (x = x_0) \wedge (y = y_0)$, en utilisant successivement l'axiome T12 relatif à la transition d'initialisation du COBEGIN . . . COEND, des règles 1, 2, 3 et enfin de la règle 3 (de justice).

(2) $p(n) - [] \rightarrow p(n-1)$, pour tout $n > 0$, pour démontrer cette propriété il faut procéder par cas sur les contrôles possibles puis à l'aide d'une règle d'induction adaptée on en déduit ce qui est demandée. Par exemple, le théorème suivant est utile dans cette preuve.

(3) $\text{at } w_1 \wedge \text{at } w_2 \wedge (x = x_0) \wedge (y = y_0) \Rightarrow p(n_0)$ où $n_0 = \text{Sup}(x_0, 0) + \text{Sup}(y_0, 0)$, puisque c'est un théorème de TH, d'après la définition de $p(n_0)$.

(4) $\text{at } \mathbb{P} \wedge (x = x_0) \wedge (y = y_0) - [] \rightarrow p(n_0)$, à l'aide de la règle 7 et du théorème (1) ci-dessus.

(5) $\bigvee_{0 \leq n \leq n_0} p(n) - [] \rightarrow p(0)$, par la règle d'induction avec le théorème (2) comme prémisse.

(6) $p(0) - [] \rightarrow (\text{after } w_1 \wedge \text{in } w_2 \wedge (\text{Sup}(x, y) = 0))$

$$\vee (\text{in } w_1 \wedge \text{after } w_2 \wedge (\text{Sup}(x, y)=0)).$$

cette propriété est vraie car les deux boucles ne peuvent que terminer dans ces cas.

$$(7) \quad (\text{after } w_1 \wedge \text{in } w_2 \wedge (\text{Sup}(x, y)=0)) \\ \vee (\text{in } w_1 \wedge \text{after } w_2 \wedge (\text{Sup}(x, y)=0)) \\ - [] \rightarrow (\text{after } w_1 \wedge \text{after } w_2 \wedge (\text{Sup}(x, y)=0)),$$

clairement les deux boucles ne peuvent que terminer.

$$(8) \quad (\text{at } \mathbb{P} \wedge (x=x_0) \wedge (y=y_0)) \vee \bigvee_{0 \leq n \leq n_0} p(n) \\ \vee (\text{after } w_1 \wedge \text{in } w_2 \wedge (\text{Sup}(x, y)=0)) \\ \vee (\text{in } w_1 \wedge \text{after } w_2 \wedge (\text{Sup}(x, y)=0)) \\ - [] \rightarrow (\text{after } w_1 \wedge \text{after } w_2 \wedge (\text{Sup}(x, y)=0)),$$

en utilisant la règle d'induction.

$$(9) \quad \text{at } \mathbb{P} \wedge (x=x_0) \wedge (y=y_0) \rightarrow \text{after } \mathbb{P} \wedge ((x=0) \vee (y=0)),$$

en appliquant la règle de l'invariance fatale. ■

Ce qui termine notre preuve où la règle de justice est utilisée pour démontrer le théorème (2). Dans cet exemple, les actions a_1 et a_2 sont déterminantes ou bien l'une d'entre elles ou les deux en même temps.

V. 2 Exemple 2

Considérons le problème classique de l'exclusion mutuelle pour deux processus :

```
Mutex := 1;
O : COBEGIN
l0 : LOOP l1 : P1(Mutex); L2 : CS1; l3 : V1(Mutex); l4 : END
||
m0 : LOOP m1 : P1(Mutex); m2 : CS1; m3 : V1(Mutex); m4 : END
COEND
1 :
```

Nous utilisons ici la récursivité de la méthode et la règle d'équité pour démontrer la propriété suivante :

$$\text{at } P_1(\text{Mutex}) - ([] \rightarrow \text{in } CS_1.$$

Nous démontrons tout d'abord le théorème suivant pour les propriétés d'invariance dont nous aurons besoin dans les preuves.

THÉORÈME 0 : Soit I l'assertion suivante :

$$(at\ 0 \wedge (mutex = 1)) \vee (at\ l_{0.1.2.3.4} \wedge at\ m_{0.1.2.3.4} \wedge (0 \leq mutex \leq 1)) \\ \wedge (\ll at\ l_{0.1.2.3.4} \gg + \ll at\ m_{0.1.2.3.4} \gg + mutex = 1)$$

où $at\ n_{0.1.2.3.4}$ et $INIT \equiv at\ O \wedge (mutex = 1)$. Alors $INIT \Rightarrow \square I$. ■

Nous ne donnons pas la preuve, afin d'aller à l'essentiel. Soit la suite d'assertions suivantes :

$$R_1 \equiv at\ n_2 \wedge (mutex = 0); \quad R_2 \equiv at\ n_3 \wedge (mutex = 0); \\ R_3 \equiv at\ n_4 \wedge (mutex = 1); \quad R_4 \equiv at\ n_1 \wedge (mutex = 1)$$

où n désigne l , si on raisonne sur les étiquettes du second processus, et m sinon.

THÉORÈME 1 : 1. $at\ l_1 \wedge I - [] \rightarrow at\ l_2$;

2. $at\ m_1 \wedge I - [] \rightarrow at\ m_2$. ■

Preuve : On peut raisonner indistinctement sur le premier ou le second processus et déduire les deux théorèmes à partir de la preuve d'un seul par processus. Considérons la preuve du premier théorème. Soit le sous-système forme du second processus et supposons que le processus 1 est bloqué sur l'opération P_1 (Mutex). Pour ce cas, n est à remplacer par m .

(1) $at\ m_2 \wedge (mutex = 0) \Rightarrow O (at\ m_3 \Rightarrow R_2)$, en utilisant les règles 1, 2 et l'axiome T15.

(2) $at\ m_2 \wedge (mutex = 0) \Rightarrow at\ m_2 \wedge (mutex = 0) u\ at\ m_3 \wedge R_2$, en appliquant la règle de l'opérateur u de numéro 5 et le théorème ci-dessus comme prémisse à celle-ci.

(3) $at\ m_2 \wedge (mutex = 0) - [] \rightarrow at\ m_3 \wedge R_2$, par la règle 10 et le théorème 2.

(4) $R_1 - [] \rightarrow R_2$, par la règle 7 et le théorème ci-dessus ainsi qu'une implication triviale.

(5) $R_2 - [] \rightarrow R_3$.

(6) $R_3 - [] \rightarrow R_4$, ces deux derniers théorèmes ont été dérivés comme le théorème 4.

(7) $(R_1 \vee R_2 \vee R_3 \vee R_4) - [] \rightarrow (R_3 \vee R_4)$, en utilisant la règle d'induction appliquée de façon très restrictive aux théorèmes 4, 5, 6.

Ces preuves ont été faites dans un système à un processus. On utilise la règle 12 pour déduire le théorème 10 qui suit :

(8) $(R_1 \vee R_2 \vee R_3 \vee R_4) \wedge \text{at } l_1 \wedge I \Leftrightarrow (R_1 \vee R_2 \vee R_3 \vee R_4) \wedge \text{at } l_1$, par définition de I et de \mathbb{TH} .

(9) $(R_3 \vee R_4) \wedge \text{at } l_1 \wedge I \Leftrightarrow (R_3 \vee R_4) \wedge \text{at } l_1$, par définition de I et de \mathbb{TH} .

(10) $(R_1 \vee R_2 \vee R_3 \vee R_4) \wedge \text{at } l_1 - [] \rightarrow (R_3 \vee R_4) \wedge \text{at } l_1$, par la règle 12 dite de la dimension et les théorèmes ci-dessus 7, 8, 9.

(11) $(R_3 \vee R_4) \wedge \text{at } l_1 \Rightarrow O(\text{at } l_2 \Rightarrow \text{true})$, à l'aide des règles 1 et 2 et de l'axiome T15).

(12) $(R_3 \vee R_4) \wedge \text{at } l_1 \wedge \text{at } m \Rightarrow O(\text{after } m \Rightarrow (R_3 \vee R_4) \wedge \text{at } l_1)$ ou m est une étiquette du second processus.

(13) $(R_1 \vee R_2 \vee R_3 \vee R_4) \wedge \text{at } l_1 - [] \rightarrow \text{at } l_2$, par la règle d'équité et les théorèmes 10, 11 et 12. ■

Nous montrons maintenant le théorème suivant qui est facilement déduit du précédent :

THÉORÈME 4 : 1. $\text{at } l_1 - ([]) \rightarrow \text{at } l_2$.

2. $\text{at } m_1 - ([]) \rightarrow \text{at } m_2$. ■

Preuve : Nous démontrons uniquement le premier puisque le système est symétrique.

(1) $\text{at } l_1 \wedge I \Leftrightarrow (R_1 \vee R_2 \vee R_3 \vee R_4) \wedge \text{at } l_1$, car c'est un théorème de \mathbb{TH} .

(2) $(R_1 \vee R_2 \vee R_3 \vee R_4) \wedge \text{at } l_1 - [] \rightarrow \text{at } l_2$, d'après le théorème précédent.

(3) $\text{INIT} \Rightarrow \square I$, d'après le théorème 0.

(4) $\text{at } l_1 - [] \rightarrow \text{at } l_2$, par la règle de l'invariance fatale et les théorèmes ci-dessus a, 2, 3.

(4) $\text{at } l_1 \rightarrow \text{at } l_2$, par la règle 13 et le théorème 5 ci-dessus. ■

Nous pourrions utiliser les treillis de preuve introduits par Owicki et Lamport [1982], pour représenter les preuves mais ceci sera rédigé ultérieurement avec le problème des philosophes. Nous avons utilisé la règle d'équité qui rend compte de l'idée « infiniment souvent » sous la forme d'une règle, alors que l'approche axiomatique d'Owicki et Lamport [1982] caractérise l'équité par un axiome et non pas une règle. Ainsi, notre approche est plus constructive.

VI. CONCLUSION

Nous avons présenté un système axiomatique basé sur la logique temporelle, FEPS. FEPS permet de prouver des propriétés de fatalité sous hypothèse

d'exécution équitable. Ce système est caractérisé par trois règles essentielles. Une règle de justice, qui axiomatise l'hypothèse de justice est construite à l'aide de l'opérateur *Until* faible et permet de construire des propriétés de fatalité du type « $p - [] \rightarrow q$ », qui sont essentielles pour les propriétés générales de fatalité. Une règle d'équité qui permet de caractériser l'idée de « infiniment souvent activable », pour un groupe d'actions atomiques nécessite une règle de la dimension, qui permet de déduire une preuve dans un système à n processus, à partir d'une preuve dans un système à $n' < n$ processus. L'aspect récursif de cette méthode correspond à l'idée suivante : en un état donné atteint, si m processus sont bloqués devant une opération $P(s)$, il est clair que le système à n processus se comporte comme le système à $n - m$ processus et donc, la preuve est faite, dans un système à $n - m$ processus. La méthode de Lehman, Pnueli et Stavi [1981] présente la même caractéristique mais, n'est pas axiomatique et s'intéresse uniquement à la terminaison équitable. La caractérisation des propriétés de fatalité a été réalisée par la construction d'un opérateur $F[q]$ pour chaque assertion q . Cet opérateur $F[q]$ est en quelque sorte un *wp* pour l'équité, comme celui de Park [1981]. Cet opérateur est la clé de la démonstration de complétude, mais il suppose l'expressivité du langage d'assertions. Il met en évidence la notion de groupe critique d'actions atomiques conduisant à une assertion donnée. Cette notion nous a permis de construire les règles indispensables, mais elle guide l'utilisateur dans sa preuve.

Les développements futurs tendront à assister l'utilisateur dans sa preuve et à le guider. Nous ferons des preuves d'algorithmes plus complexes et essaierons de les améliorer par les preuves. Notre système peut être adapté à d'autres primitives de synchronisation et rendu compositionnel (cf. Barringer, Kuiper et Pnueli [1984] et Gerth [1984]).

REMERCIEMENTS

Je remercie Monsieur P. Cousot pour les conseils qu'il a bien voulu me donner et qui m'ont permis d'améliorer la rédaction de cet article. Grâce aux remarques très pertinentes d'un référent, j'ai pu me rendre compte des lacunes de ma rédaction et je veux le remercier ici.

BIBLIOGRAPHIE

1. APT et DELPORTE [1983], *Syntax-Directed Analysis of Liveness Properties*, RR83-42, octobre 1983, L.I.T.P., Université de Paris-VII, France.
2. BARRINGER, KUIPER et PNUELI [1984], *Now You May Compose Your Temporal Logic Specification*, dans A.C.M.-Proceedings Theory of Computing.

3. BURSTALL [1974], *Program Proving as Hand Simulation with a Little Induction*, dans Proceedings I.F.I.P., 1974, p. 308-312, Amsterdam 1974, The Netherlands.
4. COUSOT [1985], *Fondements des méthodes de preuve d'invariance et de fatalité de programmes parallèles*, Thèse d'état, novembre 1985, I.N.P.L.
5. FLOYD [1967], *Assigning Meaning to Programs*, dans Proc. A.M.S. Symposium dans Applied Math. Amer. Math. Soc., 1967, p. 19-31.
6. GERTH [1984], *TRANSITION LOGIC: How to Reason About Temporal Properties of Programs in a Compositional Way*, dans RUU-CS-83-17, février 1984, Rijksuniversiteit Utrecht, The Netherlands.
7. HOARE [1969], *An Axiomatic Basis for Computer Programming*, dans Communications of A.C.M., vol. 12, 1969, p. 576-583.
8. LAMPORT [1980 a], *SOMETIME is Sometimes Better than ALWAYS*, dans Proceedings of the A.C.M. Symposium on the Principles of Programming Languages, 1980.
9. LAMPORT [1980 b], *The HOARE Logic of Concurrent Programs*, dans Acta Informatica, vol. 14, 1980, p. 21-37.
10. LEHMAN, PNUELI et STAVI [1981], *Impartiality, Justice and Fairness: the Ethics of Concurrent Termination*, dans I.C.A.L.P., 1981, L.N.C.S., n° 115, p. 264-277.
11. MANNA et PNUELI [1982], *Verification of Concurrent Programs: Proving Eventuality by Well-Founded Ranking*, dans Report N.STAN-CS-82-915, mai 1982, Department of Computer Science, Stanford University, U.S.A.
12. MANNA et PNUELI [1983 a], *How to Cook Your Temporal Proof System for Your Pet Language*, dans Proceedings of P.O.P.L., 1983, Austin, Texas, janvier 1983.
13. MANNA et PNUELI [1983 b], *Proving Precedence Properties: the Temporal Way*, dans Report N.STAN-CS-83-964, Department of Computer Science, Stanford University, U.S.A.
14. MANNA et PNUELI [1984], *Adequate Proof Principles for Invariance and Liveness Properties of Concurrent Programs*, dans Science of Computer Programming, vol. 4, 1984, p. 257-289, North-Holland.
15. OWICKI et LAMPORT [1982], *Proving Liveness Properties of Concurrent Programs*, dans ACM-TOPLAS, vol. 4, (3), 1982, p. 455-495.
16. PARK [1981], *A Predicate Transformer for Weak Fair Iteration*, dans Proc. of The sixth I.B.M. Symposium on mathematical Foundations of Computer Science, Hakone, Japan, 1981.
17. PNUELI [1977], *The Temporal Logic of Programs*, dans Proc. 18th Symposium on Foundations of Computer Science, Ri, 1977, p. 46-57.