5. PREUVES DE FATALITE

5. PREUVES DE FATALITE

5.1 RELATIONS ENTRE SEMANTIQUES CONSERVANT LA FATALITE

5.1.1 CONSERVATION DE PROPRIETES DE FATALITE PAR INCLUSION DE SEMANTIQUES

5.1.2 CONSERVATION DE PROPRIETES DE FATALITE POUR DES SEMANTIQUES CONCORDANTES A DES RELATIONS ENTRE ETATS ET ACTIONS PRES

5.2 PRINCIPES D'INDUCTION "A LA FLOYD"

5.2.1 RAPPEL DE LA METHODE DE FLOYD (DITE DES ASSERTIONS INVARIANTES ET DE L'ORDRE BIEN-FONDE) POUR DEMONTRER LA CORRECTION TOTALE DE PROGRAMMES SEQUENTIELS

5.2.1.1 Correction partielle

5.2.1.2 Absence d'erreurs à l'exécution (ou absence d'états de blocage)

5.2.1.3 Terminaison

5.2.2 LE PRINCIPE D'INDUCTION DE BASE POUR LES SEMANTIQUES CLOSES

5.2.3 PRINCIPES D'INDUCTION EQUIVALENTS POUR LES SEMANTIQUES CLOSES

5. PREUVES DE FATALITE

Nous étudions dans ce chapitre les méthodes de preuve de propriétés de fatalité des programmes (cf. 3.3) en utilisant la même approche que dans le chapitre précédent. Ceci consiste à partir des méthodes de preuves classiques pour démontrer la correction totale des programmes séquentiels (Floyd [67], Burstall [74] principalement) en les formalisant à l'aide de principes d'induction pour démontrer des propriétés de fatalité de systèmes de transition ce qui permet de considérer une classe de propriétés plutôt qu'une propriété particulière et de faire abstraction d'un langage de programmation particulier ou d'une méthode particulière de présentation de la preuve. Il est ensuite plus facile de faire des généralisations au cas des systèmes de transition non déterministes (et donc aux programmes parallèles, en particulier asynchrones) puis aux sémantiques non closes (et donc en particulier, aux programmes parallèles équitables). Cette formalisation facilite la compréhension des nombreuses variantes de ces méthodes de preuve ainsi que les preuves de correction, de complétude sémantique et d'équivalence de ces variantes.

La méthode la plus connue pour démontrer la correction totale des programmes séquentiels est la méthode de Floyd [67] dite des "assertions invariantes et de l'ordre bien-fondé" qui consiste à décomposer la preuve en une preuve de correction partielle, une preuve d'absence d'erreurs à l'exécution (ou de blocages) et une preuve de terminaison (en exhibant une quantité qui décroit à chaque pas du programme ou à chaque cycle dans une boucle mais qui ne peut pas décroître indéfiniment). Le chapitre précédent nous a déjà permis de formaliser

la partie preuves d'invariance (correction partielle, absence de blocages) et de manière générale la preuve de terminaison se formalise à l'aide de la notion d'ensemble bien-fondé.

Lorsque le non-déterminisme n'est pas fini, Dijkstra [76, 82] a montré qu'il n'est pas toujours possible d'utiliser des relations dont le rang (on dit aussi ordre) est strictement inférieur à ω et nous caractérisons les relations bien fondées qui sont nécessaires et suffisantes pour la complétude sémantique dans divers cas de non-déterminisme borné généralisant le non-déterminisme fini de Dijkstra.

Lorsque la sémantique du programme n'est pas close, la méthode d'induction sur les calculs de Floyd n'est pas applicable sans recourir à des variables auxiliaires. Une première approche consiste à appliquer la méthode de Floyd à une relation de transition auxiliaire (portant sur les états et les variables d'histoire) qui engendre exactement l'ensemble de traces original. Une seconde approche qui généralise l'utilisation de points de coupure des boucles dans la méthode de Floyd pour les programmes séquentiels, consiste à utiliser des points de coupure (où une fonction de terminaison décroit strictement) choisis dynamiquement c'est-à-dire en fonction de l'histoire des calculs cumulée dans les variables auxiliaires. Nous montrerons ensuite que ces deux approches sont en fait fortement équivalentes.

La méthode de Burstall [74] dite des "assertions intermittentes" pour démontrer la correction totale des programmes séquentiels est moins connue, souvent ignorée des ouvrages de base (nous n'avons pas trouvé de références dans Liveray [78] ou Berg et al. [82] par exemple) et est sujette à polémiques (Manna-Waldinger [78], Gries [79]). Nous nous sommes intéressés à cette méthode parce qu'étant présentée de manière très différente

de la méthode de Floyd, nous ne comprenions pas bien le rapport entre ces deux méthodes.

A partir de la déscription donnée par Burstall et Manna-Waldinger de la méthode des assertions intermittentes (principalement à l'aide d'exemples) nous dérivons un principe d'induction dont nous montrons la correction. Il est sémantiquement complet mais sous une condition qui porte sur les traces d'exécution et la propriété de fatalité considérée. En particulier cette condition est remplie quand on s'intéresse à la correction totale ou bien à des propriétés de fatalité qui s'expriment à l'aide d'assertions unaires sur des états. Cependant nous faisons la conjecture que le principe d'induction de base n'est pas complet quand on considère des propriétés de fatalité arbitraires qui sont binaires (c'est-à-dire reliant les valeurs des états à des instants différents dans le temps) et lorsqu'on ne s'autorise pas l'emploi de variables auxiliaires.

Cette conjecture nous conduit à une généralisation de la méthode de Burstall en utilisant une induction transfinie (de manière à prendre en compte le non-déterminisme non borné) et en introduisant l'utilisation de variables auxiliaires sous la forme très limitée d'assertions ternaires (qui permettent de relier l'état des variables au début du programme ainsi qu'à deux autres instants différents au cours du calcul).

A partir de ce principe d'induction généralisé nous dérivons une série de principes d'induction de manière à élargir le champ des formes de preuves permises. Ceci nous permet également de formuler la méthode de Burstall sous des formes de plus en plus abstraites pour n'en retenir finalement que l'essence.

Tous les principes d'induction considérés sont corrects et sémantiquement complets (comme le montre l'argument de Manna-Waldinger[78]

qui consiste à dire qu'une preuve à la Floyd peut s'exprimer par la
nous démontrons un résultat de
en ce sens que les propositions
dans une preuve par la méthode
des assertions intermittentes telle e nous l'avons généralisée peuvent
être choisis librement (du mo sous une condition nécessaire et
suffisante que nous stipulons avec précision).

La formalisation identique des méthodes de Floyd et Burstall nous permet de comprendre simplement leur rapport. Non seulement (comme l'ont montré Manna-Waldinger) toute preuve à la Floyd peut s'exprimer par la méthode de Burstall mais toute preuve à la Floyd est une preuve à la Burstall ! En effet, le principe d'induction qui exprime l'essence de la méthode de Burstall (convenablement généralisée comme nous le proposons) montre clairement que la méthode de Floyd est un simple cas particulier de la méthode de Burstall. Par analogie avec la programmation, ce rapport de la méthode de Burstall à celle de Floyd est similaire au rapport qu'a une programmation itérative avec un seul programme principal avec une programmation utilisant des sous-programmes éventuellement récursifs.

Pour tirer des conclusions pratiques de cette remarque, il nous reste à élaborer une méthode de présentation des preuves qui soit aussi bien utilisable pour des preuves à la Floyd qu'à la Burstall. C'est pourquoi nous proposons une présentation graphique des preuves de fatalité sous la forme de chartes de preuve qui peuvent contenir des cycles mais de manière structurée. Nous montrons que cette présentation des preuves est correcte, sémantiquement complète et convient pour démontrer des propriétés de fatalité des programmes parallèles.

Pour conclure sur la comparaison de ces méthodes, nous montrons qu'elles sont fortement équivalentes, en ce sens qu'une preuve par la méthode de Burstall peut se réécrire en une preuve par la méthode de Floyd (de la même façon qu'il est toujours possible d'éliminer la récursivité et les sous-programmes d'un programme).

Nous avons choisi de présenter ce chapitre en allant du particulier (méthode de Floyd) au général (méthode de Burstall) ce qui correspond à l'historique de la découverte de ces méthodes et à une complexité croissante des principes d'induction. Bien que moins satisfaisante pour l'esprit qu'une démarche descendante du général au particulier, cette présentation nous a semblé préférable parcequ'elle gradue les difficultés.

5.1 RELATIONS ENTRE SEMANTIQUES CONSERVANT LA FATALITE

Nous étudions dans ce paragraphe les relations entre sémantiques (définies en 2.5 et 2.6) qui conservent les propriétés de fatalité. Ceci permet en particulier de justifier très simplement les méthodes de preuve de programmes basées sur une transformation du programme.

5.1.1 CONSERVATION DE PROPRIETES DE FATALITE PAR INCLUSION DE SEMANTIQUES

De manière évidente d'après la définition 3.3.1, nous avons :

Théorème 5.1.1-1

Si $\langle S,A,\Sigma\rangle \subseteq \langle S',A',\Sigma'\rangle$, $\phi,\psi \in (S'\times S' \to \{tt,ff\})$ alors

$\Rightarrow$ [ψ est fatale sous invariance de ϕ pour $\langle S',A',\Sigma'\rangle$]

$\not\Leftarrow$ [ψ est fatale sous invariance de ϕ pour $\langle S,A,\Sigma\rangle$]

Pour voir que la réciproque ($\not\Leftarrow$) n'est pas vraie il suffit de considérer le contre-exemple suivant :

Exemple 5.1.1-1

La sémantique $\langle S,A,\Sigma\rangle$ définie dans l'exemple 2.1.2-2 ($S=\{0,1\}$, dans la sémantique $\langle S',A',\Sigma'\rangle$ définie dans l'exemple 2.1.2-3 ($S=\{0,1\}$, $A=\{a,b\}$, Wfair $\langle A\rangle(\langle S',A',\Sigma'\rangle)$). ψ définie par $\psi(0)=ff$ et $\psi(1)=tt$ est fatale pour $\langle S,A,\Sigma\rangle$ mais pas pour $\langle S',A',\Sigma'\rangle$.

Corollaire 5.1.1 ~ 2

Si $\langle S, A, \Sigma \rangle = \underset{\sim}{Efus}(\langle S, A, \Sigma \rangle)$ et $\langle S, A, \Sigma \rangle = \underset{\sim}{Retps}(\langle S, A, \Sigma \rangle)$ alors

[ψ est fatale sous invariance de ϕ pour $\underset{\sim}{Rtran}(\langle S, A, \Sigma \rangle)$]

[ψ est fatale sous invariance de ϕ pour $\langle S, A, \Sigma \rangle$]

5.1.2 CONSERVATION DE PROPRIETES DE FATALITE POUR DES SEMANTIQUES CONCORDANTES A DES RELATIONS ENTRE ETATS ET ACTIONS PRES

Théorème 5.1.2-1

Si $\simeq \langle rs, ra \rangle (\langle S, A, \Sigma \rangle, \langle S', A', \Sigma' \rangle)$

$\wedge$

$rs^{-1} \circ \phi \circ rs \Rightarrow \phi'$

$\wedge$

$rs^{-1} \circ \psi \circ rs \Rightarrow \psi'$

alors

$\Rightarrow$ [ψ est fatale sous invariance de ϕ pour $\langle S, A, \Sigma \rangle$]

$\not\Leftarrow$ [ψ' est fatale sous invariance de ϕ pour $\langle S', A', \Sigma' \rangle$]

Démonstration

($\rightarrow$) Si ψ est fatale sous invariance de ϕ pour $\langle S, A, \Sigma \rangle$ et $p' \in \Sigma'$ alors il existe $p \in \Sigma$ tel que $|p| = |p'|$ et $\forall R \in |p|. rs(p_R, p'_R)$. Comme $\exists i \in |p|. [\forall j \in i. \phi(p_0, p_j) \wedge \psi(p_0, p_i)]$ il vient $\exists i \in |p'|. [\forall j \in i. rs^{-1} \circ \phi \circ rs(p'_0, p'_j) \wedge rs^{-1} \circ \psi \circ rs(p'_0, p'_i)]$ et donc $\exists i \in |p'|. [\forall j \in i. \phi'(p'_0, p'_j) \wedge \psi'(p'_0, p'_i)]$.

($\not\leftarrow$) Choisir $S = \{0,1\}$, $A = \phi$, $\Sigma = \{0,1\}$, $S' = \{0'\}$, $A' = \phi$, $\Sigma' = \{0'\}$, $rs(0,0')$, $rs(1,0')$, $\psi(0,0)$, $\neg\psi(1,1)$, $\phi = \psi$. Nous avons $\phi' = \psi' = rs^{-1} \circ \psi \circ rs\ (0',0')$ et donc ψ' est fatale (sous invariance de ϕ') pour Σ' mais ψ n'est pas fatale (sous invariance de ϕ) pour Σ.

□

La réciproque est vraie si nous ajoutons des conditions supplémentaires :

Théorème 5.1.2≈2

Si $\simeq\langle rs, ra\rangle(\langle S, A, \Sigma\rangle, \langle S', A', \Sigma'\rangle)$

$\wedge\ rs^{-1} \subseteq \phi \subseteq rs \Rightarrow \phi' \quad \wedge \quad rs^{-1} \subseteq \psi \subseteq rs \Rightarrow \psi'$

$\wedge\ ra \subseteq \phi' \circ rs^{-1} \Rightarrow \phi \quad \wedge \quad ra \subseteq \psi' \subseteq rs^{-1} \Rightarrow \psi$

alors

[ψ est fatale sous invariance de ϕ pour $\langle S, A, \Sigma\rangle$]

$\Longleftrightarrow$

[ψ' est fatale sous invariance de ϕ' pour $\langle S', A', \Sigma'\rangle$]

Démonstration

($\Rightarrow$) cf. 5.1.2≈1

($\Leftarrow$) Si ψ' est fatale sous invariance de ϕ' pour $\langle S', A', \Sigma'\rangle$ et $p \in \Sigma$ alors $\Sigma' = \simeq\langle rs, ra\rangle[\Sigma]$ implique l'existence de $p' \in \Sigma$ tel que $\simeq\langle rs, ra\rangle(p, p')$ et $\exists i \in |p'| = |p|$. $(\forall j \in i.\ \phi'(p'_0, p'_j) \wedge \psi(p_0, p_i))$. Par conséquent nous avons $\forall j \in i.(rs(p_0, p'_0) \wedge \phi'(p'_0, p'_j) \wedge rs^{-1}(p'_j, p_j)) \wedge (rs(p_0, p'_0) \wedge \psi'(p'_0, p'_i) \wedge rs^{-1}(p'_i, p_i))$ ce qui implique $\forall j \in i.\ \phi(p_0, p_j) \wedge \psi(p_0, p_i)$.

□

Dans le cas particulier d'une concordance entre sémantiques à une fonction entre états près, nous obtenons le corollaire suivant :

Corollaire 5.1.2≈3

Si $\langle S', A', \Sigma'\rangle = \simeq\langle f_S\rangle(\langle S, A, \Sigma\rangle)$

$\wedge\ \phi(A_1, A_2) = \phi'(f_A(A_1), f_A(A_2))$

$\wedge\ \psi(A_1, A_2) = \psi'(f_A(A_1), f_A(A_2))$

alors

[ψ est fatale sous invariance de ϕ pour $\langle S, A, \Sigma\rangle$]

$\Longleftrightarrow$

[ψ' est fatale sous invariance de ϕ' pour $\langle S', A', \Sigma'\rangle$]

5.2 PRINCIPES D'INDUCTION "A LA FLOYD"

Par abstraction à partir de la méthode de Floyd [67] (méthode des assertions invariantes et de l'ordre bien-fondé pour montrer la correction totale de programmes séquentiels), nous proposons des principes d'induction pour démontrer les propriétés de fatalité de systèmes de transition. Nous démontrons que ces principes d'induction sont corrects, sémantiquement complets et équivalents. Ceci formalise la méthode de Floyd indépendemment d'un langage de programmation particulier et d'un langage d'assertions particulier et la généralise au cas de systèmes de transition non-déterministes et donc aux programmes parallèles. Considérant différentes classes de nondéterminisme borné, nous caractérisons les relations bien-fondées correspondantes qui sont nécessaires et suffisantes pour la complétude sémantique.

Quand la sémantique n'est pas close (par exemple dans le cas d'une exécution parallèle équitable), la méthode de Floyd ne peut pas s'appliquer sans utiliser des variables auxiliaires. Une première approche consiste à appliquer la méthode de Floyd à une relation de transition auxiliaire (portant sur les états et des variables d'histoire) qui engendre exactement la sémantique originale. Une seconde approche qui généralise l'utilisation de points de coupure des boucles dans la méthode de Floyd pour les programmes séquentiels, consiste à utiliser des points de coupure (où une fonction de terminaison décroît strictement) choisis dynamiquement c'est-à-dire en fonction de l'histoire des calculs cumulée dans les variables auxiliaires. Nous montrons que ces deux approches sont fortement équivalentes.

5.2.1 RAPPEL DE LA METHODE DE FLOYD (DITE DES ASSERTIONS INVARIANTES ET DE L'ORDRE BIEN-FONDE) POUR DEMONTRER LA CORRECTION TOTALE DE PROGRAMMES SEQUENTIELS

Floyd[67] considère des programmes séquentiels P_s (comme en 2.8.1) avec des états de la forme $\langle c, m\rangle \in S[\![P_s]\!]$ où $c \in C[\![P_s]\!]$ est un point de contrôle et $m \in \mathcal{M}$ est un état mémoire (affectant des valeurs aux variables). Soient $\psi \in (\mathcal{M}\times\mathcal{M} \to \{tt, ff\})$ une spécification de sortie et $\sigma \in (S[\![P_s]\!] \to \{tt, ff\})$ une fonction caractérisant les états de sortie. Soit $\bar{\psi} \in (S[\![P_s]\!] \times S[\![P_s]\!] \to \{tt, ff\})$ telle que $\bar{\psi}(\langle \underline{c}, \underline{m}\rangle, \langle \bar{c}, \bar{m}\rangle) = \psi(\underline{m}, \bar{m})$.

La correction totale est une propriété de fatalité de la forme :

$$\forall p \in \Sigma\langle S, A, t, \varepsilon\rangle.\ \exists i \in |p|.\ [(\forall j \in i.\ \neg\sigma(p_j)) \wedge \sigma(p_i) \wedge \bar{\psi}(p_0, p_i)]$$

D'après la méthode de Floyd, on démontre la correction totale en démontrant d'abord la correction partielle, puis l'absence d'erreurs à l'exécution (c'est-à-dire d'après 2.8.1.2.4, l'absence d'états de blocage) et finalement la terminaison.

5.2.1.1 Correction partielle

La méthode de Floyd[67]-Naur[66] de preuve de correction partielle consiste à d'abord associer une assertion $P_c \in (\mathcal{M}\times\mathcal{M} \to \{tt, ff\})$ à chaque point de contrôle c du programme ($P_c(\underline{m}, m)$ reliant l'état mémoire courant m à l'état mémoire initial $\underline{m}$) puis montrer que ces assertions sont invariantes et finalement montrer que l'assertion associée aux états finaux implique la spécification d'entrée-sortie.

Pour montrer que ces assertions sont invariantes, on doit d'abord montrer que l'assertion d'entrée est vraie :

$\forall \underline{c} \in C[\![Ps]\!],\ \underline{m} \in \mathcal{M}.\ [\varepsilon(\langle \underline{c}, \underline{m}\rangle) \Rightarrow P_{\underline{c}}(\langle \underline{m}, \underline{m}\rangle)]$

Puis pour chaque commande du programme, on doit montrer que si le contrôle était au point c avec P_c vraie avant l'exécution de la commande alors après exécution (si elle se termine correctement), le contrôle doit être en c' avec $P_{c'}$ vraie :

$\forall c, c' \in C[\![Ps]\!],\ \underline{m}, m, m' \in \mathcal{M}.$

$([P_c(\underline{m}, m) \wedge \neg\sigma(\langle c, m\rangle) \wedge t_{\underline{a}}(\langle c, m\rangle, \langle c', m'\rangle)] \Rightarrow P_{c'}(\underline{m}, m'))$

Finalement, on doit montrer que si l'exécution atteint un point de sortie du programme alors l'assertion associée à ce point doit impliquer la spécification :

$\forall \bar{c} \in C[\![Ps]\!],\ \underline{m}, \bar{m} \in \mathcal{M}.\ ([P_{\bar{c}}(\underline{m}, \bar{m}) \wedge \sigma(\langle \bar{c}, \bar{m}\rangle)] \Rightarrow \psi(\underline{m}, \bar{m}))$

5.2.1.2 Absence d'erreurs à l'exécution (ou absence d'états de blocage)

Si des opérations partielles sont utilisées dans un programme, alors une preuve d'absence d'erreurs à l'exécution doit montrer qu'elles ne donnent pas de résultats indéfinis. Si par convention, la sémantique opérationnelle est définie de sorte que les états conduisant à des résultats indéfinis soient des états de blocage alors une preuve d'absence d'erreurs à l'exécution consiste à montrer que les états accessibles qui ne sont pas des états finaux doivent avoir au moins un successeur :

$\forall c \in C[\![Ps]\!],\ \underline{m}, m \in \mathcal{M}.$

$([P_c(\underline{m}, m) \wedge \neg\sigma(\langle c, m\rangle)] \Rightarrow [\exists c' \in C[\![Ps]\!], m' \in \mathcal{M}.\ t_{\underline{a}}(\langle c, m\rangle, \langle c', m'\rangle)])$

5.2.1.3 Terminaison

La méthode de preuve de terminaison de Floyd consiste d'abord à associer à chaque point de contrôle $c \in C[\![Ps]\!]$ du programme, une fonction de terminaison $f_c \in (\mathcal{M} \times \mathcal{M} \to Rng(f_c))$.

Puis on montre que les valeurs de cette fonction appartiennent à un bon-ordre $\langle W, \prec \rangle$, $W \subseteq Rng(f_c)$:

$\forall c \in C[\![Ps]\!]$, $\underline{m}, m \in \mathcal{M}$.

$$([P_c(\underline{m}, m) \wedge \neg\delta(\langle c, m\rangle)] \Rightarrow f_c(\underline{m}, m) \in W)$$

Finalement, on montre qu'après chaque exécution d'une commande, la valeur courante de la fonction de terminaison associée à son point de sortie est strictement plus petite que la valeur de la fonction de terminaison associée à son point d'entrée :

$\forall c, c' \in C[\![Ps]\!]$, $\underline{m}, m, m' \in \mathcal{M}$.

$$([P_c(\underline{m}, m) \wedge \neg\delta(\langle c, m\rangle) \wedge t_a(\langle c, m\rangle, \langle c', m'\rangle)] \Rightarrow [f_{c'}(\underline{m}, m') \prec f_c(\underline{m}, m)])$$

5.2.2 LE PRINCIPE D'INDUCTION DE BASE POUR LES SEMANTIQUES CLOSES

Au lieu d'utiliser des assertions locales P_c associées aux points de contrôle $c \in C[\![Ps]\!]$, nous pouvons utiliser un invariant global J tel que (cf. 4.2.1.1) :

$$J \in (S[\![Ps]\!]^2 \to \{tt, ff\})$$

$$J(\langle \underline{\sigma}, \underline{m} \rangle, \langle c, m \rangle) = P_c(\underline{m}, m)$$

et une fonction de terminaison globale telle que :

$$f \in (S[\![Ps]\!]^2 \to \cup\{Rng(f_c) : c \in C[\![Ps]\!]\})$$

$$f(\langle \underline{\sigma}, \underline{m} \rangle, \langle c, m \rangle) = f_c(\underline{m}, m)$$

Il est alors trivial de vérifier que les conditions de vérification de Floyd (définies comme en 5.2.1) sont équivalentes aux suivantes :

Correction partielle :

$$[\ (\varepsilon(\underline{A}) \Rightarrow J(\underline{A}, \underline{A}))$$
$$\wedge ([J(\underline{A}, A) \wedge \neg\sigma(A) \wedge t_{\underline{a}}(A, A')] \Rightarrow J(\underline{A}, A'))$$
$$\wedge ([J(\underline{A}, \bar{A}) \wedge \sigma(\bar{A})] \Rightarrow \bar{\Psi}(\underline{A}, \bar{A}))]$$

- Absence d'états de blocage :

$$([J(\underline{A}, A) \wedge \neg\sigma(A)] \Rightarrow [\exists A' \in S[\![Ps]\!] . t_{\underline{a}}(A, A')])$$

- Terminaison :

$$[\ ([J(\underline{A}, A) \wedge \neg\sigma(A)] \Rightarrow f(\underline{A}, A) \in W)$$
$$\wedge ([J(\underline{A}, A) \wedge \neg\sigma(A) \wedge t_{\underline{a}}(A, A')] \Rightarrow [f(\underline{A}, A') \prec f(\underline{A}, A)])\ ]$$

Si maintenant nous posons :

$$\Phi, \Psi \in (S[\![Ps]\!]^2 \to \{tt, ff\})$$

$$\Phi(\underline{A}, A) = [\neg\sigma(A)]$$

$$\Psi(\underline{A}, \bar{A}) = [\sigma(\bar{A}) \wedge \bar{\Psi}(\underline{A}, \bar{A})]$$

alors nous pourrons vérifier aisément que la méthode de Floyd repose sur le principe d'induction suivant :

$$
\begin{array}{l}
(\exists J \in (S^2 \to \{tt, ff\}),\ f \in (S^2 \to \underline{\underline{Rng}}(f)),\ W \subseteq \underline{\underline{Rng}}(f),\ \prec \in (\underline{\underline{Rng}}(f) \times Rng(f) \to \{tt, ff\}). \\
\quad \underline{\underline{wo}}(W, \prec) \\
\quad [\forall \underline{\Delta}, \Delta \in S. \\
\qquad (\varepsilon(\underline{\Delta}) \Rightarrow J(\underline{\Delta}, \underline{\Delta})) \\
\qquad \wedge \\
\qquad (J(\underline{\Delta}, \Delta) \Rightarrow \Psi(\underline{\Delta}, \Delta) \\
\qquad\qquad \vee [\Phi(\underline{\Delta}, \Delta) \wedge f(\underline{\Delta}, \Delta) \in W \wedge \exists \Delta' \in S, a \in A.\ t_a(\Delta, \Delta') \wedge \\
\qquad\qquad\quad \forall \Delta' \in S, a \in A.\ (t_a(\Delta, \Delta') \Rightarrow [J(\underline{\Delta}, \Delta') \wedge f(\underline{\Delta}, \Delta') \prec f(\underline{\Delta}, \Delta)])])])
\end{array}
\qquad (\mathcal{F}_1')
$$

où $\underline{\underline{wo}}(W, \prec)$ caractérise les bons-ordres sur W.

5.2.3 PRINCIPES D'INDUCTION EQUIVALENTS POUR LES SEMANTIQUES CLOSES

Des variantes du principe d'induction de base sont souvent utilisées. Nous introduisons maintenant des transformations successives qui conduisent à différents principes d'induction. Nous montrons que tous ces principes d'induction sont équivalents au principe d'induction de base ($\mathcal{F}_1$).

Le co-domaine de la fonction de terminaison f peut toujours être choisi de sorte qu'il coïncide avec le sous-ensemble W de l'ordre $\prec$:

$$(\exists J \in (S^2 \to \{tt,ff\}),\ f \in (S^2 \to \underline{Rng}(f)),\ \prec \in (\underline{Rng}(f) \times \underline{Rng}(f) \to \{tt,ff\}).$$

$$\begin{array}{l} \underline{wo}(\underline{Rng}(f), \prec) \\ [\forall \underline{s}, s \in S. \\ \quad (\epsilon(\underline{s}) \Rightarrow J(\underline{s},\underline{s})) \\ \quad \wedge \\ \quad (J(\underline{s},s) \Rightarrow \Psi(\underline{s},s) \\ \qquad \vee \\ \qquad [\Phi(\underline{s},s) \wedge \exists s' \in S, a \in A.\, t_a(s,s') \wedge \\ \qquad\quad \forall s' \in S, a \in A.\, (t_a(s,s') \Rightarrow [J(\underline{s},s') \wedge f(\underline{s},s') \prec f(\underline{s},s)])])] \end{array} \qquad (\mathcal{F}_2)$$

Quand, dans les démonstrations d'équivalences, il sera nécessaire d'utiliser en même temps des objets qui sont différents dans les principes d'induction ($\mathcal{F}_1$) et ($\mathcal{F}_2$) mais auxquels pour simplifier nous avons donné le même nom (comme $J, f, \prec, \ldots$) nous utiliserons des indices (comme $J_1, J_2, f_1, f_2, \prec_1, \prec_2, \ldots$).

Théorème 5.8.3 v1

Démonstration

Choisir $J_2(\underline{a},a) = [(f_1(\underline{a},a) \in W_1 \wedge J_1(\underline{a},a)) \vee \Psi(\underline{a},a)]$, $Rng(f_2) = (W_1 \cup \{\perp\})$ avec $\perp \notin W_1$, $f_2(\underline{a},a) = ((f_1(\underline{a},a) \in W_1) \rightarrow f_1(\underline{a},a) \mid \perp)$, $w' \prec_2 w$ si et seulement si $[(w' = \perp \wedge w \in W_1) \vee (w' \in W_1 \wedge w \in W_1 \wedge w' \prec_1 w)]$.

□

Il n'est pas nécessaire d'associer une fonction de terminaison à tous les points de contrôle du programme mais seulement aux points de coupure des boucles :

$$
\begin{array}{l}
(\exists K \subseteq S,\ J \in (K \times K \times S \rightarrow \{tt, ff\}),\ f \in (K^2 \rightarrow Rng(f)),\ \prec \in (Rng(f) \times Rng(f) \rightarrow \{tt, ff\}). \\
\quad \underline{Cutset}\ \langle S, A, t, \varepsilon \rangle (K) \\
\quad \wedge\ \underline{wo}\,(Rng(f), \prec) \\
\quad \wedge\ [\forall \underline{a}, a \in K,\ a' \in S. \\
\qquad (\varepsilon(\underline{a}) \Rightarrow J(\underline{a},\underline{a},\underline{a})) \\
\qquad \wedge\ (J(\underline{a},a,a') \Rightarrow \Psi(\underline{a},a') \\
\qquad\qquad \vee\ [\Phi(\underline{a},a') \wedge \exists a'' \in S, q \in A.\ t_q(a',a'') \wedge \forall a'' \in S, q \in A. \\
\qquad\qquad\quad t_q(a',a'') \Rightarrow [\ (a'' \in K \wedge f(\underline{a},a'') \prec f(\underline{a},a) \wedge J(\underline{a},a'',a'')) \\
\qquad\qquad\qquad\qquad \vee\ (a'' \notin K \wedge J(\underline{a},a,a''))]])])
\end{array}
\qquad (\mathcal{F}_3')
$$

où $\underline{Cutset}\ \langle S, A, t, \varepsilon \rangle (K) = [(K \subseteq S) \wedge (\forall \underline{a} \in S.(\varepsilon(\underline{a}) \Rightarrow \underline{a} \in K)) \wedge \forall p \in \Sigma^\omega \langle S, A, t, tt \rangle.\ \exists i \in \omega.\ p_i \in K]$

Un ensemble de points de coupure ("cutset") est une classe d'états (qui pour simplifier, inclut les états d'entrée et) telle que s'il y avait une exécution infinie du programme, celle-ci passerait infiniment souvent par des états appartenant à l'ensemble de points de coupure.

Théorème 5.2.3 ∾ 2

$$(\mathcal{F}_2) \Rightarrow (\mathcal{F}_3)$$

Démonstration

Choisir $J_3(\underline{s}, s, s') = [J_2(\underline{s}, s') \wedge s = s']$, $\underset{\sim}{\mathrm{Rng}}(f_3) = \underset{\sim}{\mathrm{Rng}}(f_2)$, $f_3 = f_2$, $\prec_3 = \prec_2$ et $K = S$.

□

L'utilisation de bons ordres n'est pas obligatoire. Des relations bien-fondées sont suffisantes (et quelquefois plus commodes) :

$$\begin{aligned}
&(\exists J \in (S^2 \to \{tt, ff\}),\ f \in (S^2 \to \underset{\sim}{\mathrm{Rng}}(f)),\ \prec \in (\underset{\sim}{\mathrm{Rng}}(f) \times \underset{\sim}{\mathrm{Rng}}(f) \to \{tt, ff\}).\\
&\quad \underset{\sim}{\mathrm{wf}}(\underset{\sim}{\mathrm{Rng}}(f), \prec)\\
&\quad \wedge\ [\forall \underline{s}, s \in S.\\
&\qquad (\varepsilon(\underline{s}) \Rightarrow J(\underline{s}, \underline{s}))\\
&\qquad \wedge\ (J(\underline{s}, s) \to \Psi(\underline{s}, s)\\
&\qquad\qquad \vee\ [\Phi(\underline{s}, s) \wedge \exists s' \in S, a \in A.\ t_a(s, s') \wedge\\
&\qquad\qquad\quad \forall s' \in S, a \in A.\ (t_a(s, s') \Rightarrow [J(\underline{s}, s') \wedge f(\underline{s}, s') \prec f(\underline{s}, s)])])])
\end{aligned} \tag{$\mathcal{F}_4$}$$

où $\underset{\sim}{\mathrm{wf}}(W, \prec)$ caractérise les relations bien fondées sur W.

Théorème 5.2.3 ∾ 3

$$(\mathcal{F}_3) \Longrightarrow (\mathcal{F}_4)$$

Démonstration

Puisque $\underset{\sim}{wo}(\underset{\sim}{Rng}(f_3), \prec_3)$ implique $\underset{\sim}{wf}(\underset{\sim}{Rng}(f_3), \prec_3)$ nous pouvons définir :

. $\mu \in (S \times S \to \underset{\sim}{Ord})$

$$\mu(\underline{s}, \Delta') = \Big((\forall \Delta \in S. \neg J_3(\underline{s}, \Delta, \Delta') \vee \Psi(\underline{s}, \Delta')) \to 0 \mid \cap \{ \underset{\sim}{rk}(\underset{\sim}{Rng}(f_3), \prec_3)(f_3(\underline{s}, \Delta)) + 1 : \Delta \in S \wedge J_3(\underline{s}, \Delta, \Delta') \wedge \neg \Psi(\underline{s}, \Delta') \} \Big)$$

. $f_4 \in (S \times S \to \underset{\sim}{Ord} \times S)$

$f_4(\underline{s}, \Delta') = \langle \mu(\underline{s}, \Delta'), \Delta' \rangle$

. $\ll \in (S \times S \to \{tt, ff\})$ telle que $\Delta' \ll \Delta$ si et seulement si $[\exists a \in A. t_a(\Delta, \Delta') \wedge \Delta' \notin K]$

. $\prec_4 \in (\underset{\sim}{Rng}(f_4) \times \underset{\sim}{Rng}(f_4) \to \{tt, ff\})$ telle que $\langle w', \Delta' \rangle \prec_4 \langle w, \Delta \rangle$ si et seulement si $((w' < w) \vee (w' = w \wedge \Delta' \ll \Delta))$

Puisque $\underset{\sim}{wf}(\underset{\sim}{Ord}, <)$ et $\underset{\sim}{Cutset}\langle S, A, t, \varepsilon \rangle(K)$ impliquent $\underset{\sim}{wf}(S, \ll)$ nous avons $\underset{\sim}{wf}(\underset{\sim}{Rng}(f_4), \prec_4)$. Si nous choisissons $J_4(\underline{s}, \Delta') = [\exists \Delta \in S. J_3(\underline{s}, \Delta, \Delta')]$ la démonstration consiste essentiellement à montrer que nous avons :

$$[(\exists \Delta \in S. J_3(\underline{s}, \Delta, \Delta')) \wedge \neg \Psi(\underline{s}, \Delta') \wedge t_a(\Delta', \Delta'') \wedge \Delta'' \in K] \Rightarrow (\mu(\underline{s}, \Delta') > \mu(\underline{s}, \Delta''))$$

et

$$[(\exists \Delta \in S. J_3(\underline{s}, \Delta, \Delta')) \wedge \neg \Psi(\underline{s}, \Delta') \wedge t_a(\Delta', \Delta'') \wedge \Delta'' \notin K] \Rightarrow (\mu(\underline{s}, \Delta') \geq \mu(\underline{s}, \Delta''))$$

□

La fonction de terminaison peut être remplacée par une variable auxiliaire (w, n'apparaissant pas comme une variable du programme) à valeurs dans le domaine d'une relation bien-fondée et qui "décroit strictement" à chaque pas du programme.

$$(\exists W,\ \prec \in (W^2 \to \{tt, ff\}),\quad J \in (W \times S \times S \to \{tt, ff\}).$$

$$\begin{array}{l} \underset{\sim}{wf}(W, \prec) \\ \wedge\ [\forall \underline{a}, A \in S,\ w \in W. \\ \qquad (\varepsilon(\underline{a}) \Rightarrow [\exists \underline{w} \in W.\ J(\underline{w}, \underline{a}, \underline{a})]) \\ \qquad \wedge\ (J(w, \underline{a}, A) \Rightarrow \Psi(\underline{a}, A) \\ \qquad\qquad \vee\ [\Phi(\underline{a}, A) \wedge \exists A' \in S, a \in A.\ t_a(A, A') \wedge \\ \qquad\qquad\quad \forall A' \in S, a \in A.\ (t_a(A, A') \Rightarrow [\exists w' \prec w.\ J(w', \underline{a}, A')])])]) \end{array} \qquad (\mathcal{F}_5)$$

Théorème 5.2.3∨4

Démonstration

Choisir $W_5 = \underset{\sim}{Rng}(f_4)$, $\prec_5 = \prec_4$, $J_5(w, \underline{a}, A) = [w = f_4(\underline{a}, A) \wedge J_4(\underline{a}, A)]$

□

Puisque les relations bien-fondées peuvent être plongées dans des bons-ordres, l'isomorphisme de bons-ordres est une relation d'équivalence et les ordinaux sont des représentants de chaque classe d'équivalence, nous pouvons toujours utiliser le bon-ordre $<$ la classe $\underset{\sim}{Ord}$ des ordinaux pour les preuves de fatalité :

$$(\exists \Gamma \in \underset{\sim}{Ord},\ J \in (\Gamma \times S \times S \to \{tt, ff\}).$$

$$(\mathcal{F}_6.1) \qquad (\forall \Delta \in S.\ \exists \gamma \in \Gamma.\ J(\gamma, \Delta, \Delta))$$

$$(\mathcal{F}_6.2) \qquad (\forall \Delta, \Delta' \in S,\ \gamma' \in \Gamma. \qquad (\mathcal{F}_6')$$

$$J(\gamma', \Delta, \Delta') \Rightarrow$$

$$(\mathcal{F}_6.2.a) \qquad [\Phi(\Delta, \gamma) \wedge \exists \Delta'' \in S, a \in A.\ t_a(\Delta', \Delta'') \wedge \forall \Delta'' \in S, a \in A.\ [t_a(\Delta', \Delta'') \Rightarrow \exists \gamma'' < \gamma'.\ J(\gamma'', \Delta, \Delta'')]]$$

$$(\mathcal{F}_6.2.b) \qquad [\varepsilon(\Delta) \Rightarrow \Psi(\Delta, \Delta')]))$$

Théorème 5.2.3 v 5

Démonstration

Définir une fonction rang $\rho \in (W_5 \to \underset{\sim}{Ord})$ comme suit :

$$\rho(w) = \cap \{\alpha \in \underset{\sim}{Ord} : \forall w' \in W_5.\ [w' \prec_5 w \Rightarrow \rho(w') < \alpha]\}$$

(cette définition se justifie aisément par induction transfinie sur $\prec_5$, puisque $\prec_5$ est une relation bien-fondée sur W_5).

Observer que $\forall w', w \in W_5.\ [(w' \prec_5 w) \Rightarrow (\rho(w') < \rho(w))]$. Définir $\delta = \underset{\sim}{sup}^+ \{\rho(w)+1 : w \in W_5\}$.

Choisir :

$$J_6(\gamma, \underline{\Delta}, \Delta) = [\varepsilon(\underline{\Delta}) \Rightarrow ([\gamma = 0 \wedge \Psi(\underline{\Delta}, \Delta)] \vee [\exists w \in W_5.\ (J_5(w, \underline{\Delta}, \Delta) \wedge \gamma = \rho(w)+1)])]$$

□

La classe bien-fondée auxiliaire (W_1, $\underset{\sim}{Rng}(f_2)$, $\underset{\sim}{Rng}(f_3)$, $\underset{\sim}{Rng}(f_4)$ ou W_5) peut toujours être choisie comme $\langle \underset{\sim}{Ord}, < \rangle$ mais aussi comme $\langle S \times S, \prec \rangle$ où $\prec$ est une relation binaire bien-fondée sur les états convenablement choisie :

$$
\begin{array}{l}
(\exists J \in (S^2 \to \{tt, ff\}),\ \prec \in (S^2 \times S^2 \to \{tt, ff\}).\\
\quad \underset{\sim}{wf}(S^2, \prec)\\
\quad \wedge\ [\forall \underline{a}, a \in S.\\
\qquad (\varepsilon(\underline{a}) \Rightarrow J(\underline{a}, \underline{a}))\\
\qquad \wedge\ (J(\underline{a}, a) \Rightarrow \Psi(\underline{a}, a)\\
\qquad\qquad \vee\ [\Phi(\underline{a}, a) \wedge \exists a' \in S, a \in A.\, t_a(a, a') \wedge\\
\qquad\qquad\quad \forall a' \in S, a \in A.\, (t_a(a, a') \Rightarrow [J(\underline{a}, a') \wedge \langle \underline{a}, a' \rangle \prec \langle \underline{a}, a \rangle])])])
\end{array}
\qquad (\mathcal{F}_7')
$$

Théorème 5.2.3 ~ 6

$(\mathcal{F}_6') \Rightarrow (\mathcal{F}_7')$

Démonstration

Choisir $J_7(\underline{a}, a) = [\exists \gamma \in \Gamma.\, (\varepsilon(\underline{a}) \wedge J_6(\gamma, \underline{a}, a))]$, $\langle \underline{a}', a' \rangle \prec_7 \langle \underline{a}, a \rangle$ si et seulement si $[\varepsilon(\underline{a}) \wedge \underline{a}' = \underline{a} \wedge \exists \gamma \in \Gamma. J_6(\gamma, \underline{a}, a) \wedge \neg\Psi(\underline{a}, a) \wedge \forall \gamma \in \Gamma. ([J_6(\gamma, \underline{a}, a) \wedge \gamma > 0] \Rightarrow \exists \gamma' < \gamma. J_6(\gamma', \underline{a}, a'))]$.

□

Les principes d'induction $(\mathcal{F}_1')$ à $(\mathcal{F}_7')$ sont tous équivalents dans le sens que si une preuve a été faite au moyen d'un certain principe d'induction $(\mathcal{F}_i')$ comportant $J_i, \prec_i, \ldots$ la preuve peut être reéxprimée pour tout autre principe d'induction $(\mathcal{F}_j')$ puisque $J_j, \prec_j, \ldots$ peuvent être dérivés à partir de $J_i, \prec_i, \ldots$ en utilisant les règles de reécriture données dans les démonstrations $(\mathcal{F}_i') \Rightarrow (\mathcal{F}_{i \bmod 7 + 1}') \Rightarrow \cdots \Rightarrow (\mathcal{F}_j')$.
Un dernier résultat est necéssaire :

Théorème 5.2.3 ~ 7

$(\mathcal{F}_7') \Rightarrow (\mathcal{F}_1')$

Démonstration

Choisir $W_1 = Rng(f_1) = \underline{Ord}$, $\prec_1 = <$, $J_1 = J_7$ et $f_1(\underline{a}, a) = \cup\{ f_1(\underline{a}, a') + 1 : \langle \underline{a}, a' \rangle \prec_7 \langle \underline{a}, a \rangle \}$.

□

En utilisant les versions contrapositives de ces principes d'induction, nous pourrons démontrer les propriétés de fatalité des programmes par l'absurde. Par exemple $(\mathcal{F}_6)$ peut également s'écrire :

$$
\begin{array}{l}
(\exists \Gamma \in \underline{Ord},\ J \in (\Gamma \times S \times S \to \{tt, ff\}).\\
\quad [\forall \underline{a}, a, a', \bar{a} \in S,\ \alpha \in A,\ \gamma \in \Gamma.\\
\qquad (\varepsilon(\underline{a}) \Rightarrow [\exists \gamma \in \Gamma.\ J(\gamma, \underline{a}, \underline{a})])\\
\qquad \wedge ([J(\gamma, \underline{a}, a) \wedge \gamma > 0] \Rightarrow [\Phi(\underline{a}, a) \wedge \exists a' \in S, \alpha \in A.\ t_\alpha(a, a')])\\
\qquad \wedge ([J(\gamma, \underline{a}, a) \wedge \gamma > 0 \wedge t_\alpha(a, a')] \Rightarrow [\exists \gamma' < \gamma.\ J(\gamma', \underline{a}, a')])\\
\qquad \wedge (J(0, \underline{a}, \bar{a}) \Rightarrow \Psi(\underline{a}, \bar{a}))])
\end{array}
\qquad (\mathcal{F}_6')
$$

dont la version contrapositive est :

$$
\begin{array}{l}
(\exists \Gamma \in \underline{Ord},\ J \in (\Gamma \times S \times S \to \{tt, ff\}).\\
\quad [\forall \underline{a}, a, a', \bar{a} \in S,\ \alpha \in A,\ \gamma \in \Gamma.\\
\qquad (\neg\Psi(\underline{a}, \bar{a}) \Rightarrow J(0, \underline{a}, \bar{a}))\\
\qquad \wedge ([\gamma > 0 \wedge (\neg\Phi(\underline{a}, a) \vee \forall a' \in S, \alpha \in A.\ \neg t_\alpha(a, a'))] \Rightarrow J(\gamma, \underline{a}, a))\\
\qquad \wedge ([\gamma > 0 \wedge t_\alpha(a, a') \wedge \forall \gamma' < \gamma.\ J(\gamma', \underline{a}, a')] \Rightarrow J(\gamma, \underline{a}, a))\\
\qquad \wedge (\varepsilon(\underline{a}) \Rightarrow [\exists \underline{\gamma} \in \Gamma.\ \neg J(\underline{\gamma}, \underline{a}, \underline{a})])])
\end{array}
\qquad (\overline{\mathcal{F}_6'})
$$

Les versions positives et contrapositives des principes d'induction sont évidemment équivalentes :

Théorème 5.2.3∿8

$$i = 1, \ldots, 7$$

Démonstration

(⇒) Choisir $\bar{J}_i = \neg J_i$ (⇐) Choisir $J_i = \neg \bar{J}_i$

□

Exemple 5.2.3-1 (Preuve d'un programme de parcours d'arbre utilisant le principe d'induction $(\mathcal{F}_8')$)

Dans la suite (pour illustrer la méthode de Burstall [74]) nous utiliserons le programme séquentiel suivant (tiré littéralement de Burstall [74]) qui parcourt un arbre binaire à l'aide d'une pile en comptant ses feuilles externes :

La valeur de la variable Tr de type "arbre" est soit "nil" soit (lf(Tr).rg(Tr)) où lf(Tr) et rg(Tr) sont des "arbres", la valeur de Co de type "nat" est un nombre naturel et la valeur de la variable St de type "pile" est soit () soit (hd(St).tl(St)) où hd(St) est un "arbre" et tl(St) est une "pile".

```
Start:   St := ( ) ;  Co := 0;
Loop:    if Tr ≠ nil
             then begin  Push Tr onto St;
                         Tr := ff(Tr);  goto Loop
                  end
             else begin  Co := Co+1;
                         if St=( ) then goto Finish;
                         Pop Tr from St;
                         Tr := rg(Tr);  goto Loop
                  end;
Finish:
```

Dans la suite nous utiliserons cet exemple pour illustrer notre formalisation de la méthode de Burstall. Pour permettre sa comparaison avec la méthode de Floyd, nous allons montrer à l'aide de cet exemple qu'une preuve de correction totale par la méthode de Floyd consiste exactement à appliquer le principe d'induction $(\mathcal{F}_c)$:

Le système de transition $\langle S, A, t, \varepsilon \rangle$ correspondant à ce programme est défini par :

- $S = \{Start, Loop, Finish\} \times arbre \times nat \times pile$
- $A = \{\underline{a}\}$
- $t_{\underline{a}}(\langle \ell, tr, co, st\rangle, \langle \ell', tr', co', st'\rangle) =$

$$\begin{aligned}[& (\ell = Start \wedge \ell' = Loop \wedge tr' = tr \wedge co' = 0 \wedge st' = ( )) \\ \vee\ & (\ell = Loop \wedge tr \neq nil \wedge \ell' = Loop \wedge tr' = ff(Tr) \wedge co' = co \wedge st' = (tr.st)) \\ \vee\ & (\ell = Loop \wedge tr = nil \wedge st \neq ( ) \wedge \ell' = Loop \wedge tr' = rg(hd(st)) \wedge co' = co+1 \wedge st' = tl(st)) \\ \vee\ & (\ell = Loop \wedge tr = nil \wedge st = ( ) \wedge \ell' = Finish \wedge tr' = tr \wedge co' = co+1 \wedge st' = st)]\end{aligned}$$

- $\varepsilon(\langle \ell, tr, co, st\rangle) = [\ell = Start]$

La correction totale de ce programme peut être spécifiée par la fatalité de ψ pour $\langle S, A, \Sigma\langle S, A, t, \varepsilon\rangle\rangle$, telle que :

$$\psi(\langle l, tr, co, st\rangle, \langle l', tr', co', st'\rangle) = [l' = Finish \wedge co' = tips(tr)]$$

où

$$tips(tr) = (tr = mil \rightarrow 1 \mid (tips(lf(tr)) + tips(rg(tr))))$$

- La correction partielle de ce programme consiste d'abord à découvrir des assertions intermédiaires associées aux points de contrôle du programme. Ces assertions expriment les relations qu'on s'attend à trouver entre les valeurs initiales $\underline{tr}$, $\underline{co}$, $\underline{st}$ des variables Tr, Co, St et leurs valeurs tr, co, st quand le contrôle est en ces points :

$$I_{Start}(\underline{tr}, \underline{co}, \underline{st}, tr, co, st) = [tr = \underline{tr} \wedge co = \underline{co} \wedge st = \underline{st}]$$

$$I_{Loop}(\underline{tr}, \underline{co}, \underline{st}, tr, co, st) = [(tips(tr) + co + \underline{Sum}(tips \circ rg, st)) = tips(\underline{tr})]$$

$$I_{Finish}(\underline{tr}, \underline{co}, \underline{st}, tr, co, st) = [co = tips(\underline{tr})]$$

où

$f \circ g(x) = f(g(x))$ et si $f \in (arbre \rightarrow \omega)$ et st est une pile alors

$$\underline{Sum}(f, st) = (st = () \rightarrow 0 \mid (f(hd(st)) + \underline{Sum}(f, tl(st))))$$

Puis on montre que ces assertions sont invariantes, c'est-à-dire :

. I_{Start} est initialement vraie avec $tr = \underline{tr}$, $co = \underline{co}$ et $st = \underline{st}$

. Si l'assertion intermédiaire associée au point de contrôle l du programme est vraie et que le contrôle passe du point l au point l', alors l'assertion associée à l' doit être vraie :

$$I_{Start}(\underline{tr}, \underline{co}, \underline{st}, tr, co, st) \Rightarrow I_{Loop}(\underline{tr}, \underline{co}, \underline{st}, tr, 0, ())$$

$$[I_{Loop}(\underline{tr}, \underline{co}, \underline{st}, tr, co, st) \wedge tr \neq mil] \Rightarrow I_{Loop}(\underline{tr}, \underline{co}, \underline{st}, lf(tr), co, (tr.st))$$

$$[I_{Loop}(\underline{tr}, \underline{co}, \underline{st}, tr, co, st) \wedge tr = mil \wedge st \neq ()] \Rightarrow I_{Loop}(\underline{tr}, \underline{co}, \underline{st}, rg(hd(st)), co+1, tl(st))$$

$$[I_{Loop}(\underline{tr}, \underline{co}, \underline{st}, tr, co, st) \wedge tr = mil \wedge st = ()] \Rightarrow I_{Finish}(\underline{tr}, \underline{co}, \underline{st}, tr, co+1, st)$$

Finalement, l'assertion associée au point de sortie doit impliquer la spécification :

$$I_{Finish}(\underline{tr}, \underline{co}, \underline{st}, tr, co, st) \Rightarrow [co = tips(\underline{tr})]$$

- D'après Floyd [67], "proofs of termination are dealt with by showing that each step of the program decreases some entity which cannot decrease indefinitely". Nous associons donc à tout point de contrôle ℓ du programme, une fonction W_ℓ des valeurs tr, co, st des variables Tr, Co, St du programme à résultat dans le bon-ordre $\langle \omega, < \rangle$:

$$W_{Start}(tr, co, st) = size(tr)+1$$

$$W_{Loop}(tr, co, st) = size(tr) + \underline{Sum}(size \circ rg, st)$$

$$W_{Finish}(tr, co, st) = 0$$

où

$$size(tr) = (tr = nil \to 1 \mid 1 + size(lf(tr)) + size(rg(tr)))$$

et montrons qu'après chaque exécution d'une commande, la valeur courante de la fonction $W_{\ell'}$ associée à son point de "sortie" ℓ' est inférieure à la valeur antérieure de la fonction W_ℓ associée à son point d'"entrée" ℓ :

$$[tr'=tr \wedge co'=0 \wedge st'=()] \Rightarrow (W_{Start}(tr,co,st) > W_{Loop}(tr', co', st'))$$

$$[tr \neq nil \wedge tr'=lf(tr) \wedge co'=co \wedge st'=(tr.st)] \Rightarrow (W_{Loop}(tr,co,st) > W_{Loop}(tr',co',st'))$$

$$[tr=nil \wedge st\neq() \wedge tr'=rg(hd(st)) \wedge co'=co+1 \wedge st'=tl(st)] \Rightarrow$$
$$(W_{Loop}(tr,co,st) > W_{Loop}(tr',co',st'))$$

$$[tr=nil \wedge st=() \wedge tr'=tr \wedge co'=co+1 \wedge st'=st] \Rightarrow (W_{Loop}(tr,co,st) > W_{Finish}(tr',co',st'))$$

- En choisissant :

$$I=\omega$$

$$J_\ell(\gamma, \langle \underline{\ell}, \underline{tr}, \underline{co}, \underline{st}\rangle, \langle \ell, tr, co, st\rangle) =$$
$$[(\underline{\ell} = Start) \ (\gamma = W_\ell(tr,co,st) \wedge I_\ell(\underline{tr}, \underline{co}, \underline{st}, tr, co, st))]$$

les conditions de vérification de Floyd sont équivalentes à $(\mathcal{F}_\ell)$ (puisque ce programme est total, les vérifications d'absence d'erreurs à l'exécution :

$$\forall \ell \in \{Start, Loop\} . [I_\ell(\underline{tr}, \underline{co}, \underline{st}, tr, co, st) \Rightarrow$$
$$\exists \ell', tr', co', st' . t(\langle \ell, tr, co, st\rangle, \langle \ell', tr', co', st'\rangle)]$$

n'ont pas lieu d'être).

□

5.2.4 CORRECTION ET COMPLETUDE SEMANTIQUE DES PRINCIPES D'INDUCTION "A LA FLOYD" POUR LES SEMANTIQUES CLOSES

Ψ est fatale sous invariance de Φ pour $\langle S, A, \Sigma\langle S,A,t,\varepsilon\rangle\rangle$ si et seulement si un des principes d'induction est applicable.

Théorème 5.2.4~1 (Correction)

$(\mathcal{F}_1') \Rightarrow (\Psi$ est fatale sous invariance de Φ pour $\langle S, A, \Sigma\langle S,A,t,\varepsilon\rangle\rangle)$

Démonstration

Supposons par l'absurde qu'il existe $p \in \Sigma\langle S,A,t,\varepsilon\rangle$ tel que $\forall i \in |p|.\ [(\forall j \in i.\ \Phi(p_0, p_j)) \Rightarrow \neg\Psi(p_0, p_i)]$. Alors par induction sur i, $(\mathcal{F}_1')$ implique $\forall i \in |p|.\ [(\forall j \in (i+1).\ \Phi(p_0,p_j)) \wedge J_1(p_0,p_i)]$. Si $\exists m \in (\omega \vee 0).\ p \in \Sigma^m\langle S,A,t,\varepsilon\rangle$ alors $J_1(p_0,p_{m-1})$, $\neg\Psi(p_0,p_{m-1})$ et $(\mathcal{F}_1')$ impliquent $\exists s' \in S, a \in A.\ t_a(p_{m-1}, s')$, une contradiction. Sinon $p \in \Sigma^\omega\langle S,A,t,\varepsilon\rangle$ de sorte que pour tout $i \in \omega$ nous avons $J_1(p_0,p_i)$ et $t_{\#i}(p_i, p_{i+1})$ et donc d'après $(\mathcal{F}_1')$ que $f_1(p_0,p_i) \in W_1$, $f_1(p_0,p_{i+1}) \in W_1$ et $f_1(p_0,p_{i+1}) \prec_1 f_1(p_0,p_i)$ en contradiction avec ω well-founded $(W_1, \prec_1)$.

□

Théorème 5.2.4~2 (Complétude sémantique)

$(\Psi$ est fatale sous invariance Φ pour $\langle S, A, \Sigma\langle S,A,t,\varepsilon\rangle\rangle) \Rightarrow (\mathcal{F}_4')$

Démonstration

Définissons W et $\prec \in (W^2 \to \{tt, ff\})$ tels que :

- $W = \{\langle \underline{s}, s\rangle \in S^2 :\ \exists p \in \Sigma\langle S,A,t,\varepsilon\rangle, i \in |p|, k \in i.$
$[\forall j \in i.\ (\Phi(p_0,p_j) \wedge \neg\Psi(p_0,p_j)) \wedge \Psi(p_0,p_i) \wedge \underline{s} = p_0 \wedge s = p_k]\}$
- $\langle \underline{s}', s'\rangle \prec \langle \underline{s}, s\rangle \Leftrightarrow (\exists p \in \Sigma\langle S,A,t,\varepsilon\rangle, i \in |p|, k \in \omega.\ [\forall j \in i.\ (\Phi(p_0,p_j) \wedge \neg\Psi(p_0,p_j)) \wedge$
$\Psi(p_0,p_i) \wedge \underline{s}' = \underline{s} = p_0 \wedge k+1 < i \wedge s = p_k \wedge s' = p_{k+1}])$

Puisque $\langle \underline{s}', s' \rangle \prec \langle \underline{s}, s \rangle$ implique $(\underline{s}' = s \wedge \Phi(\underline{s}, s) \wedge \neg\Psi(\underline{s}, s) \wedge \exists a \in A. t_a(s, s')$ $\wedge \Phi(\underline{s}, s') \wedge \neg\Psi(\underline{s}, s'))$ nous avons $\underset{\sim}{wf}_0(W, \prec)$. (sinon il y aurait eu une chaîne $\langle \underline{s}, s_0 \rangle \prec \langle \underline{s}, s_1 \rangle \prec \cdots$ et donc une trace infinie $\underline{s}, p_1, \cdots, p_{k+1}, s_0, s_1, \cdots$ dont tous les états s satisfont $\Phi(\underline{s}, s) \wedge \neg\Psi(\underline{s}, s)$ en contradiction avec le fait que Ψ est fatale sous invariance de Φ pour $\langle S, A, \Sigma \langle S, A, t, \varepsilon \rangle \rangle$).
Définissons maintenant $\underset{\sim}{Rng}(f_4) = (W \cup \{\langle \underline{s}, s \rangle : \Psi(\underline{s}, s)\})$, $f_4(\underline{s}, s) = \langle \underline{s}, s \rangle$ et $\langle \underline{s}, s' \rangle \prec_4 \langle \underline{s}, s \rangle$ $\Leftrightarrow [(\Psi(\underline{s}', s') \wedge \underline{s}' = \underline{s} \wedge \langle \underline{s}, s \rangle \in W) \vee (\langle \underline{s}', s' \rangle \in W \wedge \langle \underline{s}, s \rangle \in W \wedge \langle \underline{s}', s' \rangle \prec \langle \underline{s}, s \rangle)]$. Nous avons $\underset{\sim}{wf}(\underset{\sim}{Rng}(f_4), \prec_4)$ de sorte qu'en choisissant $I_4(\underline{s}, s) = [\langle \underline{s}, s \rangle \in \underset{\sim}{Rng}(f_4)]$, les conditions de vérification de $(\mathcal{F}_4)$ sont vérifiées.

□

5.2.5 SUR L'UTILISATION D'HYPOTHESES d'INDUCTION ASSERTIONNELLES OU RELATIONNELLES

Si nous voulons démontrer que l'assertion $\Psi(\underline{s},s) = \psi(s)$ est fatale sous invariance de $\Phi(\underline{s},s) = \phi(s)$ pour $\langle S, A, \Sigma \langle S,A,t,\varepsilon\rangle\rangle$, les hypothèses d'induction J_i dans les principes d'induction $(\mathcal{F}_i)$, $i=1,\dots,7$, peuvent ne pas dépendre des états initiaux. Dans ce cas, le choix des f_i dans $(\mathcal{F}_i)$, $i=1,\dots,4$ comme une fonction ne dépendant pas des états initiaux est également sémantiquement complet. (Si en plus $\phi(s)=tt$, $\psi(s)=[\forall s' \in S, a \in A . \neg t_a(s,s')]$, $\langle \underset{\sim}{Rng}(f_2), \prec_2 \rangle = \langle \underset{\sim}{Ord}, < \rangle$ alors le principe d'induction $(\mathcal{F}_2)$ donne la méthode de preuve de terminaison de Lehmann-Pnueli-Stavi [81].

Théorème 5.2.5 v1

> En général, Ψ est une relation entre les états initiaux et finaux, et dans ce cas l'hypothèse d'induction doit être une relation (reliant les états initiaux et courants) pour garantir la complétude sémantique.

Démonstration

Considérons $S=\{0,1,2\}$, $A=\{a\}$, $\varepsilon(\underline{s})=[0 \le \underline{s} \le 1]$, $t_{\underline{a}}(s,s')=[(s+1 \in S) \wedge (s'=s+1)]$, $\Phi(\underline{s},s)=tt$. Alors $\Psi = t_{\underline{a}}$ est fatale de manière évidente. Supposons qu'on puisse trouver J_1 de la forme $J_1(\underline{s},s) = I(s)$ dans $(\mathcal{F}_1)$. Alors $\varepsilon(1) \Rightarrow J_1(1,1)$. Puisque $\neg\Psi(1,1)$ et $t_{\underline{a}}(1,2)$ alors nous avons $J_1(1,2) = I(2) = J_1(0,2)$. Mais $\neg\Psi(0,2)$ et $\forall s' \in S, a \in A . \neg t_{\underline{a}}(2,s')$, d'où contradiction.

□

Théorème 5.2.5 v2

> De la même manière, dans le principe d'induction $(\mathcal{F}_i)$, $i=1,\dots,4$, si f_i, $i=1,\dots,4$ est choisie comme une fonction unaire ne dépendant pas des états initiaux, $(\mathcal{F}_i)$ est (sémantiquement) incomplet.

Démonstration

Considérons $S=\omega$, $A=\{a\}$, $\varepsilon(\underline{s})=tt$, $t_a(s,s')=[s'=s+1]$ et $\Phi(\underline{s},s)=tt$. Alors de manière évidente, $\Psi(\underline{s},\bar{s})=[\bar{s}=\underline{s}+2]$ est fatale. Mais $(\mathcal{F}_1')$ ne peut être appliqué avec f_1 de la forme $f_1(\underline{s},s)=f(s)$. Par l'absurde, nous aurions $\forall s\in\omega.\ (\varepsilon(s)\Rightarrow J_1(s,s))$ de sorte que $J_1(s,s)\wedge\neg\Psi(s,s)\wedge t_{\underline{a}}(s,s+1)$ implique $f_1(s,s)\in W_1$ et $f_1(s,s+1)\prec_1 f_1(s,s)$ c'est-à-dire $f(s)\in W_1$ et $f(s+1)\prec_1 f(s)$ en contradiction avec $\underline{\underline{wo}}(W_1,\prec_1)$.

□

Par conséquent, la généralisation de la méthode de preuve de la correction partielle de Hoare[69] à la correction totale proposée par Manna-Pnueli[74] (pour laquelle la fonction de terminaison dans les boucles ne porte que sur l'état courant des variables dans la boucle sans liaison possible avec l'état initial des variables à l'entrée de la boucle) n'est pas (sémantiquement) complète. Ceci peut se corriger de la manière suivante :

Si on tient à utiliser des principes d'induction assertionnels et non relationnels, on pourra utiliser l'artifice bien connu qui consiste à utiliser des variables auxiliaires dans le programme.

Les propriétés fatales pour $\langle S,A,\Sigma\langle S,A,t,\varepsilon\rangle\rangle$ peuvent toujours aisément être démontrées en raisonnant sur $\langle S',A',\Sigma\langle S',A',t',\varepsilon'\rangle\rangle$ tel que :

$S'=S\times S$

$A'=A$

$t'_a(\langle\underline{s},s\rangle,\langle\underline{s}',s'\rangle)=[\underline{s}=\underline{s}'\wedge t_a(s,s')]$

$\varepsilon'(\langle\underline{s},s\rangle)=[\varepsilon(\underline{s})\wedge\underline{s}=s]$

car

$$\forall p\in\Sigma\langle S,A,t,\varepsilon\rangle.\ \exists i\in|p|.\ [(\forall j\in i.\ \Phi(p_0,p_j))\wedge\Psi(p_0,p_i)]$$

$\Longleftrightarrow$

$$\forall p'\in\Sigma\langle S',A',t',\varepsilon'\rangle.\ \exists i\in|p'|.\ [(\forall j\in i.\ \Phi(p'_j))\wedge\Psi(p'_i)]$$

Dans ce cas, il nous semble que l'emploi de variables auxiliaires dans les preuves ne doit pas être présenté comme une transformation de programme mais plutôt comme l'utilisation d'un principe d'induction différent. Ceci parce que les variables auxiliaires sont plus simples à introduire dans les preuves que dans les programmes (qui ont une syntaxe rigide). En outre, ceci permet de raisonner sur les méthodes de preuve de programmes qui sont indépendantes des langages de programmation.

5.2.6 SUR LE NON-DETERMINISME BORNE

Floyd [67] a noté qu'il peut être nécessaire d'utiliser d'autres bons-ordres que l'ensemble $\langle \omega, < \rangle$ des nombres naturels pour les preuves de terminaison. Dijkstra [76, p.77] a donné le contre-exemple $S = \mathbb{Z}$, $A = \{a\}$, $t_a(x,x') = [(x<0 \wedge x'>0) \vee (x>0 \wedge x' = x-1)]$, $\varepsilon(x) = [x<0]$, $\Phi(x,x) = tt$ et $\Psi(x,\bar{x}) = [\bar{x} = 0]$ pour lequel le nombre de transitions requises pour la terminaison n'a pas de borne supérieure finie. Dijkstra a montré que quand le non-déterminisme est fini, on peut faire les preuves de terminaison en utilisant $\langle \omega, < \rangle$.

5.2.6.1 Non-déterminisme $\mathfrak{m}$-borné

Un système de transition $\langle S, A, t, \varepsilon \rangle$ est dit déterministe si $\forall s \in S. [\,\mathrm{card}(\{s' \in S : \exists a \in A. t_a(s,s')\}) \leq 1\,]$ et non déterministe sinon.

Le non déterminisme est dit fini (Dijkstra dit borné) si $\forall s \in S. \exists m \in \omega. [\,\mathrm{card}(\{s' \in S : \exists a \in A. t_a(s,s')\}) \leq m\,]$ ou, et ceci est équivalent, $\forall s \in S. [\,\mathrm{card}(\{s' \in S : \exists a \in A. t_a(s,s')\}) < \omega\,]$ et infini sinon.

Le non déterminisme est dit dénombrable si $\forall s \in S. [\,\mathrm{card}(\{s' \in S : \exists a \in A. t_a(s,s')\}) \leq \omega\,]$ et non dénombrable sinon.

Plus généralement, si $\mathfrak{m} \in \mathrm{ord}$ est un cardinal, alors nous dirons que le non déterminisme d'un système de transition $\langle S, A, t, \varepsilon \rangle$ est $\mathfrak{m}$-borné si $\forall s \in S. [\,\mathrm{card}(\{s' \in S : \exists a \in A. t_a(s,s')\}) < \mathfrak{m}\,]$.

(En particulier, le non déterminisme de $\langle S, A, t, \varepsilon \rangle$ est toujours $\mathrm{card}(S)^+$-borné. En outre, le non déterminisme est dénombrable si et seulement s'il est ω_1-borné où ω_1 est le plus petit cardinal strictement plus grand que ω).

5.2.6.2 La fatalité peut être démontrée à l'aide du bon-ordre $<m^+,<>$ quand le non-déterminisme est m-borné

Le principe d'induction suivant, pour démontrer les propriétés de fatalité de sémantiques $<S,A,\Sigma<S,A,t,\varepsilon>>$ avec non déterminisme m-borné, est correct (puisque $(\mathcal{F}_8) \Rightarrow (\mathcal{F}_6')$) et sémantiquement complet :

$$
\begin{array}{l}
(\exists J \in (m^+ \times S \times S \to \{tt, ff\}). \\
\quad [\forall \underline{s}, s, s', \bar{s} \in S,\ \gamma < m^+. \\
\text{(a)} \quad\quad (\varepsilon(\underline{s}) \Rightarrow [\exists \underline{\gamma} < m^+.\ J(\underline{\gamma}, \underline{s}, \underline{s})]) \\
\quad\quad \wedge \\
\text{(b)} \quad\quad ([J(\gamma, \underline{s}, s) \wedge \gamma > 0] \Rightarrow [\Phi(\underline{s}, s) \wedge \exists s' \in S, a \in A.\ t_a(s, s')]) \\
\quad\quad \wedge \\
\text{(c)} \quad\quad ([J(\gamma, \underline{s}, s) \wedge \gamma > 0 \wedge t_a(s, s')] \Rightarrow [\exists \gamma' < \gamma.\ J(\gamma', \underline{s}, s')]) \\
\quad\quad \wedge \\
\text{(d)} \quad\quad (J(0, \underline{s}, \bar{s}) \Rightarrow \Psi(\underline{s}, \bar{s}))\ ])
\end{array}
\qquad (\mathcal{F}_8)
$$

où $m^+ = \omega$ si $m < \omega$, $m^+ = m$ si m est un cardinal infini régulier et $m^+ = m^+$ si m est un cardinal infini singulier.

Pour démontrer la complétude sémantique, nous introduisons quelques définitions que nous utiliserons par la suite.

Nous dirons que :

- un état s est <u>intermédiaire</u> pour $<S,A,t,\varepsilon>$ lorsqu'il existe une trace d'exécution passant par s pour laquelle Ψ n'est pas vraie jusqu'à s compris.
- un état s est un <u>but</u> ("goal") pour $<S,A,t,\varepsilon>$ lorsqu'il existe une trace d'exécution passant par s pour laquelle Ψ est vraie pour la première fois en s.
- un état s est <u>accessible</u> pour $<S,A,t,\varepsilon>$ lorsque s est un état intermédiaire ou un but.

Définition 5.2.6.2 : 1 (Etats intermédiaires, buts et accessibles)

- $\underline{Inter}\langle S,A,t,\varepsilon,\Phi,\Psi\rangle(\underline{s}) = \{s\in S : \exists p\in\Sigma\langle S,A,t,\varepsilon\rangle, i\in|p|.$
$(p_0=\underline{s} \wedge \forall j<i.(\Phi(p_0,p_j)\wedge\neg\Psi(p_0,p_j)) \wedge p_i=s)\}$

  $\underline{Inter}\langle S,A,t,\varepsilon,\Phi,\Psi\rangle \cup \{\underline{Inter}\langle S,A,t,\varepsilon,\Phi,\Psi\rangle(\underline{s}) : \underline{s}\in S\}$

- $\underline{Goal}\langle S,A,t,\varepsilon,\Phi,\Psi\rangle(\underline{s})$ $\{s\in S : \exists p\in\Sigma\langle S,A,t,\varepsilon\rangle, i\in|p|.$
$(p_0=\underline{s} \wedge \forall j<i.(\Phi(p_0,p_j)\wedge\neg\Psi(p_0,p_j)) \wedge p_i=s\wedge\Psi(p_0,p_i))\}$

  $\underline{Goal}\langle S,A,t,\varepsilon,\Phi,\Psi\rangle$ $\cup\{\underline{Goal}\langle S,A,t,\varepsilon,\Phi,\Psi\rangle(\underline{s}) : \underline{s}\in S\}$

- $\underline{Acc}\langle S,A,t,\varepsilon,\Phi,\Psi\rangle(\underline{s})$ - $\underline{Inter}\langle S,A,t,\varepsilon,\Phi,\Psi\rangle(\underline{s}) \cup \underline{Goal}\langle S,A,t,\varepsilon,\Phi,\Psi\rangle(\underline{s})$

  $\underline{Acc}\langle S,A,t,\varepsilon,\Phi,\Psi\rangle$ = $\underline{Inter}\langle S,A,t,\varepsilon,\Phi,\Psi\rangle \cup \underline{Goal}\langle S,A,t,\varepsilon,\Phi,\Psi\rangle$

Théorème 5.2.6.2 v 1 (Complétude sémantique)

ou invariance de Φ pour $\langle S,A,\Sigma\langle S,A,t,\varepsilon\rangle\rangle \Longrightarrow (\mathcal{F}_3)$

Démonstration

Pour tous $s\in S$ nous définissons :

- $I(\underline{s}) = \{s\in S : \underline{Inter}\langle S,A,t,\varepsilon,\Phi,\Psi\rangle(\underline{s})\}$
- $G(\underline{s}) = \{s\in S : \underline{Goal}\langle S,A,t,\varepsilon,\Phi,\Psi\rangle(\underline{s})\}$
- $A(\underline{s}) = I(\underline{s}) \cup G(\underline{s})$
- $\prec_{\underline{s}} \in (S\times S \to \{tt,ff\})$, telle que $s'\prec_{\underline{s}} s$ si et seulement si $[s\in I(\underline{s}) \wedge \exists a\in A.\, t_a(s,s')]$

Nous démontrons d'abord que $\forall \underline{s}\in S.\ \underline{wf}(A(\underline{s}),\prec_{\underline{s}})$.

Si $\neg\varepsilon(\underline{s})$ alors $A(\underline{s})=0$ et toute relation est bien fondée sur l'ensemble vide 0. Sinon, par l'absurde, supposons qu'il existe une séquence infinie q d'états dans $A(\underline{s})$ tel que $\forall i\in\omega.(q_{i+1}\prec_{\underline{s}} q_i)$. Si $s\in G(\underline{s})$ alors $s\notin I(\underline{s})$ de sorte que $\forall s'\in S.\neg(s'\prec_{\underline{s}} s)$. D'où, aucun q_i ne peut appartenir à $G(\underline{s})$. En particulier, puisque $q_0\in I(\underline{s})$ nous pouvons supposer que $q_0=\underline{s}$ (sinon il existe un préfixe $\underline{s}=p_0,\dots,p_k=q_0$ d'une certaine trace $p\in\Sigma\langle S,A,t,\varepsilon\rangle$ avec $p_{j+1}\prec_{\underline{s}} p_j$ pour $j<k$, qui peut être adjoint à la gauche de q). Nous avons $\varepsilon(q_0)$. En outre $\forall i\in\omega.(q_{i+1}\prec_{\underline{s}} q_i)$, d'où $t_{q_i}(q_i,q_{i+1})$. Il s'ensuit que $q\in\Sigma\langle S,A,t,\varepsilon\rangle$.

Mais $\forall i \in \omega$ nous avons $q_i \in I(q_0)$ et donc $\Phi(q_0,q_i) \wedge \neg\Psi(q_0,q_i)$ en contradiction avec l'hypothèse de fatalité.

Nous démontrons maintenant que $\forall \underline{a}, a \in S.(\underline{\mathrm{card}}(\underline{\mathrm{rk}}(A(\underline{a}),\prec_{\underline{a}})(a)) < m^+)$

La preuve est par induction transfinie sur la relation $\prec_{\underline{a}}$ bien fondée sur $A(\underline{a})$. Si $\{a' \in S : a' \prec_{\underline{a}} a\}$ est vide alors $\underline{\mathrm{rk}}(A(\underline{a}),\prec_{\underline{a}})(a) = 0$ donc $\underline{\mathrm{card}}(\underline{\mathrm{rk}}(A(\underline{a}),\prec_{\underline{a}})(a)) = 0 < \omega \leq m^+$. Autrement $a' \prec_{\underline{a}} a$ implique $\exists \alpha \in A.t_\alpha(a,a')$ donc $\underline{\mathrm{card}}(\{a' \in S : a' \prec_{\underline{a}} a\}) \leq \underline{\mathrm{card}}(\{a' \in S : \exists \alpha \in A.t_\alpha(a,a')\}) < m \leq m^+$ et $\underline{\mathrm{card}}(\underline{\mathrm{rk}}(A(\underline{a}),\prec_{\underline{a}})(a')) < m^+$ par hypothèse d'induction. Donc ou bien $\underline{\mathrm{card}}(\underline{\mathrm{rk}}(A(\underline{a}),\prec_{\underline{a}})(a')) < \omega$ et $\underline{\mathrm{card}}(\mathcal{P}(\underline{\mathrm{rk}}(A(\underline{a}),\prec_{\underline{a}})(a')) = \mathcal{P}(\underline{\mathrm{card}}(\underline{\mathrm{rk}}(A(\underline{a}),\prec_{\underline{a}})(a'))) < \omega \leq m^+$ ou $\underline{\mathrm{card}}(\underline{\mathrm{rk}}(A(\underline{a}),\prec_{\underline{a}})(a')) > \omega$ auquel cas $\underline{\mathrm{card}}(\mathcal{P}(\underline{\mathrm{rk}}(A(\underline{a}),\prec_{\underline{a}})(a')) = \underline{\mathrm{card}}(\underline{\mathrm{rk}}(A(\underline{a}),\prec_{\underline{a}})(a')) < m^+$. Si n est un cardinal infini régulier, alors pour tout système $\langle \mu_i : i \in I\rangle$ de cardinaux avec $\mu_i < n$ pour tout $i \in I$ et $\underline{\mathrm{card}}(I) < m$, nous avons $\bigvee_{i\in I} \mu_i < n$. D'où nous concluons que $\underline{\mathrm{card}}(\underline{\mathrm{rk}}(A(\underline{a}),\prec_{\underline{a}})(a)) = \underline{\mathrm{card}}(\bigcup\{\underline{\mathrm{rk}}(A(\underline{a}),\prec_{\underline{a}})(a')+1 : a' \prec_{\underline{a}} a\}) < m^+$.
$\underline{\mathrm{card}}(\underline{\mathrm{rk}}(A(\underline{a}),\prec_{\underline{a}})(a)) = \underline{\mathrm{card}}(\sup^+\{\underline{\mathrm{rk}}(A(\underline{a}),\prec_{\underline{a}})(a') : a' \prec_{\underline{a}} a\}) = \underline{\mathrm{card}}(\sup\{\mathcal{P}(\underline{\mathrm{rk}}(A(\underline{a}),\prec_{\underline{a}})(a')) : a' \prec_{\underline{a}} a\}) < m^+$.

Finalement nous définissons J_β tel que $J_\beta(\gamma,\underline{a},a) = [a \in A(\underline{a}) \wedge \gamma = \underline{\mathrm{rk}}(A(\underline{a}),\prec_{\underline{a}})(a)]$. Nous avons $J_\beta \in (m^+ \times S \times S \to \{tt,ff\})$ et les conditions de vérification de $(\mathcal{J}_\beta)$ sont satisfaites :

(a) $E(\underline{a}) \Rightarrow [\underline{a} \in A(\underline{a})] \Rightarrow [\exists \underline{\gamma} < m^+ . J_\beta(\underline{\gamma},\underline{a},\underline{a})]$.

(b) Si $[J_\beta(\gamma,\underline{a},a) \wedge \gamma > 0]$ alors $\underline{\mathrm{rk}}(A(\underline{a}),\prec_{\underline{a}})(a) > 0$ de sorte qu'il existe a' tel que $a' \prec_{\underline{a}} a$. Ceci implique $\exists \alpha \in A.t_\alpha(a,a')$ et $a \in I(\underline{a})$ et donc $\Phi(\underline{a},a)$.

(c) Si $[J_\beta(\gamma,\underline{a},a) \wedge \gamma > 0 \wedge t_\alpha(a,a')]$ alors $a \in I(\underline{a})$ de sorte que d'après l'hypothèse de fatalité, nous devons avoir $a' \in A(\underline{a})$ et $a' \prec_{\underline{a}} a$ et donc $\underline{\mathrm{rk}}(A(\underline{a}),\prec_{\underline{a}})(a') < \underline{\mathrm{rk}}(A(\underline{a}),\prec_{\underline{a}})(a)$ et $J_\beta(\underline{\mathrm{rk}}(A(\underline{a}),\prec_{\underline{a}})(a'),\underline{a},a')$.

(d) $I_\beta(0,\underline{a},a) \Rightarrow (\underline{\mathrm{rk}}(A(\underline{a}),\prec_{\underline{a}})(a) = 0) \Rightarrow a \in G(\underline{a}) \Rightarrow \Psi(\underline{a},a)$.

En particulier, pour démontrer les propriétés de fatalité de sémantiques $\langle S,A,\Sigma\langle S,A,t,\varepsilon\rangle\rangle$, on peut choisir des bons-ordres isomorphes à $\langle\omega,<\rangle$ quand le nondéterminisme de $\langle S,A,t,\varepsilon\rangle$ est fini et des bons-ordres isomorphes à $\langle\omega_1,<\rangle$ quand le nondéterminisme de $\langle S,A,t,\varepsilon\rangle$ est dénombrable, (ω et $\omega_1=\omega^+$ sont réguliers de sorte que $\omega^+=\omega$ et $\omega_1^+=\omega_1$).

5.2.6.3 Quels ordinaux sont nécessaires ?

Théorème 5.8.6.3 v1

Soient $\mathfrak{m}$ un cardinal régulier fini ou infini et $\langle S,A,t,\varepsilon\rangle$ un système de transition dont le nondéterminisme est $\mathfrak{m}$-borné. Supposons que nous voulions démontrer que Ψ est fatale sous invariance de Φ pour $\langle S,A,\Sigma\langle S,A,t,\varepsilon\rangle\rangle$ en utilisant le principe d'induction $(\mathcal{F}_6')$ avec $\Gamma<\mathfrak{m}^+=\mathfrak{m}$. Ceci n'est pas complet.

Démonstration

C'est évident quand $\mathfrak{m}=\omega$ et Γ est un nombre naturel, aussi nous pouvons supposer $\Gamma\geq\omega$.

Définissons $S=(\{\perp\}\cup\{\Gamma+1\})$ où $\perp\notin(\Gamma+1)$, $A=\{a_{\omega}\}$. Nous avons $\underset{\sim}{\mathrm{card}}(S)=(1+\underset{\sim}{\mathrm{card}}(\Gamma+1))=\underset{\sim}{\mathrm{card}}(\Gamma+1)=\underset{\sim}{\mathrm{card}}(\Gamma)\leq\Gamma<\mathfrak{m}^+=\mathfrak{m}$. Ainsi le nondéterminisme de $\langle S,A,t,\varepsilon\rangle$ est $\mathfrak{m}$-borné. Définissons $t_{a_{\omega}}(x,x')=[(x=\perp\wedge x'\leq\Gamma)\vee(0\leq x'<x\leq\Gamma)]$, $\varepsilon(\underline{x})=[\underline{x}=\perp]$, $\Phi(\underline{x},x)=tt$ et $\Psi(\underline{x},\bar{x})=[\bar{x}=0]$.

Une trace d'exécution $p\in\Sigma\langle S,A,t,\varepsilon\rangle$ est telle que $p_0=\perp$, $p_i\in\underset{\sim}{\mathrm{Ord}}$, $i\in(\omega\vee 0)$ et $p_1>p_2>\dots$ de sorte que nous devons avoir fatalement un $p_i=0$ puisque $\underset{\sim}{\mathrm{wo}}(\underset{\sim}{\mathrm{ord}},<)$. Ainsi Ψ est fatale sous invariance de Φ pour $\langle S,A,\Sigma\langle S,A,t,\varepsilon\rangle\rangle$.

Cependant, ceci ne peut être démontré au moyen de $(\mathcal{F}_6')$. Sinon, ayant trouvé J_6 satisfaisant les conditions de vérification de $(\mathcal{F}_6')$, nous pouvons construire une séquence infinie strictement décroissante $\alpha_0>\alpha_1>\dots$ d'ordinaux, comme suit : Posons $\alpha_0=\Gamma$. Puisque $\varepsilon(\perp)$, nous devons avoir un $\underline{\gamma}\in\Gamma$ tel que $J_6(\underline{\gamma},\perp,\perp)$. Posons $\alpha_1=\underline{\gamma}$. Puisque $\neg\Psi(\perp,\perp)$, nous avons $\neg J_6(0,\perp,\perp)$

et donc $\alpha_1 > 0$. Mais $[\alpha_1 > 0 \wedge J_6(\alpha_1, \bot, \bot) \wedge t_{\underline{a}}(\bot, \alpha_0)] \Rightarrow [\exists \gamma < \alpha_1 . J_6(\gamma, \bot, \alpha_0)]$. Posons $\alpha_2 = \gamma$. Nous avons $\Gamma = \alpha_0 > \alpha_1 > \alpha_2$ et $J_6(\alpha_2, \bot, \alpha_0)$. Puisque $\alpha_0 > 0$, nous avons $\neg \Psi(\bot, \alpha_0)$ et donc $\neg J_6(0, \bot, \alpha_0)$ et $\alpha_2 \neq 0$. Supposons que nous ayons construit la séquence jusqu'en α_{j+2} avec $\beta \geq \alpha_j > \alpha_{j+1} > \alpha_{j+2} > 0$ et $J_6(\alpha_{j+2}, \bot, \alpha_j)$. D'après $(\mathcal{F}_6')$ nous avons $[\alpha_{j+2} > 0 \wedge J_6(\alpha_{j+2}, \bot, \alpha_j) \wedge t_{\underline{a}}(\alpha_j, \alpha_{j+1})] \Rightarrow [\exists \gamma < \alpha_{j+2} . J_6(\gamma, \bot, \alpha_{j+1})]$. Posons $\alpha_{j+3} = \gamma$. Puisque $\alpha_{j+1} > 0$ nous avons $\neg \Psi(\bot, \alpha_{j+1})$ de sorte que $\beta \geq \alpha_{j+1} > \alpha_{j+2} > \alpha_{j+3} > 0$ et $J_6(\alpha_{j+3}, \bot, \alpha_{j+1})$. Et ainsi la séquence peut être prolongée indéfiniment.

□

Bien que ceci soit restreint aux cardinaux réguliers, le résultat est très général puisque le premier cardinal singulier infini est $\omega_\omega = \bigcup_{i \in \omega} \omega_i$ (qui est trop grand pour avoir un intérêt quelconque en informatique).

Un cas particulier intéressant est celui de la méthode de Knuth [68], Luckham-Suzuki [75] pour montrer la terminaison qui consiste à utiliser un compteur par boucle du programme, qui est strictement incrémenté à chaque itération dans la boucle et dont la valeur est bornée (nous utiliserons, ce qui revient au même, un seul compteur γ incrémenté à chaque pas du programme et borné par $\beta \in \omega$) :

$$
\begin{array}{l}
(\exists \beta \in \omega,\ J \in (\omega \times S \times S \to \{tt, ff\}) . \\
\quad (\forall s \in S .\ \exists \gamma \in \omega .\ J(\gamma, s, s)) \\
\quad \wedge (\forall s, s' \in S,\ \gamma' \in \omega . \\
\qquad J(\gamma', s, s') \Rightarrow [\ (\gamma' \leq \beta \ \wedge\ \Phi(s, s') \ \wedge \\
\qquad\qquad \wedge\ \exists s'' \in S, a \in A .\ t_a(s', s'') \\
\qquad\qquad \wedge\ \forall s'' \in S, a \in A . [t_a(s', s'') \Rightarrow \exists \gamma'' < \gamma' .\ J(\gamma'', s, s'')]) \\
\qquad\qquad \vee\ \Psi(s, s')])
\end{array}
\qquad (\mathcal{F}_k')
$$

La méthode est correcte car on retrouve $(\mathcal{F}'_6)$ avec $\Gamma'_6 = (\beta+1)$ et $J'_6(\delta, \Delta, \Delta') = J_R(\beta-\delta, \Delta, \Delta')$. Elle est évidemment sémantiquement complète quand le nondéterminisme est fini (car on retrouve alors $(\mathcal{F}'_8)$). Le résultat ci-dessus montre que cette méthode n'est pas sémantiquement complète quand le nondéterminisme n'est pas fini, ce qui explique qu'elle n'ait pas pu être généralisée (par exemple au cas des programmes parallèles équitables).

5.2.7 DECOMPOSITION DES CONDITIONS DE VERIFICATION

La méthode de construction d'une méthode de preuve d'invariance à partir d'une sémantique opérationnelle et d'un principe d'induction par décomposition de l'hypothèse d'induction globale en hypothèses d'induction locales proposée en 4.3 est évidemment directement applicable aux preuves de fatalité et nous n'y reviendrons pas. La plupart du temps, cette décomposition des conditions de vérification s'obtient par décomposition de de l'ensemble des états ou des actions du système de transition (Cousot-P [81]) et nous en donnerons quelques exemples.

5.2.7.1 Décomposition des conditions de vérification au moyen d'un recouvrement de l'ensemble des états du système de transition

Un recouvrement de l'ensemble des états d'un système de transition $\langle S, A, t, \varepsilon\rangle$ est une paire $\langle \mathcal{R}, \pi\rangle$ telle que :

- $\mathcal{R}$ est un ensemble fini non vide de noms de blocs
- $\pi \in (\mathcal{R} \to (S \to \{tt, ff\}))$ caractérise un recouvrement de la classe s des états, i.e. $\forall s \in S.\ \exists k \in \mathcal{R}.\ \pi_k(s)$

(La classe s des états peut être décomposée en donnant des noms aux blocs d'états jouant des rôles similaires. Par exemple un π_k, $k \in \mathcal{R}$ peut caractériser les états ayant une composante contrôle donnée).

Etant donné un recouvrement $\langle \mathcal{R}, \pi\rangle$ des états de $\langle S, A, t, \varepsilon\rangle$, une preuve de fatalité de ψ sous invariance de Φ pour $\langle S, A, \Sigma\langle S, A, t, \varepsilon\rangle\rangle$ peut être décomposée pour chaque bloc du recouvrement de l'ensemble des états :

$$(\exists \Gamma \in \underline{Ord},\ J \in (r \to (\Gamma \times S \times S \to \{tt, ff\}))).$$

$$\begin{aligned}
&[\forall k, \ell \in r,\ \underline{a}, a, a', \bar{a} \in S,\ \alpha \in A,\ \gamma \in \Gamma. \\
&\quad ([\varepsilon(\underline{a}) \wedge \pi_k(\underline{a})] \Rightarrow [\exists \underline{\gamma} \in \Gamma^{*}.\ J_k(\underline{\gamma}, \underline{a}, \underline{a})]) \\
&\quad \wedge ([J_k(\gamma, \underline{a}, a) \wedge \pi_k(a) \wedge \gamma > 0] \Rightarrow [\Phi(\underline{a}, a) \wedge \exists a' \in S, \alpha \in A.\ t_\alpha(a, a')]) \\
&\quad \wedge ([J_k(\alpha, \underline{a}, a) \wedge \pi_k(a) \wedge \gamma > 0 \wedge t_\alpha(a, a') \wedge \pi_\ell(a')] \\
&\qquad\qquad \Rightarrow [\exists \gamma' < \gamma.\ J_\ell(\gamma', \underline{a}, a')]) \\
&\quad \wedge ([J_k(0, \underline{a}, \bar{a}) \wedge \pi_k(\bar{a})] \Rightarrow \Psi(\underline{a}, \bar{a}))])
\end{aligned} \qquad (\mathcal{F}^{\Gamma}_{g.A})$$

Théorème 5.2.7.1.v1

est fatale sous invariance de Φ pour $\langle S, A, \Sigma\langle S, A, t, \varepsilon\rangle\rangle$)

Démonstration

- Correction, $(\mathcal{F}^{\Gamma}_{g.A}) \Rightarrow (\mathcal{F}'_c)$

  Choisir $\Gamma_c = \Gamma_g$, $J_c(\gamma, \underline{a}, a) = [\exists k \in r.\ (\pi_k(a) \wedge J_{gk}(\gamma, \underline{a}, a))]$.

- Complétude sémantique, $(\mathcal{F}'_c) \Rightarrow (\mathcal{F}^{\Gamma}_{g.A})$

  Choisir $\Gamma_g = \Gamma_c$, $J_{gk}(\gamma, \underline{a}, a) = [\pi_k(a) \wedge J_c(\gamma, \underline{a}, a)]$.

□

On observera que de manière équivalente $(\mathcal{F}^{\Gamma}_{g.A})$ s'obtient à partir de $(\mathcal{F}'_c)$ par la décomposition 4.3.2.4.4.

Exemple

Une version de la méthode de Floyd pour les programmes Ps, avec des états de la forme $\langle c, m\rangle$ où $c \in C[\![Ps]\!]$ est un point de contrôle et $m \in \mathcal{M}$ est un état mémoire, peut être dérivée à partir de $(\mathcal{F}^{\Gamma}_{g.A})$ en choisissant $r = C[\![Ps]\!]$, $\pi_k(\langle c, m\rangle) = [c = k]$, $\Phi(\langle \underline{c}, \underline{m}\rangle, \langle c, m\rangle) = \neg \sigma(\langle c, m\rangle)$ et $\Psi(\langle \underline{c}, \underline{m}\rangle, \langle c, m\rangle) = [\sigma(\langle c, m\rangle) \wedge \psi(\underline{m}, m)]$.

Pour comparer avec le paragraphe 5.2.1, nous pourrons poser $P_c(\underline{m}, m) = [\exists \underline{c} \in C[\![Ps]\!].\ J_{g_c}(\langle \underline{c}, \underline{m}\rangle, \langle c, m\rangle)]$.

□

5.2.7.2 Décomposition des conditions de vérification au moyen d'un recouvrement de l'ensemble des actions du système de transition

En décomposant l'hypothèse d'induction globale J dans $(\mathcal{F}_4)$ à $(\mathcal{F}_8)$ en une disjonction d'hypothèses d'induction locales correspondant à chaque action $a \in A$ du système de transition $\langle S, A, t, \varepsilon \rangle$, une preuve de fatalité de Ψ sous invariance de Φ pour $\langle S, A, \Sigma\langle S, A, t, \varepsilon\rangle\rangle$ peut être décomposée en :

- $\underset{\sim}{card}(A)$ preuves indépendantes d'invariance et de terminaison pour chaque bloc, $((\mathcal{F}_{g.t})\text{-}a\text{-}b\text{-}c\text{-}d)$
- $\underset{\sim}{card}(A) \times (\underset{\sim}{card}(A)-1)$ vérifications d'absence d'interférences entre preuves de blocs distincts, $((\mathcal{F}_{g.t})\text{-}e)$
- une preuve d'absence d'états de blocage (par exemple par l'absurde) $((\mathcal{F}_{g.t})\text{-}f)$

$$
\begin{aligned}
&(\exists \Gamma \in \underset{\sim}{Ord},\ J \in (A \to (\Gamma \times S \times S \to \{tt, ff\})).\\
&\quad [\forall a \in A.\\
&\qquad (\forall \underline{\lambda}, \lambda, \lambda' \in S,\ \gamma \in \Gamma.\\
&\text{(a)}\qquad\quad (\varepsilon(\underline{\lambda}) \Rightarrow [\exists \underline{\gamma} \in \Gamma.\ J_a(\underline{\gamma}, \underline{\lambda}, \underline{\lambda})])\\
&\text{(b)}\qquad \wedge\ ([J_a(\gamma, \underline{\lambda}, \lambda) \wedge \gamma > 0 \wedge t_a(\lambda, \lambda')] \Rightarrow [\exists \gamma' < \gamma.\ J_a(\gamma', \underline{\lambda}, \lambda')])\\
&\text{(c)}\qquad \wedge\ ([J_a(\gamma, \underline{\lambda}, \lambda) \wedge \gamma > 0] \Rightarrow \Phi(\underline{\lambda}, \lambda))\\
&\text{(d)}\qquad \wedge\ (J_a(0, \underline{\lambda}, \lambda) \Rightarrow [\Psi(\underline{\lambda}, \lambda) \vee \beta_a(\lambda)]))] \qquad (\mathcal{F}_{g.t})\\
&\quad \wedge\ [\forall a \in A,\ b \in (A \setminus a).\\
&\qquad (\forall \underline{\lambda}, \lambda, \lambda' \in S,\ \gamma \in \Gamma,\ \alpha \in \Gamma.\\
&\text{(e)}\qquad\quad ([J_a(\gamma, \underline{\lambda}, \lambda) \wedge J_b(\alpha, \underline{\lambda}, \lambda) \wedge \gamma > 0 \wedge t_b(\lambda, \lambda')] \Rightarrow\\
&\qquad\qquad\qquad [\exists \gamma' < \gamma.\ J_a(\gamma', \underline{\lambda}, \lambda')]))]\\
&\quad \wedge\ [\forall \underline{\lambda}, \lambda \in S.\\
&\text{(f)}\qquad\quad ([\forall a \in A.\ \beta_a(\lambda) \wedge \neg\Psi(\underline{\lambda}, \lambda)] \Rightarrow [\exists a \in A.\ \forall \gamma \in \Gamma.\ \neg J_a(\gamma, \underline{\lambda}, \lambda)])])
\end{aligned}
$$

où

$\beta \in (A \to (S \to \{tt, ff\}))$ caractérise les états de blocage de l'action a

$\beta_a(A) = [\forall A' \in S. \neg t_a(A, A')]$

Théorème 5.2.7.2 v1

$(\mathcal{F}'_{g.t}) \Leftrightarrow$ (Ψ est fatale sous invariance de Φ pour $<S, A, \Sigma<S, A, t, \varepsilon>>$)

Démonstration

- Correction, $(\mathcal{F}'_{g.t}) \Rightarrow (\mathcal{F}'_5)$

  Choisir $W_5 = (A \to \delta)$, $\gamma' <_5 \gamma$ si et seulement si $(\exists a \in A. [(\gamma'_a < \gamma_a) \wedge (\forall b \in (A \setminus a). \gamma'_b \leq \gamma_b)])$

  et $J_5(\gamma, \underline{A}, A) = [\forall a \in A. J_{g_a}(\gamma, \underline{A}, A)]$.

- Complétude sémantique, $(\mathcal{F}'_5) \Rightarrow (\mathcal{F}'_{g.t})$

  Choisir $J_{g.t}(\gamma, \underline{A}, A) = [(J_6(\gamma, \underline{A}, A) \wedge \neg\Psi(\underline{A}, A) \wedge \gamma > 0) \vee (\Psi(\underline{A}, A) \wedge \gamma = 0)]$.

□

Exemple

Une généralisation de la méthode de preuve de Lamport [80] à la preuve de correction totale de programmes parallèles asynchrones $[\![Pra_0 \| \ldots \| Pra_{m-1}]\!]$ peut être dérivée à partir de $(\mathcal{F}_{g.t})$. La relation de transition associée au programme est décomposée en card(A) blocs t_a correspondant à chaque processus Pra_a, ainsi J_{g_a} est un invariant global pour le processus Pra_a. Nous pouvons aussi partir d'un des principes d'induction $(\mathcal{F}_1)$ - $(\mathcal{F}_8)$ en utilisant la décomposition définie en 4.3.3.4.6.

□

5.2.7.3 Combinaison des décompositions selon les états et les actions du système de transition

Les décompositions selon l'ensemble des états et selon l'ensemble des actions peuvent être combinées et appliquées récursivement de sorte à induire des décompositions plus fines des conditions de vérification. Par exemple, nous pourrons considérer des systèmes de transition $\langle S, A, t, \varepsilon\rangle$ et des paires $\langle r, \pi\rangle$ où $t \in (A \to (S\times S \to \{tt, ff\}))$ et $\forall a\in A.\ (\pi_a \in (r_a \to (S \to \{tt, ff\})))$ avec $\forall s\in S, a\in A.\ \exists i \in r_a.\ \pi_a^i(s)$. Le principe d'induction correspondant est :

$$
\begin{aligned}
&(\exists \Gamma \in \underline{Ord},\ J.\ [\forall a\in A,\ i\in r_a.\ J_a^i \in (\Gamma\times S\times S \to \{tt, ff\})]\ \wedge\\
&\quad [\forall a\in A.\\
&\qquad (\forall i, i'\in r_a.\\
&\qquad\quad (\forall \underline{s}, s, s'\in S,\ \gamma\in\Gamma.\\
&\text{(a)}\qquad\quad ([\varepsilon(\underline{s}) \wedge \pi_a^i(\underline{s})] \Rightarrow [\exists \underline{\gamma}\in\Gamma.\ J_a^i(\underline{\gamma}, \underline{s}, \underline{s})])\\
&\text{(b)}\qquad\quad \wedge ([J_a^i(\gamma, \underline{s}, s) \wedge \pi_a^i(s) \wedge \gamma>0 \wedge t_a(s,s') \wedge \pi_a^{i'}(s')]\\
&\qquad\qquad\qquad\qquad \Rightarrow [\exists \gamma'<\gamma.\ J_a^{i'}(\gamma', \underline{s}, s')])\\
&\text{(c)}\qquad\quad \wedge ([J_a^i(\gamma, \underline{s}, s) \wedge \pi_a^i(s) \wedge \alpha>0] \Rightarrow \Phi(\underline{s}, s))\\
&\text{(d)}\qquad\quad \wedge ([J_a^i(0, \underline{s}, s) \wedge \pi_a^i(s)] \Rightarrow [\Psi(\underline{s}, s) \vee \beta_a(s)]))))]\\
&\quad \wedge [\forall a\in A,\ i, i'\in r_a,\ b\in(A\setminus a),\ j\in r_b.\\
&\qquad (\forall \underline{s}, s, s'\in S,\ \gamma, \alpha\in\Gamma.\\
&\text{(e)}\qquad\quad ([J_a^i(\gamma, \underline{s}, s) \wedge \pi_a^i(s) \wedge J_b^j(\alpha, \underline{s}, s) \wedge \pi_b^j(s) \wedge t_b(s,s') \wedge \pi_a^{i'}(s')]\\
&\qquad\qquad\qquad\qquad \Rightarrow [\exists \gamma'<\gamma.\ J_a^{i'}(\gamma', \underline{s}, s')]))]\\
&\quad \wedge [\forall \underline{s}, s\in S.\\
&\text{(f)}\qquad\quad ([\forall a\in A.\ \beta_a(s) \wedge \neg\Psi(\underline{s}, s)] \Rightarrow [\exists a\in A.\ \forall i\in r_a.\ (\pi_a^i(s) \Rightarrow \forall \gamma\in\Gamma.\ \neg J_a^i(\gamma, \underline{s}, s))])])
\end{aligned}
\qquad (\mathcal{F}^{\Phi}_{g.t_b})
$$

Théorème 5.2.7.3 v 1

$(\mathcal{F}^{\Phi}_{g.t_b}) \Leftrightarrow (\Psi$ est fatale sous invariance de Φ pour $\langle S, A, \Sigma\langle S, A, t, \varepsilon\rangle\rangle)$

Démonstration

- Correction, $(\mathcal{F}_{g.t_b}) \Rightarrow (\mathcal{F}_{g.t})$

  Choisir $J_{g.t\,a}(\gamma, \underline{\Delta}, \Delta) = [\exists i \in r_a . (\pi_a^i(\Delta) \wedge J_{g.t_b\,a}^{\,i}(\gamma, \underline{\Delta}, \Delta))]$

- Complétude sémantique, $(\mathcal{F}_{g.t}) \Rightarrow (\mathcal{F}_{g.t_b})$

  Choisir $J_{g.t_b\,a}^{\,i}(\gamma, \underline{\Delta}, \Delta) = [\pi_a^i(\Delta) \wedge J_{g.t\,a}(\gamma, \underline{\Delta}, \Delta)]$

□

Exemple

Une généralisation de la méthode de preuve de Lamport [77] (et Owicki-Gries [76]) à la preuve de correction totale de programmes parallèles asynchrones $[\![Pra_0 \| \ldots \| Pra_{m-1}]\!]$ peut être dérivée à partir de $(\mathcal{F}_{g.t_b})$. La relation de transition associée au programme est décomposée en blocs t_a correspondant à chaque processus Pra_a, r_a est l'ensemble des points de contrôle du processus Pra_a et π_a^i est vrai pour les états dont la composante contrôle pour le processus Pra_a est égale à i. Par suite, nous pouvons associer $J_{g\,a}^{\,i}$ au point i du processus Pra_a. En outre, l'exécution atomique de commandes du processus Pra_b ne peut pas modifier le contrôle dans le processus Pra_a de sorte que le seul cas à considérer dans $(\mathcal{F}_{g.t_b})$-(c) est $i=i'$. Nous pouvons aussi appliquer la décomposition 4.3.2.4.4 (ou 4.3.2.4.5) à l'un des principes d'induction $(\mathcal{F}_1)$ à $(\mathcal{F}_8)$.

□

5.2.8 PRINCIPES D'INDUCTION "A LA FLOYD" POUR DEMONTRER DES PROPRIETES DE FATALITE DE SEMANTIQUES NON CLOSES

Si nous voulons démontrer des propriétés de fatalité pour des programmes ayant une sémantique close, nous pourrons utiliser n'importe quel principe d'induction $(\mathcal{F}_1)$ à $(\mathcal{F}_9)$ pour le système de transition engendré par cette sémantique.

Comme ces principes d'induction incluent une preuve d'invariance, nous retrouvons les difficultés (et les solutions) du chapitre 4 concernant les sémantiques non closes. Il s'y ajoute la difficulté que les preuves de terminaison par la méthode de l'ordre bien fondé ne sont pas applicables aux sémantiques non closes comme le montre le contre-exemple suivant :

Exemple 5.2.8-1

Pour continuer l'exemple 5.1.1-1, si nous voulions démontrer que Ψ définie par $\Psi(0) = ff$, $\Psi(1) = tt$ est fatale pour la sémantique $\langle S, A, \Sigma \rangle$ définie dans l'exemple 2.1.2-2 ($S = \{0,1\}$, $A = \{a,b\}$, $\Sigma = \{0 \xrightarrow{b} 1, 0 \xrightarrow{a} 0 \xrightarrow{b} 1, 0 \xrightarrow{a} 0 \dots 0 \xrightarrow{a} 0 \xrightarrow{b} 1\}$) par le principe d'induction $(\mathcal{F}_2)$, nous aurons $f_2(0,0) \prec_2 f_2(0,0)$ contrairement à $wo(Rng(f_2), \prec_2)$.

□

5.2.8.1 Principes d'induction "à la Floyd" pour une sémantique non close définie par concordance avec une sémantique close

Il est toujours possible de définir une sémantique non close $\langle S, A, \Sigma\rangle$ par concordance avec une sémantique close (engendrée par un système de transition $\langle S^\#, A^\#, t^\#, \varepsilon^\#\rangle$) à une fonction $f_\Delta^\# \in (S^\# \to S)$ entre états près (cf. 2.7.2.2 v1) :

$$\langle S, A, \Sigma\rangle = \simeq\langle f_\Delta^\#\rangle(\langle S^\#, A^\#, \Sigma\langle S^\#, A^\#, t^\#, \varepsilon^\#\rangle\rangle)$$

Par conséquent, pour démontrer que $\Psi \in (S\times S \to \{tt, ff\})$ est fatale pour $\langle S, A, \Sigma\rangle$, il est toujours correct et possible (d'après 5.1.2 v3) d'utiliser l'un quelconque des principes d'induction des paragraphes 5.2.2, 5.2.3, 5.2.6.2 et 5.2.7 pour $\langle S^\#, A^\#, t^\#, \varepsilon^\#\rangle$ et $\Psi^\#$ définie par $\Psi^\#(a_0, a_1) = \Psi(f_\Delta^\#(a_0), f_\Delta^\#(a_1))$.

Par exemple, en utilisant le principe d'induction (F_6'), nous

$$(\exists S^\#, A^\#, t^\# \in (S^\# \times S^\# \to \{tt, ff\}),\ \varepsilon^\# \in (S^\# \to \{tt, ff\}),\ f_\Delta^\# \in (S^\# \to S).$$

$$\langle S, A, \Sigma\rangle = \simeq\langle f_S^\#\rangle(\langle S^\#, A^\#, \Sigma\langle S^\#, A^\#, t^\#, \varepsilon^\#\rangle\rangle)$$
$$\wedge\ [\exists \Gamma \in \underset{\sim}{Ord},\ J \in (\Gamma \times S^\# \times S^\# \to \{tt, ff\}).$$

$$(\forall a \in S^\#.\ \exists \gamma \in \Gamma.\ J(\gamma, a, a)) \qquad (F_6'^\#)$$
$$\wedge\ (\forall a, a' \in S^\#,\ \gamma' \in \Gamma.$$
$$J(\gamma', a, a') \Rightarrow$$
$$[\ (\Phi(f_\Delta^\#(a), f_\Delta^\#(a')) \wedge \exists a'' \in S^\#, q \in A^\#.\ t_q^\#(a', a'') \wedge$$
$$\forall a'' \in S^\#,\ q \in A^\#.[\ t_q^\#(a', a'') \Rightarrow \exists \gamma'' < \gamma'.\ J(\gamma'', a, a'')])$$
$$\vee\ (\varepsilon^\#(a) \Rightarrow \Psi(f_S^\#(a), f_\Delta^\#(a')))])]]$$

Quand $S^\#$, $A^\#$, $t^\#$ et $\varepsilon^\#$ sont définis en termes de S, A, t, ε (et des domaines auxiliaires), nous pouvons leur substituer leurs définitions dans les conditions de vérifications relatives au système de transition $\langle S^\#, A^\#, t^\#, \varepsilon^\# \rangle$ de façon à obtenir des conditions de vérification équivalentes relatives au système de transition original $\langle S, A, t, \varepsilon \rangle$ (et des variables auxiliaires). Alors, par construction, le principe d'induction est correct et sémantiquement complet.

Exemple 5.2.8.1-1

Reprenons l'exemple 2.7.3.2-2 où la réduction aux traces faiblement équitables $\underset{\sim}{\text{Wfair}}\langle A \rangle(\langle S, A, \Sigma\langle S, A, t, \varepsilon\rangle\rangle)$ d'une sémantique $\langle S, A, \Sigma\langle S, A, t, \varepsilon\rangle\rangle$ engendrée par un système de transition $\langle S, A, t, \varepsilon \rangle$ est définie par concordance avec la sémantique engendrée par le système de transition $\langle S^\natural, A, t^\natural, \varepsilon^\natural \rangle$ défini par :

$$S^\natural = (A \to \omega) \times S$$

$$t^\natural(\langle m, s\rangle, \langle m', s'\rangle) = [\exists a \in A.\ t_a(s, s') \wedge ([m_a > 0 \wedge m'_a < m_a \wedge \forall b \in (A \setminus a).(m'_b = m_b)] \vee [\forall b \in A.((\beta_b(s) \vee m_b = 0) \wedge m'_b > 0)])]$$

$$\varepsilon^\natural(\langle m, s\rangle) = \varepsilon(s)$$

à une fonction $f_s^\natural$ entre états près telle que :

$$f_s^\natural(\langle m, s\rangle) = s$$

En utilisant le principe d'induction $(\mathcal{F}'_6)$ pour $\langle S^\natural, A, t^\natural, \varepsilon^\natural \rangle$ et en posant $J_6(\gamma, \langle \underline{m}, \underline{s}\rangle, \langle m, s\rangle) = J_{12}(\gamma, m, \underline{s}, s)$, nous obtenons un principe d'induction pour démontrer les propriétés de fatalité de la sémantique $\underset{\sim}{\text{Wfair}}\langle A \rangle(\langle S, A, \Sigma\langle S, A, t, \varepsilon\rangle\rangle)$ quand $\underset{\sim}{\text{card}}(A) < \omega$:

$$(\exists \Gamma \in O_{rd}^{\omega},\quad J \in (\Gamma \times (A \to \omega) \times S \times S \to \{tt, ff\}).$$

$$[\forall \underline{a}, a, a', \bar{a} \in S,\ \gamma \in \Gamma,\ \underline{m}, m, m' \in (A \to \omega),\ a \in A.$$

$$(\varepsilon(\underline{a}) \Rightarrow [\exists \underline{\gamma} \in \Gamma^{\omega}.\ J(\underline{\gamma}, \underline{m}, \underline{a}, \underline{a})])$$

$$\wedge$$

$$([J(\gamma, m, \underline{a}, a) \wedge \gamma > 0] \qquad (\mathcal{F}'_{12})$$

$$\Rightarrow [\Phi(\underline{a}, a) \wedge \exists a' \in S, a \in A.\ t_a(a, a') \wedge (m_a > 0 \vee B(m, a))])$$

$$\wedge$$

$$([J(\gamma, m, \underline{a}, a) \wedge \gamma > 0 \wedge t_a(a, a') \wedge ([m_a > 0 \vee m' <_{\bar{a}} m] \vee [B(m, a) \wedge m' > 0])]$$

$$\Rightarrow [\exists \gamma' < \gamma.\ J(\gamma', m', \underline{a}, a')])$$

$$\wedge$$

$$(J(0, m, \underline{a}, \bar{a}) \Rightarrow \Psi(\underline{a}, \bar{a}))])$$

où

$B(m, a) = [\forall b \in A.\ (\beta_b(a) \vee m_b = 0)]$

$m' <_{\bar{a}} m$ si et seulement si $(m'_a < m_a \wedge \forall b \in (A \setminus a).\ (m'_b = m_b))$

$m' > 0$ si et seulement si $(\forall b \in A.\ m'_b > 0)$

Pour l'exemple 5.1.1-1, $(\mathcal{F}'_{12})$ est satisfait par $J_{12}(\gamma, m, \underline{x}, x) =$ $[(\gamma = 0 \wedge x = 1) \vee (\gamma = m_1 + m_2 > 0)]$.

□

5.2.8.2 Principes d'induction "à la Floyd" pour une sémantique non close spécifiée par un système de transition et une condition sur les traces qu'il engendre

Nous pouvons également réutiliser l'idée du paragraphe 4.2.3.1 qui consiste à cumuler l'histoire des calculs dans une variable auxiliaire. En effet,

- quand la sémantique n'est pas fermée par fusion, la condition de vérification (c) de $(\mathcal{F}_8)$ doit être affaiblie pour que le principe d'induction soit sémantiquement complet. En effet l'invariant doit être vrai pour tous les états qui peuvent être atteints en suivant le préfixe d'une trace où ψ n'est jamais satisfaite mais pas forcément pour tous les préfixes obtenus par transitions successives $\bigvee_{a \in A} t_a$.

- quand la sémantique n'est pas réduite par élimination des traces préfixes strictes, la condition (b) de $(\mathcal{F}_8)$ doit être renforcée pour que le principe d'induction soit correct. Il faut s'assurer que pour les traces finies, le but ψ est atteint avant la fin de la trace (qui peut ne pas être un état de blocage).

- quand la sémantique n'est pas fermée par limites, la condition (c) de $(\mathcal{F}_8)$ doit être affaiblie (comme le montre le contre-exemple 5.2.8.1) pour que le principe d'induction soit sémantiquement complet. On ne peut pas reprendre l'idée d'une quantité prise dans un ensemble bien-fondé à valeurs successives strictement décroissantes à chaque pas ou pour un ensemble "statique" de points de coupure (statique étant compris comme signifiant que l'ensemble des points de coupure est choisi uniquement en fonction de l'ensemble S d'états).
Cependant, il est possible de choisir l'ensemble des points de coupure dynamiquement, c'est-à-dire que le moment où la quantité décroît

strictement est choisi, non plus en fonction de l'état courant du calcul, mais en fonction de l'histoire complète du calcul.

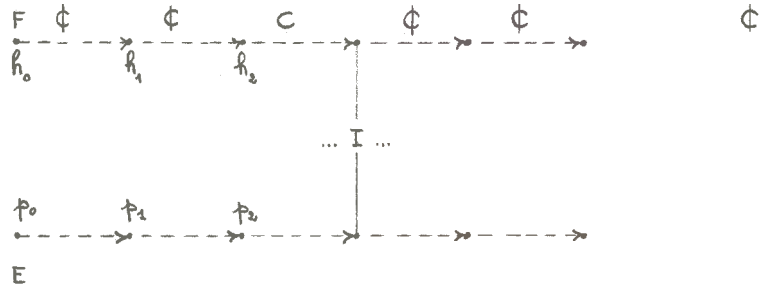
Nous commençons par le cas général, puis examinons chaque cas particulier où la sémantique n'est pas close parceque seulement l'une ou deux des conditions du théorème 2.6.8∿4 ne sont pas satisfaites.

5.2.8.2.1 Sémantique non fermée par fusion, non réduite par élimination des traces préfixes stricts et non fermée par limites

En tenant compte des remarques précédentes, nous obtenons le principe d'induction suivant (où $\langle S,A,t,\varepsilon\rangle$ est engendré par $\langle S,A,\Sigma\rangle$) :

$$
\begin{array}{l}
(\exists H,\ F\in(S\times H\to\{tt,ff\}),\ I\in(S\times H\to\{tt,ff\}),\ L,R\in(S\times H\times A\times S\to\{tt,ff\}),\\
\ C,\Phi\in(H\times A\times S\times H\to\{tt,ff\}).\\
\\
\quad \underline{Live}\langle S,A,\Sigma\rangle(A,F,I,C,\Phi)(L)\\
\quad\wedge\\
\quad \underline{Nexct}\langle S,A,\Sigma\rangle(H,F,I,C,\Phi)(R)\\
\quad\wedge\\
\quad \underline{D\text{-}cutset}\langle S,A,\Sigma\rangle(H,F,I,C,\Phi)\\
\quad\wedge\\
\quad (\exists\Gamma\in\underline{Ord},\ J\in(\Gamma\times S\times S\times H\to\{tt,ff\}).\qquad (\mathcal{F}_{13})\\
\quad\ (\forall\underline{\alpha},\alpha\in S,\ \gamma\in\Gamma,\ h\in H.\\
\quad\quad (\varepsilon(\underline{\alpha})\Rightarrow[\exists\underline{\gamma}\in\Gamma,\ \underline{h}\in H.(F(\underline{\alpha},\underline{h})\wedge J(\underline{\gamma},\underline{\alpha},\underline{\alpha},\underline{h}))]\\
\quad\quad\wedge(J(\gamma,\underline{\alpha},\alpha,h)\Rightarrow\ \Psi(\underline{\alpha},\alpha)\\
\quad\quad\quad\vee\ [\ \Phi(\underline{\alpha},\alpha)\wedge I(\alpha,h)\\
\quad\quad\quad\quad\wedge\ \exists a\in A,\alpha'\in S.[t_a(\alpha,\alpha')\wedge L(\alpha,h,a,\alpha')]\\
\quad\quad\quad\quad\wedge\ \forall a\in A,\alpha'\in S.([t_a(\alpha,\alpha')\wedge R(\alpha,h,a,\alpha')]\Rightarrow\\
\quad\quad\quad\quad\quad [\ (\exists\gamma'<\gamma,\ h'\in H.(C(h,a,\alpha',h')\wedge J(\gamma',\underline{\alpha},\alpha',h')))\\
\quad\quad\quad\quad\quad\vee(\exists h'\in H.(\Phi(h,a,\alpha',h')\wedge J(\gamma,\underline{\alpha},\alpha',h')))]]))))
\end{array}
$$

Avant de donner les définitions manquantes, remarquons que si p est le préfixe d'une trace de Σ ($\exists p' \in \Sigma . p \mapsto p'$) sur lequel le but n'est pas atteint ($\forall i \in |p| . \neg \Psi(p_0, p_i)$), alors pour tout $i \in |p|$, il existe γ_i et h_i tels que $J_{13}(\gamma_i, p_0, p_i, h_i)$ est vrai, de même que $F(p_0, h_0)$, $I(p_i, h_i)$ et $(C(h_{i-1}, b_{i-1}, p_i, h_i) \vee \Phi(h_{i-1}, b_{i-1}, p_i, h_i))$ quand $i > 0$. De la sorte F, I, C et Φ peuvent être convenablement choisis pourque h_i soit le cumul de l'histoire des calculs ayant conduit de p_0 à p_i selon le schéma suivant :

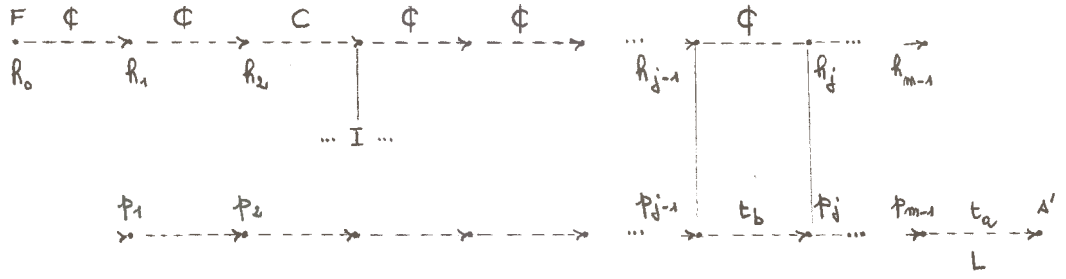


Dans (F'_{13}) nous avons :

- Live $\langle S, A, \Sigma \rangle (H, F, I, C, \Phi)(L) =$

$$(\forall m \in (\omega \setminus 0),\ p' \in \Sigma,\ p \in \Sigma^m \langle S, A \rangle,\ h \in (m \to H),\ a \in A,\ s' \in S .$$
$$[\, p \mapsto p' \wedge F(p_0, h_0) \wedge \forall j \in m . I(p_j, h_j) \wedge \forall j \in (m \setminus 0) . [\exists b \in A . t_b(p_{j-1}, p_j) \wedge$$
$$(\Phi(h_{j-1}, b, p_j, h_j) \vee C(h_{j-1}, b, p_j, h_j))] \wedge t_a(p_{m-1}, s') \wedge$$
$$L(p_{m-1}, h_{m-1}, a, s')\,] \Rightarrow [\, p \notin \Sigma\,])$$

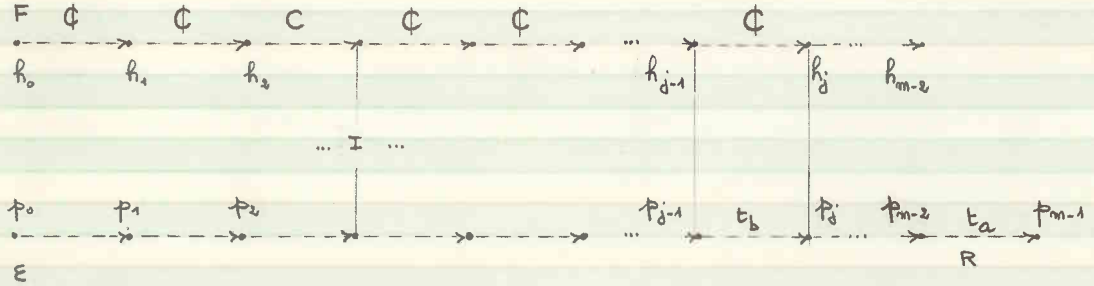
Une formule plus facile à comprendre à l'aide du schéma suivant :



(de sorte que $L(s, h, a, s')$ implique qu'il n'y a pas de trace finie p de Σ qui se termine dans l'état s et n'appartient pas aux traces de $\underset{\sim}{\text{Ret}}_{\mathcal{P}^S}(\langle S, A, \Sigma \rangle)$).

- Next$_{\langle S,A,\Sigma\rangle}(H,F,I,C,\not\subset)(R)$ =

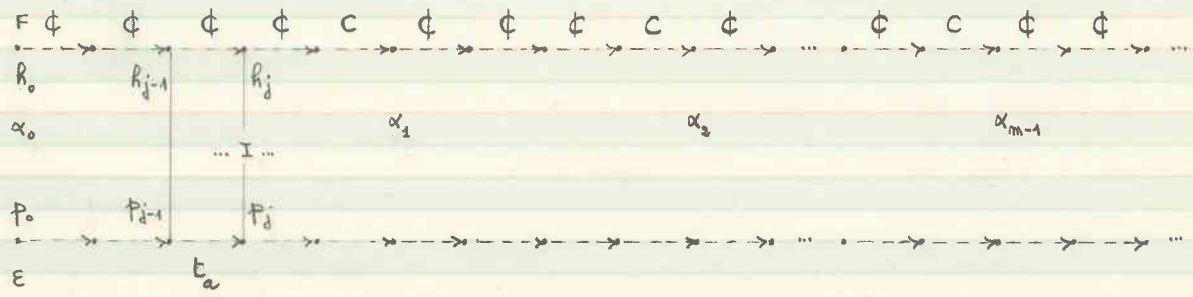
$$(\forall p' \in \Sigma,\ m \in \omega,\ p \in \Sigma^m_{\langle S,A\rangle}.[(p \mapsto p' \wedge m>1) \Rightarrow$$
$$(\forall h \in ((m-1) \to H),\ a \in A.$$
$$[F(p_0,h_0) \wedge \forall j \in (m-1).\ I(p_j,h_j) \wedge \forall j \in ((m-1) \setminus 0).[\exists b \in A.\ t_b(p_{j-1},p_j) \wedge$$
$$(\not\subset(h_{j-1},b,p_j,h_j) \vee C(h_{j-1},b,p_j,h_j))] \wedge t_a(p_{m-2},p_{m-1})]$$
$$\Rightarrow R(p_{m-2},h_{m-2},a,p_{m-1}))])$$



(de sorte que si $R(s,h,a,s')$ est vrai et p est le préfixe d'une trace de Σ se terminant dans l'état s et correspondant à l'histoire h alors $p \xrightarrow{a} s'$ est également préfixe d'une trace de Σ (et pas seulement préfixe d'une trace de la sémantique $F_{\text{lus}}(\langle S,A,\Sigma\rangle)$).

- D-cutset$_{\langle S,A,\Sigma\rangle}(H,F,I,C,\not\subset)$ =

$$[\forall p \in (\Sigma \cap \Sigma^\omega_{\langle S,A\rangle}),\ h \in (\omega \to H).$$
$$\neg[\exists m \in (\omega \setminus 0),\ \alpha \in (m \to \omega).$$
$$(\alpha_0 = 0 \wedge \forall i,j \in m.[(i<j) \Rightarrow (\alpha_i < \alpha_j)]$$
$$\wedge\ (F(p_0,h_0) \wedge \forall j \in \omega.\ I(p_j,h_j))$$
$$\wedge\ (\forall i \in (m \setminus 0).\ \exists a \in A.[t_a(p_{\alpha_i-1},p_{\alpha_i}) \wedge C(h_{\alpha_i-1},a,p_{\alpha_i},h_{\alpha_i})])$$
$$\wedge\ (\forall j \in \omega.[(\forall i \in m.\ j \neq \alpha_i) \Rightarrow (\exists a \in A.(t_a(p_{j-1},p_j) \wedge \not\subset(a_{j-1},a,p_j,h_j)))])]]$$



(Intuitivement, il n'y a pas de trace infinie p et d'histoire h sur H (dont le premier élément est défini par F, les suivants par Φ sauf pour un nombre fini de points de coupure qui sont définis par C) pour lesquelles l'invariant I est toujours vrai. (I n'a pas été incorporé dans F, C et Φ, uniquement par commodité)).

Exemple 5.2.8.2.1-1

Si $\Sigma = \Sigma\langle S, A, t, \varepsilon\rangle$ et s'il existe $K \subseteq S$ tel que cutset$\langle S, A, \Sigma\rangle(K)$ alors en choisissant $A = \{\perp\}$, $F(\underline{s}, \underline{h}) = tt$, $I(s, h) = tt$, $C(h, a, s, h') = tt$, $\Phi(h, a, s', h') = [s' \notin K]$ et $J_3(\gamma, \underline{s}, s) = J_{13}(\gamma, \underline{s}, s, \perp)$, nous obtenons une version de $(\mathcal{F}_3')$.

□

Théorème 5.2.8.2.1-1 (Correction)

$(\mathcal{F}_{13}') \Rightarrow$ (Ψ est fatale sous invariance de Φ pour $\langle S, A, \Sigma\rangle$)

Démonstration

Supposons $(\mathcal{F}_{13}')$, $p \in \Sigma$ et $\forall i \in |p|. \neg\Psi(p_0, p_i)$. Posons $Inv(d) =$ $[\exists \gamma \in (d \to \Gamma), h \in (d \to H). F(p_0, h_0) \wedge \forall j \in d. I(p_j, h_j) \wedge \forall j \in (d \setminus 0). (\exists b \in A. t_b(p_{j-1}, p_j) \wedge (\Phi(h_{j-1}, b, p_j, h_j) \vee C(h_{j-1}, b, p_j, h_j))) \wedge \forall j \in d. J_{13}(\gamma_j, p_0, p_j, h_j) \wedge \forall j \in (d \setminus 0). (\gamma_j < \gamma_{j-1})]$. Nous avons $Inv(|p|)$. Ceci parceque $\varepsilon(p_0)$ étant vrai par définition de ε, alors $F(p_0, h_0) \wedge J_{13}(\gamma_0, p_0, p_0, h_0)$ est vrai d'après $(\mathcal{F}_{13}')$. Supposons, par induction, $Inv(m-1)$ et $m \in |p|$. Nous avons $t_a(p_{m-1}, p_m)$ pour une $a \in A$, donc par définition de Next$\langle S, A, \Sigma\rangle(H, F, I, C, \Phi)(R)$, $R(p_{m-1}, h_{m-1}, a, p_m)$ est vrai et donc d'après $(\mathcal{F}_{13}')$ ou bien ils existent $\gamma_m < \gamma_{m-1}$, $h_m \in H$ tels que $C(h_{m-1}, a, p_m, h_m)$ ou bien $\gamma_m = \gamma_{m-1}$ et il existe $h_m \in H$ tel que $\Phi(h_{m-1}, a, p_m, h_m)$ et dans les deux cas $J_{13}(\gamma_m, p_0, p_m, h_m)$. De nouveau $(\mathcal{F}_{13}')$ et $\neg\Psi(p_0, p_m)$ impliquent $I(p_m, h_m)$ et donc $Inv(m)$.

Si $|p| = m \in \omega$ alors $Inv(m)$ et $(\mathcal{F}_{13}')$ impliquent $\exists a \in A, s' \in S. [t_a(p_{m-1}, s') \wedge L(p_{m-1}, h_{m-1}, a, s')]$ qui d'après la définition de Live$\langle S, A, \Sigma\rangle(H, F, I, C, \Phi)(L)$ est en contradiction avec $p \in \Sigma$.

Si $|p|=\omega$ alors $Inv(\omega)$ et la séquence infinie d'ordinaux γ_i ne peut pas être strictement décroissante de sorte qu'il y a un nombre fini d'endroits $\alpha_i \in \omega$, $i \in m$, $m \in \omega$ où γ_j décroit strictement et où C est vrai. Partout ailleurs $(\mathcal{F}'_{13})$ implique que Φ est vrai, en contradiction avec $\underline{D\text{-}cutset}\langle S,A,\Sigma\rangle(H,F,I,C,\Phi)$.

□

Théorème 5.2.8.2.1∨2 (Complétude sémantique faible)

$(\Psi$ est fatale sous invariance de Φ pour $\langle S,A,\Sigma\rangle) \Rightarrow (\mathcal{F}'_{13})$

Démonstration

Supposons que Ψ est fatale sous invariance de Φ pour $\langle S,A,\Sigma\rangle$ et définissons $H=\Sigma^{<\omega}\langle S,A\rangle$, $F(\underline{s},\underline{h})=[\underline{h}=\langle\underline{s}\rangle]$, $I(s,h)=[\forall i\in|h|.\neg\Psi(h_0,h_i)]$ et $C(h,a,s',h')=\Phi(h,a,s',h')=[h'=h\xrightarrow{a}s']$. Observons que $\forall d\in((\omega+1)\vee 0)$, $p'\in\Sigma$, $p\in\Sigma^d\langle S,A\rangle$. $p\mapsto p'$, $h\in(d\to H)$. $([F(p_0,h_0)\wedge\forall j\in d.\ I(p_j,h_j)\wedge\forall j\in(d\vee 0).[\exists b\in A.t_b(p_{j-1},p_j)\wedge(\Phi(h_{j-1},b,p_j,h_j)\vee C(h_{j-1},b,p_j,h_j))]]\Rightarrow(p=h))$. D'où $\underline{D\text{-}cutset}\langle S,A,\Sigma\rangle(H,F,I,C,\Phi)$ parceque Ψ est fatale par hypothèse. De la même manière $\underline{Live}\langle S,A,\Sigma\rangle(H,F,I,C,\Phi)(L)$ découle de $L(s,h,a,s')=[h\notin\Sigma]$ et $\underline{Next}\langle S,A,\Sigma\rangle(H,F,I,C,\Phi)(R)$ de $R(s,h,a,s')=[\exists p'\in\Sigma.(h\xrightarrow{a}\langle s'\rangle)\mapsto p']$. Alors $(\mathcal{F}'_{13})$ est satisfait par $\Gamma=\mathcal{Q}$ et $J_{13}(\gamma,\underline{s},s,h)=[(\gamma=0\wedge\Psi(\underline{s},s))\vee(\gamma=1\wedge\exists p'\in\Sigma.h\mapsto p')\wedge\exists m\in(\omega\vee 0).(|h|=m\wedge h_0=\underline{s}\wedge h_{m-1}=s\wedge[\forall i\in m.(\Phi(h_0,h_i)\wedge\neg\Psi(h_0,h_i))])]$.

□

Pour être complet, il nous faudrait examiner les 7 cas particuliers où la sémantique n'est pas close parceque seulement l'une ou deux des conditions du théorème 2.6.8∨4 ne sont pas satisfaites. Pour ce faire,

- Si la sémantique est fermée par fusion, nous prenons $R(s,h,a,s')=tt$ et $\underline{Next}\langle S,A,\Sigma\rangle(H,F,I,C,\Phi)(R)=tt$ dans $(\mathcal{F}'_{13})$.
- Si la sémantique est réduite par élimination des traces préfixes strictes, nous prenons $L(s,h,a,s')=tt$ et $\underline{Live}\langle S,A,\Sigma\rangle(H,F,I,C,\Phi)(L)=tt$ dans $(\mathcal{F}'_{13})$.

- Si la sémantique est fermée par limites, nous prenons $\Phi(h,a,s',h') = ff$ et D-cutset$\langle S,A,\Sigma\rangle(H,F,I,C,\Phi) = tt$ dans $(\mathcal{F}_{13})$.

Nous ne donnerons les démonstrations que dans 2 de ces 7 cas particuliers, les 5 autres étant similaires.

5.2.8.2.2 Sémantique non close, fermée par fusion et réduite par élimination des traces préfixes stricts

$$
\begin{array}{l}
(\exists H, F\in(S\times H\to\{tt,ff\}),\ I\in(S\times H\to\{tt,ff\}),\ C\in(H\times A\times S\times H\to\{tt,ff\}),\\
\quad \Phi\in(H\times A\times S\times H\to\{tt,ff\}).\\
\\
\quad \underset{\sim}{\text{Entry}}\langle S,A,\Sigma\rangle(E)\\
\quad \wedge\\
\quad \underset{\sim}{\text{D-cutset}}\langle S,A,\Sigma\rangle(H,F,I,C,\Phi)\\
\quad \wedge\\
\quad (\exists \Gamma\in \underset{\sim}{\text{Ord}},\ J\in(\Gamma\times S\times S\times H\to\{tt,ff\}).\qquad (\mathcal{F}'_{14})\\
\qquad (\forall \underline{s},s\in S,\ \gamma\in\Gamma,\ h\in H.\\
\qquad\quad (\varepsilon(\underline{s})\Rightarrow[\exists \underline{\gamma}\in\Gamma,\ \underline{h}\in H.\ (F(\underline{s},\underline{h})\wedge J(\underline{\gamma},\underline{s},\underline{s},\underline{h}))])\\
\qquad\quad \wedge(J(\gamma,\underline{s},s,h)\Rightarrow \Psi(\underline{s},s)\\
\qquad\qquad\qquad \vee[\Phi(\underline{s},s)\wedge I(s,h)\wedge\exists a\in A, s'\in S.\ t_a(s,s')\wedge\\
\qquad\qquad\qquad\quad \forall a\in A,\ s'\in S.\ (t_a(s,s')\Rightarrow\\
\qquad\qquad\qquad\qquad [\ (\exists\gamma'<\gamma,\ h'\in H.\ C(h,a,s',h')\wedge J(\gamma',\underline{s},s',h'))\\
\qquad\qquad\qquad\qquad\ \vee(\exists h'\in H.\ \Phi(h,a,s',h')\wedge J(\gamma,\underline{s},s',h'))]]]))
\end{array}
$$

Théorème 5.2.8.2.2 ~1 (Correction)

Si $\langle S,A,\Sigma\rangle = \underset{\sim}{\text{Retps}}(\langle S,A,\Sigma\rangle)$ alors

$(\mathcal{F}'_{14}) \Rightarrow$ (Ψ est fatale sous invariance de Φ pour $\langle S,A,\Sigma\rangle$)

Démonstration

Supposons $(\mathcal{F}_{14})$ et $p \in \Sigma$. s'il existe $i \in |p|$ tel que $\Psi(p_0, p_i)$ alors pour un tel plus petit i, il existe $\gamma \in (i \to \Gamma)$ et $h \in (i \to H)$ tels que $\forall j \in i. J_{14}(\gamma_j, p_0, p_j, h_j)$ et donc $\forall j \in i. \Phi(p_0, p_j)$. Il n'est pas possible d'avoir $\forall i \in |p|. \neg\Psi(p_0, p_i)$. Sinon il existerait $\gamma \in (|p| \to \Gamma)$ et $h \in (|p| \to H)$ tels que $E(p_0)$ donc $F(p_0, h_0)$ et $\forall j \in |p|. J_{14}(\gamma_j, p_0, p_j, h_j)$ donc $\forall j \in |p|. I(p_j, h_j)$ et en plus $\forall j \in (|p| \vee 0). \gamma_{j-1} \geq \gamma_j$. Quand p est finie, nous avons $J_{14}(\gamma_{m-1}, p_0, p_{m-1}, h_{m-1})$ où $|p| = m$. Ceci implique $\exists a \in A, s' \in S. t_a(p_{m-1}, s')$ en contradiction avec $\langle S, A, \Sigma\rangle = \underline{\text{Retps}}(\langle S, A, \Sigma\rangle)$. Quand p est infinie (auquel cas la séquence infinie d'ordinaux γ_j ne peut pas être strictement décroissante) il y a un nombre fini d'endroits $\alpha_i \in \omega$, $i \in m$, $m \in \omega$ où γ_j est strictement décroissante et C vrai. Partout ailleurs Φ serait vrai, en contradiction avec $\underline{\text{D-cutset}}\langle S, A, \Sigma\rangle(H, F, I, C, \Phi)$.

□

Théorème 5.2.8.2.2 ν 2

$\langle S, A, \Sigma\rangle = \underline{\text{Efus}}(\langle S, A, \Sigma\rangle)$ et $\langle S, A, \Sigma\rangle = \underline{\text{Retps}}(\langle S, A, \Sigma\rangle)$ alors

(Ψ est fatale sous invariance de Φ pour $\langle S, A, \Sigma\rangle$) $\Rightarrow (\mathcal{F}_{14})$

Démonstration

Supposons que Ψ est fatale sous invariance de Φ pour $\langle S, A, \Sigma\rangle$ et définissons $H = \Sigma^{<\omega}\langle S, A\rangle$, $F(\underline{s}, \underline{h}) = [\underline{h} = \langle \underline{s} \rangle]$, $I(s, h) = [\forall i \in |h|. \neg\Psi(h_0, h_i)]$, $C(h, a, s', h') = \Phi(h, a, s', h') = [h' = h \xrightarrow{a} \langle s' \rangle]$. Si nous pouvons trouver $p \in (\Sigma \wedge \Sigma^{\omega}\langle S, A\rangle)$, $h \in (\omega \to H)$, $m \in (\omega \vee 0)$, $\alpha \in (m \to \omega)$ ne satisfaisant pas la condition $\underline{\text{D-cutset}}\langle S, A, \Sigma\rangle(H, F, I, C, \Phi)$ alors nous aurons $F(p_0, h_0)$ et donc $h_0 = p_0 = p^{<1}$. Supposons par induction que $h_j = p^{<j+1}$. Alors soit $(j+1) \neq \alpha_i$ pour tout $i \in m$ auquel cas $\exists a \in A. t_a(p_j, p_{j+1}) \wedge \Phi(h_j, a, p_{j+1}, h_{j+1})$ implique $h_{j+1} = p^{<j+1} \xrightarrow{a} \langle p_{j+1} \rangle$ et $a = p_j$ d'où $h_{j+1} = p^{<j+2}$. Sinon $\exists i \in m. \alpha_i = (j+1)$ de sorte que $\alpha_i \neq 0$ donc $i \neq 0$ et $(\exists a \in A. t_a(p_j, p_{j+1}) \wedge C(h_j, a, p_{j+1}, h_{j+1}))$ implique $h_{j+1} = p^{<j+2}$. La contradiction avec le fait que Ψ est fatale sous invariance de Φ pour $\langle S, A, \Sigma\rangle$ est maintenant que $\forall j \in \omega. I(p_j, h_j)$ donc $I(p_j, p^{<j+1})$ d'où

$\neg\Psi(p_0,p_j)$. Donc D-cutset $\langle S,A,\Sigma\rangle$ (H,F,I,C,Φ).

Choisissons maintenant $\Gamma=\mathcal{Q}$ et $J_{14}(\gamma,\underline{a},a,h) = [(\gamma=0 \wedge \Psi(\underline{a},a)) \vee (\gamma=1 \wedge \exists p\in\Sigma.\ h\twoheadrightarrow p) \wedge \exists m\in(\omega\vee 0).(|h|=m) \wedge h_0=\underline{a} \wedge h_{m-1}=a \wedge [\forall i\in m.(\Phi(h_0,h_i)\wedge\neg\Psi(h_0,h_i))]]$

Alors, supposant $\langle S,A,\Sigma\rangle = Efus(\langle S,A,\Sigma\rangle)$ et $\langle S,A,\Sigma\rangle = Reltps(\langle S,A,\Sigma\rangle)$, J_{14} satisfait $(\mathcal{F}'_{14})$.

□

Le résultat de complétude ci-dessus est faible dans le sens que l'ensemble de points de coupure choisi pour la preuve de complétude dépend de la propriété Ψ dont nous prouvons la fatalité.

Pour une classe donnée de sémantiques non closes, il est quelquefois possible de trouver des variables auxiliaires et un ensemble de points de coupure dynamique correspondant qui conviennent pour la preuve de toute propriété de fatalité.

Exemple 5.2.8.2.2-2

Pour $\langle S,A,\Sigma\rangle = Wfair\langle A\rangle(\langle S,A,\Sigma\langle S,A,t,\varepsilon\rangle\rangle)$ où $card(A)<\omega$, nous pouvons choisir $H=A$, $F(a)=tt$, $I(a,h) = [\exists a'\in S.\ t_h(a,a')]$, $C(h,a,a',h') = tt$ et $\Phi(h,a,a',h') = [h\neq a \wedge h'=h]$.

Si il existe une trace p faiblement équitable et $h\in(|p|\to A)$ ne satisfaisant pas la condition D-cutset $\langle S,A,\Sigma\rangle(H,A,F,I,C,\Phi)$ alors ou bien $|p|=j\in(\omega\vee 0)$ de sorte que p_{j-1} est un état de blocage, en contradiction avec $I(p_{j-1},h_{j-1})$, ou bien $|p|=\omega$ auquel cas il existe une $a=h_{\alpha_{m-1}}$ qui est toujours prête $(\forall j\geq\alpha_{m-1}.(h_j=a)$ de sorte que $I(p_j,h_j)$ implique $\exists a'\in S.\ t_a(p_j,a'))$ et jamais activée $(\forall j>\alpha_{m-1}.\ \exists b\in A.\ (t_b(p_{j-1},p_j)\wedge\Phi(h_{j-1},b,p_j,h_j))$ d'où $b\neq a$), en contradiction avec l'hypothèse d'équité faible.

En remplaçant H,F,I,C,Φ dans $(\mathcal{F}_{14})$ par leurs définitions respectives ci-dessus, nous obtenons :

$$
\begin{aligned}
&(\exists \Gamma \in \underline{Ord},\ J \in (\Gamma \times S \times S \times A \to \{tt, ff\}).\\
&\quad (\forall \underline{\alpha}, \underline{\beta} \in S,\ \gamma \in \Gamma,\ b \in A.\\
&\qquad (\varepsilon(\underline{\alpha}) \Rightarrow [\exists \underline{\gamma} \in \Gamma,\ \underline{b} \in A.\ J(\underline{\gamma}, \underline{\alpha}, \underline{\alpha}, \underline{b})] \qquad (\mathbb{F}_{15})\\
&\qquad \wedge\\
&\qquad (J(\gamma, \underline{\alpha}, \beta, b) \Rightarrow \Psi(\underline{\alpha}, \beta)\\
&\qquad\qquad \vee [\Phi(\underline{\alpha}, \beta) \wedge \exists \beta' \in S.\ t_b(\beta, \beta') \wedge \forall a \in A, \beta' \in S.\ (t_a(\beta, \beta') \Rightarrow\\
&\qquad\qquad [(\exists \gamma' < \gamma, b' \in A.\ J(\gamma', \underline{\alpha}, \beta', b')) \vee (b \neq a \wedge J(\gamma, \underline{\alpha}, \beta', b))]]]))
\end{aligned}
$$

Une version de $(\mathbb{F}_{15})$ a été proposée par Lehmann-Pnueli-Stavi [81] et leur preuve de complétude sémantique est facile à adapter. Cette preuve est indépendante de Φ et Ψ.

□

5.2.8.2.3 Sémantique (non close) fermée par limites, non fermée par fusion et non réduite par élimination des traces préfixes stricts

$$
\begin{array}{l}
(\exists H,\ F \in (S\times H \to \{tt,ff\}),\ I \in (S\times H \to \{tt,ff\}),\ L,R \in (S\times H\times A\times S \to \{tt,ff\}),\\
\quad C \in (H\times A\times S\times H \to \{tt,ff\}).\\
\\
\quad \underline{Live}\ \langle S,A,\Sigma\rangle(H,F,I,C,ff)(L)\\
\quad \wedge\\
\quad \underline{Next}\ \langle S,A,\Sigma\rangle(H,F,I,C,ff)(R)\\
\quad \wedge\\
\quad (\exists \Gamma \in \underline{Ord},\ J \in (\Gamma\times S\times S\times H \to \{tt,ff\}).\\
\qquad (\forall \underline{s},s \in S,\ \gamma \in \Gamma,\ h \in H.\\
\qquad\quad (\mathcal{E}(\underline{s}) \Rightarrow [\exists \underline{\gamma} \in \Gamma,\ \underline{h} \in H.(F(\underline{s},\underline{h}) \wedge J(\underline{\gamma},\underline{s},\underline{s},\underline{h}))])\\
\qquad\quad \wedge\\
\qquad\quad (J(\gamma,\underline{s},s,h) \Rightarrow \Psi(\underline{s},s)\\
\qquad\qquad\qquad \vee\\
\qquad\qquad\qquad [\ \Phi(\underline{s},s) \wedge I(s,h)\\
\qquad\qquad\qquad\quad \wedge\\
\qquad\qquad\qquad\quad \exists q \in A,\ s' \in S.\ [t_q(s,s') \wedge L(s,h,q,s')]\\
\qquad\qquad\qquad\quad \wedge\\
\qquad\qquad\qquad\quad \forall q \in A,\ s' \in S.\ ([t_q(s,s') \wedge R(s,h,q,s')] \Rightarrow\\
\qquad\qquad\qquad\qquad\qquad [\exists \gamma' < \gamma,\ h' \in H.(C(h,q,s',h') \wedge J(\gamma',\underline{s},s',h'))]]))))
\end{array}
\qquad (\mathcal{F}'_{16})
$$

Théorème 5.2.8.2.3 v1 (Correction)

$(\mathcal{F}'_{16}) \Rightarrow$ (Ψ est fatale sous invariance de Φ pour $\langle S,A,\Sigma\rangle$)

La démonstration est tout à fait identique au cas général 5.2.8.2.1 v1.

Théorème 5.2.8.2.3 v2 (complétude sémantique)

Si $\langle S,A,\Sigma\rangle = \underline{Flim}(\langle S,A,\Sigma\rangle)$ alors

(Ψ est fatale sous invariance de Φ pour $\langle S,A,\Sigma\rangle$) $\Rightarrow (\mathcal{F}'_{16})$

Démonstration

Supposons que Ψ est fatale sous invariance de Φ pour $\langle S,A,\Sigma\rangle$. Posons :

$H = \Sigma^{<\omega}\langle S,A\rangle$

$F(\underline{s},\underline{h}) = [\underline{h} = \langle \underline{s}\rangle]$

$I(s,h) = [\forall i \in |h|.\ \neg\Psi(h_0,h_i)]$

$C(h,a,s',h') = [h' = h \xrightarrow{a} \langle s'\rangle]$

$L(s,h,a,s') = [h \notin \Sigma]$

$R(s,h,a,s') = [\exists p' \in \Sigma.\ (h \xrightarrow{a} \langle s'\rangle) \mapsto p']$

de sorte que les conditions Live$\langle S,A,\Sigma\rangle(H,F,I,C,\Phi)(L)$ et Next$\langle S,A,\Sigma\rangle(H,F,I,C,\Phi)(R)$ (où $\Phi(h,a,s',h') = \mathit{ff}$) sont satisfaites.

Définissons $h \prec h'$ si et seulement si $[\exists p \in \Sigma.\ (h' \mapsto h \mapsto p \wedge h \neq h' \wedge \forall i \in |h|.\ \neg\Psi(p_0,p_i))]$. Soit une suite infinie décroissante $h^0 \succ h^1 \succ h^2 \succ \cdots$. Comme la sémantique est fermée par limites, la trace limite infinie $p \in \Sigma^\omega\langle S,A\rangle$ telle que $\forall i \in \omega.\ p_i = (h^{i+1})_i$ et $p_{i+1} = (h^{i+2}_{\longrightarrow})_i$, appartient à Σ avec $\forall i \in p.\ \neg\Psi(p_0,p_i)$ en contradiction avec l'hypothèse que Ψ est fatale. La relation $\prec$ étant bien-fondée, nous définissons :

$$e(h) = \cap\{\alpha \in \mathrm{Ord} : \forall h' \in H.\ [h' \prec h \Rightarrow e(h') < \alpha]\}$$

et choisissons :

$$\Gamma = \sup^+\{e(h)+1 : h \in H\}$$

$$J_{16}(\gamma,\underline{s},s,h) = \Big([\gamma = 0 \wedge \Psi(\underline{s},s)]$$
$$\vee\ [\gamma > 0 \wedge \gamma = e(h) \wedge \exists p \in \Sigma.\ h \mapsto p \wedge h_0 = \underline{s} \wedge \exists m \in \omega.\ |h| = m \wedge h_m = s \wedge$$
$$\forall i \leq m.\ (\Phi(h_0,h_i) \wedge \neg\Psi(h_0,h_i))]\Big)$$

Alors J_{16} satisfait $(\mathcal{F}'_{16})$ car $\forall h',h \in H.\ [(h' \prec h) \Rightarrow (e(h') < e(h))]$.

□

5.2.8.3 Equivalence forte des principes d'induction ($\mathcal{F}_6'^{\#}$) et ($\mathcal{F}_{13}$)

Théorème 5.2.8.3 N 1

Toute preuve de fatalité de Ψ pour une sémantique quelconque $\langle S, A, \Sigma \rangle$ utilisant le principe d'induction ($\mathcal{F}_6'^{\#}$) peut se reécrire en une preuve de fatalité utilisant le principe d'induction ($\mathcal{F}_{13}$), et réciproquement.

Démonstration

- Ayant trouvé $S^{\#}, A^{\#}, t^{\#}, \varepsilon^{\#}, f_0^{\#}, \Gamma^{\#}$ et $J^{\#}$ satisfaisant les conditions de ($\mathcal{F}_6'^{\#}$), nous pouvons toujours reécrire cette preuve de fatalité en une preuve de fatalité utilisant le principe d'induction ($\mathcal{F}_{13}$), en posant :

$H = (S^{\#} \times S^{\#})$

$F(\underline{a}, \langle \underline{a}^{\#}, a^{\#} \rangle) = [a^{\#} = \underline{a}^{\#} \wedge \underline{a} = f_0^{\#}(\underline{a}^{\#}) \wedge \varepsilon^{\#}(\underline{a}^{\#})]$

$L(a, \langle \underline{a}^{\#}, a^{\#} \rangle, a, a') = [a = f_a^{\#}(a^{\#}) \wedge \exists a^{\#\prime} \in S^{\#}. t_a^{\#}(a^{\#}, a^{\#\prime})]$

$R(a, \langle \underline{a}^{\#}, a^{\#} \rangle, a, a') = [a = f_a^{\#}(a^{\#}) \wedge \exists a^{\#\prime} \in S^{\#}. (t_a^{\#}(a^{\#}, a^{\#\prime}) \wedge a' = f_0^{\#}(a^{\#\prime}))]$

$I(a, \langle \underline{a}^{\#}, a^{\#} \rangle) = [a = f_a^{\#}(a^{\#})]$

$C = tt$

$\Phi = ff$

$\Gamma = \Gamma^{\#}$

$J(\gamma, \underline{a}, a, \langle \underline{a}^{\#}, a^{\#} \rangle) = [\underline{a} = f_0^{\#}(\underline{a}^{\#}) \wedge a = f_a^{\#}(a^{\#}) \wedge \varepsilon(\underline{a}^{\#}) \wedge J^{\#}(\gamma, \underline{a}^{\#}, a^{\#})]$

En effet, nous avons bien :

- Live $\langle S,A,\Sigma\rangle(H,F,I,C,\Phi)(L)$

car sinon on aurait $\exists m\in(\omega\setminus 0)$, $p'\in\Sigma$, $p\in\Sigma^m\langle S,A\rangle$, $h\in(m\to S^\#\times S^\#)$. $[\ p\mapsto p'\wedge \varepsilon^\#(h_0(0))\wedge h_0(0)=h_0(1)\wedge p_0=f_a^\#(h_0(1))\wedge \forall j\in m.(p_j=f_a^\#(h_j(1)))\wedge \forall j\in(m\setminus 0).\exists b\in A.\ t_b(p_{j-1},p_j)\wedge t_a(p_{m-1},a')\wedge p_{m-1}=f_a^\#(h_{m-1}(1))\wedge \exists a^\#\in S^\#.\ t_a^\#(h_{m-1}(1),a^\#)\wedge p\in\Sigma\ ]$, donc $h_0(1)\xrightarrow{p_0}h_1(1)\dots\xrightarrow{p_{m-2}}h_{m-1}(1)$ serait à la fois une trace de $\Sigma\langle S^\#,A^\#,t^\#,\varepsilon^\#\rangle$ car $f_a^\#(h_0(1)\xrightarrow{p_0}h_1(1)\dots\xrightarrow{p_{m-2}}h_{m-1}(1))=p$ et préfixe strict d'une trace de $\Sigma\langle S^\#,A^\#,t^\#,\varepsilon^\#\rangle$ (car $\exists a^\#\in S^\#.\ t_a^\#(h_{m-1}(1),a^\#)$) en contradiction avec le fait que $\langle S^\#,A^\#,\Sigma\langle S^\#,A^\#,t^\#,\varepsilon^\#\rangle\rangle$ est close.

- Next $\langle S,A,\Sigma\rangle(H,F,I,C,\Phi)(R)$

car $\forall p'\in\Sigma, m\in\omega, p\in\Sigma^m\langle S,A\rangle.\ [(p\mapsto p'\wedge m>1)\wedge(\forall h\in((m-1)\to S^\#\times S^\#),a\in A.(\varepsilon^\#(h_0(0))\wedge h_0(0)=h_0(1)\wedge p_0=f_a^\#(h_0(1))\wedge\forall j\in(m-1).(p_j=f_a^\#(h_j(1)))\wedge\forall j\in((m-1)\setminus 0).\exists b\in A.\ t_b(p_{j-1},p_j)\wedge t_a(p_{m-2},p_{m-1}))]$ implique que $h_0(1)\xrightarrow{p_0}h_1(1)\dots\xrightarrow{p_{m-3}}h_{m-2}(1)$ est préfixe d'une trace $p^{\#\prime}\in\Sigma\langle S^\#,A^\#,t^\#,\varepsilon^\#\rangle$ telle que $p'=f_a^\#(p^{\#\prime})$. Comme $m-1\le|p|=|p^*|$ alors $\exists a^\#\in S^\#.\ t_a^\#(h_{m-2}(1),a^\#)\wedge p_{m-1}=f_a^\#(a^\#)$ donc $R(p_{m-2},h_{m-2},a,p_{m-1})$ est vrai.

- D-cutset $\langle S,A,\Sigma\rangle(H,F,I,C,\Phi)$

car sinon nous aurions une trace infinie $h_0(1)\xrightarrow{p_0}h_1(1)\dots\xrightarrow{p_{i-1}}h_i(1)\dots$ de $\Sigma\langle S^\#,A^\#,t^\#,\varepsilon^\#\rangle$ donc par $(\mathcal{F}_6^\#)$, une chaine infinie strictement décroissante d'ordinaux $\gamma_0,\dots,\gamma_i,\dots$.

$\forall \underline{s},s\in S,\ \gamma\in\Gamma$, si $(\mathcal{F}_6^\#)$,

- Si $\varepsilon(\underline{s})$ est vrai, $\exists \underline{s}^\#\in S^\#.(\underline{s}=f_s^\#(\underline{s}^\#)\wedge\varepsilon^\#(\underline{s}^\#))$ donc $\exists \underline{s}^\#\in S^\#.(\underline{s}=f_s^\#(\underline{s}^\#)\wedge\varepsilon^\#(\underline{s}^\#)\wedge\exists\underline{\gamma}\in\Gamma.\ J^\#(\underline{\gamma},\underline{s}^\#,\underline{s}^\#))$ et donc $\exists \underline{s}^\#\in S^\#.(F(\underline{s},\langle\underline{s}^\#,\underline{s}^\#\rangle)\wedge\exists\underline{\gamma}\in\Gamma.\ J(\underline{\gamma},\underline{s},\underline{s},\langle\underline{s}^\#,\underline{s}^\#\rangle)\wedge\underline{s}=f_s^\#(\underline{s}^\#))$, soit $\exists\underline{\gamma}\in\Gamma.\ \underline{h}=\langle\underline{s}^\#,\underline{s}^\#\rangle\in H.\ (F(\underline{s},\underline{h})\wedge J(\underline{\gamma},\underline{s},\underline{s},\underline{h}))$.

- si $J(\gamma,\underline{s},s,\langle\underline{s}^\#,s^\#\rangle)$ alors par définition $[\underline{s}=f_s^\#(\underline{s}^\#)\wedge s=f_s^\#(s^\#)\wedge\varepsilon^\#(\underline{s}^\#)\wedge J^\#(\gamma,\underline{s}^\#,s^\#)]$. Mais $J^\#(\gamma,\underline{s}^\#,s^\#)$ implique soit $(\varepsilon^\#(\underline{s}^\#)\Rightarrow\Psi(f_s^\#(\underline{s}^\#),f_s^\#(s^\#))$ donc $\Psi(\underline{s},s)$, soit $\Phi(f_s^\#(\underline{s}^\#),f_s^\#(s^\#))\wedge\exists a^\#\in S^\#,a\in A^\#.\ t_a^\#(s^\#,a^\#)$ et donc $[\Phi(\underline{s},s)\wedge I(\underline{s},\langle\underline{s}^\#,s^\#\rangle)\wedge\exists s'=f_s^\#(s^{\#\prime})\in S,a\in A.(t_a(s,s')\wedge L(s,\langle\underline{s}^\#,s^\#\rangle,a,s'))]$. Maintenant $[J(\gamma,\underline{s},s,\langle\underline{s}^\#,s^\#\rangle)\wedge t_a(s,s')\wedge R(s,\langle\underline{s}^\#,s^\#\rangle,a,s')]$ implique $[\underline{s}=f_s^\#(\underline{s}^\#)\wedge s=f_s^\#(s^\#)\wedge s'=f_s^\#(s^{\#\prime})\wedge\varepsilon^\#(\underline{s}^\#)\wedge J^\#(\gamma,\underline{s}^\#,s^\#)\wedge t_a^\#(s^\#,s^{\#\prime})]$ donc $[\underline{s}=f_s^\#(\underline{s}^\#)\wedge s'=f_s^\#(s^{\#\prime})\wedge\varepsilon^\#(\underline{s}^\#)\wedge\exists\gamma'<\gamma.\ J^\#(\gamma',\underline{s}^\#,s^{\#\prime})]$ c'est-à-dire $\exists\gamma'<\gamma,\ h'=\langle\underline{s}^\#,s^{\#\prime}\rangle\in H.\ J(\gamma',\underline{s},s',h')$.

- Réciproquement, ayant trouvé $H, F, I, L, R, C, \Phi, \Gamma$ et J satisfaisant les conditions de $(\mathcal{F}_{13})$, nous pouvons toujours réécrire cette preuve de fatalité en une preuve de fatalité utilisant le principe d'induction $(\mathcal{F}_6^{\#})$ de la manière suivante :

Etant donnée $p \in \Sigma$, nous construisons $\alpha \in (|p| \to \omega)$ et $p^{\#} \in \Sigma < (\omega \times \Gamma) \times S \times H, A >$ comme suit :

$\Gamma \times \omega$ bien-ordonné par l'ordre lexicographique gauche $<\gamma, n> \prec <\gamma', n'>$ si et seulement si $(\gamma < \gamma') \vee (\gamma = \gamma' \wedge n < n')$ est isomorphe à un ordinal $\Gamma^{\#} = (\omega \times \Gamma)$ par l'isomorphisme d'ordre $\underline{z}(<\gamma, n>) = (\omega \times \gamma) + n$ dont l'inverse $\underline{\gamma} \in (\Gamma^{\#} \to \Gamma)$, $\underline{n} \in (\Gamma^{\#} \to \omega)$ est tel que $\gamma = \underline{z}(<\underline{\gamma}(\gamma), \underline{n}(\gamma)>)$ et $\underline{z}(<0,0>) = 0$.

Comme $\varepsilon(p_0) \Rightarrow [\exists \gamma_0 \in \Gamma, h_0 \in H. (F(p_0, h_0) \wedge J(\gamma_0, p_0, p_0, h_0))]$, nous choisissons $\alpha_0 = 0$. Si $\neg\Psi(p_0, p_0)$ alors $I(p_0, h_0)$, si $t_{\Phi_0}(p_0, p_1)$ alors $R(p_0, h_0, \Phi_0, p_1)$ est vrai. D'après $(\mathcal{F}_{13})$ $J(\gamma_0, p_0, p_0, h_0)$ implique, ou bien $\exists \gamma_1 < \gamma_0, h_1 \in H. (C(h_0, \Phi_0, p_1, h_1) \wedge J(\gamma_1, p_0, p_1, h_1))$ auquel cas nous choisissons $\alpha_1 = 1$, $p_0^{\#} = <\underline{z}(<\gamma_{\alpha_0}, 0>), p_{\alpha_0}, h_{\alpha_0}>$, $\Phi_0^{\#} = \Phi_0$, ou bien $\exists h_1 \in H. (\Phi(h_0, \Phi_0, p_1, h_1) \wedge J(\gamma_0, p_0, p_1, h_1))$. Alors à partir de $J(\gamma_0, p_0, p_1, h_1)$ nous appliquons $(\mathcal{F}_{13})$ jusqu'à atteindre le plus petit $0 < j \in |p|$ s'il existe tel que $[\exists \gamma_j < \gamma_0, h_j \in H. (C(h_{j-1}, \Phi_{j-1}, p_j, h_j) \wedge J(\gamma_j, p_0, p_j, h_j)) \wedge \forall k \in \omega. [0 < k < j \Rightarrow \exists h_k \in H. (I(p_k, h_k) \wedge \Phi(h_{k-1}, \Phi_{k-1}, p_k, h_k) \wedge J(\gamma_0, p_0, p_k, h_k)))]]$ et nous choisissons $\alpha_1 = j$ $(\in (|p| \vee 0))$, $\forall k \in \omega. [\alpha_0 \leq k < \alpha_1 \Rightarrow (\Phi_k^{\#} = \Phi_k \wedge p_k^{\#} = <\underline{z}(<\gamma_{\alpha_0}, (\alpha_1 - 1) - k>), p_k, h_k>)]$.

Ayant construit α_i pour $i = 0, \ldots, m-1$, $m > 0$, et $p_i^{\#}$, $\Phi_i^{\#}$ pour $i = 0, \ldots, (\alpha_{m-1} - 1)$ avec $\alpha_{m-1} \in (|p| \vee 0)$, et tels que $[F(p_0, h_0) \wedge \forall i \in \alpha_{m-1}. I(p_i, h_i) \wedge \forall i \in (m \vee 0). [(C(h_{\alpha_{i-1}}, \Phi_{\alpha_{i-1}}, p_{\alpha_i}, h_{\alpha_i}) \wedge t_{\Phi_{\alpha_{i-1}}}(p_{\alpha_{i-1}}, p_{\alpha_i})) \wedge \forall k \in \omega. [\alpha_{i-1} < k < \alpha_i \Rightarrow (\Phi(h_{k-1}, \Phi_{k-1}, p_k, h_k) \wedge t_{\Phi_{k-1}}(p_{k-1}, p_k))]] \wedge J(\gamma_{\alpha_{m-1}}, p_0, p_{\alpha_{m-1}}, h_{\alpha_{m-1}})]$. $J(\gamma_{\alpha_{m-1}}, p_0, p_{\alpha_{m-1}}, h_{\alpha_{m-1}})$ implique $I(p_{\alpha_{m-1}}, h_{\alpha_{m-1}})$ si $\neg\Psi(p_0, p_{\alpha_{m-1}})$. De plus si $t_{\Phi_{\alpha_{m-1}}}(p_{\alpha_{m-1}}, p_{\alpha_{m-1}+1})$ alors $R(p_{\alpha_{m-1}}, h_{\alpha_{m-1}}, \Phi_{\alpha_{m-1}}, p_{\alpha_{m-1}+1})$. Nous appliquons $(\mathcal{F}_{13})$ jusqu'à atteindre s'il existe le plus petit j tel que $\alpha_{m-1} < j \in |p| \wedge \exists \gamma_j < \gamma_{\alpha_{m-1}}, h_j \in H. (C(h_{j-1}, \Phi_{j-1}, p_j, h_j) \wedge J(\gamma_j, p_0, p_j, h_j)) \wedge \forall k \in \omega. (\alpha_{m-1} < k < j \Rightarrow \exists h_k \in H. (I(p_k, h_k) \wedge \Phi(h_{k-1}, \Phi_{k-1}, p_k, h_k) \wedge J(\gamma_{\alpha_{m-1}}, p_0, p_k, h_k)))$, nous choisissons alors $\alpha_m = j$ $(\in (|p| \vee 0))$, $\forall k \in \omega. (\alpha_{m-1} \leq k < \alpha_m \Rightarrow (\Phi_k^{\#} = \Phi_k \wedge p_k^{\#} = <\underline{z}(<\gamma_{\alpha_{m-1}}, (\alpha_m - 1) - k>), p_k, h_k>))$. Nous avons ainsi construit α_i pour $i = 0, \ldots, m$, et $\Phi_i^{\#}$, $p_i^{\#}$ pour $i = 0, \ldots, (\alpha_m - 1)$ avec $\alpha_m \in (|p| \vee 0)$.

Si un tel j n'existe pas, nous terminons la construction comme suit : il existe (c'est évident si p est finie sinon d'après D-cutset $\langle S,A,\Sigma\rangle(H,F,I,C,\Phi)$) un plus petit $\underline{j}\in|p|$ tel que $\forall k\in\omega.\,[\alpha_{m-1}<k<\underline{j} \Rightarrow \exists h_k\in H.\,(I(p_k,h_k)\wedge\Phi(h_{k-1},p_{k-1},p_k,h_k)\wedge J(\gamma_{\alpha_{m-1}},p_0,p_k,h_k))]\wedge\neg I(p_{\underline{j}},h_{\underline{j}})$, donc $\Psi(p_0,p_{\underline{j}})$. Nous faisons le même choix que ci-dessus en remplaçant j par $\underline{j}$ et $\forall k\in|p|.\,(k\geq\underline{j}\Rightarrow(p^{\#}_k=p_k\wedge p^{\#}_k=\langle\underline{\iota}(0,0),p_k,h_k\rangle)$.

Si $\Psi(p_0,p_{\alpha_{m-1}})$ alors la construction se termine comme ci-dessus (en posant $\underline{j}=\alpha_{m-1}$).

Posons maintenant :

$\Sigma^{\#} = \{p^{\#} : p\in\Sigma\}$

$S^{\#} = S\times 2^{\Sigma^{\#}}\times\omega$

$A^{\#} = A$

$\Gamma^{\#} = \omega\times\Gamma$

$\varepsilon^{\#}(\langle\underline{a},\underline{I},\underline{m}\rangle) = [\underline{I}=\{p^{\#}\in\Sigma^{\#} : p^{\#}_{\underline{m}}(1)=\underline{a}\}\neq\phi\ \wedge\ \underline{m}=0]$

$t^{\#}_a(\langle a,I,m\rangle,\langle a',I',m'\rangle) = [m'=m+1\ \wedge\ I'=\{p^{\#}\in I : p^{\#}_m(1)=a\ \wedge\ p^{\#}_m=a\ \wedge\ p^{\#}_{m'}(1)=a'\}\neq\phi]$

$f^{\#}_a(\langle a,I,m\rangle) = a$

$J^{\#}(\gamma^{\#},\langle\underline{a},\underline{I},\underline{m}\rangle,\langle a,I,m\rangle) = [\varepsilon^{\#}(\langle\underline{a},\underline{I},\underline{m}\rangle)\Rightarrow I=\{p^{\#}\in\Sigma^{\#} : m\in|p^{\#}|\wedge p^{\#}_0(1)=\underline{a}\wedge p^{\#}_m(1)=a\wedge p^{\#}_m(0)=\gamma^{\#}\}\neq\phi]$

alors

Par construction de $S^{\#}$, $A^{\#}$, $\Sigma^{\#}$ et $f^{\#}_a$ nous avons $\langle S,A,\Sigma\rangle=\Phi^{\#}_A(\langle S^{\#},A^{\#},\Sigma^{\#}\rangle)$ et $\Sigma^{\#}=\Sigma\langle S^{\#},A^{\#},t^{\#},\varepsilon^{\#}\rangle$.

- $\forall\langle\underline{a},\underline{I},\underline{m}\rangle\in S^{\#}.\,[\varepsilon^{\#}(\langle\underline{a},\underline{I},\underline{m}\rangle)\Rightarrow(I=\{p^{\#}\in\Sigma^{\#} : p^{\#}_{\underline{m}}(1)=\underline{a}\}\neq\phi\wedge\underline{m}=0)]$ donc $\forall\langle\underline{a},\underline{I},\underline{m}\rangle\in S^{\#}.\,\exists\underline{\gamma}^{\#}=p^{\#}_{\underline{m}}(0)\in\Gamma^{\#}.\,[\varepsilon^{\#}(\langle\underline{a},\underline{I},\underline{m}\rangle)\Rightarrow I=\{p^{\#}\in\Sigma^{\#} : \underline{m}\in|p^{\#}|\wedge p^{\#}_{\underline{m}}(0)=\gamma^{\#}\wedge p^{\#}_{\underline{m}}(1)=\underline{a}\}\neq\phi]$ c'est-à-dire $\forall\langle\underline{a},\underline{I},\underline{m}\rangle\in S^{\#}.\,\exists\gamma^{\#}\in\Gamma^{\#}.\,J^{\#}(\gamma^{\#},\langle\underline{a},\underline{I},\underline{m}\rangle,\langle\underline{a},\underline{I},\underline{m}\rangle)$.

- Si $J^{\#}(\gamma^{\#},\langle\underline{a},\underline{I},\underline{m}\rangle,\langle a,I,m\rangle)$ alors $[\varepsilon^{\#}(\langle\underline{a},\underline{I},\underline{m}\rangle)\Rightarrow I=\{p^{\#}\in\Sigma^{\#} : m\in|p^{\#}|\wedge p^{\#}_0(1)=\underline{a}\wedge p^{\#}_m(1)=a\wedge p^{\#}_m(0)=\gamma^{\#}\}\neq\phi]$. Si $\gamma^{\#}=0$ alors par construction des $p^{\#}$, $\varepsilon^{\#}(\langle\underline{a},\underline{I},\underline{m}\rangle)\Rightarrow\Psi(f^{\#}_{\underline{a}}(\langle\underline{a},\underline{I},\underline{m}\rangle),f^{\#}_a(\langle a,I,m\rangle))$. Sinon $\gamma^{\#}\neq 0$ et $\exists a'\in S,\,a\in A.\,t_a(a,a')$ donc $\{p^{\#}\in I : m+1\in|p^{\#}|\}\neq\phi$ finalement $\Phi(f^{\#}_{\underline{a}}(\langle\underline{a},\underline{I},\underline{m}\rangle),f^{\#}_a(\langle a,I,m\rangle))\wedge\exists\langle a',I',m'\rangle\in S^{\#},a\in A^{\#}.\,t^{\#}_a(\langle a,I,m\rangle,\langle a',I',m'\rangle)$. Maintenant $[J^{\#}(\gamma,\langle\underline{a},\underline{I},\underline{m}\rangle,\langle a,I,m\rangle)\wedge t^{\#}_a(\langle a,I,m\rangle,\langle a',I',m'\rangle)]\Rightarrow[\varepsilon^{\#}(\langle\underline{a},\underline{I},\underline{m}\rangle)\Rightarrow$

$[\varepsilon^{\#}(<\underline{\Delta}, \underline{\Gamma}, \underline{m}>) \Rightarrow T' = \{p^{\#} \in \Sigma^{\#} : m' \in |p^{\#}| \wedge p^{\#}_{0}(1) = \underline{\Delta} \wedge p^{\#}_{m'}(1) = \Delta'\} \neq \phi]$ donc par construction des traces $p^{\#}$, $\exists \gamma^{\#\prime} < \gamma^{\#}. [\varepsilon^{\#}(<\underline{\Delta}, \underline{\Gamma}, \underline{m}>) \Rightarrow T' = \{p^{\#} \in \Sigma^{\#} : m' \in |p^{\#}| \wedge p^{\#}_{0}(1) = \underline{\Delta} \wedge p^{\#}_{m'}(1) = \Delta' \wedge \gamma^{\#\prime} = p^{\#}_{m'}(0)\} \neq \phi]$ donc $\exists \gamma^{\#\prime} < \gamma^{\#}. J^{\#}(\gamma^{\#\prime}, <\underline{\Delta}, \underline{\Gamma}, \underline{m}>, <\Delta', T', m'>)$.

□

5.3 PRINCIPES D'INDUCTION "A LA BURSTALL"

Nous formalisons la méthode des assertions intermittentes de Burstall [74] initialement conçue pour montrer la correction totale de programmes séquentiels. Nous la généralisons pour démontrer les propriétés de fatalité de programmes non-déterministes et parallèles.

Dans 5.3.1, nous dérivons à partir des exemples de Burstall [74] et Manna-Waldinger [78], un principe d'induction de base qui est une formulation très concise de la méthode des assertions intermittentes de Burstall.
Nous démontrons que la méthode est correcte. Utilisant l'induction transfinie (plutôt que finie) pour traiter le non-déterminisme infini, nous démontrons qu'elle est sémantiquement complète sous une condition suffisante (mais non nécessaire) sur les traces d'exécution et les propriétés de fatalité. Cette condition est vérifiée en particulier quand nous considérons la correction totale de programmes comme dans Burstall [74]. Elle est aussi vérifiée pour les propriétés de fatalité unaires qui ne dépendent que des états finaux (une restriction considérée par Pnueli [77], Apt-Delporte [83], Manna-Pnueli [83]).

Lorsqu'on considère des propriétés de fatalité unaires, les relations entre les valeurs initiales et finales des variables des programmes ne peuvent être exprimées qu'en affectant les valeurs initiales à des variables auxiliaires introduites dans les états. L'utilisation de variables auxiliaires a le désavantage que le programme doit être transformé. Plus important, est le fait que l'utilisation de variables auxiliaires est en un sens très souple : on peut relier des états intermédiaires quelconques lors d'un calcul et même mémoriser tout le calcul. Une telle liberté d'utilisation de variables auxiliaires n'est pas dans l'esprit de Burstall [74] et Manna-Waldinger [78] où les lemmes sont toujours de la forme "if sometime $\Phi(X_1, \ldots, X_m) \wedge X_1 = x_1 \wedge \ldots \wedge X_m = x_m$ at ℓ then sometime $\Psi(x_1, \ldots, x_m, X_1, \ldots, X_m)$ at ℓ'"

(où $X_1, \ldots, X_m$ sont les variables du programme et $x_1, \ldots, x_m$ leurs valeurs symboliques respectives au point ℓ du programme). Ceci s'exprime dans notre principe d'induction de base par l'utilisation de propriétés de fatalité binaires (mieux qu'en imposant des restrictions adéquates sur l'utilisation des variables auxiliaires qui dépendraient de la syntaxe des programmes). Cependant, nous faisons la conjecture que même pour les programmes déterministes, il existe des propriétés de fatalité pour lesquelles l'utilisation d'assertions binaires n'est pas sémantiquement complète.

Cette conjecture nous conduit, dans 5.3.2, à généraliser la méthode des assertions intermittentes de Burstall en utilisant l'induction transfinie (pour traiter le non-déterminisme infini) et des assertions intermittentes ternaires (permettant ainsi des lemmes d'une forme plus générale "if sometime $\Phi(\underline{x}_1, \ldots, \underline{x}_m, X_1, \ldots, X_m) \wedge X_1 = x_1 \wedge \ldots \wedge X_m = x_m$ at ℓ then sometime $\Psi(\underline{x}_1, \ldots, \underline{x}_m, x_1, \ldots, x_m, X_1, \ldots, X_m)$ at ℓ'" où $\underline{x}_1, \ldots, \underline{x}_m$ (respectivement $x_1, \ldots, x_m$) dénotent les valeurs des variables du programme au point d'entrée (respectivement au point ℓ)). Nous démontrons que ce principe d'induction généralisé est correct et sémantiquement complet.

Nous dérivons, dans 5.3.3, une série de principes d'induction qui sont des généralisations successives du principe d'induction ci-dessus. Ceci élargit le champ d'application de la méthode (par exemple lorsqu'on utilise des ensembles bien-ordonnés infinis d'assertions intermittentes (auxquels on peut donner des représentations finies au moyen de variables auxiliaires de terminaison), la méthode de Burstall peut être étendue de façon à incorporer la méthode de Floyd[67]). De plus, la considération de formalisations de plus en plus abstraites devraient permettre une meilleure compréhension de la méthode de Burstall (par exemple nous montrons que l'"évaluation symbolique" et l'"induction sur les données" peuvent être comprises d'une manière unifiée et réduites à une induction sur les calculs). Ces

généralisations successives introduisent plus de souplesse dans l'écriture des preuves mais pas de puissance supplémentaire puisque nous démontrons que tous les principes de preuve considérés sont corrects et sémantiquement complets donc équivalents.

Le principe d'induction "à la Floyd" comporte une induction le long des traces d'exécution tandis que le principe d'induction "à la Burstall" comporte la combinaison d'une induction le long (de parties) de traces d'exécution (en relation avec l'"évaluation symbolique" de Burstall) et une récursivité (en relation avec l'"induction sur les données" de Burstall). Ainsi le principe d'induction "à la Floyd" corspond au cas particulier du principe d'induction "à la Burstall" où la récursivité n'est pas utilisée.

L'argument de complétude consiste à démontrer que les preuves "à la Floyd" peuvent être reformulées en des preuves "à la Burstall" (i.e. l'induction sur les calculs peut être réduite à une induction sur les données). Cependant, cet argument n'est pas pleinement satisfaisant parceque le style des preuves permises est fixé. les utilisateurs de la méthode de Burstall ont besoin d'un résultat de complétude plus fort puisqu'ils veulent savoir si les lemmes qu'ils vont utiliser dans leurs preuves peuvent toujours être choisis librement. Une réponse affirmative est donnée dans 5.3.4 (avec la condition nécessaire et suffisante que chaque lemme concerne une propriété qui est fatale pour le programme mais aussi relativement aux autres lemmes qui sont utilisés dans sa preuve).

5.3.1 LE PRINCIPE D'INDUCTION DE BASE SOUS-JACENT A LA METHODE DES ASSERTIONS INTERMITTENTES DE BURSTALL

Dans ce paragraphe, nous donnons un principe d'induction de base qui est une formulation très concise de la méthode de Burstall. Dans le paragraphe suivant, nous allons supprimer un certain nombre de restrictions (dont nous croyons qu'elles engendrent des problèmes d'incomplétude) et dériver des principes d'induction plus généraux et plus abstraits qui généralisent la méthode de Burstall.

Le meilleur moyen de convaincre le lecteur que notre principe d'induction de base correspond effectivement à la méthode de Burstall serait de le dériver d'une formalisation déjà existante de la méthode. Puisqu'il n'existe pas de formalisation suffisamment générale et largement admise, le mieux que nous puissions faire est de partir des exemples originaux de Burstall [74]. Nous avons choisi une version simplifiée de la preuve de Burstall [74] du programme suivant qui calcule 2^m lorsque $n \geq 0$ (nous utilisons les notations de Manna-Waldinger [78]) :

Exemple 5.3.1-1

```
Start:  P:=1;
Loop:   if N>0 then begin P:=2×P; N:=N-1;
                          goto Loop
                    end;
Finish:
```

Ce programme définit un système de transition $\langle S, A, t, \phi \rangle$ comme suit :

Les états du programme sont de la forme $\langle c, m, p \rangle$ où l'état de contrôle c est une étiquette du programme et l'état mémoire associe des valeurs entières $m, p \in \mathbb{Z}$ aux variables N, P du programme :

$S = \{ \text{Start}, \text{Loop}, \text{Finish} \} \times \mathbb{Z} \times \mathbb{Z}$

$A = \{\underset{\sim}{a}\}$

L'exécution commence au point "start" du programme avec une valeur initiale positive m de N et une valeur arbitraire p de P. Donc

$$\Phi(\langle c, m, p\rangle) = [c = \text{Start} \wedge m \geq 0]$$

Le programme est total et déterministe (tous les états, à part les états finaux, ont un seul état successeur) :

$$t_{\underset{\sim}{a}}(\langle c, m, p\rangle, \langle c', m', p'\rangle) =$$
$$[\ (c = \text{Start} \wedge c' = \text{Loop} \wedge m' = m \wedge p' = 1)$$
$$\vee\ (c = \text{Loop} \wedge m > 0 \wedge c' = \text{Loop} \wedge m' = m-1 \wedge p' = 2 \times p)$$
$$\vee\ (c = \text{Loop} \wedge m \leq 0 \wedge c' = \text{Finish} \wedge m' = m \wedge p' = p)]$$

Ce programme calcule $P = 2^m$ quand la valeur initiale m de N est positive. La propriété de correction totale peut être exprimée formellement par :

$$\Psi(\langle c, m, p\rangle, \langle c', m', p'\rangle) = [c' = \text{Finish} \wedge p' = 2^m]$$

est fatale pour $\langle S, A, \Sigma\langle S, A, t, \Phi\rangle\rangle$.

□

Le traitement des autres exemples de Burstall est similaire mais simplement beaucoup plus long

---

5.3.1.1 Preuves de propriétés de fatalité des programmes

La correction totale du programme 5.3.1-1 est spécifiée par la proposition :

"if sometime $(m \geq 0 \wedge N = m)$ at Start then sometime $P = 2^m$ at Finish"

La preuve de cette proposition utilise le lemme suivant :

"if sometime $(m > 0 \wedge N = m \wedge P = p)$ at Loop then sometime $(N = 0 \wedge P = p \times 2^m)$ at Loop"

Burstall observe que dans les énoncés ci-dessus m et p sont des variables mathématiques alors que N et P ne le sont pas puisque leur signification dépend du contexte. L'utilisation, dans un même énoncé, de variables du programme et de variables mathématiques pourrait prêter à confusion. Cette confusion peut être évitée si nous nous débarassons des variables du programme en utilisant des variables mathématiques différentes pour dénoter les valeurs des variables du programme à différents instants du calcul. Par exemple, le lemme pourrait être écrit comme suit :

"if sometime $n \geq 0$ at Loop then sometime $(n'=0 \wedge p'=p \times 2^n)$ at Loop"

qui signifie que :

"pour tout n, si $n \geq 0$ est vrai et l'exécution du programme commence à l'étiquette Loop avec la valeur n de la variable N du programme, alors l'exécution passera fatalement en Loop avec des valeurs n' et p' des variables N et P du programme telles que $(n'=0 \wedge p'=p \times 2^n)$ est vrai".

Alors le lemme établit simplement que :

$\theta_0(\langle c, m, p\rangle, \langle c', m', p'\rangle) = [c' = \text{Loop} \wedge m'=0 \wedge p'=p \times 2^m]$

est fatale pour $\langle S, A, t, \varepsilon_0\rangle$, où :

$\varepsilon_0(\langle c, m, p\rangle) = [c = \text{Loop} \wedge m \geq 0]$

De la même manière, la proposition établit la fatalité de

$\theta_1(\langle c, m, p\rangle, \langle c', m', p'\rangle) = [c' = \text{Finish} \wedge p' = 2^m]$

pour $\langle S, A, t, \varepsilon_1\rangle$, où :

$\varepsilon_1(\langle c, m, p\rangle) = [c = \text{Start} \wedge m \geq 0]$

Plus généralement, pour démontrer que Ψ est fatale pour $\langle S,A,t,\Phi\rangle$, la méthode de Burstall consiste à découvrir des propriétés auxiliaires $\{\theta_\ell \in (S^2 \to \{tt,ff\}) : \ell \in \Lambda\}$ et des conditions initiales correspondantes $\{\varepsilon_\ell \in (S \to \{tt,ff\}) : \ell \in \Lambda\}$ (telles que $\exists \pi \in \Lambda . [\varepsilon_\pi = \Phi \wedge \theta_\pi = \Psi]$) dont on démontre la fatalité :

$$\forall \ell \in \Lambda . \forall p \in \Sigma\langle S,A,t,\varepsilon_\ell\rangle . \exists i \in |p| . \theta_\ell(p_0,p_i)$$

On ne peut utiliser qu'un nombre fini ($\mathrm{card}(\Lambda) < \omega$) de lemmes.

Remarque

Puisque Burstall [74] ne considère que des programmes totaux et déterministes, l'énoncé :

"if sometime $P(m,p)$ at L then sometime $Q(m,p,m',p')$ at L'"

peut être compris également comme :

$$\exists p \in \Sigma\langle S,A,t,\varepsilon\rangle . \exists i \in |p| . \theta(p_0,p_i)$$

où

$$\varepsilon(\langle c,m,p\rangle) = [c = L \wedge P(m,p)]$$

$$\theta(\langle c,m,p\rangle,\langle c',m',p'\rangle) = [c' = L' \wedge Q(m,p,m',p')]$$

Tous les résultats de ce paragraphe peuvent être aisément adaptés pour cette interprétation existentielle. Cependant nous avons choisi de développer l'interprétation universelle parce qu'elle est plus adaptée à la correction totale (et plus généralement aux propriétés de fatalité) de programmes parallèles.

□

5.3.1.2 Un exemple de preuve

Maintenant, nous allons essayer à l'aide de l'exemple, de saisir l'essence de la méthode de preuve de Burstall :

La preuve de la proposition θ_1 est la suivante :

Supposer :

"sometime $(N \geq 0 \wedge N = m)$ at Start" (13)

alors par évaluation symbolique :

"sometime $(N \geq 0 \wedge N = m \wedge P = 1)$ at Loop" (12)

puis d'après le lemme θ_0 :

"sometime $(N = 0 \wedge P = 2^m)$ at Loop" (11)

puis par évaluation symbolique :

"sometime $P = 2^m$ at Finish" (10)

Q.E.D.

La preuve du lemme θ_0 est par induction sur m, comme suit :

Supposer :

"sometime $(N \geq 0 \wedge N = m \wedge P = p)$ at Loop" (02)

soit $N \leq 0$ et Q.E.D.

ou $N > 0$ et alors par évaluation symbolique :

"sometime $(N > 0 \wedge N = m-1 \wedge P = p \times 2)$ at Loop" (01)

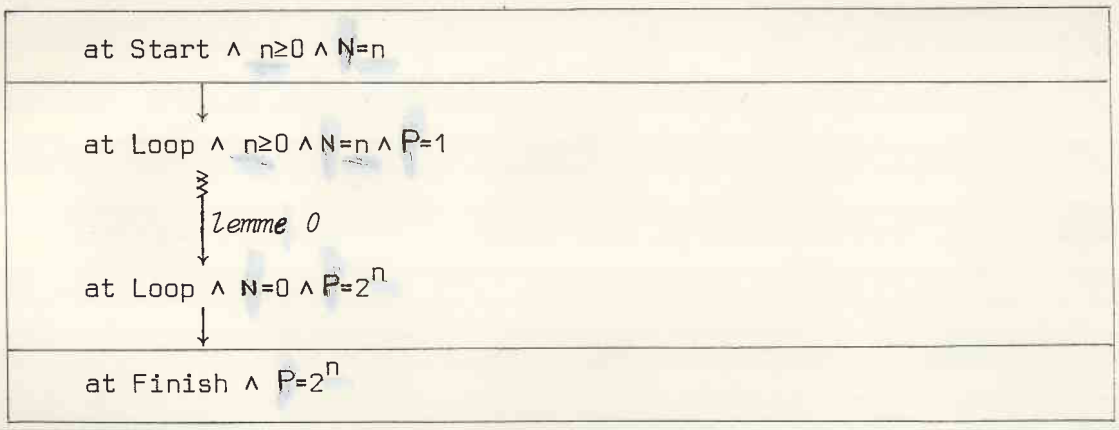
alors d'après le lemme θ_0 comme hypothèse d'induction pour $m-1$ (tel que $m > m-1 \geq 0$) :

"sometime $(N = 0 \wedge P = p \times 2^m)$ at Loop" (00)

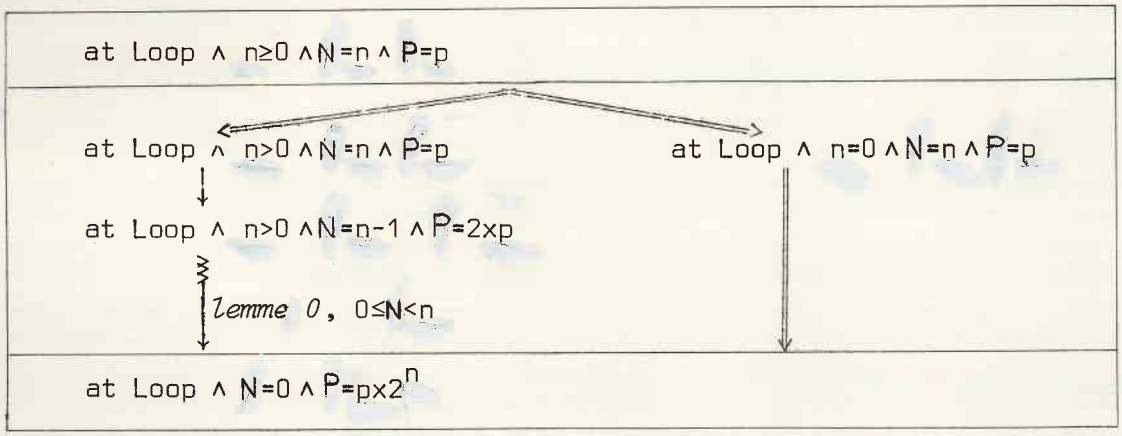
Q.E.D.

Au paragraphe 5.6 nous utiliserons des chartes de preuves permettant de présenter la preuve ci-dessus comme suit :

Proposition 1 :



Lemme 0 :



5.3.1.3 Assertions intermittentes

La preuve d'un lemme est une séquence non-vide d'assertions intermittentes dérivant les unes des autres par évaluation symbolique ou par application de lemmes. Il est clair que des séquences de dérivation infinies conduiraient à des preuves invalides (puisque les exécutions infinies ne seraient pas écartées). Par conséquent, une preuve d'une proposition θ_ℓ doit comporter un nombre fini $n_\ell+1$ d'assertions intermittentes que nous désignerons par $I^{n_\ell}_\ell, \ldots, I^1_\ell, I^0_\ell$.

Les assertions intermittentes utilisées dans la preuve n'ont pas besoin d'être distinctes comme nous le voyons dans le contre-exemple ci-dessus qui est une preuve valide de la proposition θ_1 pour tout $k \geq 1$ fini :

	"sometime $(N \geq 0 \wedge N = m)$ at Start"	(Prémisse)
	"sometime $(N \geq 0 \wedge N = m \wedge P = 1)$ at Loop"	(Evaluation symbolique)
k fois	"sometime $(N = 0 \wedge P = 2^m)$ at Loop"	(Lemme)
	...	...
	"sometime $(N = 0 \wedge P = 2^m)$ at Loop"	(Lemme)
	"sometime $P = 2^m$ at Finish"	(Conclusion)

Donc l'utilisation d'un nombre fini d'assertions intermittentes ne garantit pas que la longueur de la preuve soit finie. Par conséquent, nous devons garantir que le nombre de dérivations est fini. Pour cela, nous imposerons que toutes les occurences des assertions intermittentes utilisées dans une preuve, soient désignées par des nombres naturels et dérivées dans un ordre strictement décroissant.

Par exemple, la preuve de la proposition θ_1 comporte la découverte des assertions intermittentes suivantes :

$I_1^3(\langle c, m, p\rangle, \langle c', m', p'\rangle) = [c' = \text{Start} \wedge m' \geq 0 \wedge m' = m]$

$I_1^2(\langle c, m, p\rangle, \langle c', m', p'\rangle) = [c' = \text{Loop} \wedge m' \geq 0 \wedge m' = m \wedge p' = 1]$

$I_1^1(\langle c, m, p\rangle, \langle c', m', p'\rangle) = [c' = \text{Loop} \wedge m' = 0 \wedge p' = 2^m]$

$I_1^0(\langle c, m, p\rangle, \langle c', m', p'\rangle) = [c' = \text{Finish} \wedge p' = 2^m]$

tandis que la preuve du lemme θ_0 comporte la découverte de :

$I_0^2(\langle c, m, p\rangle, \langle c', m', p'\rangle) = [c' = \text{Loop} \wedge m' \geq 0 \wedge m' = m \wedge p' = p]$

$I_0^1(\langle c, m, p\rangle, \langle c', m', p'\rangle) = [c' = \text{Loop} \wedge m' > 0 \wedge m' = m - 1 \wedge p' = p \times 2]$

$I_0^0(\langle c, m, p\rangle, \langle c', m', p'\rangle) = [c' = \text{Loop} \wedge m' = 0 \wedge p' = p \times 2^m]$

Remarque :

D'après Burstall [74], " "sometime P at L" says that there exists a state during the execution which is at L and has property P". Autrement dit, toutes les assertions intermittentes I_ℓ^i utilisées dans la preuve du lemme θ_ℓ devraient être fatales pour $\langle S, A, \Sigma\langle S, A, t, \varepsilon_\ell\rangle\rangle$. Cette interprétation des assertions intermittentes est incorrecte. Par exemple, I_0^1 n'est jamais vrai durant l'exécution lorsque initialement N=0. Plus généralement, Burstall [74] traite les tests par cas de sorte que les assertions intermittentes utilisées dans chaque cas pourraient ne pas être fatales pour ceux des états initiaux ne correspondant pas au cas considéré. Nous choisirons une autre interprétation des assertions intermittentes de sorte que l'analyse de cas ne pose aucun problème puisque seulement la disjonction des assertions intermittentes correspondant à tous les cas, doit être fatale pour tous les états initiaux.

□

5.3.1.4 Conditions de vérification

Dans une preuve valide de fatalité de θ_ℓ pour $\langle S, A, \Sigma\langle S, A, t, \varepsilon_\ell\rangle\rangle$, les assertions intermittentes $I_\ell^{m_\ell}, \ldots, I_\ell^1, I_\ell^0$ dérivent les unes des autres suivant des règles (pour calculer l'effet d'une affectation ou d'un test, pour utiliser un lemme, etc.). Les règles informelles de Burstall [74] doivent être comprises comme des conditions de vérification que doivent vérifier les assertions intermittentes. Nous exprimons maintenant, ces conditions de vérification formellement.

5.3.1.4.1 Prémisse

Toutes les preuves dans Burstall [74] commencent par supposer la prémisse ε_ℓ de la proposition ou du lemme ϑ_ℓ qu'on démontre. Autrement dit, $I_\ell^{m_\ell}$ doit être vérifiée par les états initiaux :

$$\forall \Delta, \Delta' \in S.\ ([\varepsilon_\ell(\Delta) \wedge \Delta' = \Delta] \Rightarrow I_\ell^{m_\ell}(\Delta, \Delta'))$$

ou plus simplement :

$$\forall \Delta \in S.\ (\varepsilon_\ell(\Delta) \Rightarrow I_\ell^{m_\ell}(\Delta, \Delta))$$

Par exemple, la preuve de la proposition ϑ_1 commence par la vérification que :

$$\forall \langle c, m, p\rangle \in S.\ (\varepsilon_1(\langle c, m, p\rangle) \Rightarrow I_1^3(\langle c, m, p\rangle, \langle c, m, p\rangle)$$

(où $c = \text{start}$, $m \geq 0$ ou la condition de vérification est vraie de manière évidente)

tandis que pour le lemme ϑ_0, nous avons :

$$\forall \langle c, m, p\rangle \in S.\ (\varepsilon_0(\langle c, m, p\rangle) \Rightarrow I_0^2(\langle c, m, p\rangle, \langle c, m, p\rangle))$$

(où $c = \text{Loop}$, $m \geq 0$ dans le cas non évident)

5.3.1.4.2 Evaluation symbolique

Supposons que la preuve de la proposition ϑ_ℓ ait progressé jusqu'à atteindre l'assertion intermittente I_ℓ^i qui n'est pas la dernière. Le prochain pas peut être traité par évaluation symbolique.

Pour les programmes totaux déterministes, les règles de Burstall [74] pour calculer l'effet d'une affectation ou d'un test, vérifient que l'état courant s' satisfaisant I_ℓ^i a un successeur s'' satisfaisant une certaine assertion intermittente I_ℓ^j qui doit être prise en considération plus tard dans la preuve, d'où $j < i$:

$$[I_\ell^i(\Delta, \Delta') \wedge t_{\underset{\sim}{a}}(\Delta', \Delta'')] \Rightarrow \exists j < i.\ I_\ell^j(\Delta, \Delta'')$$

Par exemple, dans la preuve de la proposition θ_ℓ, l'affectation $P := 1$ conduit de $\langle 13 \rangle$ à $\langle 12 \rangle$ et correspond à la condition de vérification :

$$[\, I_1^3(\langle c, m, p\rangle, \langle c', m', p'\rangle) \wedge t_{\underline{a}}(\langle c', m', p'\rangle, \langle c'', m'', p''\rangle)] \Rightarrow I_1^2(\langle c, m, p\rangle, \langle c'', m'', p''\rangle)$$

(où $c' = \text{start}$, $m' \geq 0$, $m' = m$ ou la condition est vérifiée de manière évidente)

Le test $N \leq 0$ conduit de $\langle 11 \rangle$ à $\langle 10 \rangle$ et correspond à la condition de vérification :

$$[\, I_1^1(\langle c, m, p\rangle, \langle c', m', p'\rangle) \wedge t_{\underline{a}}(\langle c', m', p'\rangle, \langle c'', m'', p''\rangle)] \Rightarrow I_1^0(\langle c, m, p\rangle, \langle c'', m'', p''\rangle)$$

(où $c' = \text{Loop}$, $m' \geq 0$, $p' = 2^m$, $c'' = \text{Finish}$, $m'' = m'$, $p'' = p'$)

Dans la preuve du lemme θ_0, le corps de la boucle mène de $\langle 02 \rangle$ à $\langle 01 \rangle$. (En accord avec la sémantique opérationnelle du programme 5.3.1-1, le corps de la boucle doit être traité comme une action atomique). La condition de vérification correspondante est :

$$[\, I_0^2(\langle c, m, p\rangle, \langle c', m', p'\rangle) \wedge m' > 0 \wedge t_{\underline{a}}(\langle c', m', p'\rangle, \langle c'', m'', p''\rangle)] \Rightarrow I_0^1(\langle c, m, p\rangle, \langle c'', m'', p''\rangle)$$

(où $c' = \text{Loop}$, $m' = m$, $p' = p$, $m' > 0$, $c'' = \text{Loop}$, $m'' = m' - 1$, $p'' = 2 \times p'$ ou la condition est trivialement satisfaite)

De telles conditions de vérification ne sont pas suffisantes lorsque les affectations ou les tests comportent des fonctions partielles. Dans ce cas il faut démontrer qu'aucun état de blocage n'est accessible. Plus généralement, lorsqu'il y a non-déterminisme, l'évaluation symbolique doit garantir l'existence d'au moins un état successeur :

$$\forall \Delta, \Delta' \in S.\ [\, I_\ell^i(\Delta, \Delta') \Rightarrow \exists \Delta'' \in S.\ t_{\underline{a}}(\Delta', \Delta'')]$$

et que ces états successeurs possibles satisfont une certaine assertion intermittente qui sera considérée plus tard :

$$\forall \Delta, \Delta', \Delta'' \in S.\ [(I_\ell^i(\Delta, \Delta') \wedge t_{\underline{a}}(\Delta', \Delta'')) \Rightarrow (\exists j < i.\ I_\ell^j(\Delta, \Delta''))]$$

5.3.1.4.3 Utilisation de lemmes dans la preuve de propositions

Dans la preuve de la proposition θ_1, l'assertion intermittente ς_{11} dérive de ς_{12} par le lemme θ_0. On doit d'abord vérifier que les états courants s' satisfaisant ς_{12} satisfont également la prémisse ε_0 du lemme θ_0. Alors, appliquant le lemme, on doit montrer que tous les successeurs s'' de s' par θ_0 satisfont ς_{12}. Les conditions de vérification correspondantes sont :

$$[\, I_1^2(\langle c,m,p\rangle, \langle c',m',p'\rangle) \Rightarrow \varepsilon_0(\langle c',m',p'\rangle)]$$

$$\wedge\, [\, I_1^2(\langle c,m,p\rangle, \langle c',m',p'\rangle) \wedge \theta_0(\langle c',m',p'\rangle, \langle c'',m'',p''\rangle) \Rightarrow I_1^1(\langle c,m,p\rangle, \langle c'',m'',p''\rangle)]$$

(où dans le cas non banal $c' = \mathrm{Loop}$, $m' > 0$, $m'' = m$, $p' = 1$, $c'' = \mathrm{Loop}$, $m'' = 0$, $p'' = p' \times 2^{m'}$)

Plus généralement, la condition de vérification correspondant à l'utilisation d'un lemme dans la preuve d'une proposition est (temporairement) :

$$\forall s, s' \in S.\ [I_\ell^i(s,s') \Rightarrow (\exists \ell' \in \Lambda.\ [\varepsilon_{\ell'}(s') \wedge \forall s'' \in S.\ [\theta_{\ell'}(s',s'') \Rightarrow \exists j < i.\ I_\ell^j(s,s'')]])]$$

Observer que (contrairement au cas de l'évaluation symbolique) le test que les états courants s' satisfont la prémisse $\varepsilon_{\ell'}$ du lemme $\theta_{\ell'}$ garantit l'existence d'au moins un successeur s'' à s'. Ceci parceque'on a démontré séparément que le lemme $\theta_{\ell'}$ est fatal pour $\langle S, A, \Sigma \langle S, A, t, \varepsilon_{\ell'}\rangle\rangle$.

A propos de l'utilisation des lemmes, noter que Burstall [74] compte sur la culture mathématique de ses lecteurs et ne prend pas la peine de préciser les règles logiques élémentaires telles que les preuves des lemmes et des propositions ne doivent pas être circulaires. Cependant, de telles règles doivent être prises en compte dans la formalisation de la méthode de Burstall [74]. Une façon simple consiste à ordonner partiellement l'ensemble Λ des lemmes par un ordre bien-fondé $\prec$ tel que $\ell' \prec \ell$ est

compris comme : la preuve de fatalité de $\theta_{\ell'}$ ne dépend pas de l'hypothèse que θ_ℓ est fatal. La condition de vérification (définitive) correspondant à l'utilisation d'un lemme dans la preuve d'une proposition est maintenant :

$$\forall s,s' \in S.\,[I^i_\ell(s,s') \Rightarrow (\exists \ell' \in \Lambda.\,[\ell' \prec \ell \wedge \varepsilon_{\ell'}(s') \wedge \forall s'' \in S.\,[\theta_{\ell'}(s',s'') \Rightarrow \exists j<i.\, I^j_\ell(s,s'')]])]$$

De plus, puisque l'ensemble Λ est fini et $\prec$ est bien-fondé, nous pouvons toujours (à un isomorphisme et une fonction-rang près) choisir Λ comme un ensemble d'entiers naturels et $\prec$ comme l'ordre naturel $<$ correspondant.

5.3.1.4.4 Preuve par induction sur les données

Burstall [74] démontre les lemmes en utilisant différentes formes de l'induction mathématique, qui sont toutes équivalentes à :

$$\forall m' \in \omega.\,[(\forall m < m'.\, P(m)) \Rightarrow P(m')] \Rightarrow [\forall m' \in \omega.\, P(m')]$$

Par exemple, dans la preuve du lemme θ_0, l'assertion intermittente '00' dérive de l'assertion '01' en utilisant le lemme θ_0 comme hypothèse d'induction. Ceci est valide parce que :

$$I^1_0(\langle c,m,p\rangle, \langle c',m',p'\rangle) \Rightarrow [\varepsilon_0(\langle c',m',p'\rangle) \wedge m' < m]$$

Puis, par hypothèse d'induction, nous dérivons l'assertion intermittente I^0_0 telle que :

$$[I^1_0(\langle c,m,p\rangle, \langle c',m',p'\rangle) \wedge \theta_0(\langle c',m',p'\rangle, \langle c'',m'',p''\rangle)] \Rightarrow I^0_0(\langle c,m,p\rangle, \langle c'',m'',p''\rangle)$$

(où $c' = \text{Loop}$, $m > 0$, $m' = m-1$, $p' = p \times 2$, $c'' = \text{Loop}$, $m'' = 0$, $p'' = p' \times 2^{m'}$)

Cette condition de vérification est propre à l'exemple considéré mais en général Burstall [74] précise que l'induction est sur les données. Puisque le principe de l'induction mathématique ci-dessus s'applique aux nombres naturels, l'induction sur les données utilise une fonction f_0 des données dans les nombres naturels.

Par exemple,

$$I_0^1(\langle c,m,p\rangle,\langle c',m',p'\rangle) \Rightarrow [\varepsilon_0(\langle c',m',p'\rangle) \wedge f_0(\langle c',m',p'\rangle) < f_0(\langle c,m,p\rangle)]$$

où :

$$f_0(\langle c,m,p\rangle) = m$$

Puisque les preuves de lemmes différents sont habituellement différentes, des fonctions f_ℓ différentes peuvent être utilisées, d'où $f_\ell \in (\Lambda \to (S \to \omega))$. Nous inférons, à partir de l'exemple, que la condition de vérification pour l'utilisation d'un lemme comme hypothèse d'induction dans la preuve de ce même lemme devrait être de la forme :

$$\forall \Delta,\Delta' \in S.\,[I_\ell^i(\Delta,\Delta') \Rightarrow [\varepsilon_\ell(\Delta') \wedge f_\ell(\Delta') < f_\ell(\Delta) \wedge \forall \Delta'' \in S.\,[\theta_\ell(\Delta',\Delta'') \Rightarrow \exists j<i.\, I_\ell^j(\Delta,\Delta'')]]]$$

5.3.1.4.5 Conclusion

Commençant par la prémisse d'un lemme, une preuve de ce lemme est finie lorsqu'on a dérivé une assertion intermittente qui implique la conclusion de ce lemme :

$$\forall \Delta,\Delta' \in S.\,[\,I_\ell^i(\Delta,\Delta') \Rightarrow \theta_\ell(\Delta,\Delta')]$$

Par exemple, la preuve de la proposition θ_1 se termine par :

$$I_1^0(\langle c,m,p\rangle,\langle c',m',p'\rangle) \Rightarrow \theta_1(\langle c,m,p\rangle,\langle c',m',p'\rangle)$$

(où $c' = \text{Finish}$, $p' = 2^m$ dans le cas non trivial)

tandis que la preuve du lemme θ_0 se termine soit par :

$$[\,I_0^2(\langle c,m,p\rangle,\langle c',m',p'\rangle) \wedge m' \leq 0] \Rightarrow \theta_0(\langle c,m,p\rangle,\langle c',m',p'\rangle)$$

(où $c' = \text{Loop}$, $m' \geq 0$, $m' = m$, $p' = p$)

ou par :

$$I_0^0(\langle c,m,p\rangle,\langle c',m',p'\rangle) \Rightarrow \theta_0(\langle c,m,p\rangle,\langle c',m',p'\rangle)$$

(où $c' = \text{Loop}$, $m' = 0$, $p' = p \times 2^{m'}$)

Finalement, observons que dans une preuve, toutes les assertions intermittentes intermédiaires devraient être traitées (soit par évaluation symbolique soit en utilisant un lemme (dans la preuve d'une proposition ou comme hypothèse d'induction)) ou impliquent la conclusion.

5.3.1.5 Le principe d'induction de base formalisant la méthode des assertions intermittentes

Nous pouvons maintenant résumer ce que nous avons appris à partir de l'exemple. Pour démontrer que Ψ est fatale pour $\langle S, A, \Sigma\langle S, A, t, \Phi\rangle\rangle$, la méthode de Burstall [74] consiste à démontrer que:

$$
\begin{array}{l}
[\,\exists \Lambda \in \omega,\ \varepsilon \in (\Lambda \to (S \to \{tt, ff\})),\quad \theta \in (\Lambda \to (S^2 \to \{tt, ff\})),\quad \eta \in (\Lambda \to (S \to \omega)),\\
\quad m \in (\Lambda \to \omega).\\
\\
\quad (\exists \pi \in \Lambda.\ [\varepsilon_\pi = \Phi \wedge \theta_\pi = \Psi])\\
\quad \wedge\ (\forall \ell \in \Lambda.\ \exists I_\ell \in (m_\ell + 1 \to (S^2 \to \{tt, ff\})).\\
\qquad \forall i \le m_\ell,\ s, s' \in S.\\
\\
(P) \qquad [\varepsilon_\ell(s) \Rightarrow I_\ell^{m_\ell}(s, s)]\\
\qquad \wedge\ [I_\ell^i(s, s') \Rightarrow\\
(HS) \qquad (\exists s'' \in S, a \in A.\ (t_a(s', s'') \wedge \forall s'' \in S.\ [t_a(s', s'') \Rightarrow \exists j < i.\ I_\ell^j(s, s'')]))\\
\qquad\ \vee\\
(LI) \qquad (\exists \ell' \in \Lambda.\ [((\ell' < \ell) \vee (\ell = \ell' \wedge \eta_\ell(s') < \eta_\ell(s))) \wedge \varepsilon_{\ell'}(s') \wedge\\
\qquad\qquad \forall s'' \in S.\ (\theta_{\ell'}(s', s'') \Rightarrow \exists j < i.\ I_\ell^j(s, s''))])\\
\qquad\ \vee\\
(C) \qquad \theta_\ell(s, s')])]
\end{array}
\qquad (B_1)
$$

5.3.1.6 Questions relatives à la correction et à la complétude sémantique de la méthode de Burstall

La question de la correction et de la complétude de la méthode de Burstall a déjà été partiellement abordée. En représentant les programmes par des relations de transition dont le non-déterminisme est fini et en donnant une interprétation temporelle de la méthode des assertions intermittentes, Pnueli [77] a montré la correction et la complétude sémantique d'une version de la méthode de Burstall. Des résultats similaires de correction et de complétude ont été obtenus par Apt-Delporte [83] terministes structurés. De tels résultats ent de la remarque de Manna-Waldinger [78] entes peut être utilisée pour exprimer les preuves classiques "à la Floyd", une méthode qui est sémantiquement complète (cf. théorème 5.2.6.2 n°1).

Cependant, l'exacte portée des résultats ci-dessus devrait être interprétée avec beaucoup de précautions puisque ces preuves traitent seulement le cas d'assertions intermittentes unaires (c'est-à-dire qui portent sur les états, comme "if sometime P(s) at L then sometime Q(s') à L' ") tandis que la méthode de Burstall et le principe d'induction ($\mathbb{B}_1$) utilisent des assertions intermittentes binaires (c'est-à-dire qui relient des états, comme " if sometime P(s) at L then sometime Q(s,s') at L' "). On a souvent soutenu que les deux approches sont équivalentes car l'effet d'assertions binaires peut être obtenu en utilisant des variables auxiliaires et des assertions unaires. En effet les valeurs initiales ou intermédiaires des variables d'un programme peuvent être conservées dans des variables auxiliaires dont les valeurs font partie de l'état. En fait, l'utilisation de variables auxiliaires et d'assertions unaires est plus puissante que l'utilisation d'assertions binaires comme dans ($\mathbb{B}_1$). Ceci parce qu'en utilisant des variables

auxiliaires, on peut exprimer des relations entre les valeurs des variables à deux instants différents quelconques au cours du calcul (et même "mémoriser" les traces d'exécution entières dans des variables d'histoire). Ceci n'est pas possible avec des assertions binaires puisque, par exemple, seulement la proposition principale (et non tous les lemmes) peut dépendre des valeurs initiales des variables du programme dans le principe d'induction ($\mathcal{B}_1$). Cependant, l'utilisation d'assertions binaires semble être beaucoup plus simple puisque la question de savoir quand on doit introduire des variables auxiliaires est résolue une fois pour toutes.

Le principe d'induction ($\mathcal{B}_1$) est correct mais nous faisons la conjecture qu'il n'est pas sémantiquement complet.

5.3.1.6.1 Correction

Théorème 5.3.1.6.1~1 (Correction)

fatale pour $\langle S, A, \Sigma \langle S, A, t, \varepsilon \rangle\rangle$)

Démonstration

Nous introduirons plus tard ($\mathcal{B}_2$), une généralisation évidente de ($\mathcal{B}_1$) (de sorte que ($\mathcal{B}_1$) $\Rightarrow$ ($\mathcal{B}_2$)) et démontrerons que ($\mathcal{B}_2$) est correct.

□

5.3.1.6.2 Conjectures à propos de la complétude sémantique

Bien que le principe d'induction $(\mathcal{B}_1)$ n'utilise que l'induction sur les entiers naturels (et contrairement à $(\mathcal{F}_2^1)$, 5.2.6.3 n°1) il permet de démontrer la terminaison faible de programmes qui ne se terminent pas fortement (rappelons que, d'après Dijkstra, la terminaison est forte si on peut donner une borne finie sur le nombre de pas du programme en fonction de l'état initial et qu'elle est faible sinon). Pour le montrer nous utilisons l'exemple suivant (pris littéralement dans Dijkstra [82, p. 356]) :

Exemple 5.3.1.6.2-1

En général, le programme suivant (X et Y étant des constantes naturelles) :

```
x,y := X,Y ;
do  x>0 → x,y := x-1, un nombre naturel quelconque
[]  y>0 → y := y-1
od
```

n'a pas la propriété de terminaison forte, parce qu'aucune borne ne peut être donnée lorsque $X>0$.

La terminaison faible peut être démontrée au moyen de la méthode de Floyd [67] en utilisant l'ordre lexicographique gauche sur des paires de nombres naturels $\langle x,y\rangle$ (mais pas par simple induction sur les naturels)!

Elle peut être démontrée également en utilisant le principe d'induction $(\mathcal{B}_1)$. Nous avons $S=\mathbb{Z}^2$, $t_{\underline{2}}(\langle x,y\rangle,\langle x',y'\rangle) = [(x>0 \wedge x'=x-1) \vee (y>0 \wedge x'=x \wedge y'=y-1)]$, $\Phi(\langle x,y\rangle) = [x=X\geq 0 \wedge y=Y\geq 0]$, $\Psi(\langle x,y\rangle,\langle x',y'\rangle) = [x'=y'=0]$ et choisissons $\Lambda = X+2$, $\varepsilon_{X+1} = \Phi$, $\varepsilon_\ell(\langle x,y\rangle) = [x=\ell]$ pour $\ell \in (X+1)$, $\theta_\ell = \Psi$ pour $\ell \in (X+2)$, $\pi = X+1$, $f_\ell(\langle x,y\rangle) = y$ pour $\ell \in (X+2)$, $n_0 = 2$, $I_0^2(\langle x,y\rangle,\langle x',y'\rangle) = (\varepsilon_0(\langle x,y\rangle) \wedge \langle x',y'\rangle = \langle x,y\rangle)$ [(P), (c) quand $y'=0$, (HS) quand $y'>0$], $I_0^1(\langle x,y\rangle,\langle x',y'\rangle) = (x=0 \wedge y>0 \wedge t_{\underline{2}}(\langle x,y\rangle,\langle x',y'\rangle))$ [(LI) avec $\ell'=\ell=0$], $I_0^0 = \theta_0$ [(c)], lorsque $\ell = 1,\dots,X$, $n_\ell = 2$, $I_\ell^2(\langle x,y\rangle,\langle x',y'\rangle) =$

$(\varepsilon_\ell(\langle x,y\rangle) \wedge \langle x',y'\rangle = \langle x,y\rangle)$ [(P), (HS)], $I^1_\ell(\langle x,y\rangle,\langle x',y'\rangle) = (\varepsilon_\ell(\langle x,y\rangle) \wedge t_{\underset{\sim}{\ell}}(\langle x,y\rangle,\langle x',y'\rangle))$ [(LI) avec $\ell' = \ell - 1$ quand $x' = x-1$, (LI) avec $\ell' = \ell$ quand $x' = x$ et $y' = y-1$], $I^\circ_\ell = \theta_\ell$ [(C)], $n_{X+1} = 1$, $I^1_{X+1}(\langle x,y\rangle,\langle x',y'\rangle) = (\varepsilon_{X+1}(\langle x,y\rangle) \wedge \langle x',y'\rangle = \langle x,y\rangle)$ [(P), (LI) avec $\ell' = X$], $I^\circ_{X+1} = \theta^\circ_{X+1}$ [(C)]. (Le lecteur pourra contrôler que les conditions de vérifications sont satisfaites. Nous avons indiqué après chaque assertion intermittente, l'alternative qui devrait être choisie).

□

Comme on l'a vu dans l'exemple ci-dessus, la plus grande généralité de la méthode de Burstall restreinte aux nombres naturels (qui peut être utilisée pour démontrer la terminaison faible) par rapport à la méthode de Floyd restreinte aux nombres naturels (qui ne peut être utilisée que pour démontrer la terminaison forte) est seulement spécieuse, parceque la méthode de Burstall est implicitement fondée sur l'ordre lexicographique de paires de nombres naturels comme le montre le principe d'induction $(\mathcal{B}_1)$.

Malgré cette supériorité apparente, le rang de l'ordre lexicographique sur des paires de nombres naturels, utilisé dans le principe d'induction $(\mathcal{B}_1)$ n'est pas aussi grand qu'il est nécessaire quand on considère un non-déterminisme arbitrairement infini. Aussi nous faisons la :

Conjecture 5.3.1.6.2 v2 (Incomplétude sémantique)

pour $\langle S, A, \Sigma\langle S, A, t, E\rangle\rangle$ $\not\Rightarrow$ $(\mathcal{B}_1)$

Par analogie avec la méthode de Floyd, deux solutions peuvent être envisagées pour résoudre les problèmes d'incomplétude relatives au non-déterminisme infini. La première consiste à considérer seulement le non-déterminisme fini. L'autre consiste à considérer l'induction sur des bons-ordres arbitraires (ou à un isomorphisme près sur des ordinaux).

Cependant, nous nous hasardons à faire la conjecture que le principe d'induction $(\mathcal{B}_1)$ n'est pas complet même avec ces hypothèses simplificatrices :

Conjecture 5.3.1.6.2∿3 (Incomplétude sémantique pour le nondéterminisme fini)

$$(\Psi \text{ est fatale pour } \langle S, A, \Sigma\langle S, A, t, \epsilon\rangle\rangle \wedge \forall s \in S.\ (\underset{\sim}{\mathrm{card}}(\{s' \in S : t_{\underset{\sim}{a}}(s,s')\}) < \omega)) \not\Rightarrow (\mathcal{B}_1)$$

Conjecture 5.3.1.6.2∿4 (Incomplétude sémantique pour des bons-ordres arbitraires)

$$(\Psi \text{ est fatale pour } \langle S, A, \Sigma\langle S, A, t, \epsilon\rangle\rangle) \not\Rightarrow (\ (\mathcal{B}_1) \text{ où } f \in (\Lambda \to (S \to \Delta)),\ \Delta \in \underset{\sim}{\mathrm{Ord}})$$

Ces conjectures découlent de la remarque qu'excepté pour les exemples triviaux (qui peuvent être traités par évaluation symbolique), les preuves utilisent une relation bien-fondée sur l'ensemble des descendants des états initiaux correspondant à $\langle \Lambda \times S, \prec\rangle$ où $\langle \ell', s'\rangle \prec \langle \ell, s\rangle$ si et seulement si $(\ell' < \ell \vee (\ell' = \ell \wedge f_\ell(s') < f_\ell(s)))$. Bien qu'il existe (d'après l'hypothèse de fatalité) une relation bien-fondée sur l'ensemble des descendants de <u>chaque</u> état initial, il peut ne pas exister de telle relation bien-fondée sur l'ensemble des descendants de <u>tous</u> les états initiaux comme c'est nécessaire dans le principe d'induction $(\mathcal{B}_1)$ parceque f_ℓ ne dépend pas des états initiaux. C'est le cas pour $S = \omega$, $t_{\underset{\sim}{a}}(x,x') = [x' = x+1]$, $\Phi(x) = \#$, $\Psi(x,x') = [x' = 2x]$.

5.3.1.6.3 Un résultat de complétude sémantique partielle

Les conjectures ci-dessus n'ont que des conséquences limitées car elles ne s'appliquent pas pour un grand nombre de situations pratiques.

C'est le cas lorsque le nondéterminisme est fini et le nombre d'états initiaux est fini de sorte que (au moins en théorie) les preuves peuvent être entièrement faites par évaluation symbolique.

Des situations plus intéressantes sont celles de la correction totale des programmes séquentiels considérée par Burstall [74] ou bien les assertions intermittentes considérées par Pnueli [77] et Apt-Delporte [83].

Ces deux sortes de situations peuvent être traitées comme des cas particuliers de la situation plus générale où la fatalité de Ψ est indépendante des états initiaux pour $\langle S,A,t,\varepsilon\rangle$ (c'est-à-dire qu'aucun état intermédiaire ne peut être un but) :

Définition 5.3.1.6.3 :1 (Indépendance des états initiaux)

$$\mathrm{Isind}\langle S,A,t,\varepsilon,\Phi,\Psi\rangle = [(\mathrm{Inter}\langle S,A,t,\varepsilon,\Phi,\Psi\rangle \cap \mathrm{Goal}\langle S,A,t,\varepsilon,\Phi,\Psi\rangle) = \emptyset]$$

Quand cette condition (suffisante mais non nécessaire) d'indépendance par rapport aux états initiaux est satisfaite, nous pouvons faire des preuves de fatalité en utilisant (B_1) avec $f\in(\Lambda\to(S\to\Delta))$ pour un certain $\Delta\in\mathrm{Ord}$.

Avant de démontrer ce fait, nous devons caractériser l'ordinal Δ qui est nécessaire, autrement dit proposer une "mesure" du non-déterminisme global du programme (par opposition aux caractérisations locales du nondéterminisme comme le nondéterminisme dit fini) :

Nous démontrons d'abord le :

Lemme 5.3.1.6.3 v 2 (Existence d'une relation bien-fondée pour les preuves de fatalité (avec l'hypothèse d'indépendance des états initiaux))

$$[(\Psi \text{ est fatale sous invariance de } \Phi \text{ pour } \langle S,A,\Sigma\langle S,A,t,\varepsilon\rangle\rangle) \wedge \underset{\sim}{\text{Isind}}\langle S,A,t,\varepsilon,\Phi,\Psi\rangle] \Rightarrow \underset{\sim}{\text{wf}}(\underset{\sim}{\text{Acc}}\langle S,A,t,\varepsilon,\Phi,\Psi\rangle, t1\,\underset{\sim}{\text{Inter}}\langle S,A,t,\varepsilon,\Phi,\Psi\rangle^{-1})$$

Démonstration

Supposons par l'absurde que $\exists p \in (\omega \to \underset{\sim}{\text{Acc}}\langle S,A,t,\varepsilon,\Phi,\Psi\rangle).\ \forall i \in \omega.$ $t1\,\underset{\sim}{\text{Inter}}\langle S,A,t,\varepsilon,\Phi,\Psi\rangle(p_i,p_{i+1})$. Nous pouvons supposer $\varepsilon(p_0)$ (autrement nous pouvons adjoindre à gauche de p un préfixe $r_0,\ldots,r_k$ d'une trace de $\Sigma\langle \underset{\sim}{\text{Acc}}\langle S,A,t,\varepsilon,\Phi,\Psi\rangle, A, t1\,\underset{\sim}{\text{Inter}}\langle S,A,t,\varepsilon,\Phi,\Psi\rangle, \varepsilon\rangle$ telle que $\varepsilon(r_0)$ est vrai). Comme Ψ est fatale sous invariance pour p, il existe un plus petit i, $i \in |p|$ tel que $\Psi(p_0,p_i)$ est vrai. Alors $p_i \in \underset{\sim}{\text{Goal}}\langle S,A,t,\varepsilon,\Phi,\Psi\rangle$. Aussi $r1\,\underset{\sim}{\text{Inter}}\langle S,A,t,\varepsilon,\Phi,\Psi\rangle(p_i,p_{i+1})$ implique $p_i \in \underset{\sim}{\text{Inter}}\langle S,A,t,\varepsilon,\Phi,\Psi\rangle$ en contradiction avec $\underset{\sim}{\text{Isind}}\langle S,A,t,\varepsilon,\Phi,\Psi\rangle$.

Le nondéterminisme global de $\langle S,A,t,\varepsilon\rangle$ relativement à Φ et Ψ peut être mesuré par le rang de l'inverse de t restreinte aux états intermédiaires :

(Rang du nondéterminisme global (avec l'hypothèse d'indépendance des états initiaux))

Quand Ψ est fatale sous invariance de Φ pour $\langle S,A,\Sigma\langle S,A,t,\varepsilon\rangle\rangle$ et $\underset{\sim}{\text{Isind}}\langle S,A,t,\varepsilon,\Phi,\Psi\rangle$ est vrai, nous définissons :

$$\underset{\sim}{\text{rkgnd}}\langle S,A,t,\varepsilon,\Phi,\Psi\rangle = \underset{\sim}{\text{rk}}(\underset{\sim}{\text{Acc}}\langle S,A,t,\varepsilon,\Phi,\Psi\rangle, t1\,\underset{\sim}{\text{Inter}}\langle S,A,t,\varepsilon,\Phi,\Psi\rangle^{-1})$$

Observer que si le nondéterminisme est localement fini alors $\underset{\sim}{\text{rkgnd}}\langle S,A,t,\varepsilon,\Phi,\Psi\rangle \leq \omega$. De la même manière, si le nondéterminisme est localement dénombrable alors $\underset{\sim}{\text{rkgnd}}\langle S,A,t,\varepsilon,\Phi,\Psi\rangle \leq \omega_1$. Finalement, si t est récursive (i.e. effectivement calculable) alors $\underset{\sim}{\text{rkgnd}}\langle S,A,t,\varepsilon,\Phi,\Psi\rangle \leq \omega_1^{CK}$ (où ω_1^{CK}

est le premier ordinal non-récursif de Church-Kleene (Apt-Plotkin [82]).

Nous pouvons maintenant établir le résultat de complétude partielle concernant la méthode de Burstall :

Théorème 5.3.1.6.3 ~ 4 (Complétude sémantique partielle)

$$[(\Psi \text{ pour } \langle S,A,\Sigma\langle S,A,t,\Phi\rangle\rangle) \wedge \underline{Ind}\langle S,A,t,\Phi,\Pi,\Psi\rangle] \Rightarrow [(\mathcal{B}_1) \text{ avec } f \in (\Lambda \to (S \to \underline{rkgnd}\langle S,A,t,\Phi,\Pi,\Psi\rangle))]$$

Démonstration

Supposons que Ψ est fatale pour $\langle S,A,\Sigma\langle S,A,t,\Phi\rangle\rangle$ et $\underline{Ind}\langle S,A,t,\Phi,\Pi,\Psi\rangle$. Choisissons $\Lambda = 2$, $\varepsilon_0(s) = [s \in \underline{Acc}\langle S,A,t,\Phi,\Pi,\Psi\rangle]$, $\neg\theta_0(s,s') = [\exists p \in \Sigma\langle S,A,t,\Phi\rangle, i \in |p|.\ (\forall j<i.\ \neg\Psi(p_0,p_j)) \wedge \Psi(p_0,p_i) \wedge (\exists k \leq i.\ p_k = s) \wedge p_i = s']$, $f_0 \in (\underline{Acc}\langle S,A,t,\Phi,\Pi,\Psi\rangle \to \underline{rkgnd}\langle S,A,t,\Phi,\Pi,\Psi\rangle)$, $f_0(s) = \underline{rk}(\underline{Acc}\langle S,A,t,\Phi,\Pi,\Psi\rangle, t^{-1}\underline{Inter}\langle S,A,t,\Phi,\Pi,\Psi\rangle^{-1})$, $n_0 = 2$, $I_0^2(s,s') = [\varepsilon_0(s) \wedge s' = s]$, $I_0^1(s,s') = [\varepsilon_0(s) \wedge \neg\theta_0(s,s) \wedge \exists a \in A.\ t_a(s,s')]$, $I_0^0 = \theta_0$, $\varepsilon_1 = \Phi$, $\theta_1 = \Psi$, $n_1 = 1$, $I_1^1(s,s') = [\varepsilon_1(s) \wedge s' = s]$, $I_1^0 = \theta_1$, $\pi = 1$. Toutes les conditions de vérification sont trivialement satisfaites sauf pour $\forall s,s' \in S.((I_0^1(s,s') \wedge \neg\theta_0(s,s')) \Rightarrow (f_0(s') < f_0(s) \wedge \varepsilon_0(s') \wedge \forall s'' \in S.\ \theta_0(s',s'') \Rightarrow I_0^0(s,s'')))$.

Si $I_0^1(s,s') \wedge \neg\theta_0(s,s')$ est vrai, nous avons par définition de I_0^1, ε_0 et la fatalité de Ψ que $\exists p \in \Sigma\langle S,A,t,\Phi\rangle, i \in |p|.[(\forall j<i.\ \neg\Psi(p_0,p_j)) \wedge \Psi(p_0,p_i) \wedge \exists k.(s = p_k \wedge (k+1) < i \wedge s' = p_{k+1})]$. Puisque $s,s' \in \underline{Inter}\langle S,A,t,\Phi,\Pi,\Psi\rangle$ et $t_{p_k}(s,s')$, nous avons $f_0(s') < f_0(s) \wedge \varepsilon_0(s')$. Si $\theta_0(s',s'')$ alors $\exists q \in \Sigma\langle S,A,t,\Phi,\Pi,\Psi\rangle, i' \in |q|.[(\forall j<i'.\ \neg\Psi(p_0,p_j)) \wedge \Psi(p_0,p_{i'}) \wedge \exists k' \leq i'.(q_{k'} = s') \wedge q_{i'} = s'']$. Nous avons $\forall j.((k' < j < i') \Rightarrow \neg\Psi(p_0,q_j))$ car sinon pour le plus petit j satisfaisant $(k' < j < i') \wedge \Psi(p_0,q_j)$ nous aurions $\underline{Inter}\langle S,A,t,\Phi,\Pi,\Psi\rangle \wedge \underline{Goal}\langle S,A,t,\Phi,\Pi,\Psi\rangle$. Observer que $q_{i'} \in \underline{Goal}\langle S,A,t,\Phi,\Pi,\Psi\rangle$ de sorte que $\Psi(p_0,q_{i'})$ est vrai car sinon $q_{i'}$ serait un état intermédiaire de la trace $p_0,\ldots,p_k,q_{k'},\ldots,q_{i'},\ldots$. Puisque $s = p_k$ et $s'' = q_{i'}$, nous concluons que $\theta_0(s,s'')$ et donc $I_0^0(s,s'')$ est vrai.

□

Le résultat de complétude sémantique partielle s'applique aux propriétés de fatalité telles que les états "buts" n'ont pas d'états successeurs :

Théorème 5.3.1.6.3∨5

$$[(\Psi \text{ est fatale pour } \langle S,A,\Sigma\langle S,A,t,\Phi\rangle\rangle) \wedge \forall s\in S.(Goal\langle S,A,t,\Phi,\Vdash,\Psi\rangle(s) \Rightarrow \forall s'\in S, a\in A.\neg t_a(s,s'))] \Rightarrow Isind\langle S,A,t,\Phi,\Vdash,\Psi\rangle$$

Démonstration

Supposons $s\in Goal\langle S,A,t,\Phi,\Vdash,\Psi\rangle$. Nous avons $\forall s'\in S, a\in A.\neg t_a(s,s')$. Il s'ensuit que $s\notin Inter\langle S,A,t,\Phi,\Vdash,\Psi\rangle$ car sinon il existerait $n\in(\omega\vee 0)$, $p\in\Sigma^n\langle S,A,t,\Phi\rangle$ tels que $\forall i\leq n.\neg\Psi(p_0,p_i)$, en contradiction avec l'hypothèse de fatalité de Ψ pour $\langle S,A,\Sigma\langle S,A,t,\Phi\rangle\rangle$. □

Comme corollaire, nous obtenons que la méthode de preuve de correction totale de Burstall [74] pour les programmes séquentiels (i.e. (Φ_1) avec $f\in(\Lambda\to(S\to\omega))$) est sémantiquement complète parceque les états de sortie n'ont pas de successeurs et que les programmes considérés sont déterministes.

Le théorème 5.3.1.6.3∨4 s'applique également à Pnueli [77] et Apt-Delporte [83] parce qu'ils considèrent seulement des assertions intermittentes unaires (i.e. les assertions intermittentes relationnelles sont exprimées en utilisant des variables auxiliaires dont les valeurs font partie de l'état) :

Théorème 5.3.1.6.3∨6

$$\forall \Delta, \Delta' \in S.\ [\Psi(\Delta,\Delta') \Rightarrow (\forall \Delta'' \in S.\ \Psi(\Delta'',\Delta'))] \Longrightarrow \underline{\text{Ind}}\langle S, A, t, \Phi, \mathcal{H}, \Psi\rangle$$

Démonstration

Si $\Delta \in (\underline{\text{Inter}}\langle S, A, t, \Phi, \mathcal{H}, \Psi\rangle \cap \underline{\text{Goal}}\langle S, A, t, \Phi, \mathcal{H}, \Psi\rangle)$, alors il existe $\Delta', \Delta'' \in S$ tels que $\neg\Psi(\Delta', \Delta)$ et $\Psi(\Delta'', \Delta)$, une contradiction.

□

5.3.2 LE PRINCIPE D'INDUCTION DE BASE GENERALISANT LA METHODE DES ASSERTIONS INTERMITTENTES DE BURSTALL

Bien que le principe d'induction ($\mathcal{B}_1$) soit correct et sémantiquement complet dans un grand nombre de situations pratiques, nous faisons la conjecture qu'il n'est pas suffisamment général pour traiter certains types de propriétés de fatalité des programmes, comme celles considérées dans Manna-Waldinger [78] pour des programmes cycliques. D'où la nécessité de généraliser le principe d'induction ($\mathcal{B}_1$).

La généralisation proposée est tout à fait simple. Pour garantir l'existence des bons-ordres à utiliser dans l'induction, les lemmes et les assertions intermittentes doivent dépendre des états initiaux. Pour traiter le non déterminisme infini des bons-ordres transfinis doivent être utilisés. Ces remarques nous conduisent de ($\mathcal{B}_1$) à ($\mathcal{B}_2$) et nous démontrerons plus tard que ($\mathcal{B}_2$) est sémantiquement complet.

$$
\begin{array}{l}
[\exists \Lambda \in \omega,\ \varepsilon \in (\Lambda \to (S^2 \to \{tt,ff\})),\ \theta \in (\Lambda \to (S^3 \to \{tt,ff\})),\ \Delta \in \underset{\sim}{Ord},\ f \in (\Lambda \to (S^2 \to \Delta)),\\
\quad n \in (\Lambda \to \omega).\\
\\
\quad (\exists \pi \in \Lambda.\ \forall \underline{s}, s, s' \in S.\ (\varepsilon_\pi(\underline{s},s) = [s = \underline{s} \wedge \Phi(\underline{s})] \wedge \theta_\pi(\underline{s},s,s') = [\underline{s} = s \wedge \Psi(s,s')]))\\
\quad \wedge\\
\quad (\forall \ell \in \Lambda.\ \exists I_\ell \in (n_\ell + 1 \to (S^3 \to \{tt,ff\})).\\
\qquad \forall i \le n_\ell,\ \underline{s}, s, s' \in S.\\
\\
\text{(P)} \qquad [\varepsilon_\ell(\underline{s},s) \Rightarrow I_\ell^{n_\ell}(\underline{s},s,s)]\\
\qquad \wedge\\
\qquad [I_\ell^i(\underline{s},s,s') \Rightarrow\\
\\
\text{(HS)} \qquad\quad (\exists s'' \in S, a \in A.\ t_a(s',s'') \wedge \forall s'' \in S, a \in A.\ [t_a(s',s'') \Rightarrow \exists j < i.\ I_\ell^j(\underline{s},s,s'')])\\
\qquad\quad \vee\\
\text{(LI)} \qquad\quad (\exists \ell' \in \Lambda.\ [((\ell' < \ell) \vee (\ell' = \ell \wedge f_\ell(\underline{s},s') < f_\ell(\underline{s},s))) \wedge \varepsilon_{\ell'}(\underline{s},s') \wedge\\
\qquad\qquad\qquad \forall s'' \in S.\ (\theta_{\ell'}(\underline{s},s',s'') \Rightarrow \exists j < i.\ I_\ell^j(\underline{s},s,s'')]\\
\qquad\quad \vee\\
\text{(C)} \qquad\quad \theta_\ell(\underline{s},s,s')])]
\end{array}
\qquad (\mathcal{B}_2)
$$

Pour illustrer l'utilisation de ce principe d'induction, considérons l'exemple suivant :

Exemple 5.3.2-1

$\Psi(x,x') = [x' = 2x]$ est fatale pour $\langle \omega, \{2\}, \Sigma_{\langle \omega,\{2\}}, t, \Phi\rangle\rangle$ telle que $t_2(x,x') = [x' = x+1]$ et $\Phi(x) = \mathbb{N}$.

Observer que nous n'avons pas $\omega f(Acc_{\langle S,A,t,\Phi,\mathbb{N},\Psi\rangle}, t1Inter^{-1}_{\langle S,A,t,\Phi,\mathbb{N},\Psi\rangle})$.

La fatalité de Ψ peut être démontrée par le principe d'induction $(\mathcal{B}_2)$ en choisissant $\Lambda = 2$, $\pi = 1$, $\varepsilon_0(\underline{x},x) = [\underline{x} \leqslant x \leqslant 2\underline{x}]$, $\theta_0(\underline{x},x,x') = [\underline{x} \leqslant x \leqslant 2\underline{x} = x']$, $\Delta = \omega$, $f_0(\underline{x},x) = [2\underline{x} - x]$, $\varepsilon_1(\underline{x},x) = [\underline{x},x]$, $\theta_1(\underline{x},x,x') = [\underline{x} = x \wedge x' = 2x]$, $n_0 = 2$, $I_0^2(\underline{x},x,x') = [\underline{x} \leqslant x = x' \leqslant 2\underline{x}]$ (satisfaisant (P) et (C) quand $x' = 2\underline{x}$ ou (HS) quand $x' < 2x$), $I_0^1(\underline{x},x,x') = [\underline{x} \leqslant x < x+1 = x' \leqslant 2\underline{x}]$ (satisfaisant (LI) avec $\ell' = \ell = 0$), $I_0^0 = \theta_0$ (C), $n_1 = 1$, $I_1^1(\underline{x},x,x') = [\underline{x} = x = x']$ ((P), (LI) avec $\ell' = 0$), $I_1^0 = \theta_1$ (C).

□

Le principe d'induction $(\mathcal{B}_2)$ est une généralisation évidente de $(\mathcal{B}_1)$:

Théorème 5.3.2 v1 (Généralisation de la méthode de Burstall)

$(\mathcal{B}_1)$ $(\mathcal{B}_2)$

Avant d'aborder la question de la complétude sémantique, nous définissons quels ordinaux $\Delta \in Ord$ sont suffisants dans une preuve par $(\mathcal{B}_2)$

Définition 5.3.2 :1 (Rang du nondéterminisme global (cas général))

Lorsque Ψ est fatale sous invariance de Φ pour $\langle S,A,\Sigma_{\langle S,A,t,\varepsilon\rangle\rangle}$, définissons :

$$rkgnd_{\langle S,A,t,\varepsilon,\Phi,\Psi\rangle} = \sup^+ \{ rk(Acc_{\langle S,A,t,\varepsilon,\Phi,\Psi\rangle}(\underline{s}), t1Inter_{\langle S,A,t,\varepsilon,\Phi,\Psi\rangle}(\underline{s})^{-1}) : \underline{s} \in S \}$$

(Cette définition se justifie par le fait que pour tout $\underline{a} \in S$, $t1\,\mathrm{Inter}\langle S,A,t,\varepsilon,\Phi,\Psi\rangle^{-1}(\underline{a})$ est bien-fondée sur $\mathrm{Acc}\langle S,A,t,\varepsilon,\Phi,\Psi\rangle$, (cf. démonstration du théorème 5.2.6.2-1)).

La preuve de la complétude sémantique de $(\mathcal{B}_2)$ vient de la remarque que $(\mathcal{B}_2)$ peut être utilisé pour exprimer les preuves "à la Floyd" :

Théorème 5.3.2.2 (Complétude sémantique)

$(\Psi$ est fatale pour $\langle S,A,\Sigma\langle S,A,t,\Phi\rangle\rangle) \Longrightarrow$ ($(\mathcal{B}_2)$ avec $\Delta = \mathrm{rkgnd}\langle S,A,t,\Phi,\mathrm{tt},\Psi\rangle$)

Démonstration

Choisir $\Lambda = 2$, $\pi = 1$, $\varepsilon_1(\underline{a},\Delta) = [\underline{a} = \Delta \wedge \Phi(\underline{a})]$, $\theta_1(\underline{a},\Delta,\Delta') = [\underline{a} = \Delta \wedge \Psi(\Delta,\Delta')]$, $\varepsilon_0(\underline{a},\Delta) = [\Delta \in \mathrm{Acc}\langle S,A,t,\Phi,\mathrm{tt},\Psi\rangle(\underline{a})]$, $\theta_0(\underline{a},\Delta,\Delta') = [\exists p \in \Sigma\langle S,A,t,\Phi\rangle, i \in |p|.\ (\forall j \in i.\ \neg\Psi(p_0,p_j)) \wedge \Psi(p_0,p_i) \wedge \underline{a} = p_0 \wedge \exists k \leq i.\ p_k = \Delta \wedge p_i = \Delta')]$, f_1 est inutile, $m_1 = 1$, $I_1^1(\underline{a},\Delta,\Delta') = [\underline{a} = \Delta = \Delta' \wedge \Phi(\underline{a})]$ (satisfait (P) et (LI) avec $\ell' = 0$), $I_1^1 = \theta_1$, $m_0 = 2$, $I_0^2(\underline{a},\Delta,\Delta') = [\varepsilon_0(\underline{a},\Delta) \wedge \Delta' = \Delta]$ (satisfait (P) et (C) ou (HS)), $I_0^1(\underline{a},\Delta,\Delta') = [\varepsilon_0(\underline{a},\Delta) \wedge \neg\theta_0(\underline{a},\Delta,\Delta) \wedge \exists a \in A.\ t_a(\Delta,\Delta')]$ (satisfait (LI) avec $\ell' = 0$), $f_0(\underline{a},\Delta) = \mathrm{rk}(\mathrm{Acc}\langle S,A,t,\Phi,\mathrm{tt},\Psi\rangle(\underline{a}), t1\,\mathrm{Inter}\langle S,A,t,\Phi,\mathrm{tt},\Psi\rangle(\underline{a})^{-1})(\Delta)$ et $I_0^0 = \theta_0$ (satisfait (C)).

□

5.3.3 PRINCIPES D'INDUCTION EQUIVALENTS GENERALISANT LA METHODE DES ASSERTIONS INTERMITTENTES DE BURSTALL

Nous dérivons série de principes d'induction dont nous montrons la correcti ude sémantique donc l'équivalence au principe d'induction $(\mathcal{B}_2)$. Pour être concis, nous ne reportons pas ici toutes les alternatives concevables. En particulier, les transformations présentées en 4.2.1.2 et 5.2.3 ne seront pas répétées. Un but de la série de principes d'induction est de proposer des formalisations de plus en plus abstraites qui devraient permettre une meilleure compréhension de la méthode de Burstall. L'autre but est d'augmenter le nombre de formes de preuves permises (de manière à introduire plus de souplesse dans l'écriture des preuves, mais pas de puissance supplémentaire puisque tous ces principes d'induction sont équivalents).

Le nombre de lemmes $\langle \varepsilon_\ell, \theta_\ell \rangle$, $\ell \in \Lambda$ qui peuvent être utilisés dans le principe d'induction $(\mathcal{B}_2)$ est fini. Aussi une proposition informelle telle que

"if sometime $(\underline{x} \leq X = x \leq 2\underline{x})$ then sometime $(\underline{x} \leq x \leq 2\underline{x} = X)$"

doit être comprise comme un seul lemme de nom 0 par exemple et tel que $\varepsilon_0(\underline{x}, x) = [\underline{x} \leq x \leq 2\underline{x}]$ et $\theta_0(\underline{x}, x, x') = [\underline{x} \leq x \leq 2\underline{x} = x']$. Si nous éliminions cette restriction sur les noms des lemmes, la proposition informelle ci-dessus peut aussi être comprise comme une représentation d'un nombre infini de lemmes de noms $\underline{x}$ tels que $\varepsilon_{\underline{x}}(x) = [\underline{x} \leq x \leq 2\underline{x}]$ et $\theta_{\underline{x}}(x, x') = [\underline{x} \leq x \leq 2\underline{x} = x']$. Ce point de vue est compatible avec le fait que le seul but de l'état initial $\leq$ du programme dans le principe d'induction $(\mathcal{B}_2)$ est d'offrir la possibilité d'utiliser des bons-ordres pour l'induction sur les données qui dépendent des états initiaux du programme. Ces bons-ordres peuvent également être distingués en leur donnant des noms différents, un par état initial du programme.

Aussi, la proposition principale $\langle \Phi, \Psi \rangle$ n'a pas besoin d'être la conséquence d'un seul lemme $\langle \varepsilon_\pi, \theta_\pi \rangle$ comme dans $(\mathcal{B}_2)$ mais peut aussi être la conséquence de différents lemmes pour différents états initiaux du programme. Ces remarques conduisent au principe d'induction :

$$
\begin{aligned}
&[\exists \Lambda \in \underline{\mathrm{Ord}},\ \varepsilon \in (\Lambda \to (S \to \{tt, ff\})),\ \theta \in (\Lambda \to (S^2 \to \{tt, ff\})),\ \Delta \in \underline{\mathrm{Ord}},\ f \in (\Lambda \to (S \to \Delta)),\\
&\ n \in (\Lambda \to \omega).\\
&\quad \forall \Delta \in S.\ \exists \pi \in \Lambda.\ (\varepsilon_\pi(\Delta) = \Phi(\Delta) \wedge \forall \Delta' \in S.\ (\theta_\pi(\Delta, \Delta') = \Psi(\Delta, \Delta')))\\
&\quad \wedge (\forall \alpha \in \Lambda.\ \exists I_\alpha \in (n_\alpha + 1 \to (S^2 \to \{tt, ff\})).\\
&\qquad \forall i \leq n_\alpha,\ \Delta, \Delta' \in S.\\
&\qquad\quad [\varepsilon_\alpha(\Delta) \Rightarrow I_\alpha^{n_\alpha}(\Delta, \Delta)]\\
&\qquad\quad \wedge [I_\alpha^{i}(\Delta, \Delta') \Rightarrow\\
&\qquad\qquad (\exists \Delta'' \in S, a \in A.\ t_a(\Delta', \Delta'') \wedge \forall \Delta'' \in S, a \in A.\ [t_a(\Delta', \Delta'') \Rightarrow \exists j < i.\ I_\alpha^j(\Delta, \Delta'')])\\
&\qquad\qquad (\exists \alpha' \in \Lambda.\ [((\alpha' < \alpha) \vee (\alpha' = \alpha \wedge f_\alpha(\Delta') < f_\alpha(\Delta))) \wedge \varepsilon_{\alpha'}(\Delta') \wedge\\
&\qquad\qquad\qquad \forall \Delta'' \in S.\ (\theta_{\alpha'}(\Delta', \Delta'') \Rightarrow \exists j < i.\ I_\alpha^j(\Delta, \Delta''))]\\
&\qquad\qquad \theta_\alpha(\Delta, \Delta')])]
\end{aligned}
\tag{$\mathcal{B}_3$}
$$

Théorème 5.3.3 n°1

$$(\mathcal{B}_2) \Rightarrow (\mathcal{B}_3)$$

Démonstration

D'après l'axiome du choix, il existe un ordinal Σ et une fonction injective δ de Σ dans $S \cdot \Sigma^{\times} \Lambda_2$, bien-ordonné par l'ordre lexicographique $\langle \Delta', \ell' \rangle \prec \langle \Delta, \ell \rangle$ si et seulement si $((\Delta' < \Delta) \vee (\Delta' = \Delta \wedge \ell' < \ell))$, est isomorphe à $\Lambda_2 \dot{\times} \Sigma$, ($\dot{\times}$ est ici la multiplication d'ordinaux) par l'isomorphisme d'ordre $\underline{\iota}(\langle \Delta, \ell \rangle) = [(\Lambda_2 \dot{\times} \Delta) \dot{+} \ell]$, ($\dot{+}$ est ici l'addition d'ordinaux). Soit $\langle \underline{\xi}, \underline{\Delta} \rangle$ l'inverse de $\underline{\iota}$

de sorte que $\underline{\xi} \in ((\Lambda_2 \dot{\times} \Sigma) \to \Sigma)$, $\underline{\lambda} \in ((\Lambda_2 \dot{\times} \Sigma) \to \Lambda_2)$ et $\forall \alpha \in (\Lambda_2 \dot{\times} \Sigma).[\alpha = \underline{\iota}(\langle \underline{\xi}(\alpha), \underline{\lambda}(\alpha)\rangle)]$. Nous choisissons $\Lambda_3 = \Lambda_2 \dot{\times} \Sigma$, $\varepsilon_{3_\alpha}(\Delta) = [\varepsilon_{2_{\underline{\lambda}(\alpha)}}(\delta(\underline{\xi}(\alpha)), \Delta)]$, $\theta_{3_\alpha}(\Delta, \Delta') = [\theta_{2_{\underline{\lambda}(\alpha)}}(\delta(\underline{\xi}(\alpha)), \Delta, \Delta')]$, $\Delta_3 = \Delta_2$, $f_{3_\alpha}(\Delta) = [f_{2_{\underline{\lambda}(\alpha)}}(\delta(\underline{\xi}(\alpha)), \Delta)]$, $m_{3_\alpha} = m_{2_{\underline{\lambda}(\alpha)}}$, $I_{3_\alpha}^{i}(\Delta, \Delta') = I_{2_{\underline{\lambda}(\alpha)}}^{i}(\delta(\underline{\xi}(\alpha)), \Delta, \Delta')$. Il s'ensuit que $\varepsilon_{3_{\underline{\iota}(\langle \delta^{-1}(\Delta), \pi_2\rangle)}}(\Delta) = \Phi(\Delta)$ et $\theta_{3_{\underline{\iota}(\langle \delta^{-1}(\Delta), \pi_2\rangle)}}(\Delta, \Delta') = \Psi(\Delta, \Delta')$. Les autres conditions de vérification sont évidentes à contrôler.

□

Les noms $\alpha \in \Lambda$ des lemmes $\langle \varepsilon_\alpha, \theta_\alpha \rangle$ dans $(\mathcal{B}_3)$ sont bien-ordonnés. Pour un lemme donné $\langle \varepsilon_\alpha, \theta_\alpha \rangle$ le rôle de f_α est d'introduire un bon-ordre sur les états initiaux du lemme $\langle \varepsilon_\alpha, \theta_\alpha \rangle$. Le même effet peut être obtenu en considérant non pas un seul lemme $\langle \varepsilon_\alpha, \theta_\alpha \rangle$ mais une famille de lemmes $\{\langle \varepsilon_{\alpha, f_\alpha(\Delta)}, \theta_{\alpha, f_\alpha(\Delta)} \rangle : \Delta \in S\}$. Ce point de vue est plus abstrait du fait que nous n'avons besoin que d'un seul bon-ordre $\langle W, \prec \rangle$. Il est défini par $\langle \alpha', f_{\alpha'}(\Delta) \rangle \prec \langle \alpha, f_\alpha(\Delta) \rangle$ si et seulement si $(\alpha' < \alpha \vee (\alpha' = \alpha \wedge f_\alpha(\Delta') < f_\alpha(\Delta)))$ sur $W = \{\langle \alpha, f_\alpha(\Delta)\rangle : \alpha \in \Lambda \wedge \Delta \in S\}$. Aussi, à un isomorphisme près, nous pouvons utiliser des ordinaux et reformuler le principe d'induction $(\mathcal{B}_3)$ comme suit :

$$
\begin{aligned}
&[\exists \Lambda \in \underset{\sim}{Ord},\ \varepsilon \in (\Lambda \to (S \to \{tt, ff\})),\ \theta \in (\Lambda \to (S^2 \to \{tt, ff\})),\ n \in (\Lambda \to \omega).\\
&\quad (\forall \Delta \in S.\, \exists \pi \in \Lambda.\, [(\varepsilon_\pi(\Delta) = \Phi(\Delta)) \wedge \forall \Delta' \in S.\, (\theta_\pi(\Delta, \Delta') = \Psi(\Delta, \Delta'))])\\
&\quad \wedge (\forall \lambda \in \Lambda.\, \exists I_\lambda \in (m_\lambda + 1 \to (S^2 \to \{tt, ff\})).\\
&\qquad \forall i \in m_\lambda,\ \Delta, \Delta' \in S.\\
&\qquad\quad [\varepsilon_\lambda(\Delta) \Rightarrow I_\lambda^{m_\lambda}(\Delta, \Delta)]\\
&\qquad\quad \wedge [I_\lambda^{i}(\Delta, \Delta') \Longrightarrow\\
&\qquad\qquad (\exists \Delta'' \in S, a \in A.\, t_a(\Delta', \Delta'') \wedge \forall \Delta'' \in S, a \in A.\, [t_a(\Delta', \Delta'') \Rightarrow \exists j < i.\, I_\lambda^{j}(\Delta, \Delta'')])\\
&\qquad\qquad \vee (\exists \lambda' < \lambda.\, [\varepsilon_{\lambda'}(\Delta') \wedge \forall \Delta'' \in S.\, (\theta_{\lambda'}(\Delta', \Delta') \Rightarrow \exists j < i.\, I_\lambda^{j}(\Delta, \Delta''))])\\
&\qquad\qquad \vee \theta_\lambda(\Delta, \Delta')])]
\end{aligned}
\qquad (\mathcal{B}_4)
$$

Théorème 5.3.3 v 2

$(\mathcal{B}_3) \Rightarrow (\mathcal{B}_4)$

Démonstration

$\langle \Lambda_3, \Delta_3 \rangle$ bien-ordonné par l'ordre lexicographique gauche est isomorphe à $\Delta_3 \dot{\times} \Lambda_3$ par l'isomorphisme d'ordre $\underline{\iota}(\lambda,\kappa) = [(\Delta_3 \dot{\times} \lambda) + \kappa]$ dont l'inverse est $\langle \underline{\delta}, \underline{\lambda} \rangle$. Nous choisissons $\Lambda_4 = \Delta_3 \dot{\times} \Lambda_3$, $\varepsilon_{4_\lambda}(\Delta) = [\varepsilon_{3_{\underline{\lambda}(\lambda)}}(\Delta) \wedge f^3_{\underline{\lambda}(\lambda)}(\Delta) = \underline{\delta}(\lambda)]$, $\theta_{4_\lambda}(\Delta,\Delta') =$ $[\theta_{3_{\underline{\lambda}(\lambda)}}(\Delta,\Delta') \wedge f^3_{\underline{\lambda}(\lambda)}(\Delta) = \underline{\delta}(\lambda)]$, $m_{4_\lambda} = m_{3_{\underline{\lambda}(\lambda)}}$, $I^i_{4_\lambda}(\Delta,\Delta') = [I^i_{3_{\underline{\lambda}(\lambda)}}(\Delta,\Delta') \wedge f^3_{\underline{\lambda}(\lambda)}(\Delta) = \underline{\delta}(\lambda)]$.

□

Les propriétés de fatalité des programmes ont été spécifiées par des paires $\langle \Phi, \Psi \rangle$ où Φ est une condition sur les états initiaux et Ψ une relation entre états finaux et initiaux comme dans la méthode de Burstall [74]. Cependant, une seule relation binaire suffit car Ψ est fatale pour $\langle S, A, \Sigma \langle S, A, t, \Phi \rangle\rangle$ si et seulement si $\Psi'(\Delta,\Delta') = [\Phi(\Delta) \Rightarrow \Psi(\Delta,\Delta')]$ est fatale pour $\langle S, A, \Sigma \langle S, A, t, tt \rangle\rangle$. Aussi, nous dérivons le principe d'induction plus abstrait :

$[\exists \Lambda \in \underline{Ord},\ \theta \in (\Lambda \to (S^2 \to \{tt, ff\})),\ m \in (\Lambda \to \omega).$

$(\forall \Delta \in S.\ \exists \pi \in \Lambda.\ \forall \Delta' \in S.\ [\theta_\pi(\Delta,\Delta') = (\Phi(\Delta) \Rightarrow \Psi(\Delta,\Delta'))])$

$\wedge$ $(\forall \lambda \in \Lambda.\ \exists I_\lambda \in (m_\lambda + 1 \to (S^2 \to \{tt, ff\})).$

$\forall i \leq m_\lambda,\ \Delta, \Delta' \in S.$

(P) $\quad I^{m_\lambda}_\lambda(\Delta,\Delta)$

$\wedge$ $[I^i_\lambda(\Delta,\Delta') \Rightarrow$

(HS) $\quad (\exists \Delta'' \in S, a \in A.\ t_a(\Delta',\Delta'') \wedge \forall \Delta'' \in S, a \in A.\ [t_a(\Delta',\Delta'') \Rightarrow \exists j < i.\ I^j_\lambda(\Delta,\Delta'')])$

(LI) $\quad \vee$ $(\exists \lambda' < \lambda.\ \forall \Delta'' \in S.\ [\theta_{\lambda'}(\Delta',\Delta'') \Rightarrow \exists j < i.\ I^j_\lambda(\Delta,\Delta'')])$

(C) $\quad \vee$ $\theta_\lambda(\Delta,\Delta')]]$ $\qquad (\mathcal{B}_5)$

Théorème 5.3.3 ν3

$(\mathcal{B}_4) \Rightarrow (\mathcal{B}_5)$

Démonstration

Choisir $\Lambda_5 = \Lambda_4$, $\theta_{5_\lambda}(s,s') = [\varepsilon_{4_\lambda}(s) \Rightarrow \theta_{4_\lambda}(s,s')]$, $m_5 = m_4$ et $I_{5_\lambda}^{i}(s,s') = [\varepsilon_{4_\lambda}(s) \Rightarrow I_{4_\lambda}^{i}(s,s')]$.

□

Si dans le principe d'induction $(\mathcal{B}_5)$, nous considérons la preuve d'un lemme θ_λ donné et que cette preuve peut être faite sans (LI), alors les conditions de vérification (P), (HS) et (C) ressemblent fortement aux conditions de vérification de la méthode de preuve de Floyd [67] telle qu'elle est formalisée par le principe d'induction $(\mathcal{F}_5)$ dans le paragraphe 5.2.3. Autrement dit, dans l'hypothèse d'induction $I_\lambda^i(s,s')$, i joue le rôle d'un entier non-négatif qui décroît strictement à chaque pas du programme. Par comparaison avec la méthode de Floyd, nous observons que $(\mathcal{B}_5)$ impose deux restrictions inutiles sur i : m_λ est une borne du nombre de pas du programme et ce nombre est indépendant de l'état initial considéré, m_λ et donc i doit être un entier (de sorte que, par exemple, le nondéterminisme infini ne puisse être traité sans (LI)).

Nous supprimons d'abord la première restriction, en choisissant m_λ comme étant un ordinal :

Théorème 5.3.3 ν4

(a) $[(\mathcal{B}_i) \Rightarrow ((\mathcal{B}_i) \text{ avec } m \in (\Lambda \to Ord))]$, $i = 2,\dots,5$

(b) $[((\mathcal{B}_i) \text{ avec } m \in (\Lambda \to \underset{\sim}{Ord})) \Rightarrow ((\mathcal{B}_{i+1}) \text{ avec } m \in (\Lambda \to \underset{\sim}{Ord}))]$, $i = 2,3,4$

Démonstration

(a) est trivial parceque $\omega \in \underset{\sim}{Ord}$ donc $\omega \subseteq \underset{\sim}{Ord}$.

(b) résulte des preuves des théorèmes 5.3.3ν1, 5.3.3ν2 et 5.3.3ν3 qui n'utilisent jamais le fait que $n_\ell \in \omega$ mais seulement le fait que $\langle n_{\ell+1}, < \rangle$ est bien-fondé (ceci demeure vrai quand $n_\ell \in \underset{\sim}{Ord}$ et $n_{\ell+1}$ est l'ordinal successeur de n_ℓ).

□

Nous supprimons ensuite la seconde restriction, en choisissant un "nombre maximum de pas du programme" qui peut être différent pour chaque état initial $\underline{s}$. (Le "nombre de pas du programme" ne doit pas être interprété à la lettre mais comme $\underset{\sim}{rk}(\underset{\sim}{Acc}\langle S,A,t,\Phi,\Psi,\Psi\rangle(\underline{s}), t^{-1}\underset{\sim}{Inter}\langle S,A,t,\Phi,\Psi,\Psi\rangle(\underline{s})^{-1})(\underline{s}))$:

$$
\begin{array}{l}
[\exists \Lambda \in \underset{\sim}{Ord},\ \theta \in (\Lambda \to (S^2 \to \{tt,ff\})),\ \Delta \in \underset{\sim}{Ord},\ I \in (\Lambda \to (\Delta \times S \times S \to \{tt,ff\}))\ .\\
\quad (\forall s \in S.\ \exists \pi \in \Lambda.\ \forall s' \in S.\ [\theta_\pi(s,s') = (\Phi(s) \Rightarrow \Psi(s,s'))])\\
\quad \wedge (\forall \lambda \in \Lambda,\ s,s' \in S,\ \delta' \in \Delta.\\
\qquad [\exists \delta \in \Delta.\ I_\lambda(\delta,s,s)]\\
\qquad \wedge [I_\lambda(\delta',s,s') \Rightarrow\\
\qquad\quad (\exists s'' \in S, a \in A.\ t_a(s',s'') \wedge \forall s'' \in S, a \in A.\ [t_a(s',s'') \Rightarrow \exists \delta'' < \delta'.\ I_\lambda(\delta'',s,s'')])\\
\qquad\quad \vee (\exists \lambda' < \lambda.\ \forall s'' \in S.\ [\theta_{\lambda'}(s',s'') \Rightarrow \exists \delta'' < \delta'.\ I_\lambda(\delta'',s,s'')])\\
\qquad\quad \vee \theta_\lambda(s,s')])]
\end{array}
\qquad (\mathcal{B}_6)
$$

Théorème 5.3.3ν5

$$((\mathcal{B}_5) \text{ avec } n \in (\Lambda \to \underset{\sim}{Ord})) \Rightarrow (\mathcal{B}_6)$$

Démonstration

Choisir $\Lambda_6 = \Lambda_5$, $\theta_6 = \theta_5$, $\Delta_6 = \omega$, $I_{6\lambda}(\delta, A, A') = [\delta \in m_{5\lambda} \wedge I^{\delta}_{5\lambda}(A, A')]$.

□

Dans la suite nous utiliserons le plus souvent le principe d'induction suivant :

$[\exists \Lambda \in \mathrm{Ord},\ \theta \in (\Lambda \to (S \times S \to \{tt, ff\})),\ \pi \in \Lambda,\ \Delta \in \mathrm{Ord},\ I \in (\Lambda \to (\Delta \times S \times S \to \{tt, ff\}))$.

$(\mathcal{B}_7.1)$ $\quad \forall A, A' \in S.\ (\theta_\pi(A, A') = [\Phi(A) \Rightarrow \Psi(A, A')])$

$(\mathcal{B}_7.2)$ $\quad \wedge\ (\forall \lambda \in \Lambda.\ \forall A \in S.\ \exists \delta \in \Delta.\ I_\lambda(\delta, A, A))$

$(\mathcal{B}_7.3)$ $\quad \wedge\ (\forall \lambda \in \Lambda,\ A, A' \in S,\ \delta' \in \Delta.$ $\qquad (\mathcal{B}_7)$

$\qquad I_\lambda(\delta', A, A') \Rightarrow$

$(\mathcal{B}_7.3.a)$ $\quad [\ (\exists A'' \in S, a \in A.\ t_a(A', A'') \wedge \forall A'' \in S, a \in A.\ [t_a(A', A'') \Rightarrow \exists \delta'' < \delta'.\ I_\lambda(\delta'', A, A'')])$

$(\mathcal{B}_7.3.b)$ $\quad \vee\ (\exists \lambda' < \lambda.\ \forall A'' \in S.\ [\theta_{\lambda'}(A', A'') \Rightarrow \exists \delta'' < \delta'.\ I_{\lambda'}(\delta'', A, A'')])$

$(\mathcal{B}_7.3.c)$ $\quad \vee\ \theta_\lambda(A, A')]]$

Théorème 5.3.3 v 6

$(\mathcal{B}_6) \Rightarrow (\mathcal{B}_7)$

Démonstration

Choisir $\pi_7 = \Lambda_6$, $\Lambda_7 = (\Lambda_6 \cup \{\pi_7\}) = \Lambda_6 + 1$, $\theta_{7\lambda} = \theta_{6\lambda}$ quand $\lambda \in \Lambda_6$ et $\theta_{7\pi_7}(A, A') = [\Phi(A) \Rightarrow \Psi(A, A')]$, $\Delta_7 = (\Delta_6 \cup 2)$, $I_{7\lambda}(\delta', A, A') = I_{6\lambda}(\delta', A, A')$ quand $\lambda \in \Lambda_6$, $I_{7\pi_7}(\delta', A, A') = ff$ si $\delta' \geq 2$, $I_{7\pi_7}(1, A, A') = [A = A']$, $I_{7\pi_7}(0, A, A') = \theta_{7\pi_7}(A, A')$.

□

L'utilisation de bons-ordres (ou à des isomorphismes d'ordre près, d'ordinaux) dans $(\mathcal{B}_6)$ n'est pas obligatoire. Des relations bien-fondées peuvent aussi bien servir de base pour l'induction.

De plus, comme noté par Schwarz [77], la méthode de Burstall peut être expliquée comme la déduction mathématique de théorèmes à partir d'axiomes spécifiant l'effet des commandes élémentaires du programme. Cette explication informelle de la méthode de Burstall peut être formalisée en considérant la relation de transition dans les principes d'induction précédents comme un ensemble d'axiomes ou encore comme un lemme donné à partir duquel les autres lemmes dérivent. Une différence (qui n'est pas prise en compte par Schwarz [77] qui considère seulement des programmes totaux déterministes) est que la fatalité de t pour $\langle S, A, \Sigma\langle S, A, t, tt\rangle\rangle$ n'est vraie que pour les états qui ont au moins un successeur. De plus, le processus de déduction (que Schwarz [77] ne spécifie pas) est toujours réductible à l'induction transfinie.

Finalement, la proposition principale $\theta_\pi(s,s') = [\Phi(s) \Rightarrow \Psi(s,s')]$ peut toujours être choisie comme un des lemmes intervenant dans la preuve.

Ces remarques conduisent au principe d'induction suivant :

$$
\begin{aligned}
&[\exists \Lambda, \prec, \mu \in \Lambda, \pi \in (\Lambda \sim \mu), \Delta, <, \theta \in (\Lambda \to (S^2 \to \{tt, ff\})), I \in (\Lambda \to (\Delta \times S \times S \to \{tt, ff\})).\\
&\quad \underset{\sim}{wf}_{\sim}(\Lambda, \prec, \mu) \wedge \theta_\mu = (\exists a \in A.\, t_a) \wedge \underset{\sim}{wf}(\Delta, <) \wedge [\forall s, s' \in S.\, \theta_\pi(s,s') = [\Phi(s) \Rightarrow \Psi(s,s')]]\\
&\quad \wedge (\forall \lambda \in (\Lambda \sim \mu), s, s' \in S, \delta' \in \Delta.\\
&\qquad [\exists \delta \in \Delta.\, I_\lambda(\delta, s, s')]\\
&\qquad \wedge [I_\lambda(\delta', s, s') \Rightarrow\\
&\qquad\quad (\exists \lambda' \in \Lambda.\, [\lambda' \prec \lambda \wedge ([\lambda' = \mu] \Rightarrow [\exists s'' \in S.\, \theta_{\lambda'}(s', s'')]) \wedge\\
&\qquad\qquad \forall s'' \in S.\, (\theta_{\lambda'}(s', s'') \Rightarrow [\exists \delta'' \in \Delta.\, (\delta'' < \delta' \wedge I_\lambda(\delta'', s, s''))])])\\
&\qquad\quad \vee\\
&\qquad\quad \theta_\lambda(s, s')])]
\end{aligned}
\tag{$\mathcal{B}_8$}
$$

Remarquons que la condition $[\lambda' = \mu]$ sous laquelle $[\exists \Delta'' \in S.\ \theta_\lambda(\Delta', \Delta'')]$ devrait être vrai est optionnelle. Lorsqu'elle est absente, la condition de vérification est simplement redondante quand $\lambda' \neq \mu$.

Théorème 5.3.3.v7

$(\mathcal{B}_7) \Longrightarrow (\mathcal{B}_8)$

Démonstration

Choisir $\mu = \Lambda_7$, $\Lambda_8 = (\Lambda_7 \cup \{\mu\}) = (\Lambda_7 + 1)$, $\pi_8 = \pi_7$, $\Delta_8 = \Delta_7$, $\lambda' \prec_8 \lambda = [\lambda \in \Lambda_7 \wedge ((\lambda' = \mu) \vee (\lambda' \in \Lambda_7 \wedge \lambda' < \lambda))]$, $<_8 = <_7 = <$, $\theta_{8\lambda}(\Delta, \Delta') = [(\lambda = \mu \wedge \exists a \in A.\ t_a(\Delta, \Delta')) \vee (\lambda \in \Lambda_7 \wedge \theta_{7\lambda}(\Delta, \Delta'))]$, $I_{8\lambda}(\delta, \Delta, \Delta') = [(\lambda \in \Lambda_7 \wedge I_{7\lambda}(\delta, \Delta, \Delta')) \vee (\lambda = \mu)]$.

□

Dans le principe d'induction $(\mathcal{B}_8)$, la condition de vérification $[\exists \delta \in \Delta.\ I_\lambda(\delta, \Delta, \Delta)]$ implique que le lemme θ_λ est fatal pour $\langle S, A, \Sigma < S, A, t, \mathbb{t} \rangle$. Excepté pour la proposition principale θ_π, cette condition n'est pas nécessaire. Nous avons besoin seulement du fait que θ_λ doit être fatal pour les états particuliers pour lesquels il est utilisé. Aussi, la condition de vérification $[\exists \delta \in \Delta.\ I_\lambda(\delta, \Delta, \Delta)]$ de $(\mathcal{B}_8)$ peut être affaiblie dans :

$$[\,\exists \Lambda \in \underset{\sim}{Ord},\ \theta \in (\Lambda \to (S^2 \to \{tt, ff\})),\ \pi \in \Lambda,\ \Delta \in \underset{\sim}{Ord},\ I \in (\Lambda \to (\Delta \times S \times S \to \{tt, ff\}))\,.$$

$$(\forall A, A' \in S.\ (\theta_\pi(A,A') = [\Phi(A) \Rightarrow \Psi(A,A')]))$$

$$(\forall A \in S.\ \exists \delta \in \Delta.\ I_\pi(\delta, A, A))$$

$$(\forall \lambda \in \Lambda,\ A, A' \in S,\ \delta' \in \Delta. \qquad (\mathcal{B}_g)$$

$$[\,I_\lambda(\delta', A, A') \Rightarrow$$

$$(\exists A'' \in S, a \in A.\ t_a(A', A'') \wedge \forall A'' \in S, a \in A.\,[t_a(A', A'') \Rightarrow \exists \delta'' < \delta'.\, I_\lambda(\delta'', A, A'')])$$

$$\vee$$

$$(\exists \lambda' < \lambda.\,[\exists \delta \in \Delta.\ I_{\lambda'}(\delta, A', A') \wedge \forall A'' \in S.\,[\theta_{\lambda'}(A', A'') \Rightarrow \exists \delta'' < \delta'.\, I_\lambda(\delta'', A, A'')]])$$

$$\vee$$

$$\theta_\lambda(A, A')])]$$

Théorème 5.3.3 v8

Démonstration

Nous montrons d'abord que si $0 < \varepsilon_0 < \gamma$ et $0 < \varepsilon_1 < \gamma$ alors $((\gamma \times \delta_0) + \varepsilon_0) < ((\gamma \times \delta_1) + \varepsilon_1)$ si et seulement si $((\delta_0 < \delta_1) \vee (\delta_0 = \delta_1 \wedge \varepsilon_0 < \varepsilon_1))$.

Si $\delta_0 < \delta_1$ alors $((\gamma \times \delta_0) + \varepsilon_0) < ((\gamma \times \delta_0) + \gamma) = (\gamma \times (\delta_0 + 1)) \leq (\gamma \times \delta_1) < ((\gamma \times \delta_1) + \varepsilon_1)$. Si $(\delta_0 = \delta_1) \wedge (\varepsilon_0 < \varepsilon_1)$ alors $(\gamma \times \delta_0) = (\gamma \times \delta_1)$ et donc $((\gamma \times \delta_0) + \varepsilon_0) < ((\gamma \times \delta_1) + \varepsilon_1)$.

Si inversement $\neg((\delta_0 < \delta_1) \vee (\delta_0 = \delta_1 \wedge \varepsilon_0 < \varepsilon_1))$ alors soit $\delta_0 = \delta_1$ et $\varepsilon_0 = \varepsilon_1$ de sorte que $\alpha = ((\gamma + \delta_0) + \varepsilon_0) = ((\gamma \times \delta_1) + \varepsilon_1) = \beta$ et $\alpha \not< \beta$, ou bien $(\delta_0 > \delta_1) \vee (\delta_0 = \delta_1 \wedge \varepsilon_0 > \varepsilon_1)$ de sorte que d'après la première partie de la preuve (avec 0 et 1 interchangés) nous avons $\beta < \alpha$ donc $\alpha \not< \beta$.

Montrons maintenant qu'étant donnée une relation bien-fondée $\prec$ sur W, il existe une fonction $\rho(W, \prec)$ monotone et injective de W dans la classe $\langle \underset{\sim}{Ord}, < \rangle$ des ordinaux.

Soit $E(W, \prec) \in (\underset{\sim}{rk}(W, \prec) \to \{X : X \subseteq W\})$ défini par $E(W, \prec)(\alpha) = \{x \in W : \underset{\sim}{rk}(W, \prec)(x) = \alpha\}$. Noter que $\forall \alpha, \alpha' \in \underset{\sim}{rk}(W, \prec).\,[\alpha \neq \alpha' \Rightarrow E(W, \prec)(\alpha) \wedge E(W, \prec)(\alpha') = \phi]$ et $\forall x \in W.\ \exists \alpha \in \underset{\sim}{rk}(W, \prec).\,[x \in E(W, \prec)(\alpha)]$.

D'après l'axiome du choix, il y a un ordre total $\ll(W,\prec)(\alpha)$ sur $E(W,\prec)(\alpha)$.

Définissons $\varrho(W,\prec)(x,y) = \underset{\sim}{rk}[E(W,\prec)(rx), \ll(W,\prec)(rx)][y]$ où rx est $\underset{\sim}{rk}(W,\prec)(x)$ et $\varepsilon(W,\prec)(x) = (\varrho(W,\prec)(x,x)+1)$ de sorte que $\forall x \in W.[0 < \varepsilon(W,\prec)(x)]$. Définissons $\gamma(W,\prec) = \underset{\sim}{sup}^+\{\varepsilon(W,\prec)(x) : x \in W\}$ de sorte que $\forall x \in W.[\varepsilon(W,\prec)(x) \prec \gamma(W,\prec)]$ et $\zeta(W,\prec)(x) = ((\gamma(W,\prec) \times \underset{\sim}{rk}(W,\prec)(x)) + \varepsilon(W,\prec)(x))$.

Si $x \prec y$ alors $\underset{\sim}{rk}(W,\prec)(x) < \underset{\sim}{rk}(W,\prec)(y)$ et donc d'après le lemme $\zeta(W,\prec)(x) \prec \zeta(W,\prec)(y)$. Si $rx = \zeta(W,\prec)(x) = \zeta(W,\prec)(y) = ry$ alors $rx \not\prec ry$ et $ry \not\prec rx$ de sorte que d'après le lemme $\underset{\sim}{rk}(W,\prec)(x) = \underset{\sim}{rk}(W,\prec)(y)$ et $\varepsilon(W,\prec)(x) = \varepsilon(W,\prec)(y)$ d'où $\varrho(W,\prec)(x,x) = \varrho(W,\prec)(x,y)$. Ceci implique que nous n'avons ni $x \ll(W,\prec)(rx)\ y$ ni $y \ll(W,\prec)(rx)\ x$ et puisque $x,y \in E(W,\prec)(rx)$ qui est totalement ordonné par $\ll(W,\prec)(rx)$ nous concluons que $x = y$.

La preuve $(\mathcal{B}_\beta) \Rightarrow (\mathcal{B}_\beta)$ est maintenant immediate si nous choisissons $\Lambda_\beta = \underset{\sim}{sup}^+\{\zeta(\Lambda_\beta \cap \mu, \prec_\beta)(x) : x \in (\Lambda_\beta \cap \mu)\}$, $\pi_\beta = \zeta(\Lambda_\beta \cap \mu, \prec_\beta)(\pi_\beta)$, $\theta_{\beta_\lambda}(\Delta,\Delta') = [\exists a \in (\Lambda_\beta \cap \mu).(\lambda = \zeta(\Lambda_\beta \cap \mu, \prec_\beta)(a) \wedge \theta_{\beta_a}(\Delta,\Delta'))]$, $\Delta_\beta = \underset{\sim}{sup}^+(\zeta(\Delta_\beta, <_\beta)(x) : x \in \Delta_\beta\}$ et $I_{\beta_\lambda}(\delta,\Delta,\Delta') = [\exists a \in (\Lambda_\beta \cap \mu), d \in \Delta_\beta.(\lambda = \zeta(\Lambda_\beta \cap \mu, \prec_\beta)(a) \wedge \delta = \zeta(\Delta_\beta, <_\beta)(d) \wedge I_{\beta_a}(d,\Delta,\Delta'))]$.

□

L'utilisation des lemmes $\theta_\lambda(\Delta,\Delta')$ dans le principe d'induction $(\mathcal{B}_\beta)$ est redondante parceque nous pouvons utiliser à la place une certaine assertion intermittente $I_\lambda(\delta,\Delta,\Delta')$ pour un certain δ telle que $I_\lambda(\delta,\Delta,\Delta') \Rightarrow \theta_\lambda(\Delta,\Delta')$. Par convention, nous pouvons choisir $\delta = 0$ et donc le principe d'induction $(\mathcal{B}_\beta)$ peut être simplifié en :

$[\exists \Lambda \in \underline{Ord},\ \pi \in \Lambda,\ \Delta \in \underline{Ord},\ I \in (\Lambda \to (\Delta \times S \times S \to \{tt, ff\}))$.

$(\forall s, s' \in S.\ [I_\pi(0, s, s') = (\Phi(s) \to \Psi(s, s'))])$

$(\forall s \in S.\ \exists \delta \in \Delta.\ I_\pi(\delta, s, s))$

$(\forall \lambda \in \Lambda,\ s, s' \in S,\ \delta' \in (\Delta \vee 0).$ $\quad (_{10})$

$[I_\lambda(\delta', s, s') \Longrightarrow$

$(\exists s'' \in S, q \in A.\ t_q(s', s'') \wedge \forall s'' \in S, q \in A.\ [t_q(s', s'') \Rightarrow \exists \delta'' < \delta'.\ I_\lambda(\delta'', s, s'')])$

$\vee$

$(\exists \lambda' < \lambda.\ [\exists \delta \in \Delta.\ I_{\lambda'}(\delta, s', s') \wedge \forall s'' \in S.\ [I_{\lambda'}(0, s', s'') \Rightarrow \exists \delta'' < \delta'.\ I_\lambda(\delta'', s, s'')]])]]$

Théorème 5.3.3 ∨ 9

Démonstration

Choisir $\Lambda_{10} = \Lambda_9$, $\pi_{10} = \pi_9$, $\Delta_{10} = \Delta_9$, $I_{10_\lambda}(\delta, s, s') = [(\delta = 0 \wedge \theta_{9_\lambda}(s, s')) \vee (\delta > 0 \wedge I_{9_\lambda}(\delta, s, s'))]$.

Comme le montre cette succession de transformations, la preuve qu'en $(\mathcal{B}_{10})$, un état s' satisfaisant $I_\lambda(\delta', s, s')$ conduit fatalement à un état s'' tel que $\theta_\lambda(s, s'')$ soit vrai met en jeu une induction sur des parties de chemins d'exécution, traduite par δ' et une induction sur les données traduite par λ. Pour pouvoir établir une comparaison avec la méthode de Floyd [67], les deux cas peuvent être réduits à une induction sur les calculs, utilisant δ' pour mesurer le "nombre de pas" à faire entre s' et s'' :

$$[\exists \Gamma \in \underline{Ord},\ I \in (\Gamma \times S \times S \to \{tt, ff\}),\ \sigma \in (\Gamma \to \Gamma).$$

$$(P) \quad (\forall \Delta \in S.\ \exists \gamma \in \Gamma.\ [I(\gamma,\Delta,\Delta) \wedge \forall \Delta' \in S.\ (I(\sigma(\gamma),\Delta,\Delta') \Rightarrow [\Phi(\Delta) \Rightarrow \Psi(\Delta,\Delta')]]])$$

$$\wedge (\forall \gamma' \in \Gamma,\ \Delta,\Delta' \in S. \qquad (\mathcal{B}_{11})$$

$$[I(\gamma',\Delta,\Delta') \Rightarrow$$

$$(HS) \quad (\exists \Delta'' \in S, a \in A.\ t_a(\Delta',\Delta'') \wedge \forall \Delta'' \in S, a \in A.\ [t_a(\Delta',\Delta'') \Rightarrow \exists \gamma'' < \gamma'.\ (\sigma(\gamma'') = \sigma(\gamma') \wedge I(\gamma'',\Delta,\Delta'))])$$

$$\vee$$

$$(LI) \quad (\exists \gamma < \gamma'.\ [I(\gamma,\Delta',\Delta') \wedge \forall \Delta'' \in S.\ [I(\sigma(\gamma),\Delta',\Delta'') \Rightarrow \exists \gamma'' < \gamma'.\ (\sigma(\gamma'') = \sigma(\gamma') \wedge I(\gamma'',\Delta,\Delta''))])$$

$$\vee$$

$$(C) \quad I(\sigma(\gamma'),\Delta,\Delta')])]$$

Théorème 5.3.3∿10

$$(\mathcal{B}_{10}) \Rightarrow (\mathcal{B}_{11})$$

Démonstration

Soit $\underline{\iota}(\langle \lambda,\delta \rangle) = [(\Delta_{10} \dot{\times} \lambda) \dot{+} \delta]$ l'isomorphisme d'ordre entre $\Lambda_{10} \times \Delta_{10}$ bien-ordonné par l'ordre lexicographique gauche $\langle \lambda',\delta' \rangle \prec \langle \lambda,\delta \rangle$ si et seulement si $((\lambda' < \lambda) \vee (\lambda' = \lambda \wedge \delta' < \delta))$ et $\Gamma_{11} = (\Delta_{10} \dot{\times} \Lambda_{10})$ bien ordonné par $<$. Soit $\langle \underline{\alpha},\underline{\delta} \rangle$ l'inverse de $\underline{\iota}$ de sorte que $\forall \lambda \in \Lambda_{10},\ \delta \in \Delta_{10}.\ (\lambda = \underline{\lambda}(\underline{\iota}(\langle \lambda,\delta \rangle)) \wedge \delta = \underline{\delta}(\underline{\iota}(\langle \lambda,\delta \rangle)))$ et $\forall \gamma \in \Gamma_{11}.$ $\gamma = \underline{\iota}(\langle \underline{\alpha}(\gamma), \underline{\delta}(\gamma) \rangle)$. Choisir $I_{11}(\gamma,\Delta,\Delta') = I_{10\,\underline{\alpha}(\gamma)}(\underline{\delta}(\gamma),\Delta,\Delta')$ et $\sigma(\gamma) = \underline{\iota}(\langle \underline{\alpha}(\gamma), 0 \rangle)$.

□

En utilisant la généralisation abstraite $(\mathcal{B}_{11})$ de la méthode de Burstall, nous pouvons la comparer équitablement avec la généralisation similaire $(\mathcal{F}_6')$ de la méthode de Floyd. Pour $(\mathcal{F}_6')$, la ligne (LI) est supprimée (de sorte que nous pouvons toujours choisir $\sigma(\gamma) = 0$). Par conséquent la différence cruciale entre les méthodes de Floyd et de Burstall, n'est ni l'utilisation d'assertions invariantes au lieu d'assertions intermittentes,

ni l'utilisation d'une induction sur les calculs au lieu d'une induction sur les données mais bien l'introduction de la récursivité.

L'équivalence des principes d'induction $(\mathcal{B}_2), \ldots, (\mathcal{B}_{11})$ vient de :

Théorème 5.3.3 N11 (Correction)

est fatale pour $\langle S, A, \Sigma\langle S, A, t, \Phi\rangle\rangle$

Démonstration

Posant $\varepsilon^{\gamma}_{\Delta}(\Delta') = I(\gamma, \Delta, \Delta')$, nous démontrons par induction sur $\langle \Gamma, < \rangle$ que $\forall \gamma \in \Gamma, \Delta \in S, p \in \Sigma\langle S, A, t, \varepsilon^{\gamma}_{\Delta}\rangle . \exists i \in |p| . I(\sigma(\gamma), \Delta, p_i)$. Supposons le vrai pour $\gamma' < \gamma$. Par l'absurde, soient $\Delta \in S$, $p \in \Sigma\langle S, A, t, \varepsilon^{\gamma}_{\Delta}\rangle$ tels que $\forall i \in |p| . \neg I(\sigma(\gamma), \Delta, p_i)$. Pour obtenir une contradiction, nous construisons une séquence infinie $\langle\langle i_k, \gamma_k\rangle : k \geq 0\rangle$ telle que $\forall k \geq 0 . [I(\gamma_k, \Delta, p_{i_k}) \wedge \sigma(\gamma_k) = \sigma(\gamma) \wedge \gamma_k > \gamma_{k+1}]$. Choisir $\gamma_0 = \gamma$ et $i_0 = 0$. Si la séquence est construite jusqu'au point k alors $I(\gamma_k, \Delta, p_{i_k})$ satisfait (HS), (LI) ou (C). (C) est impossible (car $I(\sigma(\gamma_k), \Delta, p_{i_k})$ impliquerait $I(\sigma(\gamma), \Delta, p_{i_k})$). Dans le cas (HS), $\exists \Delta'' \in S, a \in A . t_a(p_{i_k}, \Delta'')$ implique que $i_{k+1} = (i_k + 1) \in |p|$. Donc $t_{p_{i_k}}(p_{i_k}, p_{i_{k+1}})$ implique $\exists \gamma_{k+1} < \gamma_k . (\sigma(\gamma_{k+1}) = \sigma(\gamma_k) = \sigma(\gamma) \wedge I(\gamma_{k+1}, \Delta, p_{i_{k+1}}))$. Dans le cas (LI), il existe $\gamma' < \gamma_k$ tel que $I(\gamma', p_{i_k}, p_{i_k})$. Donc par hypothèse d'induction $\exists j \in |p^{\geq i_k}| . I(\sigma(\gamma'), (p^{\geq i_k})_0, (p^{\geq i_k})_j)$. Si nous posons $i_{k+1} = (i_k + j)$ alors nous aurons $I(\sigma(\gamma'), p_{i_k}, p_{i_{k+1}})$ d'où $\exists \gamma_{k+1} < \gamma_k . [I(\gamma_{k+1}, \Delta, p_{i_{k+1}}) \wedge \sigma(\gamma_{k+1}) = \sigma(\gamma_k) = \sigma(\gamma)]$. Q.E.D.

Maintenant si $p \in \Sigma\langle S, A, t, \Phi\rangle$ alors $\exists \gamma \in \Gamma . I(\gamma, p_0, p_0)$ de sorte que $p \in \Sigma\langle S, A, t, \varepsilon^{\gamma}_{p_0}\rangle$ et d'après le lemme ci-dessus, $\exists i \in |p| . I(\sigma(\gamma), p_0, p_i)$. D'après (P) ceci implique $\Phi(p_0) \Rightarrow \Psi(p_0, p_i)$ d'où $\Psi(p_0, p_i)$.

□

Exemple 5.3.3-1 (Preuve d'un programme de parcours d'arbre en utilisant le principe d'induction ($\mathcal{B}_7$))

Suite à l'exemple 5.2.3-1 dans lequel nous avons démontré la correction totale du programme suivant :

```
Start:  St=(); Co:=0;
Loop:   if Tr ≠ nil
            then begin  Push Tr onto St;
                        Tr := lf(Tr); goto Loop
                 end
            else begin  Co := Co+1;
                        if St=() then goto Finish;
                        Pop Tr from st;
                        Tr := rg(Tr); goto Loop
                 end;
Finish:
```

en utilisant le principe d'induction ($\mathcal{F}_6$) correspondant à une preuve par la méthode de Floyd, nous montrons que la preuve que donne Burstall [74] de ce programme se ramène au principe d'induction ($\mathcal{B}_7$). Cette preuve peut se formuler à l'aide d'une charte de preuve (que nous formaliserons dans le paragraphe suivant) comme suit :

Posons $|Tr| = ((Tr = nil) \rightarrow 0 \mid (1 + |lf(Tr)| + |rg(Tr)|))$

Théorème 0 : sur |tr|)

[Tr=tr ∧ Co=co ∧ St=st] at Loop

tr≠nil

"Hand-simulation"

(|rg(tr)|<|tr|)

[Tr=nil ∧ Co=co+tips(tr)-1 ∧ St=st] at Loop

Théorème 1 :

[Tr=tr ∧ Co=co ∧ St=st] at Start

"Hand - simulation"

[Tr=tr ∧ Co=0 ∧ St=()] at Loop

↓ *Théorème 0*

[Tr=nil ∧ Co=tips(tr)-1 ∧ St=()] at Loop

"Hand - simulation"

[Co=tips(tr)] at Finish

La preuve du théorème i, $i=0,1$ commence par une hypothèse de la forme :

$[Tr = tr \wedge Co = co \wedge St = st]$ at L

et procède par déductions successives d'assertions intermittentes intermédiaires par évaluation symbolique, par application d'un théorème démontré auparavant ou par application récursive du théorème i. Puisque les chartes de preuve doivent être finies, chaque assertion intermittente apparaît à une distance $d \in [0, m_i]$ du point de sortie de la charte de preuve. Donc elle peut s'écrire sous la forme :

$[Tr = f_d(tr) \wedge Co = g_d(tr, co) \wedge St = h_d(tr, st)]$ at L_d

et doit être comprise comme une abbréviation de :

"if sometime $[Tr = tr \wedge Co = co \wedge St = st]$ at L
then sometime $[Tr = f_d(tr) \wedge Co = g_d(tr, co) \wedge St = h_d(tr, st)]$ at L_d"

La charte de preuve elle-même peut s'exprimer sous une forme relationnelle si nous groupons ces assertions intermittentes sous forme disjonctive :

$$PL_i(\delta', \langle \ell, tr, co, st\rangle, \langle \ell', tr', co', st'\rangle) =$$
$$[(\ell = L) \Rightarrow (\bigvee_{d=0}^{m_i} [\delta' = d \wedge \ell' = L_d \wedge tr' = f_d(tr) \wedge co' = g_d(tr, co) \wedge st' = h_d(tr, st)])]$$

Plus précisement, nous avons :

$$PL_0(\delta', \langle \ell, tr, co, st\rangle, \langle \ell', tr', co', st'\rangle) =$$
$$[(\ell = Loop) \Rightarrow$$
$$([\delta' = 4 \wedge \ell' = Loop \wedge tr' = tr \wedge co' = co \wedge st' = st]$$
$$\vee [\delta' = 3 \wedge \ell' = Loop \wedge tr' = lf(tr) \wedge co' = co \wedge st' = (tr . st)]$$
$$\vee [\delta' = 2 \wedge \ell' = Loop \wedge tr' = mil \wedge co' = (co + tips(lf(tr)) - 1) \wedge st' = (tr . st)]$$
$$\vee [\delta' = 1 \wedge \ell' = Loop \wedge tr' = rg(tr) \wedge co' = (co + tips(lf(tr)) \wedge st' = st]$$
$$\vee [\delta' = 0 \wedge \ell' = Loop \wedge tr' = mil \wedge co' = (co + tips(tr) - 1) \wedge st' = st])]$$

$$PL_1(\delta', \langle \ell, tr, co, st\rangle, \langle \ell', tr', co', st'\rangle) =$$
$$[(\ell = start) \Rightarrow$$
$$([\delta' = 3 \wedge \ell' = start \wedge tr' = tr \wedge co' = co \wedge st' = st]$$
$$\vee [\delta' = 2 \wedge \ell' = Loop \wedge tr' = tr \wedge co' = 0 \wedge st' = ()]$$
$$\vee [\delta' = 1 \wedge \ell' = Loop \wedge tr' = mil \wedge co' = (tips(tr) - 1) \wedge st' = ()]$$
$$\vee [\delta' = 0 \wedge \ell' = Finish \wedge co' = tips(tr)])]$$

Dans la preuve du théorème 1, nous utilisons le théorème 0 et dans la preuve du théorème 0 pour un arbre tr, nous supposons qu'il est vrai pour des arbres tr' tels que $|tr'| < |tr|$. Donc la preuve est par induction sur le bon-ordre $\langle (\{1,3\} \cup (\{0\} \times \omega)), \prec \rangle$ avec $\prec$ défini par

$\langle 0,m\rangle \prec 1$ pour tout $m \in \omega$ et $\langle 0,m'\rangle \prec \langle 0,m\rangle$ si et seulement si $m' < m$.
Burstall [74] parle d'"induction on data" et ne considère pas explicitement un ordre sur les théorèmes parce qu'il compte sur la culture mathématique de ses lecteurs pour éviter les erreurs comme les preuves circulaires.

Le principe d'induction (B_7) garantit que l'induction est faite correctement puisque le théorème $\theta_{\lambda'}$ ne peut être utilisé dans la preuve du théorème θ_λ que s'il a été démontré avant θ_λ (i.e. $\lambda' < \lambda$). Tous les cas particuliers tels les preuves récursives ou mutuellement récursives de théorèmes dans la méthode de Burstall peuvent être pris en compte par un choix adéquat du bon-ordre $\langle \Lambda, < \rangle$. A un isomorphisme près, nous pouvons toujours choisir $\Lambda \in \underset{\sim}{Ord}$.

Par exemple, au lieu du bon-ordre $\langle W, <\rangle$ où $W = (\{1\} \cup (\{0\} \times \omega))$, nous pouvons utiliser $\langle \Lambda, <\rangle$ où $\Lambda = \omega + 1$ à l'isomorphisme e près défini par $e(1) = \omega$ et $e(\langle 0,n\rangle) = n$.

Le lecteur peut maintenant vérifier que la preuve de Burstall [74] consiste exactement à appliquer le principe d'induction (B_7) avec :

$\Lambda = \omega + 1$

$\theta_\lambda(\langle \ell, tr, co, st\rangle, \langle \ell', tr', co', st'\rangle) =$

$\quad (\; [(\lambda = \omega \wedge \ell = Start) \Rightarrow (\ell' = Finish \wedge co' = tips(tr))]$

$\quad \wedge \; [(|tr| = \lambda < \omega \wedge \ell = Loop) \Rightarrow (\ell' = Loop \wedge tr' = mil \wedge co' = (co + tips(tr) - 1) \wedge st' = st)])$

$\Pi = \omega$

$\Delta = 5$

$I_\lambda(\delta', \langle \ell, tr, co, st\rangle, \langle \ell', tr', co', st'\rangle) =$

$\quad (\; [(\lambda = \omega) \Rightarrow PL_1(\delta', \langle \ell, tr, co, st\rangle, \langle \ell', tr', co', st'\rangle)]$

$\quad \wedge \; [(|tr| = \lambda < \omega) \Rightarrow PL_0(\langle \delta', \langle \ell, tr, co, st\rangle, \langle \ell', tr', co', st'\rangle)])$

Par exemple, à partir de :

$I_{|tr|}(3, \langle \ell, tr, co, st\rangle, \langle \ell', tr', co', st'\rangle)$

$\quad = [(\ell = Loop) \Rightarrow (\ell' = Loop \wedge tr' = ff(tr) \wedge co' = co \wedge st' = (tr.st))]$

nous vérifions que $|tr'| = |ff(tr)| < |tr|$ et en utilisant le théorème :

$$\theta_{|tr'|}(\langle \ell', tr', co', at' \rangle, \langle \ell'', tr'', co'', at'' \rangle)$$
$$= [(\ell' = Loop) \Rightarrow (\ell'' = Loop \wedge tr'' = nil \wedge co'' = (co' + tips(tr') - 1) \wedge at'' = at')]$$

nous dérivons :

$$[(\ell = Loop) \Rightarrow (\ell'' = Loop \wedge tr'' = nil \wedge co'' = (co + tips(ff(tr)) - 1 \wedge at'' = (tr.at))]$$
$$= I_{|tr|}(2, \langle \ell, tr, co, at \rangle, \langle \ell'', tr'', co'', at'' \rangle)$$

De même, à partir de :

$$I_{|tr|}(2, \langle \ell, tr, co, at \rangle, \langle \ell', tr', co', at' \rangle)$$
$$= [(\ell = Loop) \Rightarrow (\ell' = Loop \wedge tr' = nil \wedge co' = (co + tips(ff(tr)) - 1) \wedge at' = (tr.at))]$$

et

$$t_{\tilde{2}}(\langle Loop, nil, co', (tr.at) \rangle, \langle Loop, rg(tr), co'+1, at \rangle)$$

nous dérivons :

$$[(\ell = Loop) \Rightarrow (\ell'' = Loop \wedge tr'' = rg(tr) \wedge co'' = (co + tips(ff(tr))) \wedge at'' = at)]$$
$$= I_{|tr|}(1, \langle \ell, tr, co, at \rangle, \langle \ell', tr', co', at' \rangle)$$

La seule différence est que le programme ci-dessus est total, de sorte qu'on n'a pas besoin de contrôler explicitement l'absence d'erreurs à l'exécution et plus généralement d'états de blocages $(I_\lambda(\delta', \lambda, \lambda') \Rightarrow \exists \lambda'' \in S.\ t_{\tilde{2}}(\lambda', \lambda''))$.

□

5.3.4 COMPLETUDE SEMANTIQUE FORTE

L'argument de complétude sémantique donné dans le théorème 5.3.2∨2 est très faible parce qu'il consiste essentiellement à dire que (B_2) peut toujours être utilisé pour formuler les preuves "à la Floyd" (comme suggéré p anna-Waldinger [78]). Ayant étendu la méthode de Burstall de sorte qu'elle intègre la méthode de Floyd (cf. 5.3.3∨4 et $(B_6),\dots,(B_{11})$), l'argument habituel de complétude sémantique pour la méthode de Floyd peut être transcrit pour la méthode de Burstall (par exemple, $(\Psi$ est fatale pour $\langle S,A,\Sigma\langle S,A,t,\Phi\rangle\rangle)\Rightarrow((B_{11})$ avec (LI) supprimée (et $\delta(\gamma)=0)$), cf. 5.2.6.2-1). Cependant, de tels arguments de complétude ne sont pas dans l'esprit de Burstall [74] qui encourage à décomposer

partielle, absence d'états de blocage et terminaison, une décomposition qui peut également s'appliquer à tout lemme mis en jeu dans la méthode de Burstall).

Nous démontrons maintenant un résultat plus fort de complétude sémantique, montrant que les lemmes mis en jeu dans des preuves "à la Burstall" peuvent être choisis plus librement.

Nous devons d'abord introduire un principe d'induction (B_{12}) où le choix entre évaluation symbolique (HS) et induction sur les données (LI) est forcé. En particulier les lemmes qui sont utilisés dans (LI) doivent être imposés. Pour cela, nous considérons une version de (B_6) où nous introduisons une relation de choix $\zeta_\lambda(s,s',\lambda')$ de sorte que l'assertion intermittente $I_\lambda(\delta',s,s')$ peut être traitée en utilisant le lemme $\lambda'<\lambda$ si et seulement si $\zeta_\lambda(s,s',\lambda')$ est vraie. (Noter que faire dépendre ζ de δ' ne serait utile que pour imposer les lemmes identité, un cas de peu d'importance que nous excluons pour simplifier).

Pour simplifier, (HS) sera traitée dans le style de $(\mathcal{B}_8)$, comme un cas particulier de (LI) et donc la relation de transition $\exists a \in A.t_a$ sera considérée comme un lemme particulier, disons θ_0, donné comme axiome.

De plus, comme nous l'avons noté pour $(\mathcal{B}_9)$, la condition de vérification $[\exists \delta \in \Delta . I_\lambda(\delta, s, s)]$ de $(\mathcal{B}_6)$ ou $(\mathcal{B}_8)$ implique que le lemme θ_λ est fatal pour $\langle S, A, \Sigma \langle S, A, t, tt\rangle\rangle$ mais nous en avons besoin que pour les états particuliers s pour lesquels θ_λ est utilisé, c'est-à-dire lorsque $\exists \lambda' < \lambda, s' \in S . \zeta_{\lambda'}(s', s, \lambda)$.

Finalement, puisque tous les lemmes jouissent de mêmes propriétés de fatalité, nous n'avons pas vraiment besoin de distinguer une proposition principale particulière.

Ces remarques nous conduisent au principe d'induction suivant (où $\Lambda \in \underset{\sim}{Ord}$, $\theta \in (\Lambda \to (S^2 \to \{tt, ff\}))$, $\forall s, s' \in S . (\theta_0(s, s') = \exists a \in A . t_a(s, s'))$, $\zeta \in ((\Lambda \smallsetminus 0) \to (S \times S \times \Lambda \to \{tt, ff\}))$:

$$
\begin{array}{l}
[\, \exists \Delta \in \underset{\sim}{Ord},\ I \in ((\Lambda \smallsetminus 0) \to (\Delta \times S \times S \to \{tt, ff\})). \\
\quad (\forall \lambda \in (\Lambda \smallsetminus 0),\ s \in S . [(\exists \lambda' \in (\Lambda \smallsetminus 0),\ s' \in S .\ \zeta_{\lambda'}(s', s, \lambda)) \Rightarrow \exists \delta \in \Delta . I_\lambda(\delta, s, s)]) \\
\quad \wedge (\forall \lambda \in (\Lambda \smallsetminus 0),\ s, s' \in S,\ \delta' \in \Delta . \\
\qquad [\, I_\lambda(\delta', s, s') \Rightarrow \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (\mathcal{B}_{12}) \\
\qquad\quad ([\exists \lambda' < \lambda . \zeta_\lambda(s, s', \lambda')] \\
\qquad\qquad \wedge [\forall \lambda' < \lambda . (\zeta_\lambda(s, s', \lambda') \Rightarrow \\
\qquad\qquad\qquad ([\lambda' = 0 \Rightarrow \exists s'' \in S . \theta_{\lambda'}(s', s'')] \wedge \\
\qquad\qquad\qquad\qquad [\forall s'' \in S . (\theta_{\lambda'}(s', s'') \Rightarrow \exists \delta'' < \delta . I_\lambda(\delta'', s, s''))])])] \\
\qquad\quad \vee \theta_\lambda(s, s')])]
\end{array}
$$

Nous montrons d'abord que $(\mathcal{B}_{12})$ est une autre formulation des principes d'induction généralisant la méthode de Burstall:

Théorème 5.3.4v1 (Equivalence des principes d'induction)

$$(\mathcal{B}_6) \Rightarrow [\exists \Lambda, \theta, \zeta . (\forall A, A' \in S . (\theta_0(A,A') = \exists a \in A . t_a(A,A'))) \wedge (\mathcal{B}_{12})]$$

Démonstration

Choisir $\Lambda_{12} = (1 + \Lambda_6)$, $\Delta_{12} = \Delta_6$, $\forall A, A' \in S . (\theta_{12_0}(A,A') = \exists a \in A . t_a(A,A'))$, si $\lambda, \lambda' \in \Lambda_6$ alors $\theta_{12_{1+\lambda}} = \theta_{6_\lambda}$, $I_{12_{1+\lambda}}(\delta', A, A') = [I_{6_\lambda}(\delta', A, A') \wedge \forall \delta \in \Delta_6 . (I_{6_\lambda}(\delta, A, A') \quad \delta' \leq \delta)]$, $\zeta_{12_{1+\lambda}}(A, A', 0) = [\exists \delta' \in \Delta_6 . (I_{12_{1+\lambda}}(\delta', A, A') \wedge \neg \theta_{6_\lambda}(A, A') \wedge \exists A'' \in S, a \in A . t_a(A', A'') \wedge \forall A'' \in S, a \in A . (t_a(A', A'') \Rightarrow [\exists \delta'' < \delta' . I_{6_\lambda}(\delta'', A, A'')]))]$ et $\zeta_{12_{1+\lambda}}(A, A', 1+\lambda') = [\exists \delta' \in \Delta_6 . (I_{12_{1+\lambda}}(\delta', A, A') \wedge \neg \theta_{6_\lambda}(A, A') \wedge \forall A'' \in S . (\theta_{6_{\lambda'}}(A', A'') \Rightarrow [\exists \delta'' < \delta' . I_{6_\lambda}(\delta'', A, A'')]))]$.

□

Théorème 5.3.4v2 (Equivalence des principes d'induction (suite))

$$[\exists \Lambda, \theta, \zeta, \pi \in (\Lambda \vee 0) . (\forall A, A' \in S . (\zeta_\pi(A, A, \pi) = tt \wedge \theta_\pi(A, A') = [\Phi(A) \Rightarrow \Psi(A, A')] \wedge \theta_0(A, A') = \exists a \in A . t_a(A, A') \wedge (\mathcal{B}_{12}))]$$

$$\Rightarrow (\mathcal{B}_9)$$

Démonstration

Choisir $\Lambda_9 = \Lambda_{12}$, $\theta_{9_\lambda}(A, A') = (\lambda = 0 \rightarrow tt \mid \theta_{12_\lambda}(A, A'))$, $\pi_9 = \pi_{12}$, $\Delta_9 = \Delta_{12}$, $I_{9_\lambda}(\delta, A, A') = (\lambda = 0 \rightarrow ff \mid I_{12_\lambda}(\delta, A, A'))$

□

Puisque nous ne distinguerons plus une proprosition principale θ_π comme dans $(\mathcal{B}_6)$, la correction de $(\mathcal{B}_{12})$ est mieux formulée comme suit:

Condition 5.3.4 : 1 (Définition de la correction (relativement à t))

$\forall \lambda \in (\Lambda \smallsetminus 0),\ p \in \Sigma^{<S,A,t,\varepsilon_\lambda>} . \exists i \in |p| . \theta_\lambda(p_0, p_i)$ où $\varepsilon_\lambda(A) = [\exists \lambda' \in \Lambda, A' \in S . \iota_{\lambda'}(A', A, A)]$

Théorème 5.3.4 ∾ 3 (Correction (relativement à t))

(5.3.4 : 1)

Démonstration

Découle des théorèmes 5.3.4 ∾ 5 et 5.3.4 ∾ 6 que nous démontrerons plus tard.

□

Nous nous intéressons principalement à la complétude de $(\mathcal{B}_{12})$. La réciproque du théorème 5.3.4 ∾ 3 n'est pas vraie :

Théorème 5.3.4 ∾ 4 (Condition insuffisante de complétude)

$(\mathcal{B}_{12})$

Démonstration

Considérer le contre-exemple : $S = \{a, b, c\}$, $A = \{\underset{\sim}{a}\}$, $t_{\underset{\sim}{a}}(A, A') = [(A = a \wedge A' = b) \vee (A = b \wedge A' = c)]$, $\Lambda = 3$, $\theta_0 = t_{\underset{\sim}{a}}$, $\theta_1(A, A') = [A = a \wedge A' = c]$, $\theta_2(A, A') = [A = a \wedge A' = b]$, $\Delta \neq 0$, $\iota_\lambda(A, A', \lambda') = [(\lambda = 1 \wedge A = a \wedge A' \in \{a, b\} \wedge \lambda' = 0) \vee (\lambda = 2 \wedge A' = A = a \wedge \lambda' \in \{1, 2\})]$.

La condition (5.3.4 : 1) est vérifiée trivialement. Si $(\mathcal{B}_{12})$ était vrai alors nous aurions eu $\iota_2(a, a, 2)$ donc $\exists \delta_1 \in \Delta . I_2(\delta_1, a, a)$ et par $\iota_2(a, a, 1)$ et $\theta_1(a, c)$ nous aurions eu $\exists \delta_2 < \delta_1 . I_2(\delta_2, a, c)$. Mais $\neg\theta_2(a, c)$ et $\forall \lambda' < 2 . \neg \iota_2(a, c, \lambda')$, une contradiction.

□

Dans la méthode de Burstall, l'utilisation d'un lemme θ_ℓ dans la preuve de la proposition θ_p a l'effet de couvrir un certain nombre de transitions en un seul pas θ_ℓ. Aussi, θ_ℓ peut être utilisé dans la preuve de θ_p seulement si cette réduction conserve la fatalité de θ_p. Autrement dit, θ_p doit être fatal pour les "transitions" θ_ℓ résultant des lemmes utilisés dans la preuve de θ_p. Ceci s'exprime plus formellement par la condition (où $\forall_{A,A'\in S}.(\theta_0(A,A') = \exists_{a\in A}.t_a(A,A'))$) :

$$\forall \lambda \in (\Lambda \vee 0),\ A\in S.\ [(\exists \lambda' \in \Lambda, A'\in S.\ \zeta_{\lambda'}(A',A,\lambda)) \Rightarrow (\forall p \in \Sigma^{<S,\{\underline{a}\},\tau_{\lambda A},\varepsilon_A>}.\ \exists i\in|p|.\ \theta_\lambda(p_0,p_i))]$$

où $\quad \varepsilon_A(\underline{A}) = [\underline{A} = A]$

$$\tau_{\lambda A}\underline{a}(A',A'') = [\exists \lambda' < \lambda .(\zeta_\lambda(A,A',\lambda') \wedge \theta_{\lambda'}(A',A'')]$$

La condition (5.3.4 : 2) est une condition nécessaire de complétude :

Théorème 5.3.4 ∨ 5 (Correction (relativement à τ), Condition nécessaire de complétude)

($\mathcal{B}_{12}$) (5.3.4 : 2)

Démonstration

Supposons ($\mathcal{B}_{12}$). Si $\Lambda = 1$ ou $\forall \lambda', A'.\ \neg\zeta_{\lambda'}(A',A,\lambda)$ alors (5.3.4 : 2) est vrai trivialement sinon nous démontrons (5.3.4 : 2) par induction transfinie sur $\lambda \in (\Lambda \vee 0)$. Etant donné $\lambda \in (\Lambda \vee 0)$ et $A\in S$, supposons par l'absurde que $\exists \lambda' \in \Lambda$, $A' \in S$, $p \in \Sigma^{<S,\{\underline{a}\},\tau_{\lambda A},\varepsilon_A>}.(\zeta_{\lambda'}(A',A,\lambda) \wedge \forall i\in|p|.\neg\theta_\lambda(p_0,p_i))$. Pour avoir une contradiction, nous montrons qu'il est alors possible de construire une séquence infinie $<<\delta_k, i_k> : k\geq 0>$ telle que $\forall k\geq 0.\ I_\lambda(\delta_k, A, p_{i_k})$ est vrai et $<\delta_k : k\geq 0>$ est une chaîne infinie strictement décroissante d'ordinaux.

Nous avons $\zeta_{\lambda'}(A,A,\lambda)$ de sorte que par ($\mathcal{B}_{12}$) nous dérivons $I_\lambda(\delta_0,A,A)$ soit $I_\lambda(\delta_0,A,p_{i_0})$ avec $i_0 = 0$. Si nous avons construit la séquence jusqu'en k,

alors d'après (β_{12}), $I_\lambda(\delta_k, \Delta, p_{i_k})$ implique $([\exists \underline{\lambda} < \lambda . \mathcal{L}_\lambda(\Delta, p_{i_k}, \underline{\lambda})] \wedge [\forall \lambda' < \lambda . (\mathcal{L}_\lambda(\Delta, p_{i_k}, \lambda') \Rightarrow ([\lambda' = 0 \Rightarrow \exists \Delta'' \in S . \theta_{\lambda'}(p_{i_k}, \Delta'')] \wedge [\forall \Delta'' \in S . (\theta_{\lambda'}(p_{i_k}, \Delta'') \Rightarrow \exists \delta'' < \delta_k . I_\lambda(\delta'', \Delta, \Delta''))])])$ parce que nous avons supposé $\neg\theta_\lambda(p_0, p_{i_k})$ et $p_0 = \Delta$. Si $\underline{\lambda} = 0$ alors $\exists \Delta'' \in S . \theta_{\underline{\lambda}}(p_{i_k}, \Delta'')$ donc $\exists \Delta'' \in S . \tau_{\lambda\Delta}(p_{i_k}, \Delta'')$. Sinon $0 < \underline{\lambda} < \lambda$ de sorte que par hypothèse d'induction $\forall \Delta' \in S . [(\exists \lambda'', \Delta'' . \mathcal{L}_{\lambda''}(\Delta'', \Delta', \underline{\lambda})) \Rightarrow (\forall p' \in \Sigma^{<\omega}(S, A, \tau_{\underline{\lambda}\Delta'}, \varepsilon_{\Delta'}) . \exists i \in |p'| . \theta_{\underline{\lambda}}(p'_0, p'_i))]$. En particulier pour $\Delta' = p_{i_k}$, $\mathcal{L}_\lambda(\Delta, p_{i_k}, \underline{\lambda})$ est vrai et $\Sigma^{<\omega}(S, A, \tau_{\underline{\lambda}p_{i_k}}, \varepsilon_{p_{i_k}})$ n'est pas vide, d'où $\exists \Delta'' \in S . \theta_{\underline{\lambda}}(p_{i_k}, \Delta'')$ donc $\exists \Delta'' \in S . \tau_{\lambda\Delta}(p_{i_k}, \Delta'')$. Puisque p_{i_k} n'est pas un état de blocage $i_{k+1} = i_k + 1$ appartient à $|p|$ et nous avons $\tau_{\lambda\Delta}(p_{i_k}, p_{i_{k+1}})$. Il s'ensuit que $\exists \lambda' < \lambda . (\mathcal{L}_\lambda(\Delta, p_{i_k}, \lambda') \wedge \theta_{\lambda'}(p_{i_k}, p_{i_{k+1}}))$ d'où $\exists \delta_{k+1} < \delta_k . I_\lambda(\delta_{k+1}, \Delta, p_{i_{k+1}})$.

□

La condition (5.3.4 : 2) (i.e. chaque lemme est fatal relativement aux lemmes utilisés pour sa preuve) implique la condition (5.3.4 : 1) (i.e. chaque lemme est fatal relativement au système de transition) :

Théorème 5.3.4.6 (La fatalité relativement à τ implique la fatalité relativement à t)

$(5.3.4 : 2) \Longrightarrow (5.3.4 : 1)$

Démonstration

Supposons (5.3.4 : 2), nous démontrons (5.3.4 : 1) par induction transfinie sur $\lambda \in (\Lambda \cup 0)$. Supposons par l'absurde que $\exists p \in \Sigma^{<\omega}(S, A, t, \varepsilon_\lambda) . \forall i \in |p| . \neg\theta_\lambda(p_0, p_i)$. Pour avoir une contradiction, nous allons construire une séquence $\langle i_k : k > 0 \rangle$ telle que $p_{i_0} \xrightarrow{a} p_{i_1} \xrightarrow{a} \ldots$ soit un contre-exemple pour 5.3.4 : 2 c'est-à-dire $\exists \lambda', \Delta' . \mathcal{L}_{\lambda'}(\Delta', p_{i_0}, \lambda) \wedge \forall k > 0 . [\tau_{\lambda p_{i_0}}(p_{i_k}, p_{i_{k+1}}) \wedge \neg\theta_\lambda(p_{i_0}, p_{i_k})]$. Posons $i_0 = 0$. Si la séquence est construite jusqu'en i_k, alors elle peut être prolongée car $\exists i_{k+1} > i_k . \tau_{\lambda p_{i_0}}(p_{i_k}, p_{i_{k+1}})$ (et $\neg\theta_\lambda(p_{i_0}, p_{i_k})$ par hypothèse et $i_0 = 0$). Autrement $\forall j > i_k . \neg\tau_{\lambda p_{i_0}}(p_{i_k}, p_j)$ de sorte que par définition de $\tau_{\lambda p_{i_0}}$ il viendrait $\forall j > i_k . \forall \lambda' < \lambda . [\neg\mathcal{L}_\lambda(p_0, p_{i_k}, \lambda') \vee \neg\theta_{\lambda'}(p_{i_k}, p_j)]$. Si $\forall \lambda' < \lambda . \neg\mathcal{L}_\lambda(p_0, p_{i_k}, \lambda')$ alors

$\forall \lambda' \in S.\ \neg \tau_{\lambda p_{i_0}}(p_{i_k}, \lambda')$ de sorte que $p_{i_0} \xrightarrow{a} p_{i_1} \xrightarrow{a} \dots \xrightarrow{a} p_{i_k} \in \Sigma^{<S, A, \tau_{\lambda p_{i_0}}, \varepsilon_{p_{i_0}}>}$, en contradiction avec (5.3.4 : 2). Suion $\exists \lambda' < \lambda.\ \zeta_\lambda(p_0, p_{i_k}, \lambda')$ de sorte que pour ce $\lambda' < \lambda$ nous dérivons $\forall j \geq i_k.\ \neg \theta_{\lambda'}(p_{i_k}, p_j)$ donc $p_{i_k} \xrightarrow{a} p_{i_{k+1}} \dots \in \Sigma^{<S, A, t, \varepsilon_{\lambda'}>}$, en contradiction avec l'hypothèse d'induction (5.3.4 : 1).

□

Nous pouvons maintenant donner une condition nécessaire et suffisante de complétude sémantique pour $(\mathcal{B}_{1a})$:

Théorème 5.3.4 v7 (Condition nécessaire et suffisante de complétude sémantique forte)

(5.3.4 : 2) $(\mathcal{B}_{1a})$

Démonstration

Par suite de 5.3.4 v6, nous devons démontrer seulement que (5.3.4 : 2) $\Rightarrow$ $(\mathcal{B}_{1a})$. Etant donnés $\lambda \in (\Lambda \cup 0)$, $s \in S$, nous définissons :

$$\underset{\sim}{Im}_{\lambda s} = \cup\{\underset{\sim}{Inter}^{<S, \{a\}, \tau_{\lambda s}, tt, tt, \theta_\lambda>}(s) : \exists \lambda' \in \Lambda, s' \in S.\ \zeta_{\lambda'}(s', s, \lambda)\}$$

$$\underset{\sim}{Go}_{\lambda s} = \cup\{\underset{\sim}{Goal}^{<S, \{a\}, \tau_{\lambda s}, tt, tt, \theta_\lambda>}(s) : \exists \lambda' \in \Lambda, s' \in S.\ \zeta_{\lambda'}(s', s, \lambda)\}$$

$$\underset{\sim}{Ac}_{\lambda s} = \underset{\sim}{Im}_{\lambda s} \cup \underset{\sim}{Go}_{\lambda s}$$

Nous démontrons d'abord que (5.3.4 : 2) $\Rightarrow (\forall \lambda \in (\Lambda \cup 0), s \in S.\ \underset{\sim}{\omega f}(\underset{\sim}{Ac}_{\lambda s}, \tau_{\lambda s} \upharpoonright \underset{\sim}{Im}_{\lambda s}^{-1}))$.

Ceci est évident lorsque $\forall \lambda' \in \Lambda, s' \in S.\ \neg \tau_{\lambda'}(s', s, \lambda)$ puisque $Ac_{\lambda s}$ est vide. Suion, étant donnés $\lambda \in (\Lambda \cup 0)$ et $s \in S$ tels que $\exists \lambda' \in \Lambda, s' \in S.\ \zeta_{\lambda'}(s', s, \lambda)$, supposons par l'absurde que $\exists p \in (\omega \to \underset{\sim}{Ac}_{\lambda s}).\ \forall i \in \omega.\ \tau_{\lambda s} \upharpoonright \underset{\sim}{Im}_{\lambda s}(p_i, p_{i+1})$. Nous avons $\forall i \in \omega.\ (p_i \in \underset{\sim}{Im}_{\lambda s})$ de sorte que $\neg \theta_\lambda(s, p_i)$ donc $p_i \notin \underset{\sim}{Go}_{\lambda s}$. Puisque $p_0 \in \underset{\sim}{Im}_{\lambda s}$, nous pouvons supposer $p_0 = s$ (sinon nous pouvons adjoindre à gauche de p, le préfixe $r_0, \dots, r_k$ d'une certaine trace $r \in \Sigma^{<S, \{a\}, \tau_{\lambda s}, tt>}$ tel que $r_0 = s \wedge \forall j \leq k.\ \neg \theta_\lambda(r_0, r_j) \wedge r_k = p_0$ de sorte que $\forall i < k.\ \tau_{\lambda s} \upharpoonright \underset{\sim}{Im}_{\lambda s}(r_i, r_{i+1})$). Nous avons $\exists \lambda' \in \Lambda, s' \in S.\ \zeta_{\lambda'}(s', s, \lambda)$ et $p \in \Sigma^{<S, \{a\}, \tau_{\lambda s}, \varepsilon>}$ et $\forall i \in |p|.\ (p_i \notin \underset{\sim}{Go}_{\lambda s})$ donc $\neg \theta_\lambda(p_0, p_i)$, en contradiction avec (5.3.4 : 2), Q.E.D.

Supposant (5.3.4 : 2), d'après le lemme ci-dessus, nous pouvons définir :

$\Delta = \underset{\sim}{\sup}^{+}\{ \underset{\sim}{rk}(\underset{\sim}{Ac}_{\lambda\Delta}, \tau_{\lambda\Delta} \restriction \underset{\sim}{Im}_{\lambda\Delta}^{-1}) : \lambda \in (\Lambda \vee 0) \wedge \Delta \in S \}$

$I_\lambda(\delta', \Delta, \Delta') = [\Delta' \in Ac_{\lambda\Delta} \wedge \delta' = \underset{\sim}{rk}(\underset{\sim}{Ac}_{\lambda\Delta}, \tau_{\lambda\Delta} \restriction \underset{\sim}{Im}_{\lambda\Delta}^{-1})(\Delta')]$

Si $\lambda \in (\Lambda \vee 0), \Delta \in S$ et $\exists \lambda' \in (\Lambda \vee 0), \Delta' \in S . I_{\lambda'}(\Delta', \Delta, \lambda)$ alors $\Delta \in \underset{\sim}{Ac}_{\lambda\Delta}$ de sorte que $I_\lambda(\delta, \Delta, \Delta)$ est vrai avec $\delta = \underset{\sim}{rk}(\underset{\sim}{Ac}_{\lambda\Delta}, \tau_{\lambda\Delta} \restriction \underset{\sim}{Im}_{\lambda\Delta}^{-1})(\Delta)$.

Supposons $\lambda \in (\Lambda \vee 0)$, $\Delta, \Delta' \in S$, $\delta' \in \Delta$ et $I_\lambda(\delta', \Delta, \Delta')$. Nous avons $\Delta' \in \underset{\sim}{Ac}_{\lambda\Delta}$. Si $\Delta' \in \underset{\sim}{\Omega}_{\lambda\Delta}$ alors $\theta_\lambda(\Delta, \Delta')$ est vrai. Sinon $\Delta' \in \underset{\sim}{Im}_{\lambda\Delta}$ d'où, d'après (5.3.4 : 2), il existe $\Delta'' \in S$ tel que $\tau_{\lambda\Delta}(\Delta', \Delta'')$ donc un certain $\lambda' < \lambda$ tel que $I_{\lambda'}(\Delta, \Delta', \lambda') \wedge \theta_{\lambda'}(\Delta', \Delta'')$. Si $\lambda' = 0$ nous concluons que $\exists \Delta'' \in S, a \in A . t_a(\Delta', \Delta'')$ par définition de θ_0. Sinon $\lambda' \neq 0$ et si $\forall \Delta'' \in S, a \in A . \neg t_a(\Delta', \Delta'')$ alors $\langle \Delta' \rangle \in \Sigma\langle S, A, t, \varepsilon_{\lambda'} \rangle$ et donc d'après (5.3.4 : 2) et le théorème 5.3.4 ∨ 6 nous concluons à partir de (5.3.4 : 1) que $\theta_{\lambda'}(\Delta', \Delta')$ d'où $\tau_{\lambda\Delta}(\Delta', \Delta')$. Il s'ensuit d'après $\Delta' \in \underset{\sim}{Im}_{\lambda\Delta}$ que $\exists \underline{c}, \underline{\Delta}, p \in \Sigma\langle S, \{a_n\}, \tau_{\lambda\Delta}, \varepsilon_\Delta \rangle, i \in |p| .$ $(I_{\underline{\lambda}}(\underline{\Delta}, \Delta, \lambda) \wedge \forall j \leq i . \neg\theta_\lambda(p_0, p_j) \wedge p_R = \Delta')$ de sorte que la trace enfinie $p_0, \ldots, p_R, \Delta', \Delta', \ldots$ est un contre-exemple pour (5.3.4 : 2). Aussi par l'absurde, nous concluons $\exists \Delta'' \in S, a \in A . t_a(\Delta', \Delta'')$. Finalement, étant donnés $\lambda' < \lambda$ et $\Delta'' \in S$ tels que $I_\lambda(\Delta, \Delta', \lambda')$ et $\theta_{\lambda'}(\Delta', \Delta'')$ nous avons $\tau_{\lambda\Delta} \restriction \underset{\sim}{Im}_{\lambda\Delta}(\Delta', \Delta'')$ donc $\Delta'' \in \underset{\sim}{Ac}_{\lambda\Delta}$ et il existe $\delta'' = \underset{\sim}{rk}(\underset{\sim}{Ac}_{\lambda\Delta}, \tau_{\lambda\Delta} \restriction \underset{\sim}{Im}_{\lambda\Delta}^{-1})(\Delta'') < \underset{\sim}{rk}(\underset{\sim}{Ac}_{\lambda\Delta}, \tau_{\lambda\Delta} \restriction \underset{\sim}{Im}_{\lambda\Delta}^{-1})(\Delta') = \delta'$ tel que $I_\lambda(\delta'', \Delta, \Delta'')$ soit vrai.

□

5.3.5 COMPARAISON METHODOLOGIQUE DES METHODES DE FLOYD ($\mathcal{F}_6$) ET DE BURSTALL ($\mathcal{B}_7$) GENERALISEES

Comme nous l'avons vu dans les exemples 5.2.3-1 et 5.3.3-1, une différence majeure entre la méthode de Burstall [74] (et ses suivants comme Owicki-Lamport [82] ou Manna-Pnueli [82] qui présentent les preuves au moyen de formes limitées de chartes de preuves) et le principe d'induction ($\mathcal{B}_7$) est que Burstall insiste sur la présentation finie des preuves (ou de manière équivalente à considérer des chartes de preuves sans cycles et $\Delta \in \omega$ au lieu de $\Delta \in \underset{\sim}{Ord}$ dans ($\mathcal{B}_7$)) tandis que ($\mathcal{B}_7$) n'exige qu'une relation bien-fondée. En particulier, ($\mathcal{B}_7$) peut être présenté au moyen de chartes de preuves comportant des cycles de long desquels une certaine entité (représentée par δ dans le principe d'induction ($\mathcal{B}_7$)) doit décroître strictement.

L'importance de cette remarque est d'attirer l'attention sur le fait que lorsqu'elle est convenablement généralisée (comme par ($\mathcal{B}_7$)), la méthode de Burstall contient la méthode de Floyd (plus précisément le principe d'induction ($\mathcal{F}_6$)) comme un cas particulier. Ceci parceque nous pouvons choisir $\Lambda = 1 = \{0\}$, $\theta_0 = \Psi$, $\pi = 0$ dans le principe d'induction ($\mathcal{B}_7$) de sorte que ($\mathcal{B}_7$.1) est toujours vrai et ($\mathcal{B}_7$.3.b) ne s'applique jamais, auquel cas ($\mathcal{B}_7$) se réduit exactement à ($\mathcal{F}_6$).

L'avantage net du principe d'induction ($\mathcal{B}_7$) sur ($\mathcal{F}_6$) est que ($\mathcal{B}_7$) introduit la possibilité, inexistante dans ($\mathcal{F}_6$), de preuves récursives. Ceci formalise clairement la remarque de Burstall [74] que des preuves récursives peuvent être retenues pour des versions itératives de programmes récursifs. Beaucoup plus important est le fait qu'en utilisant ($\mathcal{B}_7$), une preuve peut être décomposée successivement en preuves de lemmes indépendants tandis que ($\mathcal{F}_6$) nécessite une preuve globale. L'argument traditionnel de "separation of concern" en faveur du principe d'induction ($\mathcal{F}_6$) consiste en la traditionnel décomposition de ($\mathcal{F}_6$) en preuves de correction

correction partielle, de terminaison, d'absence d'états de blocage. Comme le remarque Gries [79], tout est groupé dans la méthode de Burstall. Cependant, ceci n'est qu'une question de présentation à laquelle nous pourrons aisément remédier puisque la décomposition en preuves indépendantes de correctim partielle, de terminaison, d'absence d'états de blocage (et d'absence d'interférences dans le cas de programmes parallèles) s'appliquant à $(\mathcal{F}_6')$ peut tout aussi s'appliquer à chaque lemme θ_λ, $\lambda \in \Lambda$ dans $(\mathcal{B}_7)$.

5.4 EQUIVALENCE FORTE DES PRINCIPES D'INDUCTION $(\mathcal{F}_i)$ ET $(\mathcal{B}_j)$

Puisque, (en posant $\varepsilon = \Phi$ et $\Phi_6 = \#$), $(\mathcal{F}_6) \Leftrightarrow$ (Ψ est fatale pour $\langle S, A, \Sigma \langle S, A, t, \varepsilon \rangle\rangle$) $\Leftrightarrow (\mathcal{B}_7)$, les principes d'induction $(\mathcal{F}_6)$ et $(\mathcal{B}_7)$ sont équivalents. Ceci signifie que lorsqu'une preuve par une méthode existe, une preuve par l'autre méthode doit exister. Cependant d'un point de vue pratique, utiliser $(\mathcal{F}_6)$ pour un programme qui a une preuve "naturelle" par $(\mathcal{B}_7)$ a été considéré quelquefois comme un défi (cf. Manna-Waldinger [78], Gries [79]). Nous démontrons maintenant un résultat d'équivalence plus fort qui montre que ce défi peut toujours être relevé parceque une preuve par $(\mathcal{B}_7)$ peut être reécrite systématiquement en une preuve par $(\mathcal{F}_6)$ et vice versa, (et d'après les résultats des paragraphes 5.2.3 et 5.3.3, ceci est vrai entre $(\mathcal{F}_i)$ et $(\mathcal{B}_j)$ quelconques). La transformation entre les deux preuves est très similaire à l'élimination de la reécursivité dans les programmes et la présentation reécursive de programmes itératifs. Aussi, (comme pour les programmes), nous ne prétendons pas naturellement, que cette transformation préserve le "naturel" des preuves.

5.4.1 $(\mathcal{F}_6) \Longrightarrow (\mathcal{B}_7)$

Comme nous l'avons fait observer au paragraphe 5.3.5, $(\mathcal{B}_7)$ contient $(\mathcal{F}_6)$ comme un cas particulier (en choisissant $\Lambda = 1 = \{0\}$, $\theta_0 = \Psi$, $\pi = 0$) de sorte qu'une preuve par $(\mathcal{F}_6)$ est aussi (à des détails mineurs près) une preuve par $(\mathcal{B}_7)$.

Cependant, nous avons besoin du beaucoup plus puissant théorème 5.4.1 n° 1 pour faire une comparaison équitable entre les méthodes de Burstall et de Floyd. Ceci parceque, comme nous l'avons également fait observer

au paragraphe 5.3.5, la méthode de Burstall corrspond plus précisement au cas $\Delta \in \omega$ qu'au cas $\Delta \in \underset{\sim}{Ord}$ dans $(\mathcal{B}_7)$.

Puisque Floyd [67] et Burstall [74] utilisent l'induction mathématique seulement sous la forme d'une induction ordinaire sur les nombres naturels, on pourrait nous demander d'ajouter les restrictions $\Gamma = \omega$ dans $(\mathcal{F}_6)$ et $\Lambda = \omega$ dans $(\mathcal{B}_7)$. Nous savons que lorsque le nondéterminisme est infini, $(\mathcal{F}_6)$ n'est pas sémantiquement complet avec $\Gamma = \omega$ et nous faisons la conjecture que $(\mathcal{B}_7)$ n'est pas sémantiquement complet avec $\Lambda = \omega$ (et $\Delta \in \omega$). Cependant, considérer que $\Gamma \in \underset{\sim}{Ord}$ dans $(\mathcal{F}_6)$ et considérer que $\Lambda \in \underset{\sim}{Ord}$ dans $(\mathcal{B}_7)$ sont des généralisations respectivement des méthodes de Floyd et Burstall de même nature de sorte que nous disons qu'une conséquence du théorème 5.4.1 v 1 est qu'une preuve par la méthode de Floyd peut se reécrire en une preuve par la méthode de Burstall :

Théorème 5.4.1 v 1

Soit $\langle S, A, t, \varepsilon \rangle$, si nous avons démontré que $\Gamma \in \underset{\sim}{Ord}$, $J \in ((\Gamma \times S \times S) \to \{tt, ff\})$, $\Phi = tt$ satisfont $(\mathcal{F}_6.1)$ et $(\mathcal{F}_6.2)$ alors reécrivant la preuve, nous pouvons trouver $\Lambda \in \underset{\sim}{Ord}$, $\theta \in (\Lambda \to (S \times S \to \{tt, ff\}))$, $\pi \in \Lambda$, $\Delta \in \omega$, $I \in (\Lambda \to (\Delta \times S \times S \to \{tt, ff\}))$ satisfaisant $(\mathcal{B}_7.1)$ et $(\mathcal{B}_7.2)$ où $\Phi = \varepsilon$.

Démonstration

S étant un ensemble, il existe (d'après l'axiome du choix) un ordinal Σ et une bijection f de Σ dans S.

De plus, le produit cartésien $\Gamma \times \Sigma$ est bien ordonné par l'ordre lexicographique droit $\prec$ défini par $\langle \gamma, \sigma \rangle \prec \langle \gamma', \sigma' \rangle$ si et seulement si $\sigma < \sigma'$ ou sinon $\sigma = \sigma$ et $\gamma < \gamma'$. Alors la fonction $F(\langle \gamma, \sigma \rangle) = ((\Gamma \times \sigma) + \gamma)$ est l'unique

isomorphisme entre $\langle \Gamma \times \Sigma^*, \prec \rangle$ et $\langle \Gamma \dot{\times} \Sigma^*, < \rangle$.

Il s'ensuit que $\ell \in ((S \times \Gamma) \to \pi)$ défini par $\pi = \Gamma \dot{\times} \Sigma^*$ et $\ell(\langle \Delta, \gamma \rangle) = F(\langle \gamma, f^{-1}(\Delta) \rangle)$ est un isomorphisme. Aussi nous pouvons définir $d \in (\pi \to S)$ et $r \in (\pi \to \Gamma)$ par $d(\lambda) = \Delta$ et $r(\lambda) = \gamma$ si et seulement si $\ell(\langle \Delta, \gamma \rangle) = \lambda$.

Choisir :

$\Lambda = \pi + 1 \quad (= \pi \cup \{\pi\})$

$\theta_\lambda(\Delta, \Delta') = ([\lambda = \pi \wedge \Psi(\Delta, \Delta')] \vee [\lambda < \pi \wedge (J(r(\lambda), d(\lambda), \Delta) \Rightarrow \Psi(d(\lambda), \Delta'))])$

$\Delta = 3$

$I_\pi(2, \Delta, \Delta') = \text{ff}$

$I_\pi(1, \Delta, \Delta') = [\exists \gamma \in \Gamma.\ J(\gamma, \Delta, \Delta') \wedge \Delta' = \Delta]$

$I_\pi(0, \Delta, \Delta') = \Psi(\Delta, \Delta')$

Pour tous $\lambda < \pi$ (ou de manière equivalente $\lambda \in \pi$) :

$I_\lambda(2, \Delta, \Delta') = [J(r(\lambda), d(\lambda), \Delta) \Rightarrow (\Delta = \Delta')]$

$I_\lambda(1, \Delta, \Delta') = [J(r(\lambda), d(\lambda), \Delta) \Rightarrow (\exists a.\, r_a(\Delta, \Delta') \wedge \neg \Psi(d(\lambda), \Delta'))]$

$I_\lambda(0, \Delta, \Delta') = \theta_\lambda(\Delta, \Delta')$

□

5.4.2 $(\mathcal{B}_7) \Rightarrow (\mathcal{F}_6)$

Une conséquence du théorème 5.4.2v1 suivant est qu'une preuve par la méthode de Burstall peut se reécrire en une preuve par la méthode de Floyd. Sans surprise, la technique est analogue à la transformation d'un programme recursif en un programme itératif équivalent. En poussant cette comparaison jusqu'à la caricature, c'est comme si on remplaçait tous les théorèmes (et leurs preuves) d'un livre de mathématiques par une proposition (et sa preuve) !

Théorème 5.4.2v1

Si nous avons démontré que $\Lambda \in \underset{\sim}{Ord}$, $\theta \in (\Lambda \to (S \times S \to \{tt, ff\}))$, $\pi \in \Lambda$, $\Delta \in \underset{\sim}{Ord}$, $I \in (\Lambda \to (\Delta \times S \times S \to \{tt, ff\}))$, $\Phi = \mathcal{E}$ satisfont $(\mathcal{B}_7.1)$, $(\mathcal{B}_7.2)$ et $(\mathcal{B}_7.3)$ pour un système de transition $\langle S, A, t, \mathcal{E} \rangle$, alors reécrivant cette preuve, nous pouvons trouver $\Gamma \in \underset{\sim}{Ord}$, $J \in (\Gamma \times S \times S \to \{tt, ff\})$ satisfaisant $(\mathcal{F}_6.1)$ et $(\mathcal{F}_6.2)$ où $\Phi = tt$.

Démonstration

(a) Nous définissons $\delta m \in (\Lambda \to (S \times S \to \Delta))$ telle que pour tout $\lambda \in \Lambda$ nous avons $\underset{\sim}{dom}(\delta m(\lambda)) = \{\langle \alpha, \alpha' \rangle \in S^2 : \exists \delta' \in \Delta . I_\lambda(\delta', \alpha, \alpha')\}$ et $\delta m(\lambda)(\alpha, \alpha') = \delta$ si et seulement si $[\delta \in \Delta \wedge I_\lambda(\delta, \alpha, \alpha') \wedge \forall \delta' \in \Delta . (I_\lambda(\delta', \alpha, \alpha') \Rightarrow [\delta \leq \delta']\,)]$, de sorte que $\delta m(\lambda)(\alpha, \alpha')$ est le plus petit δ tel que $I_\lambda(\delta, \alpha, \alpha')$ soit vrai.

(b) Soient $HS \in (\Lambda \to (S^3 \to \{tt, ff\}))$ et $LI \in (\Lambda \to (S \times S \times \Delta \times S \to \{tt, ff\}))$ tels que

$$HS(\lambda)(\alpha, \alpha', \alpha'') = [I_\lambda(\delta m(\lambda)(\alpha, \alpha'), \alpha, \alpha') \wedge \exists a \in A . t_a(\alpha', \alpha'') \wedge \forall \alpha''' \in S, a \in A . (t_a(\alpha', \alpha''') \Rightarrow [\exists \delta''' < \delta m(\lambda)(\alpha, \alpha') . I_\lambda(\delta''', \alpha, \alpha''')])]$$

$$LI(\lambda)(\alpha, \alpha', \lambda', \alpha'') = [I_\lambda(\delta m(\lambda)(\alpha, \alpha'), \alpha, \alpha') \wedge \lambda' < \lambda \wedge \theta_{\lambda'}(\alpha', \alpha'') \wedge \forall \alpha''' \in S . (\theta_{\lambda'}(\alpha', \alpha''') \Rightarrow [\exists \delta''' < \delta m(\lambda)(\alpha, \alpha') . I_\lambda(\delta''', \alpha, \alpha''')])]$$

Informellement, $HS(\lambda)(s,s',s'')$ signifie que dans la preuve du lemme θ_λ, nous pouvons montrer par évaluation symbolique que si l'exécution commence dans un état s et atteint plus tard l'état s' alors l'exécution d'un pas du programme peut conduire à s''. De manière similaire, $LI(\lambda)(s,s',s',s'')$ signifie que dans la preuve du lemme θ_λ, nous pouvons montrer par induction sur les données que si l'exécution commence dans l'état s et atteint plus tard l'état s' alors d'après le lemme $\theta_{\lambda'}$, elle peut conduire à l'état s''.

(c) Nous définissons la relation $\succ$ sur $\Lambda \times S \times S$ telle que

$$\langle \lambda_1, s_1, s_1' \rangle \succ \langle \lambda_2, s_2, s_2' \rangle$$

si et seulement si

$$\begin{array}{l} [\ (\lambda_2 = \lambda_1 \wedge s_2 = s_1 \wedge HS(\lambda_1)(s_1, s_1', s_2')) \\ \vee\ (\lambda_2 = \lambda_1 \wedge s_2 = s_1 \wedge \exists \lambda' < \lambda_1 .\ LI(\lambda_1)(s_1, s_1', \lambda', s_2')) \\ \vee\ (\lambda_2 < \lambda_1)\ ] \end{array}$$

(d) La relation $\succ$ sur $\Lambda \times S \times S$ est bien-fondée.

Par l'absurde, s'il existait une chaîne infinie telle que $\forall i \geq 0 .\ (\langle \lambda_i, s_i, s_i' \rangle \succ \langle \lambda_{i+1}, s_{i+1}, s_{i+1}' \rangle)$ alors d'après (c) et (b) la chaîne $\langle \lambda_i, \delta m(\lambda_i)(s_i, s_i') \rangle$, $i \geq 0$ sera strictement décroissante pour l'ordre lexicographique gauche sur des paires d'ordinaux, une contradiction.

(e) Il s'ensuit à partir de (d) que nous pouvons définir $\delta \in (\Lambda \to (S \times S \to \underset{\sim}{Ord}))$ par induction transfinie de sorte que pour tout $\lambda \in \Lambda$, $s, s' \in S$ nous ayons :

$$\begin{array}{l} \delta(\lambda)(s,s') = \underset{\sim}{\sup} \{ \alpha + 1 : \neg \theta_\lambda(s,s') \wedge ([\exists s'' \in S .\ (HS(\lambda)(s,s',s'') \wedge \alpha = \delta(\lambda)(s,s''))] \vee \\ \qquad [\exists \lambda' < \lambda .\ ((\exists s'' \in S .\ LI(\lambda)(s,s',\lambda',s'')) \\ \qquad\qquad \wedge \alpha = (\underset{\sim}{\sup} \{ \delta(\lambda)(s,s'') : \exists s'' \in S .\ LI(\lambda)(s,s',\lambda',s'') \} + \delta(\lambda')(s',s')))]) \} \end{array}$$

Intuitivement, si l'exécution commence dans un état s et atteint un état s' alors "dans au plus $\delta(\lambda)(s,s')$ pas" l'exécution atteindra fatalement un certain état s'' satisfaisant le lemme $\theta_\lambda(s,s'')$. En particulier $\theta_\lambda(s,s')$ implique $\delta(\lambda)(s,s') = 0$.

Nous choisissons :

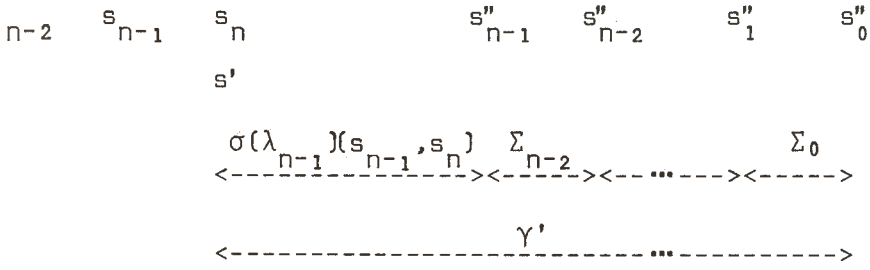
$$(f)\quad J(\gamma',\Delta,\Delta') = [\exists m\in(\omega\vee 0),\ \lambda\in(m\to\Lambda),\ \Delta_0\in S,\ \dots,\Delta_m\in S,\ \Sigma\in(m-1\to \underset{\sim}{Ord}).$$

$$\begin{aligned}
&(\Delta_0=\Delta)\wedge(\lambda_0=\pi)\wedge(\Delta_m=\Delta')\\
&\wedge\ \forall i\in(m\vee 0).\ \exists\Delta''_i\in S.\ LI(\lambda_{i-1})(\Delta_{i-1},\Delta_i,\lambda_i,\Delta''_i)\\
&\wedge\ I_{\lambda_{m-1}}(\delta m(\lambda_{m-1})(\Delta_{m-1},\Delta_m),\Delta_{m-1},\Delta_m)\\
&\wedge\ \forall i\in(m\vee 0).\ \neg\theta_{\lambda_i}(\Delta_i,\Delta_{i+1})\\
&\wedge\ \forall i\in(m\vee 0).(\Sigma_{i-1}=\underset{\sim}{sup}\{\delta(\lambda_{i-1})(\Delta_{i-1},\Delta''):LI(\lambda_{i-1})(\Delta_{i-1},\Delta_i,\lambda_i,\Delta'')\})\\
&\wedge\ \gamma'=(\Sigma_0+\dots+\Sigma_{m-2}+\delta(\lambda_{m-1})(\Delta_{m-1},\Delta_m))]
\end{aligned}$$

$$(g)\quad \Gamma = \underset{\sim}{sup}^{+}\{\gamma'\in\underset{\sim}{Ord}:\ \exists\Delta,\Delta'\in S.\ J(\gamma',\Delta,\Delta')\}$$

$J(\gamma',\Delta,\Delta')$ est choisi de façon à exprimer que "si l'exécution commence dans l'état Δ et atteint plus tard l'état Δ' alors il atteindra fatalement "dans au plus γ' pas" un état Δ'' satisfaisant $\Psi(\Delta,\Delta'')$". Puisque nous considérons le non-déterminisme non-borné, la terminaison peut être faible. Donc la phrase "dans au plus γ' pas" ne doit pas être prise littéralement lorsque nous devons recourir à des ordinaux transfinis $\gamma'>\omega$. Dans ce cas, nous pouvons démontrer la terminaison faible seulement c'est-à-dire trouver une classe bien fondée $\langle W,\prec\rangle$ et une fonction de terminaison $f\in(S\times S\to W)$ telle que $\exists a\in A.\ t_a(\Delta',\Delta'')\Rightarrow(f(\Delta,\Delta')\prec f(\Delta,\Delta''))$. Alors la phrase "dans au plus γ' pas" est une abbréviation pour la phrase plus rigoureuse "γ' est le rang $\rho(f(\Delta,\Delta'))$ de $f(\Delta,\Delta')$". De plus, nous choisissons $\langle\Gamma,<\rangle$ pour $\langle W,\prec\rangle$ et laissons f implicite. (On aurait pu définir f comme étant le plus petit ordinal γ' pour lequel $J(\gamma',\Delta,\Delta')$ est vrai).

La formule (f) peut être expliquée informellement au moyen du diagramme suivant (où seulement les chemins d'exécution commençant en Δ ont été considérés) :



Dans la preuve de fatalité du lemme θ_{λ_i}, $i=0,\ldots,m-2$, il a été montré que commençant dans l'état s_i, satisfaisant l'assertion intermittente $I_{\lambda_i}(\delta m(\lambda_i)(s_i,s_i''), s_i, s_i'')$, l'exécution atteindra l'état s_{i+1} satisfaisant $I_{\lambda_i}(\delta m(\lambda_i)(s_i,s_{i+1}''), s_i, s_{i+1}'')$. Appliquant $\theta_{\lambda_{i+1}}$ en ce point, il a été montré que l'exécution atteindra fatalement un certain état s_{i+1}'' tel que $\theta_{\lambda_{i+1}}(s_{i+1}, s_{i+1}'')$ soit vrai et satisfaisant l'assertion intermittente $I_{\lambda_i}(\delta m(\lambda_i)(s_i,s_{i+1}''), s_i, s_{i+1}'')$. Alors $LI(\lambda_i)(s_i, s_{i+1}, \lambda_{i+1}, s_{i+1}'')$ est vrai. Puis, à partir de la dernière assertion intermittente, on a déduit que $I_{\lambda_i}(\delta m(\lambda_i)(s_i,s_i''), s_i, s_i'')$ qui implique $\theta_{\lambda_i}(s_i, s_i'')$ et termine la preuve. De plus l'exécution conduit de s_{i+1}'' à s_i'' en "au plus $\sigma(\lambda_i)(s_i,s_{i+1}'')$ pas" donc "en au plus Σ_i pas" quand on considère tous les états s_{i+1}'' possibles.

Finalement, dans la preuve du lemme $\theta_{\lambda_{m-1}}$, il a été montré que commençant dans l'état s_{m-1}, l'exécution atteindra l'état $s_m = s'$ et à partir de s_m, atteindra l'état s_{n+1}'' en "au plus $\sigma(\lambda_{m-1})(s_{m-1}, s_m)$ pas".

Après "élimination de la récursivité", si l'exécution commence en $s_0 = s$ et atteint $s_1, \ldots, s_{m-1}, s_m = s'$ alors "en au plus γ pas" elle traversera $s_{m-1}'', \ldots, s_0''$ tels que $\Psi(s, s_0'')$.

(k) Preuve de $(\mathcal{F}_6.1)$:

D'après $(\mathcal{B}_7.1)$, $\forall \Delta \in S.\ \exists \delta \in \Delta.\ I_\pi(\delta, \Delta, \Delta)$ de sorte que d'après (a) nous avons $\forall \Delta \in S.\ I_\pi(\delta m(\pi)(\Delta,\Delta), \Delta, \Delta)$ qui implique $\forall \Delta \in S.\ J(\sigma(\pi)(\Delta,\Delta), \Delta, \Delta)$ en choisissant $n=1$, $\lambda_0 = \pi$, $A_0 = A_1 = \Delta$.

La preuve de $(\mathcal{F}_6.2)$ se décompose aux lemmes suivants :

(i) $\forall \lambda \in \Lambda,\ \Delta, \Delta' \in S.\ (I_\lambda(\delta m(\lambda)(\Delta,\Delta'), \Delta, \Delta') \Rightarrow [\exists \Delta'' \in S, a \in A.\ t_a(\Delta', \Delta'') \vee \theta_\lambda(\Delta, \Delta')])$

Par l'absurde supposons $I_\lambda(\delta m(\lambda)(\Delta,\Delta'), \Delta, \Delta')$, $\neg\theta_\lambda(\Delta,\Delta')$ et $\forall \Delta'' \in S, a \in A.\ \neg t_a(\Delta', \Delta'')$. La contradiction est que nous pouvons construire par induction une chaîne strictement décroissante $\lambda, \lambda_1, \lambda_2, \ldots$ d'ordinaux, comme suit :

Puisque $I_\lambda(\delta m(\lambda)(\Delta,\Delta'), \Delta, \Delta')$ est vrai mais ni $(\mathcal{B}_7.3.a)$ ni $(\mathcal{B}_7.3.c)$ ne s'appliquent, $(\mathcal{B}_7.3.b)$ implique l'existence de $\lambda_1 < \lambda$ tel que $\forall \Delta'' \in S.\ (\theta_{\lambda_1}(\Delta', \Delta'') \Rightarrow \exists \delta'' < \delta m(\lambda)(\Delta,\Delta').\ I_\lambda(\delta'', \Delta, \Delta''))$. D'après $(\mathcal{B}_7.2)$ et (a), $I_{\lambda_1}(\delta m(\lambda_1)(\Delta',\Delta'), \Delta', \Delta')$ est vrai et nous ne pouvons pas avoir $\theta_{\lambda_1}(\Delta', \Delta')$ car autrement $\exists \delta'' < \delta m(\lambda)(\Delta,\Delta').\ I_\lambda(\delta'', \Delta, \Delta')$, en contradiction avec (a).

- Supposons que nous ayons construit une séquence finie strictement décroissante $\lambda, \lambda_1, \ldots, \lambda_i$, $i \geq 0$ telle que $I_{\lambda_i}(\delta m(\lambda_i)(\Delta',\Delta'), \Delta', \Delta') \wedge \neg\theta_{\lambda_i}(\Delta', \Delta')$ soit vrai. Nous pouvons la prolonger par λ_{i+1} puisque $(\mathcal{B}_7.3.a)$ et $(\mathcal{B}_7.3.c)$ ne s'appliquant pas, $(\mathcal{B}_7.3.b)$ implique l'existence d'un $\lambda_{i+1} < \lambda_i$ tel que $\forall \Delta'' \in S.\ (\theta_{\lambda_{i+1}}(\Delta', \Delta'') \Rightarrow \exists \delta'' < \delta m(\lambda_i)(\Delta',\Delta'').\ I_{\lambda_i}(\delta'', \Delta', \Delta''))$ donc $\neg\theta_{\lambda_{i+1}}(\Delta', \Delta')$. De plus $I_{\lambda_{i+1}}(\delta m(\lambda_{i+1})(\Delta',\Delta'), \Delta', \Delta')$ dérive de $(\mathcal{B}_7.2)$ et (a). Q.E.D.

(j) $\forall \gamma' \in \Gamma,\ \Delta, \Delta' \in S.\ [(J(\gamma', \Delta, \Delta') \wedge \neg\Psi(\Delta, \Delta')) \Rightarrow \exists \Delta'' \in S, a \in A.\ t_a(\Delta', \Delta'')]$

Lorsque $J(\gamma', \Delta, \Delta')$ est vrai, nous avons $I_{\lambda_{m-1}}(\delta m(\lambda_{m-1})(\Delta_{m-1}, \Delta_m); \Delta_{m-1}, \Delta_m)$. De plus $\neg\theta_{\lambda_{m-1}}(\Delta_{m-1}, \Delta_m)$ est vrai quand $n>1$ mais aussi quand $n=1$ car $\Psi(\Delta,\Delta') = \theta_\pi(\Delta,\Delta') = \theta_{\lambda_0}(\Delta_0, \Delta_1)$. Maintenant (j) dérive de (i). Q.E.D.

(k) $\forall \lambda \in \Lambda,\ \delta \in \Delta,\ A, A' \in S.\ [I_\lambda(\delta, A, A') \Rightarrow \exists A'' \in S.\ \theta_\lambda(A, A'')]$

Par induction transfinie sur λ, supposons (k) vrai pour $\forall \lambda' < \lambda$. Nous montrons que (k) est vrai pour λ, par l'absurde. Supposons donc $\forall A'' \in S. \neg\theta_\lambda(A, A'')$. Nous avons $I_\lambda(\delta_0, A, A'_0)$ avec $\delta_0 = \delta$ et $A'_0 = A'$. Supposons avoir construit une chaîne $\delta_0 > \dots > \delta_k$ avec $I_\lambda(\delta_k, A, A'_k)$. Puisque $\neg\theta_\lambda(A, A'_k)$, alors nous avons d'après ($\mathcal{B}_7$.3.a) ou ($\mathcal{B}_7$.3.b) et l'hypothèse d'induction, $\exists \delta_{k+1} < \delta_k .\ I_\lambda(\delta_{k+1}, A, A'_{k+1})$. La contradiction est que, de cette manière, nous pouvons construire une chaîne infinie strictement décroissante d'ordinaux. Q.E.D.

(r) $\forall \gamma' \in \Gamma,\ A, A', A'' \in S.\ ([J(\gamma', A, A') \wedge \neg\Psi(A, A') \wedge \exists q \in A.\ t_q(A', A'')] \Rightarrow [\exists \gamma'' < \gamma'.\ J(\gamma'', A, A'')])$

Supposant $[J(\gamma', A, A') \wedge \neg\Psi(A, A') \wedge \exists q \in A.\ t_q(A', A'')]$ nous avons $A_m = A'$, $I_{\lambda_{m-1}}(\delta m(\lambda_{m-1})(A_{m-1}, A_m), A_{m-1}, A_m)$ et $\neg\theta_{\lambda_{m-1}}(A_{m-1}, A_m)$. ($\mathcal{B}_7$.3.c) ne s'appliquant pas, deux cas seulement sont à considérer :

(r.1) ($\mathcal{B}_7$.3.a) s'applique à $I_{\lambda_{m-1}}(\delta m(\lambda_{m-1})(A_{m-1}, A_m), A_{m-1}, A_m)$, de même ($\mathcal{B}_7$.3.b) pour un $\lambda' < \lambda_{m-1}$ tel que $\theta_{\lambda'}(A_{m-1}, A'')$.

Dans le premier cas nous avons $HS(\lambda_{m-1})(A_{m-1}, A', A'')$ et dans le second $LI(\lambda_{m-1})(A_{m-1}, A', \lambda', A'')$. Dans les deux cas (b) implique $I_{\lambda_{m-1}}(\delta m(\lambda_{m-1})(A_{m-1}, A''), A_{m-1} A'')$ et à partir de $J(\gamma', A, A')$ donc $\neg\theta_{\lambda_{m-1}}(A_{m-1}, A')$ et (e) nous dérivons $\sigma(\lambda_{m-1})(A_{m-1}, A') > \sigma(\lambda_{m-1})(A_{m-1}, A'')$.

(r.1.1) Cas $\neg\theta_{\lambda_{m-1}}(A_{m-1}, A'')$

Nous venons de démontrer que $\sigma(\lambda_{m-1})(A_{m-1}, A') > \sigma(\lambda_{m-1})(A_{m-1}, A'')$ donc $\gamma'' = (\Sigma_0 + \dots + \Sigma_{m-2} + \sigma(\lambda_{m-1})(A_{m-1}, A'')) < (\Sigma_0 + \dots + \Sigma_{m-2} + \sigma(\lambda_{m-1})(A_{m-1}, A')) = \gamma'$. Si nous posons que $J(\gamma'', A, A'')$ est égal à $J(\gamma', A, A')$ avec γ'', A'' substituées à γ', A' alors nous avons $\gamma'' < \gamma' \wedge J(\gamma'', A, A'')$.

(7.1.2) Cas $\theta_{\lambda_{m-1}}(\Delta_{m-1}, \Delta'')$

Soit ℓ le plus petit nombre naturel tel que $\ell < m$ et $\theta_{\lambda_\ell}(\Delta_\ell, \Delta'')$ soit vrai de sorte que si $\ell > 0$ alors nous avons $\neg\theta_{\lambda_{\ell-1}}(\Delta_{\ell-1}, \Delta'')$. Intuitivement, considérer la transition $t_a(\Delta', \Delta'')$ dans la preuve provoque un retour des lemmes $\theta_{\lambda_{m-1}}, \ldots, \theta_{\lambda_\ell}$ utilisés recursivement.

Notons que d'après (e), nous avons $\sigma(\lambda_j)(\Delta_j, \Delta'') = 0$ pour $j = \ell, \ldots, m-1$.

Montrons maintenant que $I_{\lambda_j}(\delta m(\lambda_j)(\Delta_j, \Delta''), \Delta_j, \Delta'')$ est vrai pour $j = \sup(\ell-1, 0), \ldots, m-1$. le cas $j = m-1$ a été déja considéré. Si $\sup(0, \ell-1) \leq j < m-1$ alors $J(\gamma', \Delta, \Delta')$ implique $\exists \Delta''' \in S.\ LI(\lambda_j)(\Delta_j, \Delta_{j+1}, \Delta_{j+1}, \Delta''')$ donc (b), $\theta_{\lambda_{j+1}}(\Delta_{j+1}, \Delta'')$ et (a) entraînent $I_{\lambda_j}(\delta m(\lambda_j)(\Delta_j, \Delta''), \Delta_j, \Delta'')$.

(7.1.2.1) Cas $\ell = 0$

Définissons $J(\gamma'', \Delta, \Delta'')$ au moyen de la formule (f) en choisissant $m = 1$, $\lambda_0 = \pi$, $\Delta_0 = \Delta$, $\Delta_1 = \Delta''$ et $\gamma'' = \sigma(\lambda_0)(\Delta, \Delta'') = 0$. $J(\gamma'', \Delta, \Delta'')$ est vrai parce qu'il revient à $I_{\lambda_0}(\delta m(\lambda_0)(\Delta_0, \Delta''), \Delta_0, \Delta'')$. De plus $J(\gamma', \Delta, \Delta')$ implique que $\gamma' = \Sigma_0 + \ldots + \Sigma_{m-2} + \sigma(\lambda_{m-1})(\Delta_{m-1}, \Delta')$. Donc $\sigma(\lambda_{m-1})(\Delta_{m-1}, \Delta') > \sigma(\lambda_{m-1})(\Delta_{m-1}, \Delta'')$ implique $\gamma' > 0 = \gamma''$.

(7.1.2.2) Cas $\ell > 0$ donc $m > 1$

Définissons $J(\gamma'', \Delta, \Delta'')$ au moyen de la formule (f) en choisissant $m = \ell$, $\lambda_0, \ldots, \lambda_{\ell-1}$, $\Delta_0, \ldots, \Delta_{\ell-1}$ et $\Sigma_0, \ldots, \Sigma_{\ell-2}$ comme définis par $J(\gamma', \Delta, \Delta')$, $\Delta_\ell = \Delta''$ et $\gamma'' = (\Sigma_0 + \ldots + \Sigma_{\ell-2} + \sigma(\lambda_{\ell-1})(\Delta_{\ell-1}, \Delta''))$. Alors $J(\gamma'', \Delta, \Delta'')$ est vrai car nous avons déjà montré que $I_{\lambda_{\ell-1}}(\delta m(\lambda_{\ell-1})(\Delta_{\ell-1}, \Delta_\ell), \Delta_{\ell-1}, \Delta_\ell)$ est vrai et $\neg\theta_{\lambda_{\ell-1}}(\Delta_{\ell-1}, \Delta_\ell)$ découle de la définition de ℓ et Δ_ℓ.

D'après $\sigma(\lambda_{m-1})(\Delta_{m-1}, \Delta') > \sigma(\lambda_{m-1})(\Delta_{m-1}, \Delta'') = 0$, nous dérivons $(\Sigma_\ell + \ldots + \Sigma_{m-2} + \sigma(\lambda_{m-1})(\Delta_{m-1}, \Delta')) > 0$. Nous savons que $\theta_{\lambda_\ell}(\Delta_\ell, \Delta'')$ est vrai et que $J(\gamma', \Delta, \Delta')$ implique $\exists \Delta'_\ell \in S.\ LI(\lambda_{\ell-1})(\Delta_{\ell-1}, \Delta_\ell, \lambda_\ell, \Delta'_\ell)$ donc d'après (b), $LI(\lambda_{\ell-1})(\Delta_{\ell-1}, \Delta_\ell, \lambda_\ell, \Delta'')$. Par définition de $\Sigma_{\ell-1}$ nous déduisons $\Sigma_{\ell-1} \geq \sigma(\lambda_{\ell-1})(\Delta_{\ell-1}, \Delta'')$. Il s'ensuit que $\gamma'' = (\Sigma_0 + \ldots + \Sigma_{\ell-2} + \sigma(\lambda_{\ell-1})(\Delta_{\ell-1}, \Delta'')) \leq (\Sigma_0 + \ldots + \Sigma_{\ell-2} + \Sigma_{\ell-1}) < (\Sigma_0 + \ldots + \Sigma_{\ell-1} + \Sigma_\ell + \ldots + \Sigma_{m-2} + \sigma(\lambda_{m-1})(\Delta_{m-1}, \Delta')) = \gamma'$.

(r.2) $(\mathcal{B}_7.3.b)$ s'applique à $I_{\lambda_{m-1}}(\delta m(\lambda_{m-1})(A_{m-1},A_m),A_{m-1},A_m)$ pour un $\lambda'<\lambda_{m-1}$ tel que $\neg\theta_{\lambda'}(A_m,A'')$.

Intuitivement, nous obtenons $J(\gamma'',A,A'')$ à partir de $J(\gamma',A,A'')$ en conservant "dans une pile" tous les lemmes applicables en A_m (sauf les $\theta_{\lambda'}$ tels que $\theta_{\lambda'}(A_m,A_m)$ ou bien $\theta_{\lambda'}(A_m,A'')$) et en considérant la transition $t_a(A_m,A'')$.

Si nous posons λ_m égal à λ', alors d'après (r.2) nous avons $\lambda_m<\lambda_{m-1}$ et $\forall A'''\in S.\,[\theta_{\lambda_m}(A_m,A''')\Rightarrow\exists\delta'''<\delta m(\lambda_{m-1})(A_{m-1},A_m).\,I_{\lambda_{m-1}}(\delta''',A_{m-1},A_m)]$. Il s'ensuit que $\neg\theta_{\lambda_m}(A_m,A_m)$ car sinon $\exists\delta'''<\delta m(\lambda_{m-1})(A_{m-1},A_m).\,I_{\lambda_{m-1}}(\delta''',A_{m-1},A_m)$ en contradiction avec la définition (a) de $\delta m(\lambda_{m-1})(A_{m-1},A_m)$. De plus $(\mathcal{B}_7.2)$ implique $I_{\lambda_m}(\delta m(\lambda_m)(A_m,A_m),A_m,A_m)$.

Supposons avoir construit une chaîne $\lambda_{m+k}<\ldots<\lambda_m<\lambda_{m-1}$ avec $k\geq 0$ telle que $\forall j\in(k+1).\,[(\forall A'''\in S.\,[\theta_{\lambda_{m+j}}(A_m,A''')\Rightarrow\exists\delta'''<\delta m(\lambda_{m+j-1})(A_{m-K^0_j},A_m).\,I_{\lambda_{m+j-1}}(\delta''',A_{m-K^0_j},A''')]\wedge\neg\theta_{\lambda_{m+j}}(A_m,A_m)\wedge I_{\lambda_{m+j}}(\delta m(\lambda_{m+j})(A_m,A_m),A_m,A_m)]$, où $K^\ell_j=(j=\ell\to 1|0)$. Si $(\mathcal{B}_7.3.b)$ s'applique à $I_{\lambda_{m+k}}(\delta m(\lambda_{m+k})(A_m,A_m),A_m,A_m)$ alors il existe $\lambda_{m+k+1}<\lambda_{m+k}$ tel que $\forall A'''\in S.\,[\theta_{\lambda_{m+k+1}}(A_m,A''')\Rightarrow\exists\delta'''<\delta m(\lambda_{m+k})(A_m,A_m).\,I_{\lambda_{m+k}}(\delta''',A_m,A''')]$ donc $\neg\theta_{\lambda_{m+k+1}}(A_m,A_m)$ d'après (a) et $I_{\lambda_{m+k+1}}(\delta m(\lambda_{m+k+1})(A_m,A_m),A_m,A_m)$ d'après $(\mathcal{B}_7.2)$.

Puisque la chaîne $\lambda_m,\ldots,\lambda_{m+k},\ldots$ d'ordinaux est strictement décroissante, elle doit être finie de sorte qu'il existe un k que nous notons K pour lequel $(\mathcal{B}_7.3.b)$ ne s'applique pas à $I_{\lambda_{m+K}}(\delta m(\lambda_{m+K}(A_m,A_m),A_m,A_m)$. $(\mathcal{B}_7.3.c)$ ne s'applique pas non plus car $\theta_{\lambda_{m+K}}(A_m,A_m)$ n'est pas vrai. Il s'ensuit que $(\mathcal{B}_7.3.a)$ s'applique de sorte que nous avons $HS(\lambda_{m+K})(A_m,A_m,A'')$. De plus, $\exists A''_{m+j+1}\in S.\,LI(\lambda_{m+j})(A_{m-K^{-1}_j},A_m,\lambda_{m+j+1},A''_{m+j+1})$ découle de $I_{\lambda_{m+j}}(\delta m(\lambda_{m+j})(A_{m-K^{-1}_j},A_m),A_{m-K^{-1}_j},A_m)$ pour $j=-1,\ldots,K-1$ et (k).

Nous devons trouver maintenant γ'' et $J(\gamma'',A,A'')$ définis d'après la formule (f) tels que $(\gamma''<\gamma'\wedge J(\gamma'',A,A''))$. Nous pouvons les dériver à partir de γ' et $J(\gamma',A,A')$ comme suit :

(r.2) $(\mathcal{B}_7.3.b)$ s'applique à $I_{\lambda_{m-1}}(\delta m(\lambda_{m-1})(A_{m-1},A_m),A_{m-1},A_m)$ pour un $\lambda'<\lambda_{m-1}$ tel que $\neg\theta_{\lambda'}(A_m,A'')$.

Intuitivement, nous obtenons $J(\delta'',A,A'')$ à partir de $J(\delta',A,A'')$ en conservant "dans une pile" tous les lemmes applicables en A_m (sauf les $\theta_{\lambda'}$ tels que $\theta_{\lambda'}(A_m,A_m)$ ou bien $\theta_{\lambda'}(A_m,A'')$) et en considérant la transition $t_a(A_m,A'')$.

Si nous posons λ_m égal à λ', alors d'après (r.2) nous avons $\lambda_m<\lambda_{m-1}$ et $\forall A'''\in S.\,[\theta_{\lambda_m}(A_m,A''')\Rightarrow\exists\delta'''<\delta m(\lambda_{m-1})(A_{m-1},A_m).\,I_{\lambda_{m-1}}(\delta''',A_{m-1},A_m)]$. Il s'ensuit que $\neg\theta_{\lambda_m}(A_m,A_m)$ car sinon $\exists\delta'''<\delta m(\lambda_{m-1})(A_{m-1},A_m).\,I_{\lambda_{m-1}}(\delta''',A_{m-1},A_m)$ en contradiction avec la définition (a) de $\delta m(\lambda_{m-1})(A_{m-1},A_m)$. De plus $(\mathcal{B}_7.2)$ implique $I_{\lambda_m}(\delta m(\lambda_m)(A_m,A_m),A_m,A_m)$.

Supposons avoir construit une chaîne $\lambda_{m+k}<\ldots<\lambda_m<\lambda_{m-1}$ avec $k\geq 0$ telle que $\forall j\in(k+1).\,[(\forall A'''\in S.\,[\theta_{\lambda_{m+j}}(A_m,A''')\Rightarrow\exists\delta'''<\delta m(\lambda_{m+j-1})(A_{m-K^0_j},A_m).\,I_{\lambda_{m+j-1}}(\delta''',A_{m-K^0_j},A''')]\wedge\neg\theta_{\lambda_{m+j}}(A_m,A_m)\wedge I_{\lambda_{m+j}}(\delta m(\lambda_{m+j})(A_m,A_m),A_m,A_m)]$, où $K^\ell_j=(j=\ell\to 1|0)$. Si $(\mathcal{B}_7.3.b)$ s'applique à $I_{\lambda_{m+k}}(\delta m(\lambda_{m+k})(A_m,A_m),A_m,A_m)$ alors il existe $\lambda_{m+k+1}<\lambda_{m+k}$ tel que $\forall A'''\in S.\,[\theta_{\lambda_{m+k+1}}(A_m,A''')\Rightarrow\exists\delta'''<\delta m(\lambda_{m+k})(A_m,A_m).\,I_{\lambda_{m+k}}(\delta''',A_m,A''')]$ donc $\neg\theta_{\lambda_{m+k+1}}(A_m,A_m)$ d'après (a) et $I_{\lambda_{m+k+1}}(\delta m(\lambda_{m+k+1})(A_m,A_m),A_m,A_m)$ d'après $(\mathcal{B}_7.2)$.

Puisque la chaîne $\lambda_m,\ldots,\lambda_{m+k},\ldots$ d'ordinaux est strictement décroissante, elle doit être finie de sorte qu'il existe un k que nous notons K pour lequel $(\mathcal{B}_7.3.b)$ ne s'applique pas à $I_{\lambda_{m+K}}(\delta m(\lambda_{m+K}(A_m,A_m),A_m,A_m)$. $(\mathcal{B}_7.3.c)$ ne s'applique pas non plus car $\theta_{\lambda_{m+K}}(A_m,A_m)$ n'est pas vrai. Il s'ensuit que $(\mathcal{B}_7.3.a)$ s'applique de sorte que nous avons $HS(\lambda_{m+K})(A_m,A_m,A'')$. De plus, $\exists A''_{m+j+1}\in S.\,LI(\lambda_{m+j})(A_{m-K^{-1}_j},A_m,\lambda_{m+j+1},A''_{m+j+1})$ découle de $I_{\lambda_{m+j}}(\delta m(\lambda_{m+j})(A_{m-K^{-1}_j},A_m),A_{m-K^{-1}_j},A_m)$ pour $j=-1,\ldots,K-1$ et (k).

Nous devons trouver maintenant δ'' et $J(\delta'',A,A'')$ définis d'après la formule (f) tels que $(\delta''<\delta'\wedge J(\delta'',A,A''))$. Nous pouvons les dériver à partir de δ' et $J(\delta',A,A')$ comme suit :

Puisque $\neg\theta_{\lambda_m}(\Delta_m, \Delta_m)$ est vrai alors il existe un plus grand nombre naturel ℓ tel que $0 < \ell \leq K$ et $\neg\theta_{\lambda_{m+\ell}}(\Delta_m, \Delta'')$. Définissons $J(\gamma'', \Delta, \Delta'')$ au moyen de la formule (f) où n est $m+\ell+1$; $\lambda_j, j \in (n+\ell+1)$, $\Delta_j, j \in (m+1)$, $\Sigma'_j, j \in (m-1)$ sont définis comme ci-dessus tandis que $\Delta' = \Delta_m = \Delta_{m+1} = \ldots = \Delta_{m+\ell}$, $\Delta_{m+\ell+1} = \Delta''$, $\Sigma'_j = \sup\{\sigma(\lambda_j)(\Delta_j, \Delta''') : LI(\lambda_j)(\Delta_j, \Delta_{j+1}, \lambda_{j+1}, \Delta''')\}$, $j = m-1, \ldots, m+\ell-1$ et $\gamma'' = (\Sigma'_0 \dot{+} \ldots \dot{+} \Sigma'_{m+\ell-1} \dot{+} \sigma(\lambda_{m+\ell})(\Delta_{m+\ell}, \Delta_{m+\ell+1}))$.

Nous avons déjà démontré que $\forall i = m, \ldots, m+\ell, \ldots, m+K.\ \exists \Delta''_i \in S.$ $LI(\lambda_{i-1})(\Delta_{i-1}, \Delta_i, \lambda_i, \Delta''_i)$ et $\forall i = m, \ldots, m+\ell-1, \ldots, m+K-1.\ \neg\theta_{\lambda_i}(\Delta_i, \Delta_{i+1})$. De plus nous avons $\neg\theta_{\lambda_{m+\ell}}(\Delta_{m+\ell}, \Delta_{m+\ell+1})$ par définition de ℓ, $\Delta_{m+\ell}$ et $\Delta_{m+\ell+1}$. Nous avons aussi $I_{\lambda_{m+\ell}}(\delta m(\lambda_{m+\ell})(\Delta_{m+\ell}, \Delta_{m+\ell+1}), \Delta_{m+\ell}, \Delta_{m+\ell+1})$ qui est vrai, il est impliqué par $HS(\lambda_{m+K})(\Delta_m, \Delta_m, \Delta'')$ quand $\ell = K$, autrement $\ell < K$ et $\theta_{\lambda_{m+\ell+1}}(\Delta_m, \Delta'')$ implique $\exists \delta'''.\ I_{\lambda_{m+\ell}}(\delta''', \Delta_{m-K^\circ_{\ell+1}}, \Delta'')$ donc d'après (a), $I_{\lambda_{m+\ell}}(\delta m(\lambda_{m+\ell})(\Delta_m, \Delta''), \Delta_m, \Delta'')$ est vrai. Nous concluons que $J(\gamma'', \Delta, \Delta'')$ est vrai.

Il reste à montrer que $\gamma'' < \gamma'$. Nous avons montré que $\forall j = -1, \ldots, K-1$ il existe un certain Δ''_{m+j+1} tel que $LI(\lambda_{m+j})(\Delta_{m-K^{-1}_j}, \Delta_m, \lambda_{m+j+1}, \Delta''_{m+j+1}) = LI(\lambda_{m+j})(\Delta_{m+j}, \Delta_{m+j+1}, \lambda_{m+j+1}, \Delta''_{m+j+1})$. De plus $\forall i = 1, \ldots, m+\ell$ nous avons $\neg\theta_{\lambda_i}(\Delta_i, \Delta_{i+1})$ de sorte que d'après (e), $\sigma(\lambda_{m+j})(\Delta_{m+j}, \Delta_{m+j+1}) > (\sup\{\sigma(\lambda_{m+j})(\Delta_{m+j}, \Delta''') : LI(\lambda_{m+j})(\Delta_{m+j}, \Delta_{m+j+1}, \lambda_{m+j+1}, \Delta''')\} \dot{+} \sigma(\lambda_{m+j+1})(\Delta_{m+j+1}, \Delta_{m+j+1})) = (\Sigma_{m+j} \dot{+} \sigma(\lambda_{m+j+1})(\Delta_{m+j+1}, \Delta_{m+j+1}))$ pour tout $j = -1, \ldots, \ell-1$. Grâce à cette inégalité et à $\Delta_{m+j+1} = \Delta_{m+j+2}$ pour $j = -1, \ldots, \ell-2$, nous obtenons $\sigma(\lambda_{m-1})(\Delta_{m-1}, \Delta_m) > (\Sigma_{m-1} \dot{+} \sigma(\lambda_m)(\Delta_m, \Delta_{m+1}) > \ldots > (\Sigma_{m-1} \dot{+} \ldots \dot{+} \Sigma_{m+\ell-1} \dot{+} \sigma(\lambda_{m+\ell})(\Delta_{m+\ell}, \Delta_{m+\ell}))$. Si $\ell = K$ alors $HS(\lambda_{m+K})(\Delta_m, \Delta_m, \Delta'')$, $\Delta_{m+\ell} = \Delta_m$, $\neg\theta_{\lambda_{m+\ell}}(\Delta_m, \Delta'')$ et (e) impliquent $\sigma(\lambda_{m+\ell})(\Delta_{m+\ell}, \Delta_{m+\ell}) > \sigma(\lambda_{m+\ell})(\Delta_{m+\ell}, \Delta'')$. Autrement $\ell < K$ et nous avons $\neg\theta_{\lambda_{m+\ell}}(\Delta_m, \Delta_m)$, $LI(\lambda_{m+\ell})(\Delta_m, \Delta_m, \lambda_{m+\ell+1}, \Delta'')$, $\Delta_{m+\ell} = \Delta_m$ de sorte que d'après (e), $\sigma(\lambda_{m+\ell})(\Delta_{m+\ell}, \Delta_{m+\ell}) = \sigma(\lambda_{m+\ell})(\Delta_m, \Delta_m) > \sigma(\lambda_{m+\ell})(\Delta_m, \Delta'') = \sigma(\lambda_{m+\ell})(\Delta_{m+\ell}, \Delta'')$. Dans les deux cas, nous concluons que $\gamma' = (\Sigma'_0 \dot{+} \ldots \dot{+} \Sigma'_{m-2} \dot{+} \sigma(\lambda_{m-1})(\Delta_{m-1}, \Delta_m)) > (\Sigma_0 \dot{+} \ldots \dot{+} \Sigma_{m-2} \dot{+} \Sigma_{m-1} \dot{+} \ldots \dot{+} \Sigma_{m+\ell-1} \dot{+} \sigma(\lambda_{m+\ell})(\Delta_{m+\ell}, \Delta'')) = \gamma''$. Q.E.D.

□

Exemple 5.4 - 1

Le système de transition $\langle S, A, t, \Pi \rangle$ correspondant au programme suivant (pris littéralement dans Dijkstra[77]) :

```
do    odd(X) and X≥3  →  X := X+1
   ▯  even(X) and X≥2  →  X := X/2
od
```

défini par :

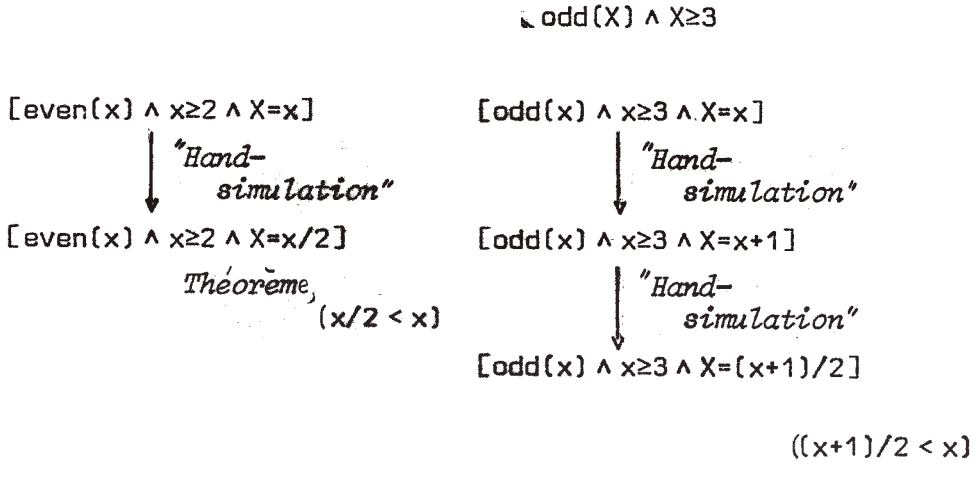
$S = \mathbb{Z}$

$A = \{a_w\}$

$t_a(x,x') = [(odd(x) \wedge x \geq 3 \wedge x' = x+1) \vee (even(x) \wedge x \geq 2 \wedge x' = x/2)]$

Une preuve que $\Psi(x,x') = [x' \leq 1]$ est fatale est donnée par la charte de preuve suivante :

Théorème : (par induction sur x)



Nous pouvons également utiliser le principe d'induction $(\mathcal{B}_7)$ avec :

$\Lambda = \omega+1$

$\theta_\omega = \psi$

$\theta_\lambda(x,x') = [(\lambda = \underline{\lambda}(x)) \Rightarrow (x' \leqslant 1)]$ quand $\lambda<\omega$ et $\underline{\lambda}(x) = (x \leqslant 1 \rightarrow 0 | x-1)$

$\pi = \omega$

$\Delta = 4$

$I_\omega(\delta,x,x') = [(\delta=1 \wedge x=x') \vee (\delta=0 \wedge x' \leqslant 1)]$

$I_\lambda(\delta,x,x') = [(\lambda=\underline{\lambda}(x)) \Rightarrow ((\delta=3 \wedge x'=x) \vee (\delta=2 \wedge t_{\underline{a}}(x,x')) \vee (\delta=1 \wedge odd(x) \wedge t_{\underline{a}}^2(x,x')) \vee (\delta=0 \wedge x' \leqslant 1))]$

D'après la définition (a) de $\delta m \in (\Lambda \rightarrow (S \times S \dashrightarrow \Delta))$, nous obtenons :

$\delta m(\lambda)(x,x') = 3$ ssi $[\lambda<\omega \wedge x=x' \wedge x'>1]$

$= 2$ ssi $[\lambda<\omega \wedge t_{\underline{a}}(x,x') \wedge x'>1]$

$= 1$ ssi $[(\lambda=\omega \wedge x=x'>1) \vee (\lambda<\omega \wedge odd(x) \wedge t_{\underline{a}}^2(x,x') \wedge x'>1)]$

$= 0$ ssi $[x' \leqslant 1]$

Le développement de (b) donne :

$HS(\omega)(x,x',x'') = [x=x' \wedge t_{\underline{a}}(x',x'') \wedge x'' \leqslant 1]$

quand $\lambda<\omega$,

$HS(\lambda)(x,x',x'') = [[(x=x') \vee (t_{\underline{a}}(x,x') \wedge [(\lambda=\underline{\lambda}(x)) \Rightarrow (odd(x) \vee x'' \leqslant 1)]) \vee (odd(x) \wedge t_{\underline{a}}^2(x,x') \wedge [(\lambda=\underline{\lambda}(x)) \Rightarrow (x'' \leqslant 1)])] \wedge t_{\underline{a}}(x',x'')]$

De la même manière :

$LI(\omega)(x,x',\lambda',x'') = [x'=x>1 \wedge \lambda'=\underline{\lambda}(x') \wedge x'' \leqslant 1]$

quand $\lambda<\omega$,

$LI(\lambda)(x,x',\lambda',x'') = [[(x=x') \vee t_{\underline{a}}(x,x') \vee (odd(x) \wedge t_{\underline{a}}^2(x,x'))] \wedge x'>1 \wedge \lambda>\lambda'=\underline{\lambda}(x') \wedge x'' \leqslant 1]$

(Noter que lorsque $(\mathcal{B}_7.3.a)$ et $(\mathcal{B}_7.3.b)$ sont tous deux vrais (par exemple quand $x=x'=2$ et $x''=1$) alors $HS(\lambda)(x,x',x'')$ et $LI(\lambda)(x,x',\lambda',x'')$ le sont aussi car il n'y a pas moyen de savoir laquelle de l'évaluation symbolique ou de l'induction sur les données a été utilisée).

Nous pouvons maintenant déterminer $\sigma(\lambda)(x,x')$. Parceque $LI(\lambda)(x,x',\lambda',x'')$ implique $\theta_\lambda(x,x'')$ donc $\sigma(\lambda)(x,x'')=0$, la formule (e) se réduit à :

$$\sigma(\lambda)(x,x') = \sup\{\alpha+1 : \neg\theta_\lambda(x,x') \wedge ([HS(\lambda)(x,x',x'') \wedge \alpha=\sigma(\lambda)(x,x'')] \vee [\exists\lambda'\in\Lambda, x''\in S.\ LI(\lambda)(x,x',\lambda',x'') \wedge \alpha=\sigma(\lambda')(x',x')])\}$$

Ce n'est pas nécessaire de chercher une définition non-récursive de σ, car nous n'aurons besoin que des propriétés suivantes :

- $t_{\underset{\sim}{a}}(x,x') \Rightarrow [\neg\theta_{\underline{\lambda}(x)}(x,x) \wedge HS(\underline{\lambda}(x))(x,x,x')] \Rightarrow [\sigma(\underline{\lambda}(x))(x,x') < \sigma(\underline{\lambda}(x))(x,x)]$

- $[even(x) \wedge t_{\underset{\sim}{a}}(x,x') \wedge t_{\underset{\sim}{a}}(x',x'')] \Rightarrow [\neg\theta_{\underline{\lambda}(x')}(x',x') \wedge HS(\underline{\lambda}(x'))(x',x',x'') \wedge \neg\theta_{\underline{\lambda}(x)}(x,x')$
  $\wedge \exists x'''.\ LI(\underline{\lambda}(x))(x,x',\underline{\lambda}(x'),x''')] \Rightarrow [\sigma(\underline{\lambda}(x'))(x',x'') < \sigma(\underline{\lambda}(x'))(x',x') < \sigma(\underline{\lambda}(x))(x,x')]$

- $[odd(x) \wedge t_{\underset{\sim}{a}}(x,x') \wedge t_{\underset{\sim}{a}}(x',x'')] \Rightarrow [(\exists x'''.\ LI(\underline{\lambda}(x))(x,x'',\underline{\lambda}(x''),x''') \vee \theta_{\underline{\lambda}(x)}(x,x''))$
  $\wedge HS(\underline{\lambda}(x))(x,x',x'')] \Rightarrow [(\sigma(\underline{\lambda}(x''))(x'',x'') < \sigma(\underline{\lambda}(x))(x,x') \vee \sigma(\underline{\lambda}(x''))(x'',x'')=0)$
  $\wedge \sigma(\underline{\lambda}(x))(x,x'') < \sigma(\underline{\lambda}(x))(x,x')] \Rightarrow [\sigma(\underline{\lambda}(x''))(x'',x'') < \sigma(\underline{\lambda}(x))(x,x')]$

- $[odd(x) \wedge t^2_{\underset{\sim}{a}}(x,x') \wedge t_{\underset{\sim}{a}}(x',x'')] \Rightarrow [\neg\theta_{\underline{\lambda}(x')}(x',x') \wedge HS(\underline{\lambda}(x'))(x',x',x'') \wedge \neg\theta_{\underline{\lambda}(x)}(x,x')$
  $\wedge \exists x'''.\ LI(\underline{\lambda}(x))(x,x',\underline{\lambda}(x'),x''')] \Rightarrow [\sigma(\underline{\lambda}(x'))(x',x'') < \sigma(\underline{\lambda}(x))(x,x')]$.

Dans la définition (f) de $J(\gamma',x,x')$, le cas $m=1$ se réduit à $[(x=x' \vee x'\leq 1) \wedge \gamma'=\sigma(\omega)(x,x')]$. Autrement $m>1$ et pour $\forall i\in(m\vee 0)$ nous avons $\Sigma_{i-1}=0$ parceque $LI(\lambda_{i-1})(s_{i-1},s_i,\lambda_i,s'')$ implique $s''<1$ donc $\sigma(\lambda_{i-1})(s_{i-1},s'')=0$. De plus, λ est une fonction de s car $\lambda_0=\omega$ et $LI(\lambda_{i-1})(s_{i-1},s_i,\lambda_i,s''_i)$ impliquent $\lambda_i=\underline{\lambda}(s_i)$ pour tout $i\in(m\vee 0)$. Quand $i=1$, ceci implique aussi $s_0=s_1=x>1$. Quand $i\in(2,\dots,m-1)$, les termes $\exists s''_i\in S.\ LI(\lambda_{i-1})(s_{i-1},s_i,\lambda_i,s''_i)$ sont de la forme :

$$\underline{\lambda}(s_{i-1})>\underline{\lambda}(s_i) \wedge s_i>1 \wedge [(s_{i-1}=s_i) \vee t_{\underset{\sim}{a}}(s_{i-1},s_i) \vee (odd(s_{i-1}) \wedge t^2_{\underset{\sim}{a}}(s_{i-1},s_i))]$$
$$= (s_i>1) \wedge [(even(s_{i-1}) \wedge t_{\underset{\sim}{a}}(s_{i-1},s_i)) \vee (odd(s_{i-1}) \wedge t^2_{\underset{\sim}{a}}(s_{i-1},s_i))]$$

parceque $s_{i-1}=s_i$ ou $s_i=(s_{i-1}+1)$ (quand $odd(s_{i-1}) \wedge t_{\underset{\sim}{a}}(s_{i-1},s_i)$ ne sont pas compatibles avec $\underline{\lambda}(s_{i-1})>\underline{\lambda}(s_i)>s_i>1$. Le terme $\neg\theta_{\lambda_i}(s_i,s_{i+1})$ revient à $s_{i+1}>1$ pour $i\in(m\vee 0)$. Il s'ensuit que :

$$J(\gamma', x, x') = \Big([(x = x' \vee x' \leq 1) \wedge \gamma' = \sigma(\omega)(x, x')]$$
$$\vee [\exists m > 1,\ \Delta \in (m+1 \to S).\ ((x = \Delta_0 = \Delta_1 > 1) \wedge (\Delta_m = x' > 1)$$
$$\wedge\ \forall i \in (2, \ldots, m-1).\ [(even(\Delta_{i-1}) \wedge t_{\underline{a}}(\Delta_{i-1}, \Delta_i)) \vee (odd(\Delta_{i-1}) \wedge t^2_{\underline{a}}(\Delta_{i-1}, \Delta_i))]$$
$$\wedge\ [(\Delta_{m-1} = \Delta_m) \vee t_{\underline{a}}(\Delta_{m-1}, \Delta_m) \vee (odd(\Delta_{m-1}) \wedge t^2_{\underline{a}}(\Delta_{m-1}, \Delta_m))]$$
$$\wedge\ \gamma' = \sigma(\underline{\lambda}(\Delta_{m-1}))(\Delta_{m-1}, \Delta_m))])$$

Si nous posons $t'(x, x')$ égal à $[(even(x) \wedge t_{\underline{a}}(x, x')) \vee (odd(x) \wedge t^2_{\underline{a}}(x, x'))]$, ceci peut s'écrire plus simplement comme suit :

$$J(\gamma', x, x') = [\exists x'' \in S.\ (t'^*(x, x'') \wedge ((x' > 1) \Rightarrow [(x'' = x') \vee t_{\underline{a}}(x'', x') \vee (odd(x'') \wedge t^2_{\underline{a}}(x'', x'))]$$
$$\wedge\ \gamma' = \sigma(\underline{\lambda}(x''))(x'', x'))))]$$

Noter que cette formule met en évidence l'essence de la preuve par la méthode de Burstall qui consiste à considérer un pas pour les états pairs et deux pas pour les états impairs. Il reste à montrer que $J(\gamma', x, x')$ satisfait $(\mathcal{F}_6.1)$ (ce qui est évident) et $(\mathcal{F}_6.2)$. De manière évidente, si $\neg\Psi(x, x')$ alors $x' > 1$ donc $\exists x''' \in S.\ t_{\underline{a}}(x', x''')$. Mais aussi, si $t_{\underline{a}}(x', x'')$ alors quatre cas doivent être considérés :

- Si $x' = x''$ alors $t'^*(x, x') \wedge t_{\underline{a}}(x', x''')$ implique $J(\sigma(\underline{\lambda}(x'))(x', x'''), x', x'')$ et $\sigma(\underline{\lambda}(x'))(x', x''') < \sigma(\underline{\lambda}(x'))(x'', x')$.

- Si $even(x'')$ et $t_{\underline{a}}(x'', x')$ alors $t'^*(x, x'') \wedge (even(x'') \wedge t_{\underline{a}}(x'', x')) \wedge t_{\underline{a}}(x', x''')$ implique $t'^*(x, x') \wedge t_{\underline{a}}(x', x''')$ donc $J(\sigma(\underline{\lambda}(x'))(x', x'''), x, x''')$ et $\sigma(\underline{\lambda}(x'))(x', x''') < \sigma(\underline{\lambda}(x''))(x'', x')$.

- Si $odd(x'')$ et $t_{\underline{a}}(x'', x')$ alors $t'^*(x, x'') \wedge (odd(x'') \wedge t_{\underline{a}}(x'', x') \wedge t_{\underline{a}}(x', x'''))$ implique $t'^*(x, x''') \wedge (x''' = x'')$ donc $J(\sigma(\underline{\lambda}(x'''))(x''', x'''), x, x''')$ et $\sigma(\underline{\lambda}(x'''))(x''', x''') < \sigma(\underline{\lambda}(x''))(x'', x')$.

- Si $odd(x'')$ et $t^2_{\underline{a}}(x'', x')$ alors $t'^*(x, x'') \wedge (odd(x'') \wedge t^2_{\underline{a}}(x'', x')) \wedge t_{\underline{a}}(x', x''')$ implique $t'^*(x, x') \wedge t(x', x''')$ donc $J(\sigma(\underline{\lambda}(x'))(x', x'''), x, x''')$ et $\sigma(\underline{\lambda}(x'))(x', x''') < \sigma(\underline{\lambda}(x''))(x'', x')$.

□

5.5 CHARTES DE PREUVE

Ayant montré que la méthode de Floyd est un cas particulier de la méthode de Burstall (après des généralisations adéquates), il nous reste à étudier une présentation uniforme des preuves par l'une ou l'autre des méthodes. A cet effet nous utiliserons une présentation graphique des preuves.

L'idée de présenter les preuves de programmes par des diagrammes acycliques fut introduite par Lamport [77] et développée ultérieurement par Owicki-Lamport [82] et Manna-Pnueli [82]. Cependant ces méthodes n'étaient pas sémantiquement complètes à cause d'un certain nombre de restrictions (comme l'impossibilité de faire des inductions infinies ou la restriction à des programmes dont le nombre d'état est fini (et petit), etc.). Notre formalisation est plus générale du fait qu'elle consiste à introduire des chartes de preuve bien-structurées présentant eventuellement des cycles et dont nous démontrons la correction et la complétude sémantique.

Finalement, nous montrons que les chartes de preuve sont adéquats pour démontrer les propriétés de fatalité des programmes parallèles.

5.5.1 DEFINITION D'UNE CHARTE DE PREUVE D'UN PROGRAMME

Une charte de preuve pour un système de transition $\langle S, A, E, \Phi \rangle$ sera formalisée au moyen d'un ensemble fini de graphes finis étiquetés bien-structurés et ayant une seule entrée et une seule sortie. Nous écrivons $I^E \rightsquigarrow I^S$ pour dénoter un tel graphe avec un sommet d'entrée unique noté I^E et un sommet de sortie unique noté I^S.

Nous définissons l'ensemble [illegible] ecrites par une grammaire. Les graphes élémentaires sont de la [illegible] J où I est le sommet d'entrée, J le sommet de sortie et il n'existe qu'un seul arc entre I et J. Il y a différents types d'arcs (que nous représentons différemment), certains pouvant être étiquetés (l'étiquette est alors écrite sur l'arc correspondant). Pour composer ces graphes, nous utiliserons les opérations de composition suivantes :

- Si I ⇝ J et K ⇝ L sont deux graphes tels que J=K alors I ⇝ J ⇝ L dénote le graphe composé obtenu en confondant le sommet de sortie J avec le sommet d'entre K. Il n'y a pas d'autres fusions possibles des sommets des graphes originaux et le sommet d'entrée (respectivement de sortie) graphe composé est le sommet étiqueté I (respectivement L).

- Si I ⇝ J et K ⇝ L sont deux graphes tels que I=K et J=L alors I ⇉ J dénote le graphe composé où les sommets d'entrée (respectivement de sortie) sont confondus, les autres sommets restant deux à deux distincts.

- Si I ⇝ J et K ⇝ L sont deux graphes tels que I=K alors la boucle  est le graphe composé avec le sommet d'entrée I confondu avec K, un sommet de sortie J et un nouvel arc reliant le sommet L au sommet d'entrée I.

Nous écrirons $I(s_0, \underline{s}, \vec{s}, s)$ (respectivement $I(s_0, \underline{s}, \vec{s}.\tilde{s}, s)$ et $I(s_0, \underline{s}, \underline{s})$) pour signifier que l'étiquette I associée à un sommet d'un graphe appartient à $(S \times S \times S^m \times S \to \{tt, ff\})$ où m=m (respectivement m=n+1, m=0), n étant le nombre de boucles imbriquées du graphe contenant ce sommet. D'une manière informelle, s_0 est la valeur de l'état d'entrée du programme, $\underline{s}$ (respectivement $\vec{s}_i$) est la valeur de l'état correspondant à l'entrée du graphe (respectivement à l'entrée de la i ème boucle imbriquée du graphe) et s est la valeur de l'état courant.

Définition 5.5.1 : 1 (Chartes de preuve)

Une charte de preuve pour $\langle S,A,t,\# \rangle$ est une paire $\langle (\Lambda,\vdash), \{(G_\ell,(f_\ell,W_\ell,\prec_\ell)) : \ell \in \Lambda\}\rangle$ où $\langle \Lambda,\vdash\rangle$ est un ensemble fini bien-fondé (de noms de graphes) et où $\forall \ell \in \Lambda$, $f_\ell \in (S^2 \to W_\ell)$, $wf(W_\ell,\prec_\ell)$ et G_ℓ est une charte bien-formée $I_\ell^{\varepsilon_\ell}(s_0,\underline{s},s) \rightsquigarrow\rightsquigarrow I_\ell^{\sigma_\ell}(s_0,\underline{s},s)$ générée par la grammaire de graphes suivante :

$J(s_0,\underline{s},\vec{s},s) \rightsquigarrow\rightsquigarrow K(s_0,\underline{s},\vec{s},s)$ ::=

$J(s_0,\underline{s},\vec{s},s) \longrightarrow K(s_0,\underline{s},\vec{s},s)$

si $\forall s_0,\underline{s},s \in S, \vec{s} \in S^n . [J(s_0,\underline{s},\vec{s},s) \Rightarrow (\exists s' \in S, a \in A . t_a(s,s') \wedge \forall s' \in S, a \in A . (t_a(s,s') \Rightarrow K(s_0,\underline{s},\vec{s},s')))]$

| $J(s_0,\underline{s},\vec{s},s) \overset{\ell'}{\longrightarrow\!\!\!\!\!\!\rightarrow} K(s_0,\underline{s},\vec{s},s)$

si $\ell' \vdash \ell \wedge \forall s_0,\underline{s},s \in S, \vec{s} \in S^n . [J(s_0,\underline{s},\vec{s},s) \Rightarrow (I_{\ell'}^{\varepsilon\ell'}(s_0,s,s) \wedge \forall s' \in S . (I_{\ell'}^{\sigma\ell'}(s_0,s,s') \Rightarrow K(s_0,\underline{s},\vec{s},s')))]$

| $J(s_0,\underline{s},\vec{s},s) \overset{\ell,(f_\ell,W_\ell,\prec_\ell)}{\longrightarrow\!\!\!\!\!\!\rightarrow} K(s_0,\underline{s},\vec{s},s)$

si $\forall s_0,\underline{s},s \in S, \vec{s} \in S^n . [J(s_0,\underline{s},\vec{s},s) \Rightarrow (f_\ell(s_0,s) \prec_\ell f_\ell(s_0,\underline{s}) \wedge I_\ell^{\varepsilon\ell}(s_0,s,s) \wedge \forall s' \in S . (I_\ell^{\sigma\ell}(s_0,s,s') \Rightarrow K(s_0,\underline{s},\vec{s},s')))]$

| $J(s_0,\underline{s},\vec{s},s) \Longrightarrow K(s_0,\underline{s},\vec{s},s)$

si $\forall s_0,\underline{s},s \in S, \vec{s} \in S^n . [J(s_0,\underline{s},\vec{s},s) \Rightarrow K(s_0,\underline{s},\vec{s},s)]$

| $J(s_0,\underline{s},\vec{s},s)$ → $L^1(s_0,\underline{s},\vec{s},s)$, $L^2(s_0,\underline{s},\vec{s},s)$, ... , $L^p(s_0,\underline{s},\vec{s},s)$ → $K(s_0,\underline{s},\vec{s},s)$

si $\forall s_0,\underline{s},s \in S, \vec{s} \in S^n . [J(s_0,\underline{s},\vec{s},s) \Rightarrow \overset{p}{\underset{i=1}{\vee}} L^i(s_0,\underline{s},\vec{s},s)]$

| $J(s_0,\underline{s},\vec{s},s) \rightsquigarrow\rightsquigarrow L(s_0,\underline{s},\vec{s},s) \rightsquigarrow\rightsquigarrow K(s_0,\underline{s},\vec{s},s)$

| $J(s_0,\underline{s},\vec{s},s)$ → $K(s_0,\underline{s},\vec{s},s)$, $L(s_0,\underline{s},\vec{s}.\tilde{s},s) \rightsquigarrow\rightsquigarrow M(s_0,\underline{s},\vec{s}.\tilde{s},s)$, $(f,W,\prec)$

si $f \in (S^3 \to W) \wedge Wf(W,\prec) \wedge \forall s_0,\underline{s},s,\tilde{s} \in S, \vec{s} \in S^n . ($
$[J(s_0,\underline{s},\vec{s},s) \Rightarrow (K(s_0,\underline{s},\vec{s},s) \vee L(s_0,\underline{s},\vec{s}.s,s))]$
$\wedge$
$[M(s_0,\underline{s},\vec{s}.\tilde{s},s) \Rightarrow ([f(s_0,\underline{s},s) \prec f(s_0,\underline{s},\tilde{s}) \wedge L(s_0,\underline{s},\vec{s}.s,s)] \vee K(s_0,\underline{s},\vec{s},s))])$

(Observons que les chartes de preuve sont des graphes reductibles, donc que nous aurions pu aussi formaliser à l'aide d'un langage bien structuré).

Nous pouvons démontrer que Ψ est fatale pour $\langle S, A, \Sigma\langle S, A, t, \#\rangle\rangle$ en démontrant que :

Condition 5.5.1 : 2 (Preuves par chartes)

Il existe une charte de preuve $\langle\langle \Lambda, \vdash\rangle, \{(I_\ell^{\varepsilon_\ell}(A_0, \underline{A}, A) \rightsquigarrow\rightsquigarrow\rightsquigarrow I_\ell^{\sigma_\ell}(A_0, \underline{A}, A), (f_\ell, W_\ell, \prec_\ell)) : \ell \in \Lambda\}\rangle$ et $\pi \in \ell$ tels que :

$\forall A_0, \underline{A}, A \in S. \left[I_\pi^{\varepsilon_\pi}(A_0, \underline{A}, \underline{A}) = [A_0 = \underline{A} \wedge \Phi(\underline{A})] \quad \wedge \quad I_\pi^{\sigma_\pi}(A_0, \underline{A}, A) = [A_0 = \underline{A} \wedge \Psi(\underline{A}, A)] \right]$

5.5.2 CORRECTION ET COMPLETUDE SEMANTIQUE DES PREUVES PAR CHARTES

Théorème 5.5.2 - 1 (Correction des preuves par chartes)

$(5.5.1 : 2) \Longrightarrow ((\beta_2)$ avec $n \in (\Lambda \to \underbrace{Ord}))$

Démonstration

Soit $\langle\langle \Lambda, \vdash\rangle, \{(G_\ell, (f_\ell, W_\ell, \prec_\ell)) : \ell \in \Lambda\}\rangle$ une charte de preuve. Puisque $\underbrace{wf}(W_\ell, \prec_\ell)$ nous pouvons supposer, sans perte de généralité que $W_\ell \in \underbrace{Ord}$ et $\prec_\ell = <$ (autrement nous pouvons utiliser des fonctions-rang). Nous pouvons également supposer que $\Lambda \in w$ et $\vdash = <$ puisque Λ est fini. Chaque graphe G_ℓ étant fini, nous pouvons supposer que ses sommets prennent leurs noms dans un ensemble fini N_ℓ, le sommet j étant étiqueté par $J_\ell^j \in (S \times S \times S^{e(j)} \times S \to \{\#, \#\#\})$ où $e(j)$ est le nombre de boucles renfermant j. Soient ε_ℓ et σ_ℓ les noms

respectifs du sommet d'entrée unique et du sommet de sortie unique de G_ℓ.

Pour tout $\ell \in \Lambda$, nous considérons l'ensemble T_ℓ de tuples $\langle j, \Delta_0, \underline{\Delta}, \vec{\Delta}, \Delta \rangle$ tels que $j \in N_\ell$, $\Delta_0, \underline{\Delta}, \Delta \in S$, $\vec{\Delta} \in S^{e(j)}$ et $J_\ell^j(\Delta_0, \underline{\Delta}, \vec{\Delta}, \Delta)$ soit vrai. Définissons la relation binaire $\ll_\ell$ sur T_ℓ comme suit : $\langle j', \Delta'_0, \underline{\Delta}', \vec{\Delta}', \Delta' \rangle \ll_\ell \langle j, \Delta_0, \underline{\Delta}, \vec{\Delta}, \Delta \rangle$ si et seulement si :

soit $\quad J_\ell^j \longrightarrow J_\ell^{j'} \wedge s'_0 = s_0 \wedge \underline{s}' = \underline{s} \wedge \vec{s}' = \vec{s} \wedge \exists a \in A.\, t_a(\Delta, \Delta')$

ou $\quad J_\ell^j \overset{\ell'}{\rightsquigarrow} J_\ell^{j'} \wedge s'_0 = s_0 \wedge \underline{s}' = \underline{s} \wedge \vec{s}' = \vec{s} \wedge J_{\ell'}^{\sigma_{\ell'}}(s_0, s, s')$

ou $\quad J_\ell^j \overset{\ell, (f_\ell, W_\ell, \prec_\ell)}{\rightsquigarrow} J_\ell^{j'} \wedge s'_0 = s_0 \wedge \underline{s}' = \underline{s} \wedge \vec{s}' = \vec{s} \wedge J_\ell^{\sigma_\ell}(s_0, s, s')$

ou $\quad [(J_\ell^j \Longrightarrow J_\ell^{j'}) \vee (J_\ell^j \circ\!\!\Longrightarrow J_\ell^{j'}) \vee (J_\ell^j \rightarrowtail\!\!\circ\!\!\Longrightarrow J_\ell^{j'})] \wedge s'_0 = s_0 \wedge \underline{s}' = \underline{s} \wedge \vec{s}' = \vec{s} \wedge s' = s$

ou $\quad J_\ell^j \circ\!\!\Longrightarrow J_\ell^{j'} \wedge s'_0 = s_0 \wedge \underline{s}' = \underline{s} \wedge \vec{s}' = \vec{s}.s \wedge s' = s$

sinon $\quad J_\ell^j \rightarrowtail\!\!\circ\!\!\Longrightarrow J_\ell^{j'} \wedge s'_0 = s_0 \wedge \underline{s}' = \underline{s} \wedge \vec{s} = \vec{s}'.s \wedge s' = s$

Supposons que $\langle \langle j_k, \Delta_{0k}, \underline{\Delta}_k, \vec{\Delta}_k, \Delta_k \rangle : k \geq 0 \rangle$ est une séquence infinie strictement décroissante pour $\ll_\ell$. Il s'ensuit que $\langle j_k, k \geq 0 \rangle$ est un chemin infini dans le graphe fini G_ℓ, c'est donc un cycle. Alors il existe un sommet j de G_ℓ (de type $J_\ell^j \rightarrowtail\!\!\circ$) tel que la séquence $\langle \langle j, \Delta_{0i_k}, \underline{\Delta}_{i_k}, \vec{\Delta}_{i_k}, \vec{\Delta}'_{i_k} . \tilde{\Delta}_{i_k}, \Delta_{i_k} \rangle : k \geq 0 \rangle$ d'éléments de $\langle \langle j_k, \Delta_{0k}, \underline{\Delta}_k, \vec{\Delta}_k, \Delta_k \rangle : k \geq 0 \rangle$ tels que $j_k = j$ est infinie. Ceci est en contradiction avec $\forall k \geq 0.\, (f(\Delta_{0i_k}, \underline{\Delta}_{i_k}, \Delta_{i_k}) \prec f(\Delta_{0i_k}, \underline{\Delta}_{i_k}, \tilde{\Delta}_{i_k}))$, $f \in (S^3 \to W)$ et $\underset{\sim}{\omega f}(W, \prec)$. Par l'absurde, nous avons $\underset{\sim}{\omega f}(T_\ell, \ll_\ell)$.

Nous choisissons $\Lambda_2 = \Lambda$, $\varepsilon_{2\ell}(\Delta_0, \Delta) = J_\ell^{\varepsilon_\ell}(\Delta_0, \Delta, \Delta)$, $\sigma_{2\ell}(\Delta_0, \underline{\Delta}, \Delta) = J_\ell^{\sigma_\ell}(\Delta_0, \underline{\Delta}, \Delta)$, $\Delta_2 = \underset{\sim}{\sup}^+ \{ \underset{\sim}{rk}(W_\ell, \prec_\ell) : \ell \in \Lambda \}$, $f_{2\ell}(\Delta_0, \Delta) = \underset{\sim}{rk}(W_\ell, \prec_\ell)(f_\ell(\Delta_0, \Delta))$, $n_\ell = (\underset{\sim}{rk}(T_\ell, \ll_\ell) + 1)$, $\pi_2 = \pi$, $I_{2\ell}^i(\Delta_0, \underline{\Delta}, \Delta) = [\exists j \in N_\ell, \vec{\Delta} \in S^{e(j)} . (J_\ell^j(\Delta_0, \underline{\Delta}, \vec{\Delta}, \Delta) \wedge i = \underset{\sim}{rk}(T_\ell, \ll_\ell)(\langle j, \Delta_0, \underline{\Delta}, \vec{\Delta}, \Delta \rangle))]$ quand $i < m_\ell$ et $I_{2\ell}^{m_\ell}(\Delta_0, \underline{\Delta}, \Delta) = J_\ell^{\varepsilon_\ell}(\Delta_0, \underline{\Delta}, \Delta)$.

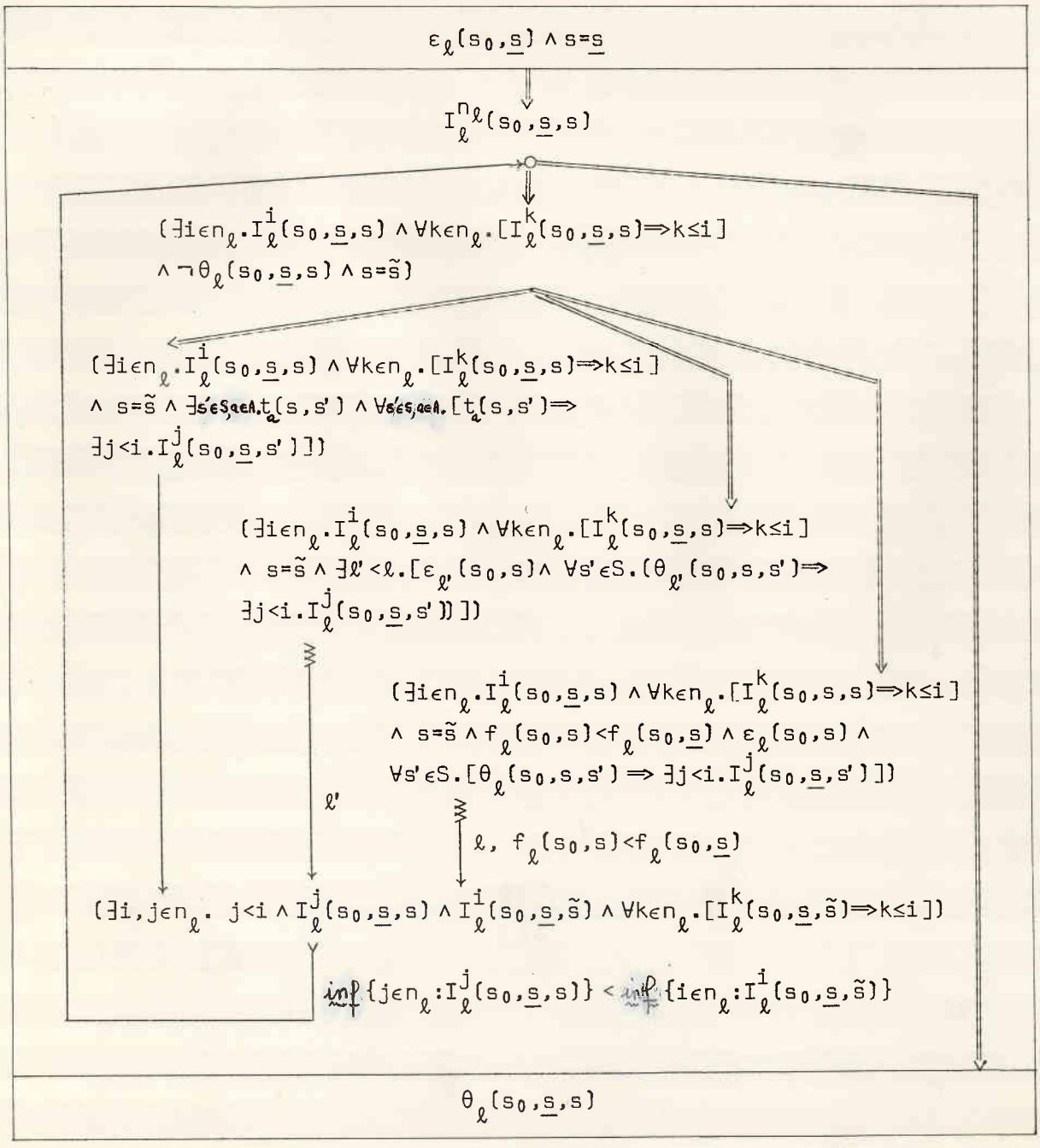
□

Théorème 5.5.2.N.2 (Complétude sémantique des preuves par chartes)

$(\mathcal{B}_2) \Longrightarrow (5.5.1:2)$

Démonstration

Nous pouvons représenter une preuve par $(\mathcal{B}_2)$ par la charte de preuve $\langle\langle\Lambda,\vdash\rangle,(G_\ell,(f_\ell,0_{\mathrm{rd}},<)):\ell\in\Lambda\rangle$ où chaque graphe G_ℓ, $\ell\in\Lambda$ est la charte suivante :



□

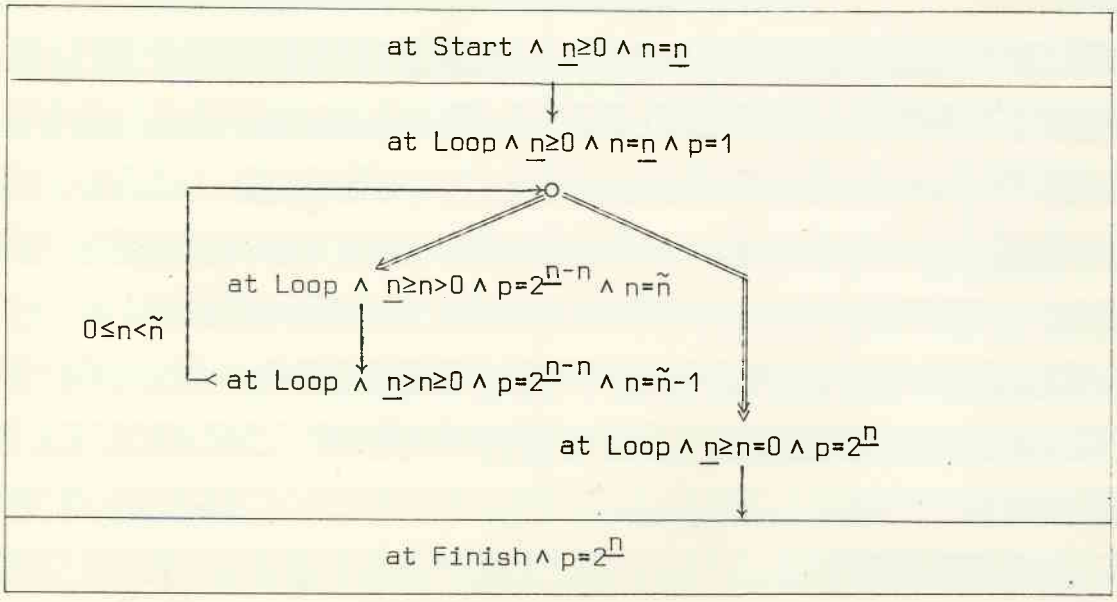
5.5.3 EXEMPLES DE PRESENTATION DE PREUVES PAR CHARTES

Les chartes permettent aussi bien de présenter des preuves par la méthode de Burstall que par la méthode de Floyd, généralisant les deux méthodes en les unifiant. Elles permettent également de traiter le cas des programmes parallèles.

5.5.3.1 Présentation de preuves "à la Floyd" par des chartes

Nous dirons qu'une preuve de fatalité par une charte est une preuve "à la Floyd" quand la charte de preuve a la forme de la charte (ou organigramme) du programme. Dans ce cas les assertions utilisées dans la charte de preuve sont des invariants au sens de Floyd.

Par exemple, une preuve "à la Floyd" de correction totale du programme 5.3.1-1 peut également être présentée comme suit :



5.5.3.2 Preuve de propriétés de fatalité de programmes parallèles asynchrones

Puisque nous pouvons représenter un programme parallèle par un système de transition non-déterministe, les chartes de preuve peuvent également s'appliquer aux preuves de fatalité de programmes parallèls.

Exemple 5.5.3.2-1 (Correction totale d'un programme parallèle)

Considérons une version parallèle asynchrone du programme 5.3.1-1 qui calcule 2^m quand $m \geq 0$:

```
1:  N1:=0; N2:=N;
2:  [[
      11:  P1:=1
      12:  if N1+1 < N2 then
      13:       T1:=N1+1; P1:=2xP1;
      14:       N1:=T1;
      15:  fi;  goto 12;
      16:
    ||
      21:  P2:=1;
      22:  if N1+1 < N2 then
      23:       T2:=N2-1; P2:=2xP2;
      24:       N2:=T2;
      25:  fi;  goto 22;
      26:
    ]];
3:  P :=  if N1+1=N2 then 2xP1xP2 else P1xP2 fi;
4:
```

Nous écrivons at_j (respectivement at_{ij}) à la place de $c=i$ (respectivement $c_i=j$) où c (respectivement c_i) est le point de contrôle du programme (respectivement du processus i lorsque le contrôle est dans la commande parallèle). Nous écrivons $in\,E$ pour $\vee\{at\,\ell : \ell \in E\}$, et

Dans la charte de preuve de correction totale

P: (at1 ∧ n_0>0 ⇝ at4 ∧ p=2^{n_0}), nous distinguons deux cas :

- le cas $n_0 \leq 1$ se traite par le lemme

Z0: (at1 ∧ $0 \leq n_0 \leq 1$ ⇝ at4 ∧ p=2^{n_0}). Ce lemme peut être démontré par évaluation symbolique et nous laissons le lecteur faire la charte de preuve correspondante.

- le cas principal $n_0 > 1$ se traite par le lemme

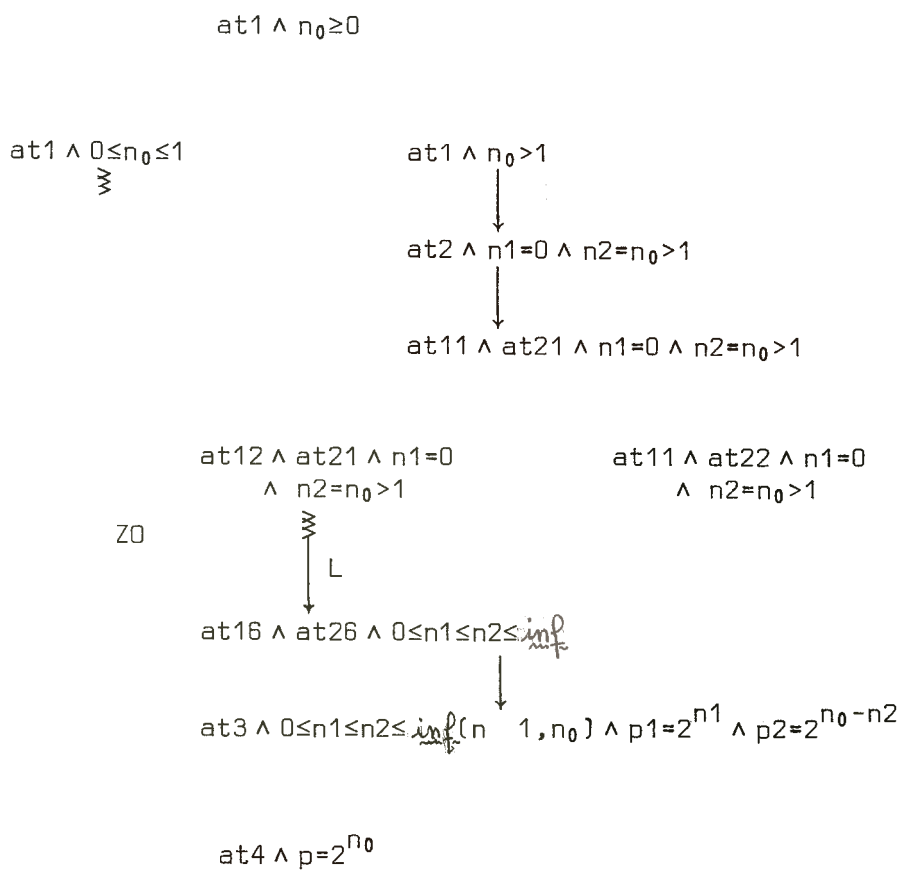
L: (Atloop ∧ n1=$\underline{n}1$ ∧ n2=$\underline{n}2$ ∧ ($\underline{n}1$+1)<$\underline{n}2$ ∧ Inv ⇝ at16 ∧ at26 ∧ $\underline{n}1 \leq n1 \leq n2 \leq \underline{inf}(n1+1, \underline{n}2)$ ∧ p1=2^{n1} ∧ p2=2^{n_0-n2}) où Atloop remplace ([at12 ∧ in{21,...,25}] ∨ [in{11,...,15} ∧ at22])

et Inv est l'invariant suivant :

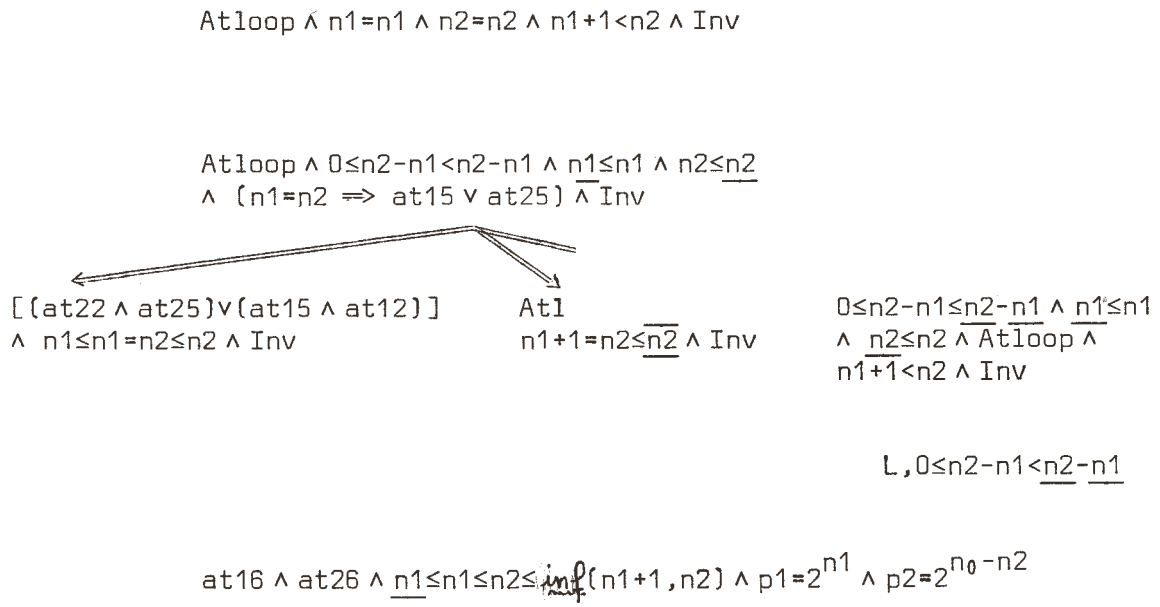
Inv = [(at11 ⇒ n1=0 | p1=2^{n1} × (at14 → 2|1)) ∧ (at14 ⇒ t1= n1+1) ∧

(at21 ⇒ n2=n_0 | p2=2^{n_0-n2} × (at24 → 2|1)) ∧ (at24 ⇒ t2= n2-1)]

P est la charte de preuve suivante :

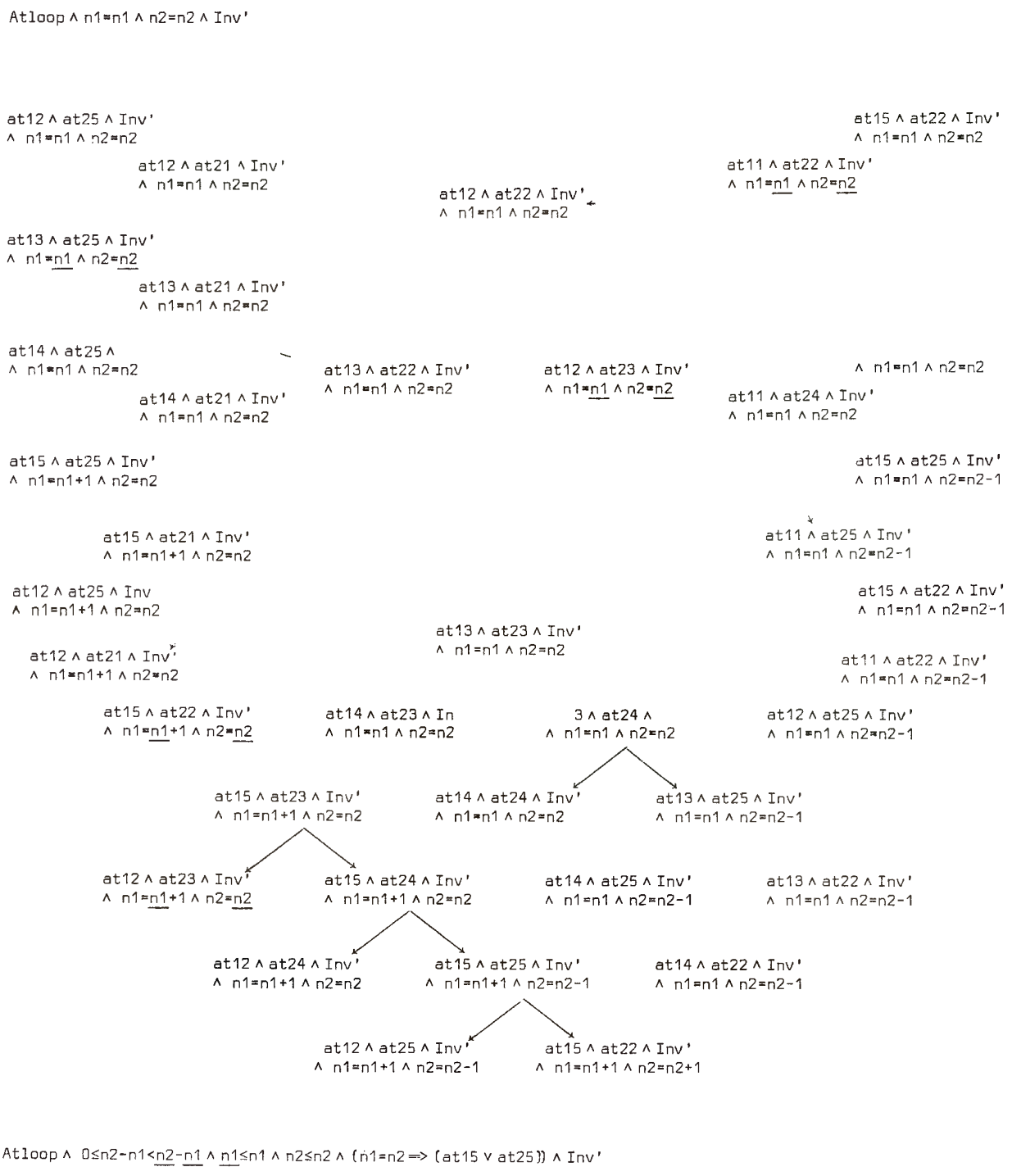


La preuve du lemme L se fait par induction sur $(m2-m1)$ qui décroit strictement à chaque itération de boucle dans l'un des deux processus. Cette itération est décrite par le lemme I: $(Atloop \wedge m1=\underline{m}1 \wedge m2=\underline{m}2 \wedge (\underline{m}1+1<m2) \wedge Inv \rightsquigarrow Atloop \wedge 0\leq(m2-m1)<(\underline{m}2-m1) \wedge \underline{m}1\leq m1 \wedge m2\leq\underline{m}2 \wedge (\underline{m}1+1)<\underline{m}2 \wedge Inv \wedge (m1=m2 \Rightarrow (at15 \vee at25)))$. Nous avons $m1\leq m2\leq(m1+1)$. Le cas $m1=m2$ se traite par le lemme E: $(in\{12,15,16\} \wedge in\{22,25,26\} \wedge m1=\underline{m}1 \wedge p1=\underline{p}1 \wedge m2=\underline{m}2 \wedge p2=\underline{p}2 \wedge (m1+1)\geq m2 \rightsquigarrow at16 \wedge at26 \wedge m1=\underline{m}1 \wedge p1=\underline{p}1 \wedge m2=\underline{m}2 \wedge p2=\underline{p}2)$. Sa preuve est triviale par évaluation symbolique et nous laissons sa charte au lecteur. Le cas $m2=(m1+1)$ se traite par le lemme B: $(Atloop \wedge m1=\underline{m}1 \wedge (m1+1)=m2=\underline{m}2 \wedge Inv \rightsquigarrow at16 \wedge at26 \wedge \underline{m}1\leq m1\leq m2\leq \inf(m1+1,\underline{m}2) \wedge p1=2^{m1} \wedge p2=2^{n_0-m2})$. La charte de preuve L est la suivante :



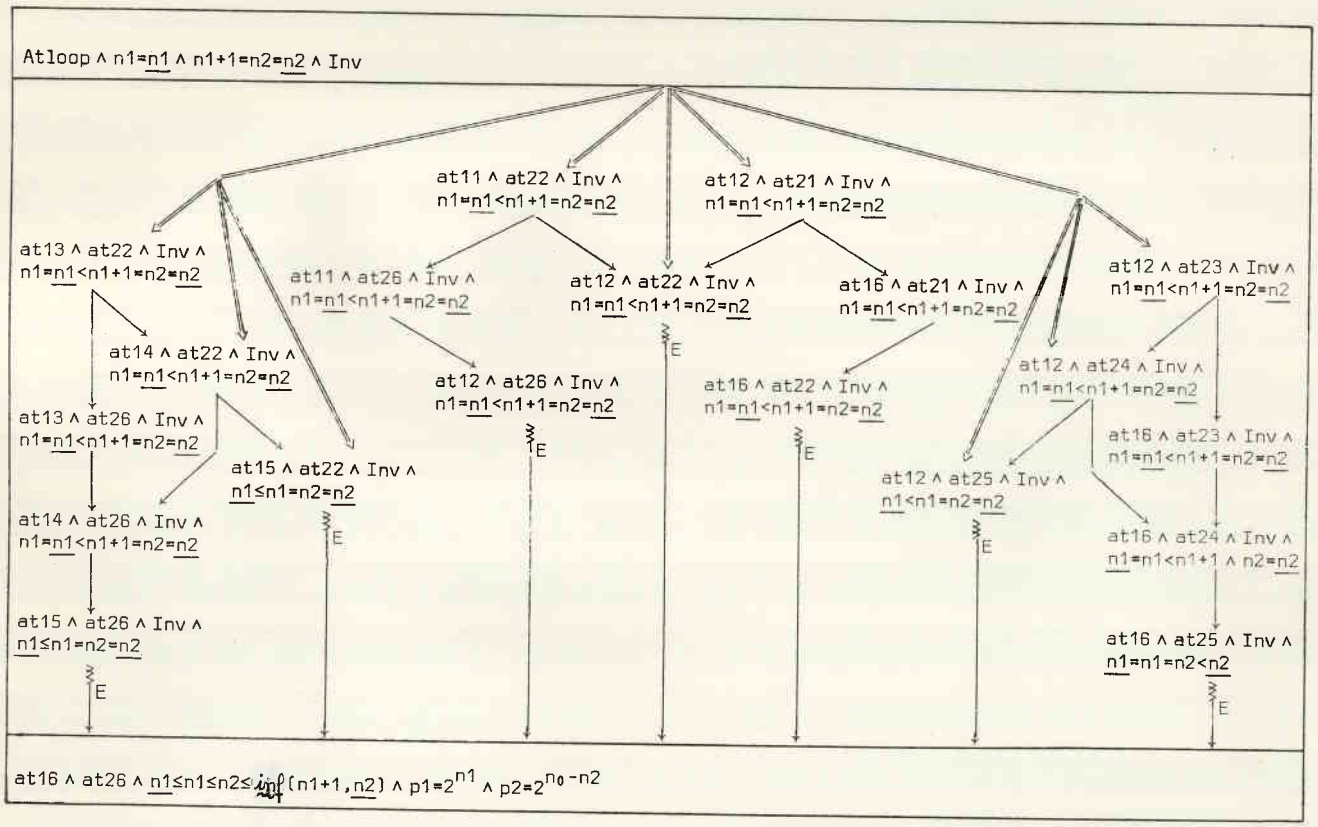
Les preuves des lemmes I et B ne présentent aucune difficulté et nous pouvons les faire entièrement par évaluation symbolique.

Nous posons $Inv' = (Inv \wedge (\underline{m}1+1)<\underline{m}2)$ dans la charte de preuve I :



Atloop ∧ $0 \le n2-n1 < n2-n1 \wedge n1 \le n1 \wedge n2 \le n2 \wedge (n1=n2 \Rightarrow (at15 \vee at25)) \wedge Inv'$

B


Atloop ∧ n1=n1 ∧ n1+1=n2=n2 ∧ Inv
at11 ∧ at22 ∧ Inv ∧ n1=n1<n1+1=n2=n2
at12 ∧ at21 ∧ Inv ∧ n1=n1<n1+1=n2=n2
at13 ∧ at22 ∧ Inv ∧ n1=n1<n1+1=n2=n2
at11 ∧ at26 ∧ Inv ∧ n1=n1<n1+1=n2=n2
at12 ∧ at22 ∧ Inv ∧ n1=n1<n1+1=n2=n2
at16 ∧ at21 ∧ Inv ∧ n1=n1<n1+1=n2=n2
at12 ∧ at23 ∧ Inv ∧ n1=n1<n1+1=n2=n2
at14 ∧ at22 ∧ Inv ∧ n1=n1<n1+1=n2=n2
at12 ∧ at26 ∧ Inv ∧ n1=n1<n1+1=n2=n2
at16 ∧ at22 ∧ Inv ∧ n1=n1<n1+1=n2=n2
at12 ∧ at24 ∧ Inv ∧ n1=n1<n1+1=n2=n2
at13 ∧ at26 ∧ Inv ∧ n1=n1<n1+1=n2=n2
at15 ∧ at22 ∧ Inv ∧ n1≤n1=n2=n2
at16 ∧ at23 ∧ Inv ∧ n1=n1<n1+1=n2=n2
at12 ∧ at25 ∧ Inv ∧ n1<n1=n2=n2
at14 ∧ at26 ∧ Inv ∧ n1=n1<n1+1=n2=n2
at16 ∧ at24 ∧ Inv ∧ n1=n1<n1+1 ∧ n2=n2
at15 ∧ at26 ∧ Inv ∧ n1≤n1=n2=n2
at16 ∧ at25 ∧ Inv ∧ n1=n1=n2<n2
E
at16 ∧ at26 ∧ n1≤n1≤n2≤inf{n1+1, n2} ∧ p1=2^n1 ∧ p2=2^n0-n2

□

5.5.3.3 Preuve de propriétés de fatalité de programmes parallèles faiblement équitables

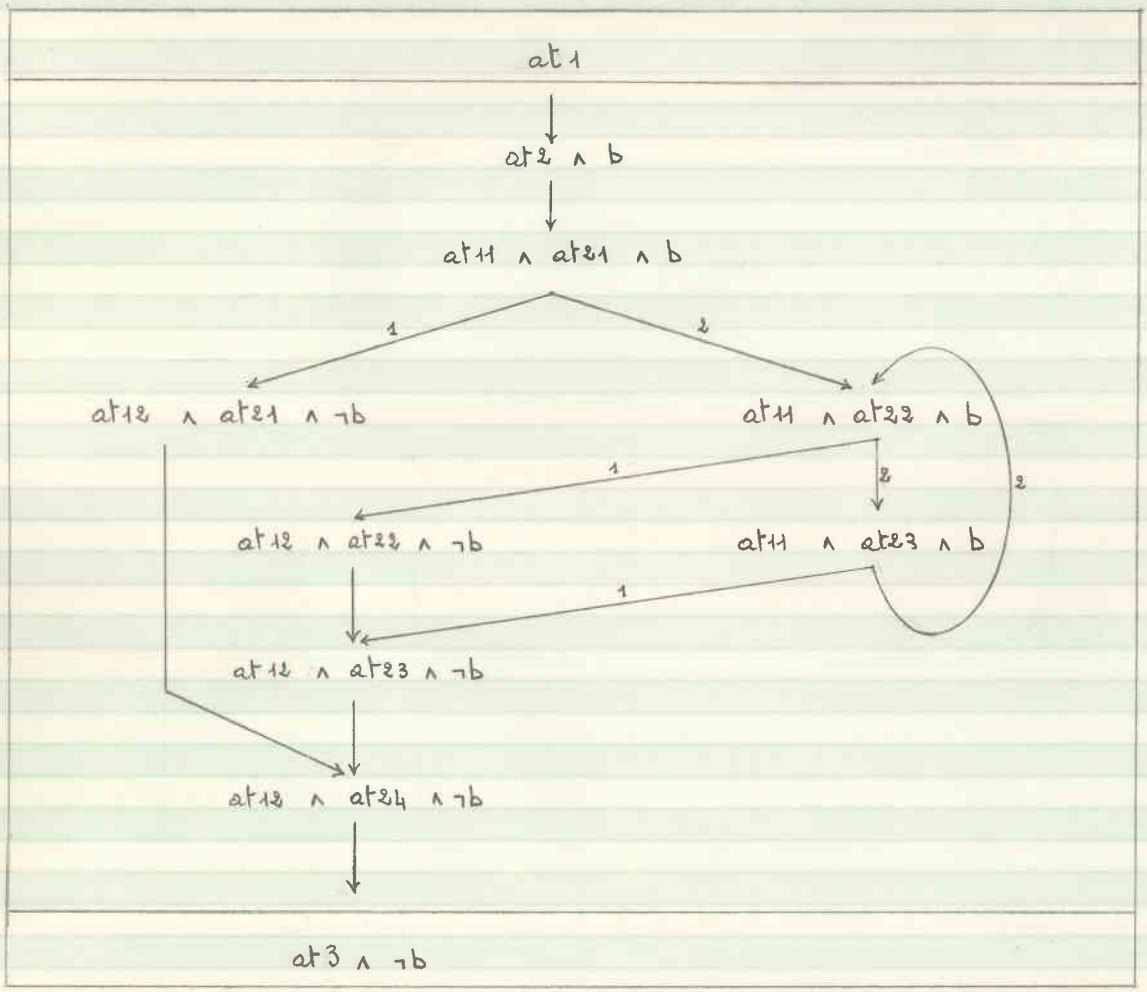
Les chartes de preuve peuvent également s'appliquer aux preuves de programmes parallèles faiblement équitables. Pour ce faire, nous notons i sur chaque arc de la charte correspondant à l'évaluation symbolique d'un pas du processus Prw_i d'un programme parallèle faiblement équitable $[\![Prw_1 \| \dots \| Prw_{m-1}]\!]$. Pourque la méthode soit complète, nous autorisons la présence de cycles dans la définition 5.5.1:1 pour lesquels il n'y a pas de preuve de terminaison à la Floyd par un triplet $\langle f, v, \prec \rangle$. Dans ce cas, il faudra simplement démontrer qu'il y a un processus toujours actif et jamais activé le long de ce cycle (si le cycle fait intervenir un lemme, ceci devra être démontré par une preuve d'invariance séparée). La terminaison découle alors de manière évidente de l'hypothèse d'équité faible. D'après le principe d'induction $(\mathcal{F}'_{15})$, cette approche est complète.

Exemple 5.5.3.3-1

Pour l'exemple classique :

```
1:
    B: true ;
2:
    [[ 11:
           B:= false ;
       12:
    ||  21:
           while B do
       22:
               skip;
       23:
           od;
       24:
    ]];
3:
```

nous avons la charte de preuve suivante :



Le long du cycle, le processus 1 est toujours activable et jamais activé.

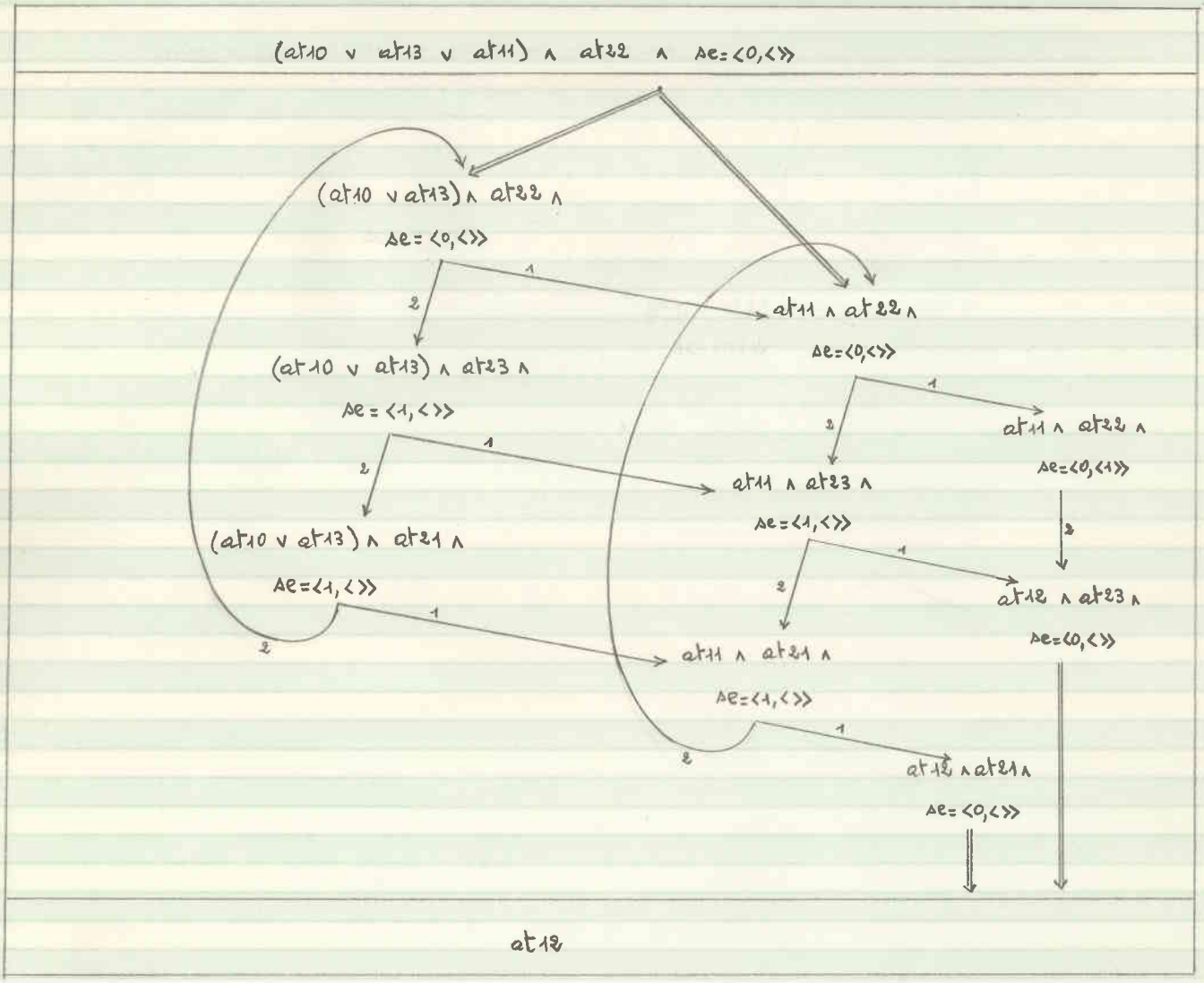
□

5.5.3.4 Preuve de propriétés de fatalité de programmes parallèles synchrones

En ce qui concerne les programmes parallèles synchrones, l'essentiel du travail a été fait au paragraphe 2.8.5. D'après 2.8.5.2.5 nous n'avons à considérer que les traces faiblement équitables engendrées par un système de transition. Par conséquent, la méthode des preuves par chartes du paragraphe précédent est directement applicable !

Exemple 5.5.3.4 n°1

Considérons le programme synchrone 2.8.5.4 qui réalise une section critique. Supposons que le deuxième processus soit en section critique. Nous démontrons que le premier processus entrera fatalement en section critique au moyen de la charte de preuve suivante :



Le long des deux cycles correspondant à une boucle du processus 2, le processeus 1 est toujours actif et jamais activé, elle est donc finie. □

L'utilisation explicite de la file d'attente des sémaphores peut être critiquée. En plus des arguments developpés au paragraphe 2.8.5, 2.6 qui expliquent pourquoi la définition des sémaphores de Dijkstra doit être prise à la lettre, on pourra comparer la simplicité de la preuve comparée à celle de Manna-Pnueli [83].

5.6 REFERENCES

APT K.R., DELPORTE C. [83], "An axiomatization of the intermittent assertion method," Rapport de Recherche 82-70, LITP, (Paris), (Jan. 1983), 21p.

APT K.R., OLDEROG E.R. [82], "Proof rules dealing with fairness", (extended Abstract), Proc. Logics of Programs, lect. notes in Comp. Sci. 131, Springer-Verlag, (1982), 1-8.

APT K.R., PLOTKIN G.D. [82], "Countable nondeterminisme and random assignment" Research report 82-7, Dept. Comp. Sci., Edimburg U., (Feb. 1982), 40p.

BERG H.K., BOEBERT W.E., FRANTA W.R., MOHER T.G. [82], "Formal methods of program verification and specification", Prentice-Hall, (1982).

BURSTALL R.M. [74], "Program proving as hand simulation with a little induction", IFIP 74, North-Holland Pub. Co., (1974), 308-312.

COUSOT P. [81], "Semantic foundations of program analysis", in Program Flow Analysis, Theory and Applications, S.S. Muchnick - N.D. Jones (Eds), Prentice-Hall, (1981), 303-342.

COUSOT P., COUSOT R. [80], "Reasoning about program invariance proof methods", Rapport de Recherche CRIN-80-P050, (1980).

COUSOT P., COUSOT R. [82], " "A la Floyd" induction principles for proving inevitability properties of programs", Rapport de Recherche LRIN-82-04 à paraître dans "Algebraic methods in Programming", (M. Nivat & J. Reynolds, eds), Cambridge University Press.

COUSOT P., COUSOT R. [83a], " "A la Burstall" induction principles for proving inevitability properties of programs", Rapport de Recherche LRIM-83-08, (Nov. 1983).

COUSOT P., COUSOT R. [83b], " SOMETIME = ALWAYS+RECURSION = ALWAYS, on the equivalence of the intermittent and invariant assertions methods for proving inevitability properties of programs", Rapport de Recherche LRIM-83-03, (Juillet 1983).

Dijkstra E.W. [76], " A discipline of programming", Prentice-Hall, (1976).

Dijkstra E.W. [77], " A sequel to EWD 592", EDW 600, (Jan. 1977).

Dijkstra E.W. [82], " Selected writings on computing : a personal perspective", Springer-Verlag, (1982).

Floyd R.W. [67], " Assigning meaning to programs", Proc. Symp. Applied Math., Vol. 19, AMS, Providence, R.I., (1967), 19-32.

Gries D. [79], " Is SOMETIME ever better than ALWAYS ? ", TOPLAS 1, 2, (1979).

HAREL D. [79], " First order dynamic logic", Lect. Notes in Comp. Sci. 68, Springer-Verlag, (1979).

HOARE C.A.R. [69], "An axiomatic basis of computer programming", CACM 12, 10 (1969), 576-580, 583.

KNUTH D.E. [68], " The art of computer programming", vol. 1, Addison-Wesley, New-York, (1968).

LAMPORT L. [77], "Proving the correctness of multiprocess programs", IEEE Trans. on Soft. Eng., SE3, 2 (1977), 125-143.

LAMPORT L. [80], " The Hoare logic of concurrent programs", Acta Informatica 14 (1980), 21-37.

LEHMANN D., PNUELI A., STAVI J. [81], "Impartiality, justice and fairness : the ethics of concurrent termination", Proc. 8th Colloq. on Automata, Languages and Programming, Lect. Notes in Comp. Sci. 115, Springer-Verlag. (1981), 264-277.

LIVERCY C. [78], "Théorie des programmes", Dunod, (1978).

LUCKHAM D.C., SUZUKI N. [75], "Automatic program verification IV : proof of termination within a weak logic of programs", Comp. sci. Report 522, Standford U., (1975).

MANNA Z., PNUELI A. [74], "Axiomatic approach to total correctness", Acta Informatica 3, (1974), 243-263.

MANNA Z., PNUELI A. [82], "Verification of concurrent programs : proving eventualities by well-founded ranking", STAN-CS-82-915, Comp. sci. Dept., Standford U., (May 1982).

MANNA Z., PNUELI A. [83], "Verification of concurrent programs : a temporal proof system", stan-cs-83-967, Comp. sci. Dept., standford U., (June 1983).

MANNA Z., WALDINGER R.J. [78], "Is SOMETIME sometimes better than ALWAYS ?, intermittent assertions in proving program correctness", CACM 21, 8 (1978), 159-172.

NAUR P. [66], "Proof of algorithms by general snapshots", BIT 6, (1966), 310-316.

OWICKI S., GRIES D. [76], "An axiomatic proof technique for parallel programs I", Acta Informatica, 6 (1976), 319-340.

OWICKI S., LAMPORT L. [82], "Proving liveness properties of concurrent programs", TOPLAS 4, 3 (July 1982), 455-495.

PARK D. [81], "A predicate transformer for weak fair iteration", Proc. 6th IBM Symp. on Math. Foundations of Comp. sci., Logical aspects of programs, Hakone, Japon (1981), 259-275.

PNUELI A. [77], "The temporal logic of programs", Proc. 18th Symp. on Found. of Comp. Sci., Providence, R.I., (1977), 46-57.

SCHWARZ J. [76], "Event-based reasoning - a system for proving correct termination of programs", Proc. ICALP 3, Edinburgh, (1976) 131-146.

6. CONCLUSION

6. CONCLUSION

6.1 SEMANTIQUE OPERATIONNELLE

6.2 PROPRIETES D'INVARIANCE ET DE FATALITE DES PROGRAMMES

6.3 PREUVES D'INVARIANCE ET DE FATALITE

6.4 REFERENCES

6. CONCLUSION

Pour conclure, nous terminons par quelques commentaires sur chaque chapitre de la thèse, en particulier pour comparer notre approche avec celle des logiques formelles (de Hoare, modales, temporelles, dynamiques, algorithmiques, etc...) qui sont le formalisme le plus souvent utilisé pour les preuves de programmes.

6.1 SEMANTIQUE OPERATIONNELLE

Les logiques formelles ont été le plus souvent interpretées à l'aide de systèmes de transition (cf. 2.2) puis, à la suite de Lamport [80], à l'aide d'ensembles de traces. Cependant, le choix de $\langle S, A, \Sigma \rangle$ est différent de celui que nous avons fait (cf. 2.1) :

- Pratt [82], par exemple, ne considère que des traces finies en ajoutant que les calculs infinis s'obtiennent comme limites de traces finies. Ceci revient à choisir $\underline{\text{Flim}}\,(\langle S, A, \Sigma \cap \Sigma^{<\omega}\langle S, A\rangle\rangle)$ et ne permet notamment pas de décrire des programmes parallèles faiblement équitables.

- Au contraire, Manna-Pnueli [81a] ne considèrent que des traces infinies, nos traces finies étant prolongées en répétant indéfiniment le dernier état (ils auraient pu également répéter indéfiniment un état spécial "indéfini"). Cette approche nous semble avoir l'inconvénient de masquer les blocages (de sorte que, par exemple, pour une preuve de fatalité relationnelle ($\Phi = \Psi$), la condition $(\mathcal{F}_0)$-(b), assurant qu'aucun état de blocage ne peut être atteint avant le but, devient inutile et

se retrouve cachée dans la condition $(\mathcal{F}_s)$-(d) puisque le but ne peut pas être satisfait par un état indéfini).

- Lamport [80] et Emerson [81] choisissent $\underset{\sim}{Suff}(\langle S, A, \Sigma\rangle)$ avec, dit le premier, l'idée que ce choix est nécessaire pour exprimer que "la façon dont le calcul évolue dans le futur ne dépend que de son état courant". Cet argument nous semble incomplet, la notion de sémantique close (cf. 2.6.8) étant mieux adaptée. Quand la sémantique est fermée par suffixes, il est possible d'exprimer certaines propriétés de la sémantique sans recourir aux indices pour désigner des états dans les traces et donc sans utiliser (directement) la notion de temps. Par exemple, la fatalité de l'assertion $\Psi \in (S \to \{tt, ff\})$ pour $\langle S, A, \Sigma'\rangle = \underset{\sim}{Suff}(\langle S, A, \Sigma\rangle)$ peut s'exprimer par :

$$\forall p \in \Sigma'.\ \exists p' \in \Sigma'.\ (p' \twoheadrightarrow p \wedge \Psi(p'_0))$$

L'inconvénient principal de la fermeture par suffixes nous semble être que la notion d'états initiaux (cf. 2.3 : 1) disparaît.

- Hoare [81] choisit $\underset{\sim}{Pref}^{<\omega}(\langle S, A, \Sigma\rangle)$ ce qui revient à considérer non seulement les calculs complets ou maximaux mais également des calculs en cours ou initiaux. La condition de fermeture par préfixes corspond à l'idée que le préfixe d'un calcul en cours est également un calcul en cours. De plus, l'idée de Hoare est qu'il est possible de raisonner sur des comportements infinis (c'est-à-dire $\underset{\sim}{Flim}(\underset{\sim}{Pref}^{<\omega}(\langle S, A, \Sigma\rangle))$ en n'utilisant que des traces finies (c'est-à-dire $\underset{\sim}{Pref}^{<\omega}(\langle S, A, \Sigma\rangle)$), ce qui, comme nous l'avons vu précédemment pour Pratt [82] n'est pas toujours possible (quand $\underset{\sim}{Flim}(\underset{\sim}{Pref}^{<\omega}(\langle S, A, \Sigma\rangle)) \neq \underset{\sim}{Pref}(\langle S, A, \Sigma\rangle)$). Finalement, quand on raisonne sur la fermeture par préfixes d'une sémantique, il n'est plus possible d'exprimer, par exemple, qu'il peut survenir des pannes aléatoires car dans ce cas certains états peuvent être le dernier état d'une trace finie terminée (quand il y a une panne)

qui est également préfixe strict d'une autre trace (quand la panne n'a pas lieu). Avec le point de vue de Hoare, le cas d'arrêt sur panne doit se traiter par passage dans un état indéfini n'ayant pas de successeurs.

- Pour éviter les inconvénients inhérents aux propositions précédentes, Arnold-Nivat [82] choisissent de décrire l'ensemble de traces Σ par trois ensembles à savoir ce qu'ils appelent les "comportements initiaux" (c'est-à-dire les préfixes finis $\underline{Pref}^{<\omega}(<S,A,\Sigma>)$ des traces de Σ), les "comportements finis terminés" (c'est-à-dire les traces finies $\Sigma \cap \Sigma^{<\omega}<S,A>$ de Σ) et les "comportements infinis" (c'est-à-dire les traces infinies $\Sigma \cap \Sigma^{\omega}<S,A>$ de Σ). Il faut alors ajouter un certain nombre de conditions de cohérence (les comportements initiaux sont fermés par préfixes, les comportements finis terminés sont des comportements initiaux, les préfixes finis des comportements infinis sont des comportements initiaux).

Par comparaison avec ces modèles de sémantique opérationnelle basés sur la notion de traces, notre choix s'avère plus simple et aussi expressif. En utilisant des opérateurs convenablement définis sur les sémantiques, nous pourrons retrouver tous les modèles précédemment cités comme cas particuliers.

Nous avons modélisé le parallélisme par le mélange nondéterministe d'actions atomiques. En principe il est possible de déterminer l'ordre réel des actions et comme deux actions atomiques concurrentes ne peuvent avoir aucune influence l'une sur l'autre, nous pouvons supposer qu'elles ont été exécutées dans n'importe quel ordre. Pour des actions qui ne sont pas atomiques il est alors nécessaire de

les subdiviser en actions atomiques plus petites.

Ce point de vue est bien adapté à l'expression de l'exclusion mutuelle mais pas à celle de la simultanéité (deux actions s'exécutent en même temps). Pour ce faire, on peut considérer qu'une action pour un programme parallèle est un vecteur d'actions pour chacun de ses processus, une action spéciale pouvant (comme dans Arnold-Nivat [82]) dénoter qu'un processus est inactif à l'instant. On peut évidemment avoir besoin de décrire non seulement des simultanéités mais également des recouvrements. Pour ce faire, on peut imaginer de marquer le début et la fin de chaque action par un événement $a \in A$ de la sémantique $\langle S, A, \Sigma \rangle$.

Pour décrire l'ensemble de traces Σ, nous avons utilisé l'intermédiaire d'un système de transition. De manière plus générale, on peut imaginer (à la suite de Lamport [78]) de définir un ordre partiel sur des événements (marquant le début et la fin de chaque action) satisfaisant un certain nombre de contraintes d'ordonnancement (du genre les événements relatifs à un même processus sont totalement ordonnés, la fin de l'envoi d'un message précède le début de sa réception, etc ...). On peut évidemment combiner ces différentes idées en décrivant un programme parallèle à l'aide d'un système de transition pour chaque processus, d'un ordre partiel sur des événements exprimant des conditions de synchronisation et d'une condition globale sur les traces exprimant des hypothèses d'équité.

6.2 PROPRIETES D'INVARIANCE ET DE FATALITE DES PROGRAMMES

Nous avons étudié les propriétés d'invariance conditionnelle et de fatalité sous invariance à cause de leur importance et de leur simplicité. En combinant ces deux types de propriétés, il est possible d'exprimer la plupart des propriétés relatives à la correction des programmes.

Il aurait cependant été aisé de le généraliser mais ce, au prix de complications qui nous ont semblées inutiles.

Par exemple, nous aurions pu prendre comme état initial non pas l'état de départ de la trace mais un état intermédiaire quelconque :

Pour l'invariance relationnelle, nous aurions alors la définition :

$$\forall p \in \Sigma.\ \forall i \in |p|.\ \forall j \in |p|.\ (j \geq i \Rightarrow \Psi(p_i, p_j))$$

tandis que la fatalité relationnelle serait définie par :

$$\forall p \in \Sigma.\ \forall i \in |p|.\ \exists j \in |p|.\ (j \geq i \wedge \Psi(p_i, p_j))$$

mais ce type de formule revient à considérer une propriété d'invariance ou de fatalité telles que nous les avons définies pour la fermeture par suffixes $\underset{\sim}{Suff}(\langle S, A, \Sigma \rangle)$.

Il est également possible de considérer des propriétés relatives au passé plutôt qu'au futur comme par exemple (Clarke-al. [81]) :

$$\forall p \in \Sigma,\ i \in |p|.\ \exists k \in |p|.\ [k \leq i \wedge \Psi(p_k, p_i) \wedge \forall j \in \omega.\ [k < j \leq i \Rightarrow \Phi(p_j, p_i)]]$$

mais là encore il suffit de raisonner sur une sémantique transformée pour adapter les principes d'induction.

Les logiques temporelles (Lamport [80]) du type "linear type logic" comprennent des formules du genre

$\models \Box \psi$ et $\models \Diamond \psi$

qui s'interprètent, pour une sémantique donnée $\langle S, A, \Sigma \rangle$, respectivement par

$\forall p \in \Sigma.\ \forall i \in |p|.\ \psi(p^{\geq i})$ et $\forall p \in \Sigma.\ \exists i \in |p|.\ \psi(p^{\geq i})$

Les logiques temporelles (Lamport [80]) du type "branching type logic" comprennent des formules du genre :

$\models$ ALL ψ $\models$ SOME ψ $\models$ POT ψ et $\models$ INEV ψ

qui s'interprètent respectivement par les formules :

$\forall A \in S.\ \forall p \in \Sigma.\ \forall i \in |p|.\ [(p_i = A) \Rightarrow \forall j \in |p|.\ [(j \geq i) \Rightarrow \psi(p_j)]]$

$\forall A \in S.\ \exists p \in \Sigma.\ \exists i \in |p|.\ [(p_i = A) \wedge \forall j \in |p|.\ [(j \geq i) \Rightarrow \psi(p_j)]]$

$\forall A \in S.\ \exists p \in \Sigma.\ \exists i \in |p|.\ [(p_i = A) \wedge \exists j \in |p|.\ [(j \geq i) \wedge \psi(p_j)]]$

$\forall A \in S.\ \forall p \in \Sigma.\ \forall i \in |p|.\ [(p_i = A) \Rightarrow \exists j \in |p|.\ [(j \geq i) \wedge \psi(p_j)]]$

Ces logiques sont utilisées pour faire des spécifications et faire des preuves. En ce qui concerne les spécifications il s'agit d'exprimer des propriétés des traces. Ceci peut se faire avec la logique ordinaire qui permet d'exprimer des assertions arbitraires sur les traces d'exécution. Les défenseurs des logiques temporelles pensent que cette généralité n'est pas nécessaire et qu'on peut se restreindre à quelques modalités bien choisies. Sans entrer dans l'énumération de toutes les modalités possibles, nous nous contentons d'une comparaison avec les logiques linéaire et arborescente de Lamport [80], brièvement décrites ci-dessus :

- Remarquons qu'il n'est pas possible de lier l'état courant p_i à l'état initial p_0 sauf en utilisant des variables auxiliaires (comme dans la preuve du théorème 4.2.1.4 v 2). Ceci nous ramène

à des discussions antérieures (cf. 4.3.2.4.5, 5.3.1.6) et à notre préférence pour l'utilisation explicite de variables auxiliaires dans la preuve plutôt qu'à l'utilisation implicite de variables auxiliaires dans un programme transformé.

- Les propriétés universelles de la "linear type logic" sont bien adaptées pour exprimer des propriétés de correction de programmes nondéterministes exécutés en profondeur (une alternative est choisie puis menée à son terme) ou de programmes parallèles. C'est ce genre de propriétés que nous avons retenues (cf. Ch.3). En s'inspirant des logiques du type "branching time", il est également possible de considérer des propriétés de la forme :

$$\exists p \in \Sigma.\ \forall i \in |p|.\ \Phi(p_0, p_i)$$

$$\exists p \in \Sigma.\ \exists i \in |p|.\ \Phi(p_0, p_i)$$

Nous n'avons pas étudié ces propriétés existentielles car il est très facile d'adapter ce que nous avons fait pour les propriétés universelles correspondantes et nous avons voulu éviter ce qui peut paraître comme d'évidentes redites.

- Il est bien connu qu'il existe des propriétés des programmes qu'on peut spécifier avec une logique de type "linear time" et pas avec une logique de type "branching time" et inversement (Lamport [80], Graf [84]). Il existe également des propriétés de programmes qui ne peuvent s'exprimer ni par l'une ni par l'autre de ces logiques (Clarke-al [81]). Ceci explique la multiplication de propositions concernant l'introduction de nouvelles modalités et montre que le choix des quelques modalités auxquelles on peut se restreindre n'est pas si facile. C'est pourquoi nous avons utilisé au chapitre 3 une méthode de spécification qui nous semble tout à fait générale dans la mesure où nous raisonnons explicitement sur

l'ensemble des traces p de la sémantique en utilisant des indices $i \in |p|$ pour désigner l'état p_i du calcul à un instant i quelconque.

6.3 PREUVES D'INVARIANCE ET DE FATALITE

On peut également utiliser ces logiques pour faire des preuves et ceci nous amène à une comparaison avec les chapitres 4 et 5. En logique, les preuves se font à l'aide de systèmes formels comportant des axiomes (dont certains expriment la sémantique du programme) et des règles d'inférence.

- Il semble difficile d'utiliser cette méthode pour formaliser des méthodes de preuve de programmes indépendamment du langage de programmation utilisé. Par exemple Apt-Delporte [83] ont essayé de formaliser la méthode des assertions intermittentes de Burstall à l'aide d'une logique temporelle pour en démontrer la correction et la complétude arithmétique. Cette étude est faite pour la correction totale de programmes itératifs (du style de ceux considérés en 2.8.1) et on ne voit pas, par exemple, comment généraliser à d'autres propriétés du même genre ou au cas des programmes parallèles. La formalisation des méthodes de preuve à l'aide de nos principes d'induction nous semble donc plus abstraite et générale.

- Pour formaliser l'induction utilisée dans les preuves de programmes, les règles d'inférence pour ces logiques incluent généralement une règle très générale d'induction mathématique de la forme (Manna [81]) :

$$[[\psi(0) \wedge \forall n \in \omega . [\psi(n) \Rightarrow \psi(n+1)]] \Rightarrow \forall n \in \omega . \psi(n)]$$

ou même l'induction transfinie sur les ordinaux (qui est plus générale):

$$[[\forall \alpha \in \delta.((\forall \beta \in \alpha.\ \psi(\beta)) \Rightarrow \psi(\alpha))] \Rightarrow \forall \alpha \in \delta.\ \psi(\alpha)]$$

Les autres formes d'induction peuvent évidemment s'en déduire (puisque c'est ce principe d'induction que nous avons utilisé pour démontrer la correction de tous nos principes d'induction). Cette forme de généralité permet par exemple à Manna-Pnueli [81a,b] de formaliser les méthodes de Floyd et Burstall en ajoutant (presque exclusivement) que les assertions considérées (pour les programmes séquentiels) sont de la forme :

$$\vdash (at\ l \wedge \phi) \Rightarrow \Diamond (at\ l' \wedge \psi)$$

A notre avis, cette forme de généralités n'apprend pas grand chose pour mieux comprendre ces méthodes alors que notre formalisation du chapitre 5 est beaucoup plus précise. De plus pour arriver à un système de déduction utilisable en pratique, il est nécessaire de déduire des axiomes et règles d'inférence de base, un grand nombre de règles dérivées (il y a 24 axiomes, 4 règles d'inférence de base et 62 règles dérivées dans Manna [81]). Cette profusion de règles est à contraster avec la concision des principes d'induction du chapitre 5.

- Il est bien évident qu'en pratique, on s'intéresse à la preuve simultanée de plusieurs propriétés d'un programme. Dans ce cas, on s'efforce d'éviter les répétitions en combinant autant que faire se peut les différentes preuves. Nous avons abordé à plusieurs reprises ce problème (cf. par exemple, la discussion en 4.3.2.2.6) qui semble être ignoré dans le domaine des logiques temporelles (qui ne permettent guère de démontrer $\vdash \Box\phi \wedge \Diamond\psi$ qu'en démontrant $\vdash \Box\phi$ et $\vdash \Diamond\psi$ séparément).

- Par contre, les logiques temporelles permettent d'exprimer aisément des propriétés du type $\vdash \Box\Diamond\psi$ combinant plusieurs modalités, (d'où l'avantage de faire porter ψ (aussi bien que $\Diamond\psi$) sur des (suffixes de) traces plutôt que sur des (paires d') états). Malheureusement, il n'est en général pas prévu d'axiomes ou de règles d'inférence pour les preuves comportant de telles combinaisons de modalités. Nous devons dire que nous n'avons pas nous non plus, abordé ce problème.

- Ceci nous amène enfin au problème de la structuration des preuves en particulier pour les gros programmes. Nous avons proposé diverses méthodes de décompositions des preuves, basées sur la sémantique des programmes (cf. par exemple 4.3.2.4 et 5.2.7) et généralisé l'idée de Burstall, classique en mathématiques, de décomposition de la preuve d'un théorème à l'aide de lemmes. Cette étude devrait pouvoir être approfondie notamment dans le cas des programmes distribués (pour éviter les phénomènes d'interférence) et des programmes modulaires (pour faire correspondre les lemmes aux modules). Pour explorer cette voie, on pourrait s'inspirer de Jones [85].

6.4 REFERENCES

APT K.R., DELPORTE C. [83], " An axiomatization of the intermittent assertion method", Rapport de Recherche 82-70, LITP, Paris, (Jan. 1983), 21p.

ARNOLD A., NIVAT M. [82], " Comportements de processus ", Rapport de Recherche 82-12, LITP, Paris, (Février 1982), 34p.

CLARKE Ed., FRANCEZ N., GUREVICH Y., SISTLA A. [81], " Can message buffers be characterized in linear temporal logic ", Rapport de Recherche, Harvard U., (1981), 14p.

EMERSON E.A. [81], " Alternative semantics for temporal logics ", Rapport de Recherche TR-182, Texas U., Austin, (Oct. 1981).

GRAF S. [84], " On Lamport's comparison between linear and branching time temporal logic ", RAIRO Inf. Théorique, 18, 4 (1984), 345-353.

HOARE C.A.R. [81], " A calculus of total correctness for communicating processes ", SCP 1 (1981), 49-72.

JONES C.B. [85], " The role of proof obligations in software design ", TAPSOFT 85, Lect. Notes in Comp. Sci. 186, (1985), 27-41.

LAMPORT L. [78], " Time, clocks and the ordering of events in a distributed system ", CACM 21, 7 (July 1978), 558-565.

LAMPORT L. [80], " 'Sometime' is sometimes 'not never' ", 7th annual ACM Symp. on principles of programming languages, (1980), 174-185.

MANNA Z. [81], " Verification of sequential programs : temporal axiomatization ", Rapport de Recherche STAN-CS-81-877, Dept. of Comp. Sci., Stanford U., (Sept. 1981), 45p.

MANNA Z., PNUELI A. [81a], "Vérification of concurrent programs, part I : the temporal logic framework", Rapport de Recherche STAN-CS-81-836, Dept. of Comp. Sci., Stanford U., (1981), 62 p.

MANNA Z., PNUELI A. [81b], "Vérification of concurrent programs, part II : temporal proof principles", Rapport de Recherche STAN-CS-81-843, Dept. of Comp. Sci., Stanford U., (1981), 51 p.

PRATT V.R. [82], "On the composition of processes", 9th ACM annual symp. on principles of programming Languages, (1982), 213-223.

ANNEXE I :

NOTATIONS MATHEMATIQUES

I.1 NOTATIONS DE LOGIQUE

Nous utilisons la logique du premier ordre avec egalité de manière intuitive. Les variables logiques sont notées $x, x, \ldots$, les prédicats $P, Q, \ldots$ et nous utilisons la convention habituelle que $P(x)$ signifie que x peut apparaître comme variable libre dans P. Nous utilisons les symboles logiques $P \vee Q$ (disjonction),

, $P \Rightarrow Q$ (implication), $P \Leftrightarrow Q$

$\forall x.\, P(x)$ (quantification universelle)

elle) et diverses formes de

s notons $P \Leftarrow Q$, $P \not\Rightarrow Q$, $P \not\Leftarrow Q$,

$x \neq y$, ... respectivement pour $Q \Rightarrow P$, $\neg(P \Rightarrow Q)$, $\neg(Q \Rightarrow P)$, $\neg(x = y)$, ... $(P \Rightarrow Q | R)$ est l'abréviation de $((P \wedge Q) \vee (\neg P \wedge R))$, $x = (P \rightarrow y | z)$ celle de $(P \Rightarrow x = y | x = z)$, $\exists! x.\, P(x)$ celle de $\exists y. \forall x. (x = y \Leftrightarrow P(x))$ où y n'est pas libre dans P. Nous écrivons $\forall x_0 \in X_0, \ldots, x_i \in X_i, \ldots .\ P(x_0, \ldots, x_i, \ldots)$ pour $\forall x_0 . \ldots \forall x_i . \ldots (x_0 \in X_0 \Rightarrow \ldots (x_i \in X_i \Rightarrow \ldots P(x_0, \ldots, x_i, \ldots) \ldots) \ldots)$ et de même avec les

Par abus de notation les qu

omis. Les valeurs de vérité tt et ff

vec leurs interprétations, c'est-à-dire

ions des domaines de valeurs de

e $\{tt, ff\}$ des valeurs de vérité.

Dans ce cas, nous notons $P(x_0, \ldots, x_i, \ldots)$ où $x_0, \ldots x_i, \ldots$ sont les seules variables qui peuvent apparaître libres dans P et dans certains cas l'équivalence $P \Leftrightarrow Q$ st noteé $P = Q$.

I.2 NOTATIONS ENSEMBLISTES ELEMENTAIRES

Nous utilisons l'axiomatisation de la théorie des ensembles de Zermelo-Fraenkel et admettons l'axiome du choix. Les prédicats de cette théorie sont ceux d'une logique avec égalité où les notions de classes $X, X', X_0, \ldots, X_i, \ldots, x, x', x_0, \ldots, x_i, \ldots$ et d'appartenance $\in$ sont considérées comme primitives. X est un ensemble si et seulement si il existe Y tel que $X \in Y$. $\mathrm{Set}(X)$ est l'abréviation de $\exists Y.(X \in Y)$.

Si $P(x)$ est un prédicat et X n'est pas libre dans P, alors $\{x : P(x)\}$ est l'unique classe X telle que $\forall x.(x \in X \Leftrightarrow (\mathrm{Set}(x) \wedge P(x)))$ c'est-à-dire la classe de tous les ensembles x tels que $P(x)$. Nous écrivons $\{x_0, \ldots, x_i, \ldots\}$ pour $\{x : x = x_0 \vee \ldots \vee x = x_i \vee \ldots\}$. Plus généralement si $\tau(x_0, \ldots, x_i, \ldots)$ est un terme et $P(x_0, \ldots, x_i, \ldots)$ un prédicat de la théorie des ensembles dans lesquels X n'apparaît pas, nous définissons $\{\tau(x_0, \ldots, x_i, \ldots) : P(x_0, \ldots, x_i, \ldots)\} = \{X : \exists x_0, \ldots, x_i, \ldots . (P(x_0, \ldots, x_i, \ldots) \wedge X = \tau(x_0, \ldots, x_i, \ldots))\}$. $\{\tau \in Y : P\}$ est l'abréviation de $\{\tau : \tau \in Y \wedge P\}$.

L'inclusion est définie par $X \subseteq Y \Leftrightarrow \forall z.(z \in X \Rightarrow z \in Y)$, l'inclusion stricte par $X \subset Y \Leftrightarrow (X \subseteq Y \wedge X \neq Y)$. Nous écrivons également $X \supseteq Y$ pour $Y \subseteq X$, $X \not\subseteq Y$ pour $\neg(X \subseteq Y)$, etc ... .

L'ensemble vide zéro noté 0 ou $\emptyset$ est $\{x : x \neq x\}$.

Nous définissons l'union $X \cup Y = \{x : x \in X \vee x \in Y\}$, l'union infinie $\cup X = \{x : \exists y \in X.(x \in y)\}$ et $\bigcup_{i \in I} A_i = \cup\{A_i : i \in I\}$ quand A est une famille indexée d'ensembles. De même pour l'intersection $X \cap Y = \{x : x \in X \wedge x \in Y\}$, $\cap X = \{x : \forall y \in X.(x \in y)\}$ et $\bigcap_{i \in I} A_i = \cap\{A_i : i \in I\}$. La différence de classes est $X \setminus Y = \{x : x \in X \wedge x \notin Y\}$ avec la notation particulière $X \setminus x = X \setminus \{x\}$ pour la différence avec un singleton. La puissance est $\mathcal{P}^X = \{x : x \subseteq X\}$. Si z est la paire ordonnée $\langle x, y \rangle = \{\{x\}, \{x, y\}\}$, nous notons $z_0 = z(0) = x$ et $z_1 = z(1) = y$. Ceci permet de définir le produit cartésien $X \times Y = \{\langle x, y \rangle : x \in X \wedge y \in Y\}$.

Une classe r est une relation (binaire) si tout membre x de r est une paire ordonnée c'est-à-dire que $\forall x.[(x\in r)\Rightarrow(\exists y,z.(x=\langle y,z\rangle))]$. Le domaine d'une relation r est $\mathrm{dom}(r)=\{x:\exists y.(\langle x,y\rangle\in r)\}$, son co-domaine est $\mathrm{rng}(r)=\{y:\exists x.(\langle x,y\rangle\in r)\}$, son champ est $\mathrm{fld}(r)=\mathrm{dom}(r)\cup\mathrm{rng}(r)$.

que $\forall x,y,z.((\langle x,y\rangle\in f\wedge\langle x,z\rangle\in f)\Rightarrow(y=z))$
n fonctionnelle $f(x)$, fx ou f_x pour
que $\langle x,y\rangle\in f$. Nous notons $X\to Y$
X dans Y c'est-à-dire

$\{f: f$ est une fonction $\wedge\ \mathrm{dom}(f)\subseteq X\wedge\mathrm{rng}(f)\subseteq Y\}$ et $X\to Y$ la classe des fonctions totales de X dans Y c'est-à-dire $\{f: f\in(X\to Y)\wedge\mathrm{dom}(f)=X\}$. La composition fonctionnelle est définie par $fog(x)=f(g(x))$.
Si $f\in(X_0\times\dots\times X_m\to Y)$ nous écrivons $f(x_0,\dots,x_m)$ au lieu de $f(\langle\dots\langle\langle x_0,x_1\rangle,x_2\rangle,\dots,x_m\rangle)$. $f\in(X\to Y)$ est injective (respectivement surjective, bijective) si et seulement si $\forall x,y\in\mathrm{dom}(f).[(x\neq y)\Rightarrow(f(x)\neq f(y))]$ (respectivement $\mathrm{rng}(f)=Y$, f est injective et surjective). La restriction $f\restriction X$ de la fonction f à X est $f\cap(X\times\mathrm{rng}(f))$. Si $\tau(x)$ est un terme ensembliste (désignant une classe) et $P(x)$ un prédicat alors $\langle\tau(x):P(x)\rangle$ dénote la fonction $\{\langle x,\tau(x)\rangle:P(x)\}$.

Dans la suite nous décrirons souvent les ensembles X au moyen de leurs fonctions caractéristiques également notées X tellesque $X\in(\mathcal{U}\to\{tt,ff\})$, $\mathcal{U}=\{x:tt\}$ est l'univers et $\forall x.(X(x)=(x\in X))$. Les opérateurs logiques sont étendus point par point : $X\wedge Y(x)=X(x)\wedge Y(x)$, etc. En particulier pour les définissons $\mathrm{rel}(X,r)=[r\in(X\times X\to\{tt,ff\})]$ et écrirons $\langle x,y\rangle\in r$, $x\,r\,y$ et $r(x,y)$. La restriction (respectivement gauche, droite) de la relation r à X est $r\restriction X(x,y)=[x\in X\wedge r(x,y)\wedge y\in X]$ (respectivement $r1X(x,y)=[x\in X\wedge r(x,y)]$, $r\restriction X(x,y)=[r(x,y)\wedge y\in X]$). La composition de relations est $r\circ s(x,y)=\exists z.[r(x,z)\wedge s(z,y)]$. La relation identité est $\underline{1}(x,y)=[x=y]$. Ayant introduit l'ensemble ω des entiers naturels, nous définissons $r^0=\underline{1}$, $r^{m+1}=r\circ r^m$, la fermeture transitive $r^+=\exists m\in(\omega\vee 0).r^m$ et la fermeture transitive

réflexive $r^* = \mathbb{1} \vee r^+$. L'inverse d'une relation est $r^{-1}(x,y) = r(y,x)$. L'image $r[X]$ de X par r est $\{y : \exists x \in X.\ (x\,r\,y)\}$. Un morphisme f de la relation r dans la relation r' st tel que $[f \in (\mathrm{fld}(r) \to \mathrm{fld}(r')) \wedge \forall a,b \in \mathrm{fld}(r).\ [a\,r\,b = (fa)\,r'\,(fb)]]$. C'st un isomorphisme si f est bijective, épimorphisme si f st surjective, monomorphisme si f est injective, un endomorphisme si $\mathrm{fld}(r) = \mathrm{fld}(r')$ et un automorphisme si c'st un endomorphisme et un isomorphisme.

r est une relation d'équivalence sur X si elle est réflexive (i.e. $\forall x \in X.\ x\,r\,x$), symétrique (i.e. $\forall x,y \in X.\ x\,r\,y \Rightarrow y\,r\,x$) et transitive (i.e. $\forall x,y,z \in X.\ (x\,r\,y \wedge y\,r\,z) \Rightarrow x\,r\,z$). r induit une partition (i.e. tout élément de X appartient à exactement un seul élément de la partition) de X en classes d'équivalence $[x]_r = \{y : y \in X \wedge x\,r\,y\}$. L'ensemble des classes d'équivalence de X pour r est appelé l'ensemble quotient de X modulo r et est noté X/r.

I.3 ORDRES

Une relation d'ordre (partiel) strict est irreflexive et transitive $\mathrm{apo}(W,\prec) = (\mathrm{rel}(W,\prec) \wedge [\forall x \in W.\ \neg(x \prec x)] \wedge [\forall x,y,z \in W.\ (x \prec y \wedge y \prec z) \Rightarrow x \prec z])$. La relation d'ordre reflexif $\preccurlyeq$ correspondant à $\prec$ est $x \preccurlyeq y = [x \prec y \vee x = y]$. Une relation d'ordre reflexif est reflexive, transitive et antisymétrique $\mathrm{rpo}(W,\preccurlyeq) = (\mathrm{rel}(W,\preccurlyeq) \wedge [\forall x \in W.\ x \preccurlyeq x] \wedge [\forall x,y,z \in W.\ (x \preccurlyeq y \wedge y \preccurlyeq z) \Rightarrow x \preccurlyeq z] \wedge [\forall x,y,z \in W.\ (x \preccurlyeq y \wedge y \preccurlyeq x) \Rightarrow x = y])$. Nous notons $\succ$ (respectivement $\succcurlyeq$) l'inverse de $\prec$ (respectivement $\preccurlyeq$) et $\mathrm{po}(W,r) = \mathrm{apo}(W,r) \vee \mathrm{rpo}(W,r)$.

Si $\mathrm{rpo}(W,\preccurlyeq)$, $X \subseteq W$ et $a \in W$ alors a est un majorant (respectivement majorant strict, minorant, minorant strict) de X pour $\preccurlyeq$ si et seulement si $\forall x \in X.\ x \preccurlyeq a$ (respectivement $\forall x \in X.\ x \prec a$, $\forall x \in X.\ a \prec x$, $\forall x \in X.\ a \prec x$). a est le plus grand (respectivement plus petit) élement de X pour $\preccurlyeq$ si c'est un majorant (respectivement minorant) de X pour $\preccurlyeq$ et $a \in X$. a est un élément maximal (respectivement minimal) de X pour $\preccurlyeq$ si $a \in X \wedge \forall x \in X.\ \neg(a \prec x)$ (respectivement $\forall x \in X.\ \neg(x \prec a)$). a est la borne supérieure (respectivement inférieure) de X pour $\preccurlyeq$ si a st le plus petit (respectivement grand) des majorants (respectivement minorants) de X pour $\preccurlyeq$ c'st-à-dire $[\forall x \in X.\ x \preccurlyeq a \wedge \forall y \in W.\ (\forall x \in X.\ x \preccurlyeq y) \Rightarrow (a \preccurlyeq y)]$ (respectivement $[\forall x \in X.\ a \preccurlyeq x \wedge \forall y \in W.\ (\forall x \in X.\ y \preccurlyeq x) \Rightarrow (y \preccurlyeq a)]$). Si elle existe la borne supérieure ou supremum de X pour $\preccurlyeq$ st notée $\sup(W,\preccurlyeq)X$ (ou $\sqcup X$). La borne inférieure ou infimum est notée $\inf(W,\preccurlyeq)X$ (ou $\sqcap X$). La borne supérieure (respectivement inférieure) stricte de X pour $\preccurlyeq$ est notée $\sup^+(W,\preccurlyeq)X$ (respectivement $\inf^+(W,\preccurlyeq)X$) si elle existe. C'st le plus petit (respectivement grand) des majorants (respectivement minorants) stricts de X pour $\preccurlyeq$.

Un ordre r (strict ou reflexif) sur W est linéaire ou total si et seulement si deux éléments quelconques distincts sont comparables c'est-à-dire $\mathrm{lo}(W,r) = [\mathrm{po}(W,r) \wedge \forall x,y \in W.\ [(x \neq y) \Rightarrow (x\,r\,y \vee y\,r\,x)]]$. Une relation $\prec$ sur W est bien-fondée si et seulement si toute partie non vide a un élément minimal. $\mathrm{wf}(W,\prec) = [\mathrm{rel}(W,\prec) \wedge \forall X \subseteq W.\ [X \neq 0 \Rightarrow \exists y \in X.\ (\neg \exists z \in X.\ z \prec y)]]$. Ayant admis

l'axiome du choix, ceci est équivalent à l'assertion que toute chaine strictement décroissante est finie. Nous écrivons $\underset{\sim}{\omega fi}(W,\prec,\mu)$ quand $\prec$ est une relation bien fondée sur W avec un élément minimal μ, $\underset{\sim}{\omega fi}(W,\prec,\mu) = [\underset{\sim}{\omega f}(W,\prec) \wedge \mu \in W \wedge \forall x \in W. \neg(x \prec \mu)]$. Une relation $\prec$ de bon-ordre sur W est linéaire et bien-fondée $\underset{\sim}{\omega o}(W,\prec) = [\underset{\sim}{lo}(W,\prec) \wedge \underset{\sim}{\omega f}(W,\prec)]$.

Une classe partiellement ordonnée est une paire $\langle W,\prec\rangle$ telle que $\underset{\sim}{po}(W,\prec)$. L'addition de classes partiellement ordonnées est définie par $\langle W_0,\prec_0\rangle \oplus \langle W_1,\prec_1\rangle = \langle W,\prec\rangle$ si et seulement si $W = \{\langle 0,w_0\rangle : w_0 \in W_0\} \cup \{\langle 1,w_1\rangle : w_1 \in W_1\}$ et $\langle i,x\rangle \prec \langle j,y\rangle$ si et seulement si $[(i=j=0 \wedge x \prec_0 y) \vee (i<j) \vee (i=j=1 \wedge x \prec_1 y)]$ (c'est-à-dire que dans $\langle W,\prec\rangle$ les éléments de W_0 sont ordonnés comme dans $\langle W_0,\prec_0\rangle$, tous les membres de W_0 précèdent ceux de W_1 et les éléments de W_1 sont ordonnés comme dans $\langle W_1,\prec_1\rangle$). La multiplication de classes ordonnées est définie par $\langle W_0,\prec_0\rangle \otimes \langle W_1,\prec_1\rangle = \langle W,\prec\rangle$ si et seulement si $W = W_0 \times W_1$ et $\prec$ est l'ordre lexicographique droit, $\langle x,y\rangle \prec \langle x',y'\rangle$ si et seulement si $[y \prec_1 y' \vee (y=y' \wedge x \prec_0 x')]$.

Soit $\langle W,\preccurlyeq\rangle$ une classe partiellement ordonnée. $X \subseteq W$ est dirigé si et seulement si $\forall x,x' \in X. \exists x'' \in X. (x \preccurlyeq x'' \wedge x' \preccurlyeq x'')$. $\langle W,\preccurlyeq\rangle$ est un $\underset{\sim}{spo}$ (ordre partiel complet ou encore ensemble inductif) si W est un ensemble possédant un plus petit élément et tout ensemble dirigé $X \subseteq W$ admet une borne supérieure. $\langle W,\preccurlyeq,\wedge,\vee,\bot,\top\rangle$ est un treillis complet si pour toute partie non vide $X \subseteq W$, sa borne supérieure $\underset{\sim}{sup}(W,\preccurlyeq)X$ (notée $\vee X$) et sa borne inférieure $\underset{\sim}{inf}(W,\preccurlyeq)X$ (notée $\wedge X$) existent. Ceci entraîne en particulier que W admet un plus petit élément $\bot = \underset{\sim}{inf}(W,\preccurlyeq)W$ et un plus grand élément $\top = \underset{\sim}{sup}(W,\preccurlyeq)W$. Un treillis booléen complet $\langle W,\preccurlyeq,\wedge,\vee,\bot,\top,\neg\rangle$ est un treillis distributif complémenté ($\neg x$ désigne un complément de x) complet.

Soient $\langle W,\preccurlyeq\rangle$ une classe partiellement ordonnée et $e \in (W \to W)$. e est une rétraction sur W s'il est monotone (i.e. $\forall x,y \in W. (x \preccurlyeq y \Rightarrow e(x) \preccurlyeq e(y))$) et idempotent (i.e. $e = e \circ e$ c'est-à-dire $\forall x \in W. e(x) = e(e(x))$). e est une préfermeture supérieure (dualement inférieure) sur W s'il est monotone, idempotent et satisfait

l'axiome de connectivité supérieure (i.e. $\forall x \in W.\ \varphi(\sup(W,\preceq)\{x,\varphi(x)\}) = \varphi(x)$) (dualement l'axiome de connectivité inférieure (i.e. $\forall x \in W.\ \varphi(\inf(W,\preceq)\{x,\varphi(x)\}) = \varphi(x)$). φ est une fermeture supérieure (dualement inférieure) sur W s'il est monotone, idempotent et extensif (i.e. $\forall x \in W.\ x \preceq \varphi(x)$) (dualement réductif (i.e. $\forall x \in W.\ \varphi(x) \preceq x$)).

Soient $\langle W_1, \preceq_1\rangle$ et $\langle W_2, \preceq_2\rangle$ deux classes partiellement ordonnées et (α,γ) une paire de fonctions monotones $\alpha \in (W_1 \to W_2)$ et $\gamma \in (W_2 \to W_1)$ et telles que $\alpha\gamma \preceq_1 \underline{1}$ et $\underline{1} \preceq_2 \gamma\alpha$. (α,γ) est une correspondance de Galois.
En définissant les fonctions partielles $\vartheta_1 \in ((W_2 \to W_1) \to (W_1 \to W_2))$ et $\vartheta_2 \in ((W_1 \to W_2) \to (W_2 \to W_1))$ comme suit : $\vartheta_1(\gamma)(x) = \wedge\{y \in W_2 : x \preceq_1 \gamma(y)\}$, $\vartheta_2(\alpha)(y) = \vee\{x \in W_1 : \alpha(x) \preceq_2 y\}$, nous avons les résultats suivants : $[\alpha\gamma \preceq_1 \underline{1}$ et $\underline{1} \preceq_2 \gamma\alpha] \Leftrightarrow [\alpha$ est un v-morphisme complet et $\gamma = \vartheta_2(\alpha)] \Leftrightarrow [\forall x \in W_1, y \in W_2.\ (\alpha(x) \preceq_2 y) \Leftrightarrow (x \preceq_1 \gamma(y))] \Leftrightarrow [\vartheta_1(\gamma) = \alpha$ et $\vartheta_2(\alpha) = \gamma] \Leftrightarrow$ $[\alpha \preceq_2 \vartheta_2(\gamma)$ et $\vartheta_2(\alpha) \preceq_1 \gamma] \Leftrightarrow [\gamma$ est un $\wedge$-morphisme complet et $\alpha = \vartheta_1(\gamma)]$.

Si (α,γ) est une correspondance de Galois entre $\langle W_1, \preceq_1\rangle$ et $\langle W_2, \preceq_2\rangle$ alors nous avons $[\alpha\gamma = \underline{1}] \Leftrightarrow [\alpha$ est surjective$] \Leftrightarrow [\gamma$ est injective$]$ et aussi $[\gamma\alpha = \underline{1}] \Leftrightarrow [\alpha$ est injective$] \Leftrightarrow [\gamma$ est surjective$]$.

Soient $\langle W_1, \preceq_1, \wedge_1, \vee_1, \bot_1, \top_1, \neg_1\rangle$ et $\langle W_2, \preceq_2, \wedge_2, \vee_2, \bot_2, \top_2, \neg_2\rangle$ deux treillis booléens. (Si f est une fonction d'un treillis booléen dans un treillis booléen, nous définissons $\tilde{f}$ telle que $\tilde{f}(x) = \neg(f(\neg x))$). Alors $[(\alpha,\gamma)$ est une correspondance de Galois entre W_1 et W_2, si et seulement si $(\tilde{\gamma},\tilde{\alpha})$ est une correspondance de Galois entre W_2 et $W_1]$ et [f est surjective (respectivement injective) si et seulement si $\tilde{f}$ est surjective (respectivement injective)].

I.4 ORDINAUX

Les ordinaux constituent une extension dans l'infini de l'ensemble ω des entiers naturels muni de l'ordre naturel $<$:

$$0, 1, 2, \dots, \omega, \omega+1, \omega+2, \dots, \omega+\omega = \omega\times 2, \omega\times 2+1, \omega\times 2+2, \dots, \omega\times 2+\omega = \omega\times 3, \omega\times 3+1, \dots,$$
$$\omega\times 4, \dots, \omega\times\omega = \omega\uparrow 2, \dots, \omega\uparrow 3, \dots, \omega\uparrow\omega, \dots, \varepsilon_0 = \underbrace{\omega\uparrow\omega\uparrow\omega\dots}_{\omega \text{ fois}}, \dots, \omega_1^{CK}, \dots, \omega_1, \dots, \aleph_1, \dots, \aleph_\omega \dots$$

Nous utilisons la définition de Zermelo-Von Neumann des ordinaux. Une classe X est transitive si tout membre d'un membre de X est membre de X, $\mathrm{tran}(X) = [\forall y\in X, x\in y.\ x\in X] = [\cup X \subseteq X]$. X est un ordinal si et seulement si X est transitif et tout membre de X est transitif : $\mathrm{ord}(X) = [\mathrm{tran}(X) \wedge \forall x\in X.\ \mathrm{tran}(x)]$. Nous notons également $\mathrm{ord} = \{x : \mathrm{ord}(x)\}$ la classe des ordinaux et nous utilisons le plus souvent des lettres grecques $\alpha, \delta, \lambda, \dots$ pour désigner des ordinaux (mais plutôt des lettres latines $n, m, \dots$ pour des entiers). La relation $\alpha<\beta = (\mathrm{ord}(\alpha) \wedge \mathrm{ord}(\beta) \wedge \alpha\in\beta) = (\mathrm{ord}(\alpha) \wedge \mathrm{ord}(\beta) \wedge \alpha\subset\beta)$ est une relation de bon-ordre pour ord dont l'infimum est zéro, l'ensemble vide, noté 0. Nous définissons le successeur $S\alpha$ d'un ordinal α comme l'ordinal $S\alpha = \alpha\cup\{\alpha\} = \alpha+1$. Nous posons comme d'habitude $1 = S0 = \{0\}$, $2 = S1 = \{0,1\} = \{0,\{0\}\}$, $3 = S2 = \{0,1,2\}$ ... et plus généralement $\alpha = \{\beta\in\mathrm{ord} : \beta<\alpha\}$. α est un ordinal successeur si et seulement si $\mathrm{succ}(\alpha) = [\alpha\in\mathrm{ord} \wedge \exists\beta\in\mathrm{ord}.\ \alpha = S\beta] = [\alpha\in\mathrm{ord} \wedge \cup\alpha<\alpha]$. Nous notons $\alpha-1$ le prédécesseur de α. Nous avons $\forall\alpha,\beta\in\mathrm{ord}.\ \neg(\alpha<\beta<S\alpha)$. Si α n'est pas un ordinal successeur, c'est un ordinal limite et nous posons $\alpha-1=\alpha$. Les ordinaux limites sont caractérisés par $\mathrm{limit}(\alpha) = [\mathrm{ord}(\alpha) \wedge \forall\beta<\alpha.\ S\beta<\alpha] = [\mathrm{ord}(\alpha) \wedge \cup\alpha=\alpha] = [\mathrm{ord}(\alpha) \wedge \forall\beta<\alpha.\ \exists\gamma.\ \beta<\gamma<\alpha]$. Nous posons $\alpha\leqslant\beta = [\alpha<\beta \vee \alpha=\beta] = [\alpha<S\beta] = [\alpha\subseteq\beta]$. Pour tout $X\subseteq\mathrm{ord}$, $\sup X = \sup(\mathrm{ord},\leqslant)X = \cup X$ (respectivement $\sup^+ X = \sup^+(\mathrm{ord},\leqslant)X$) est le supremum (respectivement strict) de X pour $\leqslant$. Si X n'a pas de plus grand élément alors c'est un ordinal limite et $\sup^+ X = \sup X$. Si X a un plus grand élément α alors $\sup X = \alpha$ et $\sup^+ X = S\alpha$. Pour tout ensemble non vide $X\subseteq\mathrm{ord}$, $\cap X$ est le plus petit élément de X pour $\leqslant$. L'ensemble des entiers naturels est $\omega = \cap\{x : 0\in x \wedge \forall x\in X.\ Sx\in X\} = \{\alpha : \mathrm{nat}(\alpha)\}$ où $\mathrm{nat}(\alpha) =$

$[\underset{\sim}{ord}(\alpha) \wedge (\alpha \neq 0 \Rightarrow (\neg \underset{\sim}{limit}(\alpha) \wedge \forall \beta \in (\alpha \vee 0). \neg \underset{\sim}{limit}(\beta)))]$, $\omega = \cap\{\lambda \in \underset{\sim}{ord} : \lambda \neq 0 \wedge \underset{\sim}{limit}(\lambda)\}$. L'ensemble des entiers est $\mathbb{Z} = \omega \cup \{\dot{-}m : 0 < m \in \omega\}$ (où $\dot{-}m$ est la paire $\langle 0, m \rangle$).

Si $\underset{\sim}{wf}(W, \prec)$ alors les preuves par induction transfinie sur $\prec$ sont de la forme $[\forall \alpha \in W. ((\forall \beta \prec \alpha. P(\beta)) \Rightarrow P(\alpha))] \Rightarrow [\forall \alpha \in W. P(\alpha)]$. En particulier pour les ordinaux nous avons $[P(0) \wedge \forall \alpha \in \underset{\sim}{ord}. [P(\alpha) \Rightarrow P(\mathcal{S}\alpha)] \wedge \forall \alpha \in \underset{\sim}{ord}. [(\underset{\sim}{limit}(\alpha) \wedge \forall \beta < \alpha. P(\beta)) \Rightarrow P(\alpha)]] \Rightarrow [\forall \alpha \in \underset{\sim}{ord}. P(\alpha)]$ tandis que pour les entiers nous avons
$\forall \alpha \in \omega. [P(0) \wedge \forall \alpha \in \omega. [P(\alpha) \Rightarrow P(\alpha + 1)]] \Rightarrow [\forall \alpha \in \omega. P(\alpha)]$.

Si $\underset{\sim}{wf}(W, \prec)$ alors les définitions par récurrence transfinie sont de la forme $F(x_1, \ldots, x_m, w) = G(x_1, \ldots, x_m, w, \{\langle z, F(x_1, \ldots, x_m, z)\rangle : z \in W \wedge z \prec w\})$. En particulier pour les ordinaux ces définitions prennent pouvent la forme $F(x_1, \ldots, x_m, 0) = f(x_1, \ldots, x_m)$, $F(x_1, \ldots, x_m, \mathcal{S}\alpha) = G(x_1, \ldots, x_m, \alpha, F(x_1, \ldots, x_m, \alpha))$ et $\underset{\sim}{limit}(\alpha) \Rightarrow [F(x_1, \ldots, x_m, \alpha) = H(x_1, \ldots, x_m, \alpha, \{F(x_1, \ldots, x_m, \beta) : \beta < \alpha\})]$.

Nous utilisons les ordinaux comme modèles des ordres bien fondés. Si $\underset{\sim}{wf}(W, \prec)$ alors nous définissons le rang de $x \in W$ par $\underset{\sim}{rk}(W, \prec)(x) = \underset{\sim}{sup}^+\{\underset{\sim}{rk}(W, \prec)(y) : y \in W \wedge y \prec x\}$ et le rang de $\langle W, \prec \rangle$ par $\underset{\sim}{rk}[W, \prec] = \underset{\sim}{sup}^+\{\underset{\sim}{rk}(W, \prec)(x) : x \in W\}$. Nous avons $\underset{\sim}{rk}(W, \prec) \in (W \to \underset{\sim}{rk}[W, \prec])$, $\forall x, y \in W. [(x \prec y) \Rightarrow \underset{\sim}{rk}(W, \prec)(x) \prec \underset{\sim}{rk}(W, \prec)(y)]$ et $\underset{\sim}{rk}(W, \prec)$ est un isomorphisme de W sur $\underset{\sim}{rk}[W, \prec]$ quand $\underset{\sim}{wo}(W, \prec)$.

L'addition d'ordinaux est définie par $\alpha + 0 = \alpha$, $\alpha + \mathcal{S}\beta = \mathcal{S}(\alpha + \beta)$ et $\alpha + \gamma = \bigcup_{\delta < \gamma} (\alpha + \delta)$ quand γ st un ordinal limite non nul. Si $\underset{\sim}{wo}(W_0, \prec_0)$ et $\underset{\sim}{wo}(W_1, \prec_1)$ alors $\underset{\sim}{+}$ défini par $\underset{\sim}{+}(\langle 0, w \rangle) = \underset{\sim}{rk}(W_0, \prec_0)(w)$, $\underset{\sim}{+}(\langle 1, w \rangle) = \underset{\sim}{rk}[W_0, \prec_0] + \underset{\sim}{rk}(W_1, \prec_1)(w)$ et l'unique isomorphisme de $\langle W_0, \prec_0 \rangle \oplus \langle W_1, \prec_1 \rangle$ sur $\underset{\sim}{rk}[W_0, \prec_0] + \underset{\sim}{rk}[W_1, \prec_1] = \underset{\sim}{rk}[\langle W_0, \prec_0 \rangle \oplus \langle W_1, \prec_1 \rangle]$.

La multiplication d'ordinaux est définie par $\alpha \times 0 = 0$, $\alpha \times \mathcal{S}\beta = (\alpha \times \beta) + \alpha$ et $\alpha \times \gamma = \bigcup_{\delta < \gamma} (\alpha \times \delta)$ quand γ st un ordinal limite non nul. Si $\underset{\sim}{wo}(W_0, \prec_0)$ et $\underset{\sim}{wo}(W_1, \prec_1)$ alors $\underset{\sim}{x}$ défini par $\underset{\sim}{x}(\langle x, y \rangle) = (\underset{\sim}{rk}[W_0, \prec_0] \times \underset{\sim}{rk}(W_1, \prec_1)(y)) + \underset{\sim}{rk}[W_1, \prec_1]$ est l'unique isomorphisme de $\langle W_0, \prec_0 \rangle \otimes \langle W_1, \prec_1 \rangle$ sur $\underset{\sim}{rk}[W_0, \prec_0] \times \underset{\sim}{rk}[W_1, \prec_1] = \underset{\sim}{rk}[\langle W_0, \prec_0 \rangle \otimes \langle W_1, \prec_1 \rangle]$.

L'exponentiation d'ordinaux est définie par $\alpha \uparrow 0 = 1$, $\alpha \uparrow \beta' = (\alpha \uparrow \beta) \times \alpha$ et $\alpha \uparrow \gamma = \bigcup_{\delta < \gamma} \alpha \uparrow \delta$ quand γ est un ordinal limite non nul.

I.5 SEQUENCES

Une séquence sur A est une fonction f telle que $\mathrm{dom}(f)$ est un ordinal et $\mathrm{rng}(f) \subseteq A$. Nous appelons A l'alphabet de la séquence f. La longueur $\mathrm{dom}(f)$ de la séquence f est également notée $|f|$. Nous notons $A^{\ll\lambda} = \cup\{(\alpha \to A) : \alpha \in (\lambda \vee 0)\}$ (respectivement $A^{\ll\lambda} = \cup\{(\alpha \to A) : \alpha \in ((\lambda+1) \vee 0)\}$) la classe des séquences non vides de longueur inférieure (respectivement ou égale) à λ et $A^{\leq\lambda} = \cup\{(\alpha \to A) : \alpha \in (\lambda+1)\}$) la classe des séquences (vides et ou égale) à λ.

finies car $|f| < \omega$. La séquence vide
ngueur 0 est 0 également notée $\langle\rangle$.
ssemblistes, alors le k-tuple $\langle a,b,\ldots,t\rangle$
dénote la séquence finie $\{\langle 0,a\rangle, \langle 1,b\rangle, \ldots, \langle k-1,t\rangle\}$ de longueur k. Les notions de paire ordonnée $\langle x,y\rangle = \{\{x\},\{x,y\}\}$ et de séquence de longueur 2 $\langle \alpha,y\rangle = \{\langle 0,x\rangle, \langle 1,y\rangle\}$ sont confondues parceque nous n'utilisons que leur propriété commune $(\langle x,y\rangle = \langle u,v\rangle) \Rightarrow (x=u \wedge y=v)$. Les séquences infinies de longueur ω sont notées $\langle f_i : i \in \omega\rangle$ ou $\langle f_0, \ldots, f_i, \ldots\rangle$ tandis que les séquences transfinies de longueur $\lambda > \omega$ sont notées $\langle f_i : i \in \lambda\rangle$ ou $\langle f_0, \ldots, f_i, \ldots\rangle_{i\in\lambda}$.

Le préfixe $f^{<\pi}$ (respectivement $f^{\leq\pi}$) d'une séquence $f \in (\alpha \to A)$ est la séquence f quand $\pi \geq \alpha$ (respectivement $\pi+1 \geq \alpha$) sinon c'est la séquence $f' \in (\pi \to A)$ telle que $\forall i \in \pi.\ f'_i = f_i$ (respectivement $f' \in (\pi+1 \to A)$ telle que $\forall i \in (\pi+1).\ f'_i = f_i$). Le suffixe $f^{>\pi}$ (respectivement $f^{\geq\pi}$) d'une séquence $f \in (\alpha \to A)$ est la séquence vide si $\pi+1 \geq \alpha$ (respectivement $\pi \geq \alpha$) sinon $\pi+1 < \alpha$ (respectivement $\pi < \alpha$) et c'est la séquence $f' \in (\beta \to A)$ telleque β est l'unique ordinal positif tel que $(\pi+1)+\beta = \alpha$ que nous notons $\beta = \alpha - (\pi+1)$ et $\forall i \in \beta.\ f'_i = f_{(\pi+1)+i}$ (respectivement $\pi+\beta = \alpha$ que nous notons $\beta = \alpha - \pi$ et $\forall i \in \beta.\ f'_i = f_{\pi+i}$). La tranche $f^{<\alpha,\beta>}$ (respectivement $f^{\leq\alpha,\beta\geq}$, $f^{\leq\alpha,\beta>}$, $f^{\ll\alpha,\beta\geq}$) est $(f^{<\beta})^{>\alpha}$ (respectivement $(f^{\leq\beta})^{\geq\alpha}$, $(f^{<\beta})^{\geq\alpha}$, $(f^{\leq\beta})^{>\alpha}$).

Nous définissons l'opération de concaténation $\hat{}$ sur $A^{\leq\omega}$ par $f\hat{}g = f$ si $|f| = \omega$ sinon $f\hat{}g = f \cup \{\langle |f|+i,\ g(i)\rangle : i \in |g|\}$.

I.6 CARDINAUX

X est équipotent avec Y noté $X \approx Y$ si et seulement si il existe une bijection entre X et Y. $\underline{m}$ est un cardinal si et seulement si $[\underline{m} \in ord \wedge \forall \beta \in \underline{m}. \neg(\beta \approx \underline{m})]$. Nous notons $card(X)$ le cardinal de l'ensemble X c'est-à-dire l'unique cardinal $\underline{m}$ équipotent à X. Pour tout ordinal α, α^+ est le plus petit cardinal strictement supérieur à α. Un ensemble est fini si $card(X) < \omega$, dénombrable si $card(X) \leq \omega$, infini si $card(X) \geq \omega$.

Un cardinal $\underline{m}$ est dit régulier si $\forall \Gamma \subseteq \underline{m}$, si $card(\Gamma) < \underline{m}$ alors $\nu\Gamma < \underline{m}$, sinon il est dit singulier.

$\underline{m}^+ = \omega$ si $\underline{m} < \omega$, $\underline{m}^+ = \underline{m}$ si $\underline{m}$ est un cardinal infini régulier et $\underline{m}^+ = \underline{m}^+$ si $\underline{m}$ est un cardinal infini singulier. (Pour tout cardinal $\underline{m}$, m^+ est régulier (puisque ω est régulier et supposant l'axiome du choix, pour tout cardinal infini $\underline{m}$, m^+ est régulier)).

ANNEXE II :

INDEX DES NOTATIONS MATHEMATIQUES

Les notations ci-dessous sont introduites et définies dans l'annexe I.

LOGIQUE

$=$	égalité
$\neq$	différent
$\neg$	non logique
$\Rightarrow$	implication logique
$\Leftarrow$	implication logique inverse
$\not\Rightarrow$	négation de l'implication logique
$\not\Leftarrow$	négation de l'implication logique inverse
$\Leftrightarrow$	équivalence logique
$\vee$	ou (inclusif) logique
$\wedge$	et logique
ff	valeur de vérité fausse
tt	valeur de vérité vraie
$\underline{\underline{\mathcal{U}}}$	univers ($\underline{\underline{\mathcal{U}}} = \{x : \mathit{tt}\}$)
$X, X', X_i, \ldots, x, x', \ldots, x_i, \ldots$	variables logiques
$P, Q, \ldots$	Prédicats
$P(x_0, \ldots, x_i, \ldots)$	un prédicat avec $x_0, \ldots, x_i \ldots$ comme (seules) variables libres possibles
$(P \Rightarrow Q \mid R)$	si P alors Q sinon R, $((P \wedge Q) \vee (\neg P \wedge R))$
$(P \rightarrow a \mid b)$	désigne la valeur a si P est vrai, sinon b
$\forall$	pour tout
$\forall x_0, \ldots, x_i, \ldots \cdot P$	abbréviation de $\forall x_0 \cdot (\ldots (\forall x_i \cdot (\ldots P \ldots)) \ldots)$
$\forall x_0 \in X_0, \ldots, \forall x_i \in X_i, \ldots \cdot P$	abbréviation de $\forall x_0, \ldots, x_i, \ldots \cdot ((x_0 \in X_0 \wedge \ldots \wedge x_i \in X_i \wedge \ldots) \Rightarrow P)$
$\exists$	il existe

$\exists!$	il existe un unique
$\exists x_0,\dots,x_i,\dots . P$	abbréviation de $\exists x_0.(\dots(\exists x_i.(\dots . P\dots))\dots)$
$\exists x_0 \in X_0,\dots,x_i \in X_i,\dots . P$	abbréviation de $\exists x_0,\dots,x_i,\dots .(x_0 \in X_0 \wedge \dots \wedge x_i \in X_i \wedge \dots \wedge P)$

ENSEMBLES

$\in$	est membre de
$\notin$	n'est pas membre de
$\cup$	$X \cup Y$ union binaire
	$\cup X$ union infinie de tous les membres de X
	$\bigcup_{i \in I} A_i = \cup \underset{\sim}{\mathrm{rng}}(A)$ où $A \in (I \to \underset{\sim}{\mathrm{rng}}(A))$
$\cap$	$X \cap Y$ intersection binaire
	$\cap X$ intersection infinie de tous les membres de X
	$\bigcap_{i \in I} A_i = \cap \underset{\sim}{\mathrm{rng}}(A)$ où $A \in (I \to \underset{\sim}{\mathrm{rng}}(A))$
$\subseteq$	inclusion large
$\subset$	inclusion stricte
$\supseteq$	$X \supseteq Y \iff Y \subseteq X$
$\supset$	$X \supset Y \iff Y \subset X$
$\nsubseteq$	$X \nsubseteq Y \iff \neg(X \subseteq Y)$
$\not\subset$	$X \not\subset Y \iff \neg(X \subset Y)$
$\nsupseteq$	$X \nsupseteq Y \iff \neg(X \supseteq Y)$
$\not\supset$	$X \not\supset Y \iff \neg(X \supset Y)$
$0, \varnothing$	zéro ou ensemble vide
$\setminus$	différence de classes
$\sim$	différence avec un singleton, $X \sim x = X \setminus \{x\}$
$\times$	produit cartésien
.	tel que (dans $\exists x.P(x)$), on a (dans $\forall x.P(x)$)
:	tel que (dans $\{x : P(x)\}$)
2^X	puissance de X
$X/_r$	ensemble quotient de X modulo r
$[x]_r$	classe d'équivalence de x pour r
$\langle x, y \rangle$	paire ordonnée

$\forall x.\, P(x)$	pour tout x, on a $P(x)$
$\exists x.\, P(x)$	il existe x tel que $P(x)$
$\{x : P(x)\}$	la classe de tous les x tels que $P(x)$
$\{\tau(x_0,\dots,x_i,\dots) : P(x_0,\dots,x_i,\dots)\}$	la classe de tous les $\tau(x_0,\dots,x_i,\dots)$ tels que $P(x_0,\dots,x_i,\dots)$
$\{\tau \in X : P\}$	abbréviation de $\{\tau : \tau \in X \wedge P\}$
$\underline{i+}$	l'unique isomorphisme de l'addition $\langle W_0,\prec_0\rangle \oplus \langle W_1,\prec_1\rangle$ de classes bien fondées sur $\mathrm{rk}[W_0,\prec_0] + \mathrm{rk}[W_1,\prec_1]$
$\underline{ix}$	l'unique isomorphisme de la multiplication $\langle W_0,\prec_0\rangle \otimes \langle W_1,\prec_1\rangle$ de classes bien fondées sur $\mathrm{rk}[W_0,\prec_0] \times \mathrm{rk}[W_1,\prec_1]$
$\underline{\mathrm{set}}(X)$	X est un ensemble
$\underline{\mathrm{tran}}(X)$	X est une classe transitive, $[\forall y \in X, x \in y.\, x \in X]$

RELATIONS

$\neg$	négation de relation $(\neg r)(x,y) = \neg r(x,y)$
$\Rightarrow$	implication de relations $(r \Rightarrow r')(x,y) = (r(x,y) \Rightarrow r'(x,y))$
$\vee$	union de relations $(r \vee r')(x,y) = r(x,y) \vee r'(x,y)$
$\wedge$	intersection de relations $(r \wedge r')(x,y) = r(x,y) \wedge r'(x,y)$
$\underline{1}$	relation identité
$x\, r\, y$	x est en relation avec y selon la relation binaire r
$r(x,y)$	"
$\langle x,y \rangle \in r$	"
$r \mid X$	restriction gauche de la relation r, $r \cap (X \times \underline{rng}(r))$
$r \restriction X$	restriction droite de la relation r, $r \cap (\underline{dom}(r) \times X)$
$r \mid X$	restriction de la relation r, $r \cap (X \times X)$
$r \underline{\circ} s$	composition de relations, $r \underline{\circ} s\,(x,y) = \exists y.[r(x,y) \wedge s(y,y)]$
r^0	relation identité, $\underline{1}$
r^m	$r \underline{\circ} r \underline{\circ} \dots \underline{\circ} r$, m fois
r^+	fermeture transitive d'une relation, $\exists m \in (\omega \setminus 0).\, r^m$
r^*	fermeture transitive reflexive, $\exists m \in \omega.\, r^m$
r^{-1}	inverse de la relation r, $x\, r^{-1} y = y\, r\, x$
$r[X]$	image de X par r, $\{y : \exists x \in X.\, r(x,y)\}$
$\underline{dom}(r)$	domaine de la relation r, $\{x : \exists y.\, x\, r\, y\}$
$\underline{fld}(r)$	champ de la relation r, $\underline{dom}(r) \cup \underline{rng}(r)$
$\underline{rng}(r)$	co-domaine de la relation r, $\{y.\, \exists x.\, x\, r\, y\}$
$\underline{rel}(X,r)$	r est une relation sur X, $r \in (X \times X \to \{tt, ff\})$

FONCTIONS

$\circ$	composition fonctionnelle, $f \circ g(x) = f(g(x))$
$(X \to Y)$	classe des fonctions partielles de X dans Y
$(X \to Y)$	classe des fonctions totales de X dans Y
$\langle \tau(x) : x \in I \rangle$	la fonction f sur I telle que $\forall x.\ f(x) = \tau(x)$
$f \restriction X$	restriction de f à X, $f \cap (X \times \mathrm{rng}(f))$
$f(x)$	notation fonctionnelle de y tel que $\langle x, y \rangle \in f$, s'il existe
f_x	"
f_x	"
$f(x_0, \ldots, x_{m-1})$	$f(\langle x_0, \ldots, x_{m-1} \rangle)$
$f[x \mid y]$	la fonction f' telle que $f'(x) = y$ et $f'(z) = f(z)$ si $z \neq x$

ORDRES

$\prec$	une relation d'ordre strict
$\preccurlyeq$	la relation d'ordre reflexif correspondant à $\prec$
$\succ$	la relation inverse de $\prec$
$\succcurlyeq$	la relation inverse de $\preccurlyeq$
$\oplus$	addition de classes partiellement ordonnées
$\otimes$	multiplication de classes partiellement ordonnées
$\bot$	infimum d'un treillis complet
$\top$	supremum d'un treillis complet
$\langle W, \preccurlyeq \rangle$	classe partiellement ordonnée
$\langle W, \preccurlyeq, \wedge, \vee, \bot, \top \rangle$	treillis complet
$\langle W, \preccurlyeq, \wedge, \vee, \bot, \top, \neg \rangle$	treillis booléen complet
(α, γ)	demi-correspondance, quasi-correspondance ou correspondance de Galois
$\underset{\sim}{cpo}$	ordre partiel complet
$\underset{\sim}{lo}(W, r)$	la relation r est un ordre linéaire ou total sur W
$\underset{\sim}{po}(W, r)$	la relation r est un ordre partiel (strict ou reflexif) sur W
$\underset{\sim}{rk}[W, \prec]$	rang de $\langle W, \prec \rangle$ pour la relation bien fondée $\prec$ sur W, $\underset{\sim}{sup}^+\{\underset{\sim}{rk}(W,\prec)(x) : x \in W\}$
$\underset{\sim}{rk}(W, \prec)(x)$	rang de x pour la relation bien fondée $\prec$ sur W, $\underset{\sim}{sup}^+\{\underset{\sim}{rk}(W,\prec)(y) : y \in W \wedge y \prec x\}$
$\underset{\sim}{rpo}(W, \preccurlyeq)$	la relation $\preccurlyeq$ est un ordre partiel reflexif sur W
$\underset{\sim}{apo}(W, \prec)$	la relation $\prec$ est un ordre partiel strict sur W
$\underset{\sim}{sup}(W, \preccurlyeq) X$	borne supérieure de X pour $\preccurlyeq$ sur W
$\underset{\sim}{sup}^+(W, \preccurlyeq) X$	borne supérieure stricte de X pour $\preccurlyeq$ sur W
$\underset{\sim}{wf}(W, \prec)$	la relation $\prec$ est bien fondée sur W, $[rel(W,\prec) \wedge \forall X \subseteq W.\, [X \neq 0 \Rightarrow \exists y \in X.\, (\forall z \in X.\, \neg z \prec y)]]$
$\underset{\sim}{wfi}(W, \prec, \mu)$	la relation $\prec$ est bien fondée sur W avec un element minimal μ, $[wf(W,\prec) \wedge \mu \in W \wedge \forall x \in W.\, \neg(x \prec \mu)]$
$\underset{\sim}{wo}(W, \prec)$	la relation $\prec$ est un bon-ordre sur W, $[\underset{\sim}{lo}(W,\prec) \wedge \underset{\sim}{wf}(W,\prec)]$

ORDINAUX

$<$	inférieur strict sur les ordinaux, $\alpha<\beta = (\underset{\sim}{ord}(\alpha) \wedge \underset{\sim}{ord}(\beta) \wedge \alpha\in\beta)$
$\leqslant$	inférieur ou égal sur les ordinaux, $\alpha\leqslant\beta = (\alpha<\beta)\vee(\alpha=\beta)$
$>$	inverse de $<$
$\geqslant$	inverse de $\leqslant$
$\not<$	négation de $<$
$\not\leqslant$	négation de $\leqslant$
$\not>$	négation de $>$
$\not\geqslant$	négation de $\geqslant$
1	$= \underset{\sim}{\mathcal{S}}0 = \{0\}$
2	$= \underset{\sim}{\mathcal{S}}1 = \{0,1\} = \{0,\{0\}\}$
3	$= \underset{\sim}{\mathcal{S}}2 = \{0,1,2\} = \{0,\{0\},\{0,\{0\}\}\}$
...	...
$\alpha-1$	ordinal predecesseur de α, c'est α si $\underset{\sim}{limit}(\alpha)$ sinon β tel que $\underset{\sim}{\mathcal{S}}\beta=\alpha$
$\underset{\sim}{\mathcal{S}}\alpha$	ordinal successeur de α, $\underset{\sim}{\mathcal{S}}\alpha = \alpha\cup\{\alpha\} = \alpha+1$
$\cup X$	supremum de $X\subseteq \underset{\sim}{ord}$ pour $\leqslant$
$\cap X$	plus petit élément de $X\subseteq\underset{\sim}{ord}$ pour $\leqslant$ quand $X\neq 0$
ω	ensemble des entiers naturels
$+, \dot{+}$	addition d'ordinaux, $\alpha+\beta = \alpha\cup \underset{\sim}{sup}\{(\alpha+\gamma)+1 : \gamma<\beta\}$
$\times, \dot{\times}$	multiplication d'ordinaux, $\alpha\times\beta = \underset{\sim}{sup}\{(\alpha\times\gamma)+\alpha : \gamma<\beta\}$
$\uparrow$	exponentiation d'ordinaux
$\underset{\sim}{limit}(\alpha)$	caractérise un ordinal limite
$\underset{\sim}{limit}$	$= \{\alpha : \underset{\sim}{limit}(\alpha)\}$
$\underset{\sim}{nat}(x)$	caractérise les entiers naturels
$\underset{\sim}{ord}(\alpha)$	caractérise les ordinaux, $[\underset{\sim}{tran}(\alpha) \wedge \forall\beta\in\alpha.\ \underset{\sim}{tran}(\beta)]$
$\underset{\sim}{ord}$	classe des ordinaux, $\{\alpha : \underset{\sim}{ord}(\alpha)\}$
$\underset{\sim}{succ}(\alpha)$	caractérise un ordinal successeur, $[\underset{\sim}{ord}(\alpha) \wedge \exists\beta\in\underset{\sim}{ord}.\ \alpha=\underset{\sim}{\mathcal{S}}\beta]$
$\underset{\sim}{succ}$	$=\{\alpha : \underset{\sim}{succ}(\alpha)\}$
$\underset{\sim}{sup}X$	supremum d'une classe d'ordinaux, $\underset{\sim}{sup}(\underset{\sim}{ord},\leqslant)X = \cup X$
$\underset{\sim}{sup}^{+}X$	supremum strict d'une classe d'ordinaux, $\underset{\sim}{sup}^{+}(\underset{\sim}{ord},\leqslant)X$

SEQUENCES

$\langle\rangle$	séquence vide 0
$\langle f_0,\dots,f_{m-1}\rangle$	séquence finie de longueur m
$\langle f_0,\dots,f_i,\dots\rangle$	sequence infinie de longueur ω
$\langle f_i : i\in\lambda\rangle$	sequence transfinie de longueur λ
$\langle f_0,\dots,f_i,\dots\rangle_{i\in\lambda}$	"
$\lvert f\rvert$	longueur de la séquence f, $\lvert f\rvert = \underline{\mathrm{dom}}(f)$
$\hat{}$	concaténation de séquences
$A^{<\lambda}$	classe des séquences non vides sur A de longueur inférieure à λ, $\cup\{(\alpha\to A):\alpha\in(\lambda\setminus 0)\}$
$A^{\leqslant\lambda}$	classe des séquences non vides sur A de longueur inférieure ou egale à λ, $\cup\{(\alpha\to A):\alpha\in(\lambda+1\setminus 0)\}$
$A^{<\lambda}$	classe des séquences sur A de longueur inférieure à λ, $\cup\{(\alpha\to A):\alpha\in\lambda\}$
$A^{\leqslant\lambda}$	classe des séquences sur A de longueur inférieure ou égale à λ, $\cup\{(\alpha\to A):\alpha\in\lambda+1\}$
$f^{<\pi}$	préfixe $\langle f_0,\dots,f_{\pi-1}\rangle$ d'une séquence f (si $\pi<\lvert f\rvert$ sinon f)
$f^{\leqslant\pi}$	préfixe $\langle f_0,\dots,f_\pi\rangle$ d'une séquence f (si $\pi+1<\lvert f\rvert$ sinon f)
$f^{>\pi}$	suffixe $\langle f_{\pi+1},\dots\rangle$ d'une sequence f (si $\pi+1<\lvert f\rvert$ sinon $\langle\rangle$)
$f^{\geqslant\pi}$	suffixe $\langle f_\pi,\dots\rangle$ d'une séquence f (si $\pi<\lvert f\rvert$ sinon $\langle\rangle$)
$f^{<\alpha,\beta>}$	tranche d'une séquence f, $(f^{<\beta})^{>\alpha}$
$f^{\leqslant\alpha,\beta\geqslant}$	tranche d'une séquence f, $(f^{\leqslant\beta})^{\geqslant\alpha}$
$f^{\leqslant\alpha,\beta>}$	tranche d'une séquence f, $(f^{<\beta})^{\geqslant\alpha}$
$f^{<\alpha,\beta\geqslant}$	tranche d'une séquence, f, $(f^{\leqslant\beta})^{>\alpha}$

CARDINAUX

$\approx$	équipotence
α^+	le plus petit cardinal strictement supérieur à α
$\mathfrak{m}^+$	$\underline{\mathfrak{m}}^+ = \omega$ si $\underline{\mathfrak{m}} < \omega$, $\underline{\mathfrak{m}}^+ = \underline{\mathfrak{m}}$ si $\underline{\mathfrak{m}}$ est un cardinal infini régulier, $\underline{\mathfrak{m}}^+ = \underline{\mathfrak{m}}^+$ si $\underline{\mathfrak{m}}$ est un cardinal infini singulier
$\underset{\sim}{\mathrm{card}}(X)$	cardinal de X, unique cardinal équipotent à X

ANNEXE III :

INDEX DES NOTATIONS INFORMATIQUES

Les notations informatiques sont introduites par chapitre puis classées dans chaque chapitre comme suit : symbole, ordre alphabétique des lettres grecques, ordre alphabétique des lettres latines. Le numéro de paragraphe qui suit chaque notation, représente le paragraphe où elle a été introduite.

REFERENCES ET NOTATIONS TYPOGRAPHIQUES

$n_0 \cdot \ldots \cdot n_{k-1} \cdot n_k$	paragraphe n_k du paragraphe $n_0 \cdot \ldots \cdot n_{k-1}$
$n_0 \cdot \ldots \cdot n_k : m$	définition m du paragraphe $n_0 \cdot \ldots \cdot n_k$
$n_0 \cdot \ldots \cdot n_k \sim m$	théorème, lemme ou corollaire m du paragraphe $n_0 \cdot \ldots \cdot n_k$
$n_0 \cdot \ldots \cdot n_k - m$	exemple m du paragraphe $n_0 \cdot \ldots \cdot n_k$
□	fin d'une démonstration de théorème ou de lemme, d'un exemple.
x [yy]	référence bibliographique à l'auteur (ou aux auteurs) x et l'année 19yy

AUTRES NOTATIONS

?	affectation aléatoire ($V_i := ?$), 2.8.1.1
	reception d'un message sur rendez-vous (Ch ? V), 2.8.3.1
!	envoi d'un message sur rendez-vous (Ch ! E), 2.8.3.1
|	alternative syntaxique, 2.8
≡	identité syntaxique, 2.8.1.1
	relation d'équivalence entre systèmes de transition, 2.5.2

$:=$	affectation à une variable de programme, 2.8.1.1
$:$	suit une étiquette de programme, 2.8.1.1
$;$	composition séquentielle de commandes, 2.8.1.1
$\wedge$	relation de concaténation (de traces), 2.1.1
$\xrightarrow{a}$	relation de concaténation (de traces) via l'action a, 2.1.1
$\longrightarrow$	dérivation syntaxique, 2.8
$\rightarrowtail$	relation de préfixe entre traces, 2.6.1
$\twoheadrightarrow$	relation de suffixe entre traces, 2.6.2
$[\![\ldots \| \ldots \| \ldots]\!]$	composition parallèle de processus séquentiels, 2.8.2
ε	caractérise les états initiaux, $(\varepsilon \in (S \to \{tt, ff\}))$, 2.2
$\varepsilon[\![Pr]\!]$	caractérise les états initiaux du programme Pr, 2.8.1.2.3
$\varepsilon\langle S, A, \Sigma\rangle$	caractérise les états initiaux engendrés par la sémantique $\langle S, A, \Sigma\rangle$ $(\varepsilon(s) = [\exists p \in \Sigma.\ p_0 = s])$, 2.3
$\varepsilon\ell[\![Pps]\!]$	caractérise les états initiaux associés au programme asynchrone Pps par la sémantique libérale, 2.8.5.3
$\sigma\alpha(p_i, p_j)$	nombre de fois qu'une action appartenant à α est exécutée entre p_i et p_j, $(\sigma \in (2^A \to (\Sigma \times \omega \to \omega)))$, 2.8.5.2.6
Σ	ensemble de traces, 2.1.1
$\Sigma[\![Pr]\!]$	ensemble de traces associé au programme Pr, 2.1.2
$\Sigma\langle S, A\rangle$	ensemble des traces sur un ensemble S d'états et un ensemble A d'actions, (abbréviation de $\Sigma^{\leq\omega}\langle S, A\rangle$), 2.1.1
$\Sigma\langle \mathcal{S}, \mathcal{A}\rangle$	ensemble des traces sur les ensembles $\mathcal{S}$ d'états et $\mathcal{A}$ d'actions, 2.5.1
$\Sigma^n\langle S, A\rangle$	ensemble des traces de longueur n sur S et A, $\{\langle n, s, a\rangle : n \in (\omega+1) \wedge s \in (n \to S) \wedge a \in (n-1 \to A)\}$, 2.1.1
$\Sigma^{<\ell}\langle S, A\rangle$	ensemble des traces sur S et A de longueur strictement inférieure à ℓ, $(\bigcup_{n \in (\ell+0)} \Sigma^n\langle S, A\rangle)$, 2.1.1
$\Sigma^{\leq\ell}\langle S, A\rangle$	ensemble des traces sur S et A de longueur inférieure ou égale à ℓ, $(\Sigma^{<\ell}\langle S, A\rangle \cup \Sigma^{\ell}\langle S, A\rangle)$, 2.1.1

$\Sigma^{\omega}\langle S,A\rangle$ ensemble des traces infinies sur S et A, 2.1.1

$\Sigma^{<\omega}\langle S,A\rangle$ ensemble des traces finies sur S et A, 2.1.1

$\Sigma^{\leq\omega}\langle S,A\rangle$ ensemble des traces sur S et A, 2.1.1

$\Sigma\langle S,A,t,\varepsilon\rangle$ ensemble des traces complètes engendrées par le système de transition $\langle S,A,t,\varepsilon\rangle$, (abbréviation de $\bigcup_{n\in((\omega+1)\setminus 0)}\Sigma^{n}\langle S,A,t,\varepsilon\rangle$), 2.4

$\Sigma^{n}\langle S,A,t,\varepsilon\rangle$ ensemble des traces complètes finies de longueur $n\in(\omega\setminus 0)$ engendrées par le système de transition $\langle S,A,t,\varepsilon\rangle$, $(\{p\in\Sigma^{n}\langle S,A\rangle : \varepsilon(p_0)\wedge\forall i\in|\natural|\cdot t_{\natural i}(p_i,p_{i+1})\wedge\forall a\in A,\forall s\in S.\neg t_a(p_{n-1},s)\})$, 2.4

$\Sigma^{\omega}\langle S,A,t,\varepsilon\rangle$ ensemble des traces infinies engendrées par le système de transition $\langle S,A,t,\varepsilon\rangle$, $(\{p\in\Sigma^{\omega}\langle S,A\rangle : \varepsilon(p_0)\wedge\forall i\in\omega.\, t_{\natural i}(p_i,p_{i+1})\})$, 2.4

$\Sigma^{<\ell}\langle S,A,t,\varepsilon\rangle$ ensemble des traces de longueur strictement inférieure à ℓ engendrées par le système de transition $\langle S,A,t,\varepsilon\rangle$, $(\bigcup_{n\in(\ell\setminus 0)}\Sigma^{n}\langle S,A,t,\varepsilon\rangle)$, 2.4

$\Sigma^{<\omega}\langle S,A,t,\varepsilon\rangle$ ensemble des traces finies engendrées par le système de transition $\langle S,A,t,\varepsilon\rangle$, 2.4

$\Sigma^{\leq\ell}\langle S,A,t,\varepsilon\rangle$ ensemble des traces de longueur inférieure ou égale à ℓ engendrées par le système de transition $\langle S,A,t,\varepsilon\rangle$, $(\Sigma^{<\ell}\langle S,A,t,\varepsilon\rangle\cup\Sigma^{\ell}\langle S,A,t,\varepsilon\rangle)$, 2.4

$\Sigma^{\leq\omega}\langle S,A,t,\varepsilon\rangle$ ensemble des traces complètes engendrées par le système de transition $\langle S,A,t,\varepsilon\rangle$, (noté aussi $\Sigma\langle S,A,t,\varepsilon\rangle$), 2.4

$\Sigma\ell[\![Pps]\!]$ ensemble de traces associé au programme synchrone Pps par la sémantique libérale, 2.8.5.2

a action (commande, processus, ...), 2.1.1

$\underline{a}$ action unique, 2.5.3.3

A ensemble d'actions, 2.1.1

$\mathcal{A}$ ensemble des actions, 2.1.2

Alt commande alternative, (B; Ch!e then Prc | B; Ch?v then Prc), 2.8.3.1

$A[\![Pr]\!]$	ensemble d'actions associé au programme Pr, 2.1.2
$<A,\Sigma>$	sémantique concordante à $<S,A,\Sigma>$ à l'annulation des états près, 2.5.3.2
$Al[\![Pps]\!]$	ensemble d'actions associé au programme synchrone par la sémantique libérale, 2.8.5.3
$Ar[\![Pps]\!]$	ensemble réduit d'actions associé au programme synchrone Pps, 2.8.5.2.5
$\underset{\sim}{Acc}<S,A,t,\varepsilon>$	caractérise les états accessibles du système de transition $<S,A,t,\varepsilon>$, 2.5.2
B	expression booléenne d'un programme, 2.8.1.2.4
$\mathbb{B}$	ensemble des expressions booléennes, 2.8.1.1
$\mathbb{B}$	sémantique des expressions booléennes, $\mathbb{B}\in(\mathcal{E}\to((\mathcal{V}\to\mathcal{D})\to\{tt,ff\}))$, 2.8.1.2.4
$\mathbb{B}[\![B]\!](M)$	valeur de l'expression booléenne B dans l'état mémoire M, 2.8.1.2.4
$\underset{\sim}{Blo}<S,A,t,\varepsilon>$	caractérise les états de blocage du système de transition $<S,A,t,\varepsilon>$, 2.5.1
$C[\![Pps]\!]$	ensemble des états de contrôle associé au programme synchrone Pps, 2.8.5.2.1
Ch	un canal de communication, 2.8.3.2.2
$\underline{Ch}$	action correspondant à une communication sur le canal ch, 2.8.3.2.2
$\mathcal{C}$	ensemble des commandes séquentielles, $(\mathcal{C}\to \underline{skip}\mid \mathcal{V}:=\mathcal{E}\mid \mathcal{V}:=?\mid \underline{if}\ \mathbb{B}\ \underline{then}\ Ps_1\ \underline{else}\ Ps_2\ \underline{fi}\mid \underline{while}\ \mathbb{B}\ \underline{do}\ Ps\ \underline{od})$, 2.8.1.1
$\mathcal{C}a$	ensemble des commandes des processus parallèles asynchrones, $(\mathcal{C}a\to \underline{skip}\mid \mathcal{V}:=\mathcal{E}\mid \mathcal{V}:=?\mid \underline{if}\ \mathbb{B}\ \underline{then}\ Pra_1\ \underline{else}\ Pra_2\ \underline{fi}\mid \underline{while}\ \mathbb{B}\ \underline{do}\ Pra\ \underline{od}\mid \$Ps\$)$, 2.8.2.1
$\mathcal{C}c$	ensemble des commandes des processus parallèles communicants, $(\mathcal{C}c\to \underline{skip}\mid \mathcal{V}:=\mathcal{E}\mid \mathcal{V}:=?\mid \underline{if}\ \mathbb{B}\ \underline{then}\ Prc_1\ \underline{else}\ Prc_2\ \underline{fi}\mid \underline{while}\ \mathbb{B}\ \underline{do}\ Prc\ \underline{od}\mid \$Ps\$\mid Ch!\mathcal{E}\mid Ch?\mathcal{V}\mid \underline{se}\ Alt_0\ \underline{or}\ \dots\ \underline{or}\ Alt_{\ell-1}\ \underline{es})$, 2.8.3.1
$\mathcal{C}h$	ensemble des canaux de communication, 2.8.3.1
$\mathcal{C}s$	ensemble des commandes des processus parallèles synchrones, $(\mathcal{C}s\to \underline{skip}\mid \mathcal{V}:=\mathcal{E}\mid \mathcal{V}:=?\mid \underline{if}\ \mathbb{B}\ \underline{then}\ Prs_1\ \underline{else}\ Prs_2\ \underline{fi}\mid \underline{while}\ \mathbb{B}\ \underline{do}\ Prs\ \underline{od}\mid p(\mathcal{L}e)\mid v(\mathcal{L}e))$, 2.8.5.1

cond	$cond[\![Ps]\!](L,L')(M)$ est la condition sur l'état mémoire M pour que le contrôle passe de L à L' dans l'exécution d'un pas de Ps, 2.8.1.2.4
$\mathfrak{D}$	domaine des valeurs des variables des programmes, 2.8.1.2.1
E	expression d'un programme, 2.8.1.2.4
$\mathbb{E}$	sémantique des expressions, $(\mathbb{E} \in (\mathfrak{E} \to ((\mathcal{V} \to \mathfrak{D}) \to \mathfrak{D})))$, 2.8.1.2.4
$\mathbb{E}[\![E]\!](M)$	valeur de l'expression E dans l'état mémoire M, 2.8.1.2.4
$\mathfrak{E}$	ensemble des expressions, 2.8.1.1
Enabled (a,i,p,Σ)	l'action a est activable au point $i \in \|p\|$ d'une trace p de Σ, $([i \in \|p\| \wedge \exists q \in \Sigma.(i \in \|q\| \wedge q^{<i} = p^{<i} \wedge q_i = a)])$, 2.6.4
Efus	extension par fusions, 2.6.5
Efus $(<S,A,\Sigma>)$	extension par fusions de la sémantique $<S,A,\Sigma>$, $(<S,A,\{p^\frown q : p \in \Sigma^{<\omega}<S,A> \wedge q \in \Sigma^{\leq\omega}<S,A> \wedge \exists p',q' \in \Sigma.(p \twoheadrightarrow p' \wedge q \twoheadrightarrow q')\}>)$, 2.6.5
$\simeq <f_S>$	concordance à une fonction f_S des états près, 2.5.3.1
$\simeq <f_S>(<S,A,\Sigma>)$	sémantique concordante à $<S,A,\Sigma>$ à la fonction f_S des états près, $(<f_S[S], A, \{<m, f_S(a), a> : <m,a,a> \in \Sigma\}>)$, 2.5.3.1
Ffus	fermeture par fusions, 2.6.5
Ffus $(<S,A,\Sigma>)$	fermeture par fusions de la sémantique $<S,A,\Sigma>$, $(\bigcup_{m>0} \text{Efus}^m)$, 2.6.5
Flim	fermeture par limites, 2.6.7
Flim $(<S,A,\Sigma>)$	fermeture par limites de la sémantique $<S,A,\Sigma>$, $(<S,A,\Sigma \cup \{p \in \Sigma^\omega <S,A> : \forall m \in \omega.\exists q \in \Sigma.\ p^{<m} \twoheadrightarrow q\}>)$, 2.6.7
$f[x \leftarrow y]$	substitution syntaxique, (f où y est substitué à x), 4.3.2.1.4.2
if ... then ... else ... fi	composition alternative de commandes, 2.8.1
if ... then ... fi	
Isem (se)	valeur initiale du sémaphore se, $(\text{Isem} \in (\mathcal{V}_e \to \mathfrak{D}))$, 2.8.5.1

L	étiquette d'un programme, 2.8.1.1
$\mathcal{L}$	ensemble des étiquettes, 2.8.1.1
$\mathcal{LP}$	langage de programmation, 2.1
M	un état mémoire, $(M \in \mathcal{M})$, 2.8.1.2.1
$\mathcal{M}$	ensemble des états mémoires, $(\mathcal{M} = (\mathcal{V} \to \mathcal{D}))$, 2.8.1.2.1
$\langle m, \Delta, a \rangle$	trace de longueur n $(n \in \omega+1)$ où Δ est la séquence d'états sur S $(\Delta \in (n \to S))$ et a la séquence d'actions sur A $(a \in (n-1 \to A))$, 2.1.1
p	trace, $(p = p_0$ si $\lvert p\rvert = 1$, $p = \langle p_i \xrightarrow{\sharp_i} p_{i+1} : i \in \lvert\sharp\rvert\rangle$ si $\lvert p\rvert > 1)$, 2.1.1
$\lvert p\rvert$	longueur de la trace p en nombre d'états, 2.1.1
$\lvert\sharp\rvert$	longueur de la trace p en nombre d'actions, 2.1.1
p_i	ième état de la trace p, 2.1.1
$\sharp_i$	ième action de la trace p, 2.1.1
$p^{<m}$	préfixe $\langle p_0 \dots p_{m-1} \rangle$ d'une trace p, 2.1.1
$p^{\leq m}$	préfixe $\langle p_0 \dots p_m \rangle$ d'une trace p, 2.1.1
$p^{>m}$	suffixe $\langle p_{m+1} \dots \rangle$ d'une trace p, 2.1.1
$p^{\geq m}$	suffixe $\langle p_m \dots \rangle$ d'une trace p, 2.1.1
$p^{<m,n>}$	tranche d'une trace p, $((p^{<m})^{>m})$, 2.1.1
$p^{\leq m,n\geqslant}$	tranche d'une trace p, $((p^{\leq m})^{\geq m})$, 2.1.1
$p^{\leq m,n>}$	tranche d'une trace p, $((p^{<m})^{\geq m})$, 2.1.1
$p^{<m,n\geqslant}$	tranche d'une trace p, $((p^{\leq m})^{>m})$, 2.1.1
$p \leftrightarrow q$	la trace p est préfixe de la trace q, $(\exists i \in \lvert q\rvert . p = q^{\leq i})$, 2.6.1
$p \twoheadrightarrow q$	la trace p est suffixe de la trace q, $(\exists i \in \lvert q\rvert . p = q^{>i})$, 2.6.2
$p \xrightarrow{a} q$	concaténation des traces p et q par l'action a, 2.1.1
$\sharp(Se)$	prendre sur un sémaphore Se, 2.8.5.1
$\sharp(Se,i)$	action qui correspond à l'exécution de la commande $\sharp(Se)$ par le processus Pro_i et au passage de ce sémaphore par ce processus, 2.8.5.2.2

action correspondant à un pas d'exécution du prélude d'un programme parallèle, 2.8.2.2.2

action correspondant à un pas d'exécution du postlude d'un programme parallèle, 2.8.2.2.2

P propriété d'un programme, $(Pe((\underset{\sim}{Spec} \times \underset{\sim}{Sem}\langle \mathcal{S}, \mathcal{A}\rangle) \to \{tt, ff\}))$, 2.5

Ppa programme parallèle asynchrone, 2.8.2.1

Ppc programme parallèle communicant, 2.8.3.1

Pps programme parallèle synchrone, 2.8.5.1

Ppw programme parallèle faiblement équitable, 2.8.4.1

Pr programme, 2.1.2

Pra processus asynchrone, 2.8.2.1

Prc processus parallèle communicant par envois de messages sur rendez-vous, 2.8.3.1

Prs processus synchrone, 2.8.5.1

Ps programme séquentiel, 2.8.1.1

{Ps} une liste de commandes séquentielles exécutées de manière indivisible, 2.8.2.1

$\mathcal{P}pa$ ensemble des programmes parallèles asynchrones,

$(\mathcal{P}pa \longrightarrow Ps \,[\![\, Pra_0 \,\|\, \ldots \,\|\, Pra_{m-1} \,]\!]; Ps' \quad (m>1))$, 2.8.2.1

$\mathcal{P}pc$ ensemble des programmes parallèles communicants,

$(\mathcal{P}pc \longrightarrow Ps \,[\![\, Prc_0 \,\|\, \ldots \,\|\, Prc_{m-1} \,]\!]; Ps' \quad (m>1))$, 2.8.3.1

$\mathcal{P}ps$ ensemble des programmes parallèles synchrones,

$(\mathcal{P}ps \longrightarrow Ps \,[\![\, Prs_0 \,\|\, \ldots \,\|\, Prs_{m-1} \,]\!]; Ps' \quad (m>1))$, 2.8.5.1

$\mathcal{P}pw$ ensemble des programmes parallèles faiblement équitables,

$(\mathcal{P}pw \longrightarrow \mathcal{P}pa)$

ensemble des programmes

$(\mathcal{P}r \longrightarrow Ps \mid Ppa \mid Ppc \mid Ppw \mid Pps)$, 2.8

$\mathcal{P}ra$ ensemble des processus asynchrones,

$(\mathcal{P}ra \longrightarrow \mathcal{L}_0 : Ca_0 ; \ldots ; \mathcal{L}_{m-1} : Ca_{m-1} ; \mathcal{L}_m : \quad (m>1))$, 2.8.2.1

ensemble de processus communicant par envois de messages sur rendez-vous

$(\mathcal{P}rc \longrightarrow \mathcal{L}_0 : Cc_0 ; \ldots ; \mathcal{L}_{m-1} : Cc_{m-1} ; \mathcal{L}_m : \quad (m>1))$, 2.8.3.1

$\mathcal{P}rs$	ensemble des processus synchrones,
	$(\mathcal{P}rs \longrightarrow \mathcal{L}_0 : \mathcal{C}s_0 ; \ldots ; \mathcal{L}_{m-1} : \mathcal{C}s_{m-1} ; \mathcal{L}_m :$ $(m>1)$), 2.8.5.1
$\mathcal{P}s$	ensemble des programmes séquentiels,
	$(\mathcal{P}s \longrightarrow \mathcal{L}_0 : \mathcal{C}_0 ; \ldots ; \mathcal{L}_{m-1} : \mathcal{C}_{m-1} ; \mathcal{L}_m :$ $(m>0)$), 2.8.1.1
$\lfloor\!\lfloor \mathcal{P}s \rfloor\!\rfloor$	ensemble des listes de commandes séquentielles exécutées de manière indivisible, 2.8.2.1
$\mathcal{P}ref$	fermeture par préfixes, 2.6.1
$\mathcal{P}ref^{<\omega}$	préfermeture par préfixes finis, 2.6.1
$\mathcal{P}ref(\langle S,A,\Sigma\rangle)$	fermeture par préfixes de la sémantique $\langle S,A,\Sigma\rangle$,
	$(\langle S, A, \{p \in \Sigma^{<\omega}\langle S,A\rangle . \exists q \in \Sigma . p \leftrightarrow q\}\rangle)$, 2.6.1
$\mathcal{P}ref^{<\omega}(\langle S,A,\Sigma\rangle)$	préfermeture par préfixes finis de la sémantique $\langle S,A,\Sigma\rangle$,
	$(\langle S, A, \{p \in \Sigma^{<\omega}\langle S,A\rangle . \exists q \in \Sigma . p \leftrightarrow q\}\rangle)$, 2.6.1
Q	file d'attente associée à un sémaphore,
	$(Q \in (\mathcal{N} \to n^{<\omega})$ où n est le nombre de processus synchrones), 2.8.5.2.1
ra	relation entre actions, $(ra \in (\mathcal{A} \times \mathcal{A} \to \{tt, ff\}))$, 2.5.3
rs	relation entre états, $(rs \in (\mathcal{S} \times \mathcal{S} \to \{tt, ff\}))$, 2.5.3
$\simeq\langle rs, ra\rangle$	concordance aux relations rs entre états et ra entre actions près, 2.5.3
$\simeq\langle rs, ra\rangle(p,q)$	concordance aux relations rs entre états et ra entre actions près entre les traces p et q, $([\lvert p\rvert = \lvert q\rvert \wedge \forall i \in \lvert p\rvert . rs(p_i, q_i) \wedge \forall i \in \lvert\!\lfloor p \rfloor\!\rvert . ra(\lfloor p \rfloor_i, \lfloor q \rfloor_i)])$, 2.5.3
$\simeq\langle rs, ra\rangle(\langle S,A,\Sigma\rangle, \langle S',A',\Sigma'\rangle)$	concordance aux relations rs entre états et ra entre actions près entre sémantiques, $([S' = rs[S] \wedge A' = ra[A] \wedge \Sigma' = \simeq\langle rs, ra\rangle[\Sigma]])$, 2.5.3
$\simeq\langle rs, ra\rangle(\langle S,A,t,\varepsilon\rangle, \langle S',A',t',\varepsilon'\rangle)$	concordance aux relations rs entre états et ra entre actions près entre systèmes de transition, $(\simeq\langle rs, ra\rangle(\langle S,A,\Sigma\langle S,A,t,\varepsilon\rangle\rangle, \langle S',A',\Sigma'\langle S',A',t',\varepsilon'\rangle\rangle))$, 2.5.3
$Redai\ \langle A'\rangle(p)$	trace dérivée de $p \in \Sigma\langle S,A\rangle$ par réduction des actions inobservables $A \sim A'$, 2.5.4.2
$Redei\ \langle S'\rangle(p)$	trace dérivée de $p \in \Sigma\langle S,A\rangle$ par réduction des états inobservables $S \sim S'$, 2.5.4.1

Redei $\langle s' \rangle (\langle S,A,\Sigma \rangle)$	sémantique dérivée de $\langle S,A,\Sigma \rangle$ par réduction des états inobservables $s \nu s'$, $(\langle S', A^{*\omega}, \text{Redei } \langle s' \rangle [\Sigma] \rangle)$, 2.5.4.1
Redei $\langle s' \rangle (\langle S,A,t,\varepsilon \rangle, \langle S',A',t',\varepsilon' \rangle)$	réduction des états inobservables $s \nu s'$ entre systèmes de transition, $([\text{Redei } \langle s' \rangle (\langle S,A,\Sigma\langle S,A,t,\varepsilon\rangle\rangle) = \langle S',A',\Sigma\langle S',A',t',\varepsilon'\rangle\rangle])$, 2.5.4.1
Redeaa $(\langle S,A,\Sigma \rangle)$	Réduction aux états et actions accessibles de la sémantique $\langle S,A,\Sigma \rangle$, $(\langle \{s \in S : \exists p \in \Sigma, i \in \lvert p \rvert \cdot p_i = s\}, \{a \in A : \exists p \in \Sigma, j \in \lvert p \rvert \cdot p_j = a\}, \Sigma \rangle)$, 2.6.3
Redeaa $(\langle S,A,t,\varepsilon \rangle, \langle S',A',t',\varepsilon' \rangle)$	réduction aux états et actions accessibles entre systèmes de transition $([\text{Redeaa } (\langle S,A,\Sigma\langle S,A,t,\varepsilon\rangle\rangle) = \langle S',A',\Sigma\langle S',A',t',\varepsilon'\rangle\rangle])$, 2.6.3
Retps	réduction par élimination des traces préfixes stricts, 2.6.6
Retps $(\langle S,A,\Sigma \rangle)$	réduction par élimination des traces préfixes stricts de la sémantique $\langle S,A,\Sigma \rangle$, $(\langle S,A,\{p \in \Sigma : \forall p \in \Sigma \cdot (p \mapsto q) \Rightarrow (p = q)\} \rangle)$, 2.6.6
Rtran	rétraction par transitions, 2.6.8
Rtran $(\langle S,A,\Sigma \rangle)$	rétraction par transitions de la sémantique $\langle S,A,\Sigma \rangle$, $(\langle S,A,\Sigma\langle S,A,t\langle S,A,\Sigma\rangle,\varepsilon\langle S,A,\Sigma\rangle\rangle\rangle)$, 2.6.8
s	état, 2.1.1
$\hat{s}$	état unique, 2.5.3.2
S	ensemble d'états, 2.1.1
$S[\![Pr]\!]$	ensemble non vide d'états associé au programme Pr, 2.1.2
Se	sémaphore, 2.8.5
Sp	spécification d'un programme, 2.5
Sl $[\![Pps]\!]$	ensemble d'états associé au programme synchrone Pps par la sémantique libérale, 2.8.5.3
$\mathcal{S}$	ensemble des états, 2.1.2
$\mathcal{S}e$	ensemble des sémaphores, $(\mathcal{S}e \subseteq \mathcal{S})$, 2.8.5
$\langle S,\Sigma \rangle$	sémantique concordante à $\langle S,A,\Sigma \rangle$ à l'annulation des actions près, 2.5.3.3
$\langle S,A,\Sigma \rangle$	sémantique, 2.1.2
$\langle S,A,\Sigma\langle S,A,t,\varepsilon\rangle\rangle$	sémantique engendrée par le système de transition $\langle S,A,t,\varepsilon \rangle$, 2.4
$\langle S[\![Pr]\!], A[\![Pr]\!], \Sigma[\![Pr]\!] \rangle$	sémantique associée au programme Pr, 2.1.2 - 2.8.1.2

$\langle S, A, t, \varepsilon \rangle$	système de transition, 2.2
$\langle S, A, t\langle S,A,\Sigma\rangle, \varepsilon\langle S,A,\Sigma\rangle\rangle$	système de transition engendré par la sémantique $\langle S,A,\Sigma\rangle$, 2.3
$\langle S[\![Pr]\!], A[\![Pr]\!], t[\![Pr]\!], \varepsilon[\![Pr]\!]\rangle$	système de transition associé au programme Pr, 2.8.1.2
skip	commande nulle, 2.8.1.1
Sem $\langle \mathcal{S}, \mathcal{A}\rangle$	ensemble des sémantiques sur les ensembles $\mathcal{S}$ d'états et $\mathcal{A}$ d'actions ($\{\langle S,A,\Sigma\rangle : S\subseteq\mathcal{S} \wedge A\subseteq\mathcal{A} \wedge \Sigma\subseteq\Sigma\langle S,A\rangle\}$), 2.1.2
$\langle$Sem$\langle\mathcal{S},\mathcal{A}\rangle, \subseteq, \langle\emptyset,\emptyset,\emptyset\rangle, \langle\mathcal{S},\mathcal{A},\Sigma\langle\mathcal{S},\mathcal{A}\rangle\rangle\rangle$	treillis complet des sémantiques, 2.5.1
Sfair $(\langle S,A,\Sigma\rangle)$	réduction d'une sémantique $\langle S,A,\Sigma\rangle$ aux traces fortement équitables, (Sfair $\langle A\rangle(\langle S,A,\Sigma\rangle)$), 2.6.4
Sfair $\langle\alpha\rangle(\langle S,A,\Sigma\rangle)$	réduction d'une sémantique $\langle S,A,\Sigma\rangle$ aux traces fortement équitables pour un ensemble α d'actions, $(\langle S,A,\{p\in\Sigma: \lvert p\rvert=\omega \Rightarrow \neg(\exists a\in\alpha, i\in\omega.(\forall j\geq i.\exists k\geq j.$ Enabled$(a,k,p,\Sigma))\wedge\forall j\geq i.\sharp_j\neq a)\}\rangle)$, 2.6.4
Spec	ensemble des spécifications, 2.5
succ	succ $[\![Ps]\!](L)(M,M')$ est la condition pour que l'état mémoire M' soit successeur de l'état mémoire M après exécution d'un pas de Ps au point de contrôle L, 2.8.1.2.4
Suff	fermeture par suffixes, 2.6.2
Suff$(\langle S,A,\Sigma\rangle)$	fermeture par suffixes de la sémantique $\langle S,A,\Sigma\rangle$, $(\langle S,A,\{p\in\Sigma^{\leq\omega}\langle S,A\rangle : \exists q\in\Sigma.\ p\rightarrow^* q\}\rangle)$, 2.6.2
se .. or .. or .. es	commande alternative, 2.8.3.1
t	relation de transition, $t\in(A\rightarrow(S\times S\rightarrow\{tt,ff\}))$, 2.2
t^*	fermeture transitive réflexive de t, $(t^*(s,s') = \bigcup_{m\geq 0} t^m(s,s')$ avec $t^0(s,s')=[s=s']$, $t^{m+1}(s,s')=[\exists a\in A, s''\in S.(t_a(s,s'')\wedge t^m(s'',s'))])$, 2.6.2
$t\,\vert\,S_d, S_i, S_f\,\vert_{\langle\rangle}(s,s')$	relation de transition entre un état s de S_d et un état s' de S_f par aucune action sur des états intermédiaires de S_i, $([s'=s \wedge s'\in S_f])$, 2.5.4
$t[\![Pr]\!]$	relation de transition associée au programme Pr, 2.1.2

$t1S_d, S_i, S_f\,{}^{\vdash}\!\!<a_0,\ldots,a_m>(s,s')$ relation de transition entre un état s de S_d et un état s' de S_f par les actions $a_0 \ldots a_m$ sur des états intermédiaires de S_i, $([\exists s \in (m+2 \to S).(s_0 = s \in S_d \wedge \forall j \in (m+1 \setminus 0).s_j \in S_i \wedge s_{m+1} \in S_f \wedge \forall j \in (m+1).t_{a_j}(s_j, s_{j+1}))])$, 2.5.4

$t<S,A,\Sigma>$ relation de transition engendrée par la sémantique $<S,A,\Sigma>$, $([\exists p \in \Sigma, i \in |p| .(p_i = s \wedge p_{i+1} = s' \wedge a = A_i)] = t_a(s,s'))$, 2.3

$tl[\![Pps]\!]$ relation de transition associée au programme synchrone Pps par la sémantique libérale, 2.8.5.3

Tran $<\mathcal{S},\mathcal{A}>$ ensemble des systèmes de transitions sur les ensembles $\mathcal{S}$ d'états et $\mathcal{A}$ d'actions, $(\{<S,A,t,\varepsilon>: S \in \mathcal{S} \wedge A \in \mathcal{A} \wedge t \in (A \to (S \times S \to \{tt, ff\})) \wedge \varepsilon \in (S \to \{tt, ff\})\})$, 2.2

$\underline{v}(se)$ rendre le sémaphore se, 2.8.5.1

$\underline{v}(se,i)$ action qui correspond à l'exécution de la commande $\underline{v}(se)$ par le processus Pr_i qui libère le sémaphore alors qu'aucun autre processus n'était en attente sur ce sémaphore, 2.8.5.2.2

$\underline{vp}(se,i,j)$ action qui correspond à l'exécution de la commande $\underline{v}(se)$ par le processus Pr_i qui libère le sémaphore et permet au processus Pr_j qui était en attente de le passer, 2.8.5.2.2

VA ensemble des variables auxiliaires, 4.1

$\mathcal{V}$ ensemble des variables, 2.8.1.1

$\underline{w}(se,i)$ action qui corrspond à l'exécution de la commande $\underline{p}(se)$ par le processus Pr_i qui provoque sa mise en attente devant le sémaphore se, 2.8.5.2.2

Wfair $(<S,A,\Sigma>)$ réduction d'une sémantique $<S,A,\Sigma>$ aux traces faiblement équitables, (Wfair $<A>(<S,A,\Sigma>)$), 2.6.4

Wfair $<\alpha>(<S,A,\Sigma>)$ réduction d'une sémantique $<S,A,\Sigma>$ aux traces faiblement équitables pour un ensemble α d'actions, $(<S,A,\{p \in \Sigma: |p| = \omega \Rightarrow \neg(\exists a \in \alpha, i \in \omega. \forall j \geq i .(\underline{\text{Enabled}}(a,k,p,\Sigma) \wedge p_j \neq a))\}>)$, 2.6.4

while ... do ... od composition itérative de commandes, 2.8.1.1

The first part of Radhia Cousot's thesis is available at
https://pcousot.github.io/publications/RadhiaCousotTheseEsSciences.PDF

The second part of Radhia Cousot's thesis is available at
https://pcousot.github.io/publications/RadhiaCousotTheseEsSciences2.PDF

The third part of Radhia Cousot's thesis is available at
https://pcousot.github.io/publications/RadhiaCousotTheseEsSciences3.PDF