Équivalence des sémantiques opérationnelles à petits et grands pas pour le langage IMP

Julie Parreaux

2018-2019

Référence du développement : Nielson & Nielson [1, p.41]

Leçons où on présente le développement : 930 (Sémantique).

1 Introduction

Nous avons définie deux sémantiques opérationnelles pour les instructions du langage IMP qui nous permettent de savoir comment on calcul. Ces deux sémantiques diffère par les pas de calcul qu'elle modélise : la sémantique à grand pas ne déroule pas les instructions contrairement à la sémantique à petit pas. Certaines propriétés sur les langages de programmation sont donc plus facilement exprimable par l'une ou l'autre de ces sémantiques, voir même inexprimable par l'une des sémantiques. Cependant, dans le cas particulier du langage IMP, nous allons voir qu'elles sont équivalentes : elles expriment donc les même propriété. Outre la motivation consistant à définir deux sémantiques équivalentes, ce résultat est intéressant par les deux inductions qu'il met en place dans sa preuve.

Remarques sur le développement

Ce développement demande une bonne compréhension des sémantiques opérationnelles à grands pas et à petits pas. Les propriétés sur les sémantiques sont à énoncer dans le plan (au tableau si on a le temps) mais pas à montrer.

1. Présentation des règles des sémantiques (à écrire dans un coin du tableau).
2. Preuve que la sémantique à grands pas implique celle à petit pas.
3. Preuve que la sémantique à petits pas implique celle à grands pas.

2 Les sémantiques opérationnelles à petits et grands pas

Sémantique opérationnelle à grands pas La sémantique opérationnelle à grands pas [1, p.20] est une sémantique permettant de décrire comment on calcul à l'aide de grandes étapes (dans une boucle par exemple, on ne détaille pas toutes les exécutions mais seulement le résultat si elle termine).

Définition. On définit la sémantique à grands pas des instructions du langage IMP à l'aide de règle du type $\langle S, s\rangle \rightarrow s'$ avec $s, s' \in State$.

$[skip_{NS}]$ $\langle \mathtt{skip}, s\rangle \rightarrow s$

$[ass_{NS}]$ $\langle x := a, s\rangle \rightarrow s\,[x \mapsto \mathcal{A}[\![a]\!]_s]$

$[comp_{NS}]$ $\dfrac{\langle S_1, s\rangle \rightarrow s' \quad \langle S_2, s'\rangle \rightarrow s''}{\langle S_1; S_2, s\rangle \rightarrow s''}$

$[if^{tt}_{NS}]$ $\dfrac{\langle S_1, s\rangle \rightarrow s'}{\langle \text{ if } b \text{ then } S_1 \text{ else } S_2, s\rangle \rightarrow s'}$ si $\mathcal{B}[\![b]\!]_s = tt$

$[if^{ff}_{NS}]$ $\dfrac{\langle S_2, s\rangle \rightarrow s'}{\langle \text{ if } b \text{ then } S_1 \text{ else } S_2, s\rangle \rightarrow s'}$ si $\mathcal{B}[\![b]\!]_s = ff$

$[while^{tt}_{NS}]$ $\dfrac{\langle S, s\rangle \rightarrow s' \quad \langle \text{while } b \text{ do } S, s'\rangle \rightarrow s''}{\langle \text{while } b \text{ do } S, s\rangle \rightarrow s''}$ si $\mathcal{B}[\![b]\!]_s = tt$

$[while^{ff}_{NS}]$ $\langle \text{while } b \text{ do } S, s\rangle \rightarrow s$ si $\mathcal{B}[\![b]\!]_s = ff$

Définition. On appelle arbre de dérivation la succession d'applications des règles et des axiomes.

Définition. Le langage IMP est déterministe sous la sémantique à grands pas, on définit donc une fonction sémantique : $\mathcal{S}_{NS} : Stm \to (State \hookrightarrow State)$ par

$$\mathcal{S}_{NS}[\![S]\!]s = \begin{cases} s' & \text{si } \langle S, s\rangle \to s' \\ \text{undef} & \text{sinon} \end{cases}$$

Limite : La sémantique à grands pas ne permet pas toujours d'avoir la terminaison de notre instruction.

Sémantique opérationnelle à grands pas La sémantique opérationnelle à petits pas [1, p.33] est une sémantique permettant de décrire comment on calcul à l'aide de grandes étapes. Elle permet de plus, grâce à ses petits itérations de décider si une instruction termine ou non.

Définition. On définit la sémantique à petits pas des instructions du langage IMP à l'aide de règle du type $\langle S, s\rangle \Rightarrow \gamma$ avec $s \in State$ et γ de la forma $\langle S', s'\rangle$ ou s' avec $s' \in State$ et $S' \in Stm$.

$[skip_{SOS}]$ $\quad \langle \text{skip}, s\rangle \Rightarrow s$ $\qquad [ass_{SOS}]$ $\quad \langle x := a, s\rangle \Rightarrow s\,[x \mapsto \mathcal{A}[\![a]\!]_s]$

$[comp^1_{SOS}]$ $\quad \dfrac{\langle S_1, s\rangle \Rightarrow \langle S_1', s'\rangle}{\langle S_1; S_2, s\rangle \Rightarrow \langle S_1'; S_2, s'\rangle}$ $\qquad [comp^2_{SOS}]$ $\quad \dfrac{\langle S_1, s\rangle \Rightarrow s'}{\langle S_1; S_2, s\rangle \Rightarrow \langle S_2, s'\rangle}$

$[if^{tt}_{SOS}]$ $\quad \langle \text{ IF } b \text{ THEN } S_1 \text{ ELSE } S_2, s\rangle \Rightarrow \langle S_1, s'\rangle$ $\qquad$ si $\mathcal{B}[\![b]\!]_s = tt$

$[if^{ff}_{SOS}]$ $\quad \langle \text{ IF } b \text{ THEN } S_1 \text{ ELSE } S_2, s\rangle \Rightarrow \langle S_2, s'\rangle$ $\qquad$ si $\mathcal{B}[\![b]\!]_s = ff$

$[while_{SOS}]$ $\quad \langle \text{while } b \text{ do } S, s\rangle \to \langle \text{ IF } b \text{ THEN } (S; \text{while } b \text{ do } S) \text{ ELSE skip}, s\rangle$

Définition. Une séquence de dérivation pour $S \in Stm$ et $s \in State$ est soit une séquence finie $\gamma_0 \Rightarrow \cdots \Rightarrow \gamma_k$ avec $\gamma_0 = \langle S, s\rangle$ et γ_k une configuration finale ; soit une séquence infinie $\gamma_0 \Rightarrow \gamma_1 \Rightarrow \ldots$ avec $\gamma_0 = \langle S, s\rangle$.

Définition. Le langage IMP est déterministe sous la sémantique à petits pas, on définit donc une fonction sémantique : $\mathcal{S}_{SOS} : Stm \to (State \hookrightarrow State)$ par

$$\mathcal{S}_{SOSS}[\![S]\!]s = \begin{cases} s' & \text{si } \langle S, s\rangle \Rightarrow^* s' \\ \text{undef} & \text{sinon} \end{cases}$$

3 Équivalence de ces sémantique pour le langage IMP

Théorème. *Pour toute instruction S du langage IMP,* $\mathcal{S}_{SOS}[\![S]\!] = \mathcal{S}_{NS}[\![S]\!]$

Démonstration. Ce théorème exprime la propriété suivante : l'exécution d'une instruction S à partir d'un état s termine dans l'une des sémantique (sémantique à grands pas) si et seulement si elle termine dans l'autre (sémantique à petits pas). Commençons par énoncer quelques propriétés sur la sémantique à petits pas.

Propriétés sur la sémantique à petits pas On énonce ici des résultats sur le comportement de la composition sous la sémantique à petits pas. En effet, dans une composition l'instruction suivante n'affecte pas son instruction précédente (lemme 1). De plus, elle peut être décomposée en deux étapes : elle étudie la première instruction puis la seconde (lemme 2)

Lemme 1. *Soient* $S_1, S_2 \in Stm$, $s, s' \in State$ *et* $k \in \mathbb{N}$ *tels que* $\langle S_1, s\rangle \Rightarrow^k s'$, *alors* $\langle S_1; S_2, s\rangle \Rightarrow^k \langle S_2, s'\rangle$. *L'exécution de* S_1 *n'est pas affectée par l'exécution de l'instruction suivante.*

Démonstration. On raisonne par induction sur k.

Initialisation ($k = 0$) Impossible ($\langle S_1, s\rangle \neq s'$)

Héridité Soit k tel que le lemme soit vrai. Soit $s \in State$ tel que $\langle S_1, s\rangle \Rightarrow^{k+1} s'$. Soient S_1, S_2 deux instructions du langage IMP. On distingue les cas selon l'instruction S_1 :

Cas $[ass_{SOS}]$ ok (on est dans le cas $k = 0$) Soit $s \in State$ tel que $\langle x := a, s\rangle \Rightarrow s[x \mapsto \mathcal{A}[\![a]\!]_s]$. Par la règle $[comp^2_{SOS}]$, on a $\langle x := a; S_2, s\rangle \Rightarrow \langle S_2, s[x \mapsto \mathcal{A}[\![a]\!]_s]\rangle$.

Cas $[skip_{SOS}]$ ok (on est dans le cas $k = 0$) Soit $s \in State$ tel que $\langle \mathsf{skip}, s\rangle \Rightarrow s$. Par la règle $[comp^2_{SOS}]$, on a $\langle \textsc{skip}; S_2, s\rangle \Rightarrow s$.

Cas $[comp^1_{SOS}]$ Soient S_1', S_2' deux instructions et $s, s'' \in State$ tels que $\left\langle S_1'; S_2', s\right\rangle \Rightarrow^{k+1} s'$. En appliquant la règle $[comp^1_{SOS}]$, on a $\left\langle S_1'; S_2', s\right\rangle \Rightarrow \left\langle S_1''; S_2', s\right\rangle \Rightarrow^k s'$. En appliquant l'hypothèse d'induction à $\left\langle S_1''; S_2', s\right\rangle \Rightarrow^k s'$, on obtient $\left\langle S_1''; S_2'; S_2, s\right\rangle \Rightarrow^k s'$ et donc $\langle S_1; S_2, s\rangle \Rightarrow^{k+1} s'$.

Cas $[comp^2_{SOS}]$ Soient S_1', S_2' deux instructions et $s, s'' \in State$ tels que $\left\langle S_1'; S_2', s\right\rangle \Rightarrow^{k+1} s'$. En appliquant la règle $[comp^1_{SOS}]$, on a $\left\langle S_1'; S_2', s\right\rangle \Rightarrow \langle S_2', s\rangle \Rightarrow^k s'$. En appliquant l'hypothèse d'induction à $\langle S_2', s\rangle \Rightarrow^k s'$, on obtient $\langle S_2'; S_2, s\rangle \Rightarrow^k s'$ et donc $\langle S_1; S_2, s\rangle \Rightarrow^{k+1} s'$.

Cas $[if^{tt}_{SOS}]$ Soient S_1', S_2' deux instructions et $s, s'' \in State$ tels que $\left\langle \mathsf{if}\ b\ \mathsf{then}\ S_1'\ \mathsf{else}\ S_2', s\right\rangle \Rightarrow^{k+1} s'$ et b tel que $\mathcal{B}[\![b]\!] = tt$. On applique alors $[if^{tt}_{SOS}]$, ce qui donne la séquence de dérivation suivante $\left\langle \mathsf{if}\ b\ \mathsf{then}\ S_1'\ \mathsf{else}\ S_2', s\right\rangle \Rightarrow \left\langle S_1', s\right\rangle \Rightarrow^k s'$. On applique l'hypothèse de récurrence à la dérivation $\left\langle S_1', s\right\rangle \Rightarrow^k s'$, on obtient $\left\langle S_1'; S_2, s\right\rangle \Rightarrow^k s'$. On obtient donc $\langle S_1; S_2, s\rangle \Rightarrow^{k+1} s'$.

Cas $[if^{ff}_{SOS}]$ Analogue au cas $[if^{tt}_{SOS}]$. Soient S_1', S_2' deux instructions et $s, s'' \in State$ tels que $\left\langle \mathsf{if}\ b\ \mathsf{then}\ S_1'\ \mathsf{else}\ S_2', s\right\rangle \Rightarrow^{k+1} s'$ et b tel que $\mathcal{B}[\![b]\!] = ff$. On applique alors $[if^{ff}_{SOS}]$, ce qui donne la séquence de dérivation suivante $\left\langle \mathsf{if}\ b\ \mathsf{then}\ S_1'\ \mathsf{else}\ S_2', s\right\rangle \Rightarrow \langle S_2', s\rangle \Rightarrow^k s'$. On applique l'hypothèse de récurrence à la dérivation $\langle S_2', s\rangle \Rightarrow^k s'$, on obtient $\langle S_2'; S_2, s\rangle \Rightarrow^k s'$. On obtient donc $\langle S_1; S_2, s\rangle \Rightarrow^{k+1} s'$.

Cas $[while_{SOS}]$ Soient S une instruction et $s, s'' \in State$ tels que $\langle \mathsf{while}\ b\ \mathsf{do}\ S, s\rangle \Rightarrow^{k+1} s'$. On applique alors $[while_{SOS}]$, ce qui donne la séquence de dérivation suivante $\langle \mathsf{while}\ b\ \mathsf{do}\ S, s\rangle \Rightarrow \langle \mathsf{if}\ b\ \mathsf{then}\ (S;\ \mathsf{while}\ b\ \mathsf{do}\ S)\ \mathsf{else}\ \mathsf{skip}, s\rangle \Rightarrow^k s'$. On applique l'hypothèse de récurrence à la dérivation $\Rightarrow \langle \mathsf{if}\ b\ \mathsf{then}\ (S;\ \mathsf{while}\ b\ \mathsf{do}\ S)\ \mathsf{else}\ \mathsf{skip}, s\rangle \Rightarrow^k s'$, on obtient $Rightarrow \langle (\mathsf{if}\ b\ \mathsf{then}\ (S;\ \mathsf{while}\ b\ \mathsf{do}\ S)\ \mathsf{else}\ \mathsf{skip}); S_2, s\rangle \Rightarrow^k s'$. On obtient donc $\langle S_1; S_2, s\rangle \Rightarrow^{k+1} s'$.

□

Lemme 2. *Soient $S_1, S_2 \in Stm$, $s, s'' \in State$ et $k \in \mathbb{N}$ tels que $\langle S_1; S_2, s\rangle \Rightarrow^k s''$, alors il existe $s' \in State$ et $k_1, k_2 \in \mathbb{N}$ tels que $k = k_1 + k_2$ et $\langle S_1, s\rangle \Rightarrow^{k_1} s'$, $\langle S_2, s'\rangle \Rightarrow^{k_2} s''$. L'exécution de la séquence peut être décomposée en deux parties : la première puis la deuxième instruction.*

Démonstration. On raisonne par induction sur k.

Initialisation ($k = 0$) Impossible ($\langle S_1; S_2, s\rangle \neq s''$)

Héridité Soit k tel que le lemme soit vrai. Soit $s \in State$ tel que $\langle S_1; S_2, s\rangle \Rightarrow^{k+1} s''$. On alors $\langle S_1; S_2, s\rangle \Rightarrow \gamma \Rightarrow^k s''$. On distingue deux cas :

Cas $\langle S_1, s\rangle \Rightarrow \langle S_1', s'\rangle$ Dans ce cas, $\gamma = \left\langle S_1'; S_2, s'\right\rangle \Rightarrow^k s''$. On applique l'hypothèse d'induction et on trouve k_1' et k_2. On pose $k_1 = k_1' + 1$. D'où le résultat.

Cas $\langle S_1, s\rangle \Rightarrow s'$ Dans ce cas, $\gamma = \langle S_2, s'\rangle \Rightarrow^k s''$. On a alors $k_1 = 1$ et $k_2 = k$.

□

Équivalence sous la sémantique à grands pas On énonce un résultat d'équivalence que nous utiliserons dans la preuve.

Lemme 3. *Pour toute instruction S, les instructions $\mathsf{if}\ b\ \mathsf{then}\ (S;\ \mathsf{while}\ b\ \mathsf{do}\ S)\ \mathsf{else}\ \mathsf{skip}$ et $\mathsf{while}\ b\ \mathsf{do}\ S$ sont sémantiquement équivalence sous la sémantique à grands pas.*

Démonstration. Supposons qu'il existe $s' \in State$ tel que $\langle \mathsf{while}\ b\ \mathsf{do}\ S, \rangle \to s'$. On distingue deux cas :

— Si on est dans le cas d'application de la règle $[while^{ff}_{NS}]$. On a alors $\mathcal{B}[\![b]\!]_s = ff$ et $s' = s$. On obtient l'arbre de dérivation en appliquant les règles $[if^{ff}_{NS}]$ et $[skip_{NS}]$.

— Si on est dans le cas d'application de la règle $[while^{tt}_{NS}]$, on construit les deux arbres de dérivations qui vont bien : pour while, on applique la règle $[while^{tt}_{NS}]$ et dans le cas du if, on applique les règles $[if^{tt}_{NS}]$ et $[while^{tt}_{NS}]$.

Réciproquement, on raisonne de même. Si on est dans le cas de l'application de la règle $[if^{ff}_{NS}]$, alors on peut appliquer la règle $[while^{ff}_{NS}]$ ce qui nous donne la concordance. Sinon, on applique la règle $[while^{tt}_{NS}]$ pour le while et les règles $[if^{tt}_{NS}]$ et $[while^{tt}_{NS}]$ pour le if. □

Étape 2 : grands pas implique petit pas Soient S une instruction du lange IMP et $s, s' \in State$, montrons que $\langle S, s\rangle \to s'$ implique $\langle S, s\rangle \Rightarrow^* s'$. On raisonne par induction sur la structure de l'arbre de dérivation de $\langle S, s\rangle \to s'$. On pose $\mathcal{H} = "\forall s, s' \in State, \langle S, s\rangle \to s'$ implique $\exists k \in \mathbb{N}, \langle S, s\rangle \Rightarrow^* s'"$ l'hypothèse d'induction.

Cas $[ass_{NS}]$ Soit $s \in State$ tel que $\langle x := a, s\rangle \to s[x \mapsto \mathcal{A}[\![a]\!]_s]$ (donnée par la règle $[ass_{NS}]$ de la sémantique à grands pas). Par la règle $[ass_{SOS}]$ (sémantique à petit pas), on a $\langle x := a, s\rangle \Rightarrow s[x \mapsto \mathcal{A}[\![a]\!]_s]$. Donc $\mathcal{H}$ est vraie pour $[ass_{NS}]$.

Cas $[skip_{NS}]$ Analogue au cas $[ass_{NS}]$. (Soit $s \in State$ tel que $\langle \mathsf{skip}, s\rangle \to s$ (donnée par la règle $[ass_{NS}]$ de la sémantique à grands pas). Par la règle $[skip_{SOS}]$ (sémantique à petit pas), on a $\langle \mathsf{skip}, s\rangle \Rightarrow s$. Donc $\mathcal{H}$ est vraie pour $[skip_{NS}]$.)

Cas $[comp_{NS}]$ Soient S_1 et S_2 deux instructions du langage IMP et $s, s', s'' \in State$ tels qu'on ait

$$[comp_{NS}]\ \frac{\langle S_1, s\rangle \to s' \qquad \langle S_2, s'\rangle \to s''}{\langle S_1; S_2, s\rangle \to s''}$$

En appliquant l'hypothèse d'induction aux deux prémisses $\langle S_1, s\rangle \to s'$ et $\langle S_2, s'\rangle \to s''$ nous donne $\langle S_1, s\rangle \Rightarrow^* s'$ et $\langle S_2, s'\rangle \Rightarrow^* s''$. Par le lemme 1, on a $\langle S_1; S_2, s\rangle \Rightarrow^* \langle S_2, s'\rangle$. On a ainsi la séquence de dérivation $\langle S_1; S_2, s\rangle \Rightarrow^* \langle S_2, s'\rangle \Rightarrow^* s''$. Donc $\langle S_1; S_2, s\rangle \Rightarrow^* s''$ et $\mathcal{H}$ est vraie pour $[comp_{NS}]$.

Cas $[if^{tt}_{NS}]$ Soient S_1 et S_2 deux instructions du langage IMP et $s, s', s'' \in State$ tels qu'on ait

$$[if^{tt}_{NS}]\frac{\langle S_1, s\rangle \to s'}{\langle \text{ IF } b \text{ THEN } S_1 \text{ ELSE } S_2, s\rangle \to s'} \text{ si } \mathcal{B}[\![b]\!]_s = tt$$

Comme $\mathcal{B}[\![b]\!]_s = tt$, par la règle $[if^{tt}_{SOS}]$, on a $\langle \text{ IF } b \text{ THEN } S_1 \text{ ELSE } S_2, s\rangle \Rightarrow \langle S_1, s\rangle$. En appliquant l'hypothèse d'induction à la prémisse $\langle S_1, s\rangle \to s'$, on a $\langle S_1, s\rangle \Rightarrow^* s'$. On a ainsi la séquence de dérivation $\langle \text{ IF } b \text{ THEN } S_1 \text{ ELSE } S_2, s\rangle \Rightarrow \langle S_1, s\rangle \Rightarrow^* s'$. Donc $\langle \text{ IF } b \text{ THEN } S_1 \text{ ELSE } S_2, s\rangle \Rightarrow^* s'$ et $\mathcal{H}$ est vraie pour $[if^{tt}_{NS}]$.

Cas $\left[if^{ff}_{NS}\right]$ Analogue au cas $[if^{tt}_{NS}]$. Soient S_1 et S_2 deux instructions du langage IMP et $s, s', s'' \in State$ tels qu'on ait

$$[if^{ff}_{NS}]\frac{\langle S_2, s\rangle \to s'}{\langle \text{ IF } b \text{ THEN } S_1 \text{ ELSE } S_2, s\rangle \to s'} \text{ si } \mathcal{B}[\![b]\!]_s = ff$$

Comme $\mathcal{B}[\![b]\!]_s = ff$, par la règle $[if^{ff}_{SOS}]$, on a $\langle \text{ IF } b \text{ THEN } S_1 \text{ ELSE } S_2, s\rangle \Rightarrow \langle S_2, s\rangle$. En appliquant l'hypothèse d'induction à la prémisse $\langle S_2, s\rangle \to s'$, on a $\langle S_2, s\rangle \Rightarrow^* s'$. On a ainsi la séquence de dérivation $\langle \text{ IF } b \text{ THEN } S_1 \text{ ELSE } S_2, s\rangle \Rightarrow \langle S_2, s\rangle \Rightarrow^* s'$. Donc $\langle \text{ IF } b \text{ THEN } S_1 \text{ ELSE } S_2, s\rangle \Rightarrow^* s'$ et $\mathcal{H}$ est vraie pour $\left[if^{ff}_{NS}\right]$.

Cas $[while^{tt}_{NS}]$ Soient S une instruction du langage IMP et $s, s', s'' \in State$ tels que

$$[while^{tt}_{NS}]\frac{\langle S, s\rangle \to s' \qquad \langle \mathsf{while}\ b\ \mathsf{do}\ S, s'\rangle \to s''}{\langle \mathsf{while}\ b\ \mathsf{do}\ S, s\rangle \to s''} \text{ si } \mathcal{B}[\![b]\!]_s = tt$$

En appliquant l'hypothèse d'induction aux prémisses $\langle S, s\rangle \to s'$ et $\langle \mathsf{while}\ b\ \mathsf{do}\ S, s'\rangle \to s''$, on a $\langle S, s\rangle \Rightarrow^* s'$ et $\langle \mathsf{while}\ b\ \mathsf{do}\ S, s'\rangle \Rightarrow^* s''$. Par le lemme 1 appliqué à $\langle S, s\rangle \Rightarrow^* s'$, on a $\langle S; \mathsf{while}\ b\ \mathsf{do}\ S, s\rangle \Rightarrow^* s''$. Par les règles $[while^{tt}_{NS}]$ et $[if^{tt}_{NS}]$, on a la séquence de dérivation : $\langle \mathsf{while}\ b\ \mathsf{do}\ S, s\rangle \Rightarrow \langle \text{ IF } b \text{ THEN } (S; \mathsf{while}\ b\ \mathsf{do}\ S) \text{ ELSE } \mathsf{skip}, s\rangle \Rightarrow \langle S; \mathsf{while}\ b\ \mathsf{do}\ S, s\rangle$. On a ainsi la séquence de dérivation $\langle \mathsf{while}\ b\ \mathsf{do}\ S, s\rangle \Rightarrow^* \langle S; \mathsf{while}\ b\ \mathsf{do}\ S, s\rangle \Rightarrow^* s''$. Donc $\langle \mathsf{while}\ b\ \mathsf{do}\ S, s\rangle \Rightarrow^* s''$ et $\mathcal{H}$ est vraie pour $\left[while^{tt}_{NS}\right]$.

Cas $\left[while^{ff}_{NS}\right]$ Évident Soient S une instruction du langage IMP et $s \in State$ tels que $[while^{ff}_{NS}]$ $\langle$while b do $S, s\rangle \rightarrow s$ si $\mathcal{B}[\![b]\!]_s = ff$. Par la règle $[while_{SOS}]$, on a $\langle$while b do $S, s\rangle \Rightarrow \langle$ IF b THEN $(S;$while b do $S)$ ELSE skip$, s\rangle$. Comme $\mathcal{B}[\![b]\!]_s = ff$, la règle $[if^{ff}_{SOS}]$ donne la séquence de dérivation suivante : $\langle$while b do $S, s\rangle \Rightarrow \langle$ IF b THEN $(S;$while b do $S)$ ELSE skip$, s\rangle \Rightarrow \langle$skip$, s\rangle$. On applique enfin la règle $[skip_{SOS}]$, ce qui nous donne : $\langle$while b do $S, s\rangle \Rightarrow^* s$ et $\mathcal{H}$ est vraie pour $\left[while^{ff}_{NS}\right]$.

Étape 3 : petits pas implique grands pas Réciproquement, soient S une instruction du lange IMP, $s, s' \in State$ et $k \in \mathbb{N}$, montrons que $\langle S, s\rangle \Rightarrow^k s'$ implique $\langle S, s\rangle \rightarrow s'$. Contrairement à l'implication précédente, on ne peut pas raisonner de manière structurelle sur la sémantique (dans le cas du while on n'a pas un sous terme strict). On raisonne par induction sur k. On pose $\mathcal{H}_k = $ " $\langle S, s\rangle \Rightarrow^k s'$ implique $\langle S, s\rangle \rightarrow s'$" l'hypothèse d'induction.

Initialisation ($k = 0$) Impossible car $\langle S, s\rangle$ n'est pas égal à s'.

Hérédité Soit $k_0 \in \mathbb{N}$ tel que $\forall k \leq k_0$, $\langle S, s\rangle \Rightarrow^k s'$ implique $\langle S, s\rangle \rightarrow s'$. Montrons que si $\langle S, s\rangle \Rightarrow^{k_0+1} s'$ alors $\langle S, s\rangle \rightarrow s'$. On distingue les cas selon l'instruction S.

Cas $[ass_{SOS}]$ ok (on est dans le cas $k_0 = 0$) Soit $s \in State$ tel que $\langle x := a, s\rangle \Rightarrow s[x \mapsto \mathcal{A}[\![a]\!]_s]$. Par la règle $[ass_{NS}]$, on a $\langle S, s\rangle \rightarrow s[x \mapsto \mathcal{A}[\![a]\!]_s]$. On a donc $\mathcal{H}_{k_0+1}$.

Cas $[skip_{SOS}]$ ok (on est dans le cas $k_0 = 0$) Soit $s \in State$ tel que $\langle$skip$, s\rangle \Rightarrow s$. Par la règle $[skip_{NS}]$, on a $\langle$SKIP$, s\rangle \rightarrow s$. On a donc $\mathcal{H}_{k_0+1}$.

Cas $[comp^1_{SOS}]$ **et** $[comp^2_{SOS}]$ Soient S_1, S_2 deux instructions et $s, s'' \in State$ tels que $\langle S_1; S_2, s\rangle \Rightarrow^{k_0+1} s'$. En appliquant le lemme 2 à $\langle S_1; S_2, s\rangle \Rightarrow^{k_0+1} s$, il existe $s' \in State$ et $k_1, k_2 \in \mathbb{N}$ tels que $k_0 + 1 = k_1 + k_2$ et $\langle S_1, s\rangle \Rightarrow^{k_1} s'$, $\langle S_2, s'\rangle \Rightarrow^{k_2} s''$. On applique l'hypothèse d'induction à $\langle S_1, s\rangle \Rightarrow^{k_1} s'$ et à $\langle S_2, s'\rangle \Rightarrow^{k_2} s''$, on obtient $\langle S_1, s\rangle \rightarrow s'$ et $\langle S_2, s'\rangle \rightarrow s''$. Par la règle $[comp_{NS}]$, on a $\langle S_1; S_2, s\rangle \rightarrow s$ et $\mathcal{H}_{k_0+1}$ est vraie.

Cas $[if^{tt}_{SOS}]$ Soient S_1, S_2 deux instructions et $s, s'' \in State$ tels que $\langle$ if b then S_1 else $S_2, s\rangle \Rightarrow^{k_0+1} s'$ et b tel que $\mathcal{B}[\![b]\!] = tt$. On applique alors $[if^{tt}_{SOS}]$, ce qui donne la séquence de dérivation suivante $\langle$ if b then S_1 else $S_2, s\rangle \Rightarrow \langle S_1, s\rangle \Rightarrow^{k_0} s'$. On applique l'hypothèse de récurrence à la dérivation $\langle S_1, s\rangle \Rightarrow^{k_0} s'$, on obtient $\langle S_1, s\rangle \rightarrow s'$. On applique la règle $[if^{tt}_{NS}]$, $\langle$ if b then S_1 else $S_2, s\rangle \rightarrow s'$. On a donc $\mathcal{H}_{k_0+1}$.

Cas $[if^{ff}_{SOS}]$ Analogue au cas $[if^{tt}_{SOS}]$. Soient S_1, S_2 deux instructions et $s, s'' \in State$ tels que $\langle$ if b then S_1 else $S_2, s\rangle \Rightarrow^{k_0+1} s'$ et b tel que $\mathcal{B}[\![b]\!] = ff$. On applique alors $[if^{ff}_{SOS}]$, ce qui donne la séquence de dérivation suivante $\langle$ if b then S_1 else $S_2, s\rangle \Rightarrow \langle S_2, s\rangle \Rightarrow^{k_0} s'$. On applique l'hypothèse de récurrence à la dérivation $\langle S_2, s\rangle \Rightarrow^{k_0} s'$, on obtient $\langle S_2, s\rangle \rightarrow s'$. On applique la règle $[if^{ff}_{NS}]$, $\langle$ if b then S_1 else $S_2, s\rangle \rightarrow s'$. On a donc $\mathcal{H}_{k_0+1}$.

Cas $[while_{SOS}]$ Soient S une instruction et $s, s'' \in State$ tels que $\langle$ while b do $S, s\rangle \Rightarrow^{k_0+1} s'$. On applique alors $[while_{SOS}]$, ce qui donne la séquence de dérivation suivante $\langle$ while b do $S, s\rangle \Rightarrow \langle$ if b then $(S;$ while b do $S)$ else skip$, s\rangle \Rightarrow^{k_0} s'$. On applique l'hypothèse de récurrence à la dérivation $\Rightarrow \langle$ if b then $(S;$ while b do $S)$ else skip$, s\rangle \Rightarrow^{k_0} s'$, on obtient $Rightarrow \langle$ if b then $(S;$ while b do $S)$ else skip$, s\rangle \rightarrow s'$. Par équivalence sémantique de if b then $(S;$ while b do $S)$ else skip et de while b do S dans la sémantique à grands pas, $\langle$ while b do $S, s\rangle \rightarrow s'$. On a donc $\mathcal{H}_{k_0+1}$.

D'où l'équivalence □

Références

[1] H. R. Nielson ; F. Nielson. *Semantics with application*.