Équivalence entre sémantiques opérationnelles

Leçons : 930

Références : NIELSON & NIELSON, *Semantics with applications* (p.40) [?] et LESESVRE–MONTAGNON–LE BARBENCHON–PIERRON, *131 développements pour l'oral* (p. 830) [?]

On notera $\langle S, s\rangle \to s'$ pour la sémantique à grands pas (NS) (voir appendice)
On notera $\langle S, s\rangle \Longrightarrow s'$ pour un pas de la sémantique à petits pas (SOS) (voir appendice)

Introduction :

Nous allons prouver que la sémantique opérationnelle à petits pas et la sémantique opérationnelle à grands pas sont équivalentes. Elles sont équivalentes mais ont toutes deux des avantages différents. [1]

Théorème 1. Pour toute instruction S, on a $\mathscr{S}_{SOS}[\![S]\!] = \mathscr{S}_{NS}[\![S]\!]$

Lemme 1. Si $\langle S_1, s\rangle \Longrightarrow^* s'$, alors

$$\langle S_1; S_2, s\rangle \Longrightarrow^* \langle S_2, s'\rangle$$

Démonstration. Par récurrence sur le nombre de pas.
Initialisation : Si $k = 1$, alors $\langle S_1, s\rangle \Longrightarrow s'$, d'où par la règle de composition [comp $_{SOS}$], on a

$$\langle S_1; S_2, s\rangle \Longrightarrow \langle S_2, s'\rangle$$

ce qui prouve le cas d'initialisation.
Hérédité : Supposons que l'on ait le résultat pour tout $k \in \mathbb{N}^*$. Si $\langle S_1, s\rangle \Longrightarrow^{k+1} s'$, on a alors $\langle S_1, s\rangle \Longrightarrow \langle S_1', s''\rangle$. Ainsi $\langle S_1', s''\rangle \Longrightarrow^k s'$ puis, en appliquant l'hypothèse de récurrence, cela donne

$$\langle S_1'; S_2, s''\rangle \Longrightarrow^* s'$$

On utilise alors la règle de composition [comp $_{SOS}$] puisque l'on a $\langle S_1, s\rangle \Longrightarrow \langle S_1', s''\rangle$ pour avoir

$$\langle S_1; S_2, s\rangle \Longrightarrow \langle S_1'; S_2, s''\rangle \Longrightarrow^* s'$$

D'où le résultat souhaité.
Conclusion : Ainsi, pour tout $k \in \mathbb{N}^*$, si $\langle S_1, s\rangle \Longrightarrow^k s'$, alors

$$\langle S_1; S_2, s\rangle \Longrightarrow^* \langle S_2, s'\rangle$$

□

1. par exemple, la sémantique à grands pas permet d'être plus concis car on n'explique pas chaque pas de calcul, cependant sur un langage qui aurait des exceptions (division par zero par exemple) la sémantique à petit pas permet d'aller calculer jusqu'à l'exception dans une boucle while alors que la sémantique à grands pas bloquera dès le début de la boucle while car il ne trouverait pas le résultat final.

Lemme 2. Si $\langle S_1; S_2, s\rangle \Longrightarrow^k s'$ alors il existe $k_1, k_2 \in \mathbb{N}$ tels que $k_1 + k_2 = k$ et

$$\langle S_1, s\rangle \Longrightarrow^{k_1} s'' \text{ et } \langle S_2, s''\rangle \Longrightarrow^{k_2} s'$$

Démonstration. Par récurrence sur le nombre de pas.
Initialisation : On ne peut pas avoir $k = 1$[2], donc on initialise la récurrence pour $k = 2$.Si $\langle S_1; S_2, s\rangle \Longrightarrow^2 s'$ c'est que $\langle S_1, s\rangle \Longrightarrow s''$ et $\langle S_2, s''\rangle \Longrightarrow s'$ (par la règle de composition [comp $_{SOS}$]). Donc on a bien $k_1 = 1$ et $k_2 = 1$ avec $k_1 + k_2 = 2$, ce qui prouve le cas d'initialisation.
Hérédité : Soit $k > 2$, on suppose que la propriété est vraie pour tout $l < k$. On suppose que $\langle S_1; S_2, s\rangle \Longrightarrow^k s'$.

- Si $\langle S_1, s\rangle \Longrightarrow s''$, alors, par [comp $_{SOS}$], on a $\langle S_2, s''\rangle \Longrightarrow^{k-1} s'$ et comme $(k-1)+1 = k$, on a le résultat souhaité
- Si $\langle S_1, s\rangle \Longrightarrow \langle S_1', s''\rangle$, alors, par [comp $_{SOS}$], on a $\langle S_1' S_2, s''\rangle \Longrightarrow^{k-1} s'$. Par hypothèse de récurrence, on a

  $$\langle S_1', s''\rangle \Longrightarrow^{k_1} s''' \text{ et } \langle S_2, s'''\rangle \Longrightarrow^{k_2} s'$$

  avec $k_1 + k_2 = k - 1$. Ce qui donne

  $$\langle S_1, s\rangle \Longrightarrow \langle S_1', s''\rangle \Longrightarrow^{k_1} s'''$$

  D'où

  $$\langle S_1, s\rangle \Longrightarrow^{k_1+1} s'''$$

  De plus, on a $k_1 + 1 + k_2 = (k-1) + 1 = k$. Donc on a le résultat souhaité

Conclusion : Pour tout $k \in \mathbb{N}$[3] si $\langle S_1; S_2, s\rangle \Longrightarrow^k s'$ alors il existe $k_1, k_2 \in \mathbb{N}$ tels que $k_1 + k_2 = k$ et

$$\langle S_1, s\rangle \Longrightarrow^{k_1} s'' \text{ et } \langle S_2, s''\rangle \Longrightarrow^{k_2} s'$$

□

Démonstration du théorème.
Étape 1 : *Montrons que* $\langle S, s\rangle \to s'$ *implique* $\langle S, s\rangle \Longrightarrow^* s'$
On va prouver cela par induction sur la structure de l'arbre de dérivation[4].
cas de base : La propriété est vraie car les règles pour skip et l'assignation sont les mêmes dans les deux sémantiques.
cas inductif :

→ cas *comp* : Par la règle [comp $_{NS}$], on a $\dfrac{\langle S_1, s\rangle \to s_1' \qquad \langle S_2, s_1'\rangle \to s'}{\langle S_1; S_2, s\rangle \to s'}$
Par hypothèse d'induction, on a $\langle S_1, s\rangle \Longrightarrow^{k_1} s_1'$ et $\langle S_2, s_1'\rangle \Longrightarrow^{k_2} s'$
On a donc

$$\langle S_1; S_2, s\rangle \Longrightarrow^{k_1} \langle S_2, s_1'\rangle \Longrightarrow^{k_2} s'$$

On a utilisé le lemme 1 pour la première flèche. On a donc ainsi

$$\langle S_1; S_2, s\rangle \Longrightarrow^* s'$$

2. car il faut au moins 2 pas pour transformer $S_1; S_2, s$ en un état s'
3. on peut dire que $k \in \mathbb{N}$ car la propriété sera vraie puisque l'hypothèse ne se produit jamais
4. attention, on ne peut pas le faire par induction sur la formule directement car pour le cas *while*, on a le même terme qui réapparaît .

→ cas *if* : Par la règle $[\text{if}\ {}^{tt}_{NS}]$, on a $\dfrac{\langle S_1, s\rangle \to s'}{\langle\ \text{if } b \text{ then } S_1 \text{ else } S_2, s\rangle \to s'}$ si $\mathcal{B}[\![b]\!]_s = tt$

Par hypothèse d'induction, on a

$$\langle S_1, s\rangle \Longrightarrow^k s'$$

Or la règle $[\text{if}\ {}^{tt}_{SOS}]$ nous donne $\langle\ \text{if } b \text{ then } S_1 \text{ else } S_2, s\rangle \Longrightarrow \langle S_1, s\rangle$, d'où

$$\langle\ \text{if } b \text{ then } S_1 \text{ else } S_2, s\rangle \Longrightarrow \langle S_1, s\rangle \Longrightarrow^* s'$$

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

Par hypothèse d'induction, on a

$$\langle S_2, s\rangle \Longrightarrow^k s'$$

Or la règle $[\text{if}\ {}^{ff}_{SOS}]$ nous donne $\langle\ \text{if } b \text{ then } S_1 \text{ else } S_2, s\rangle \Longrightarrow \langle S_2, s\rangle$, d'où

$$\langle\ \text{if } b \text{ then } S_1 \text{ else } S_2, s\rangle \Longrightarrow \langle S_2, s\rangle \Longrightarrow^* s'$$

→ cas *while* : Par la règle $[\text{while}\ {}^{tt}_{NS}]$, on a $\dfrac{\langle S, s\rangle \to s' \quad \langle \text{while } b \text{ do } S, s'\rangle \to s''}{\langle \text{while } b \text{ do } S, s\rangle \to s''}$ si $\mathcal{B}[\![b]\!]_s = tt$

Par hypothèse d'induction, on a

$$\langle S, s\rangle \Longrightarrow^* s' \text{ et } \langle \text{while } b \text{ do } S, s'\rangle \Longrightarrow^* s''$$

Or la règle du $[\text{while}\ {}_{SOS}]$ nous donne

$$\langle \text{while } b \text{ do } S, s\rangle \Longrightarrow \langle\ \text{if } b \text{ then } (S; \text{while } b \text{ do } S) \text{ else skip}, s\rangle$$

D'où

$$\begin{aligned}
\langle \text{while } b \text{ do } S, s\rangle &\Longrightarrow \langle\ \text{if } b \text{ then } (S; \text{while } b \text{ do } S) \text{ else skip}, s\rangle \\
&\Longrightarrow \langle S; \text{while } b \text{ do } S, s\rangle \quad \text{car } \mathcal{B}[\![b]\!]_s = tt \\
&\Longrightarrow^* \langle \text{while } b \text{ do } S, s'\rangle \quad \text{en appliquant le lemme 1 et en utilisant} \\
&\qquad\qquad\qquad\qquad \text{l'hypothèse } \langle S, s\rangle \Longrightarrow^* s' \\
&\Longrightarrow^* s'' \quad \text{en utilisant l'hypothèse } \langle \text{while } b \text{ do } S, s'\rangle \Longrightarrow^* s''
\end{aligned}$$

Par la règle $[\text{while}\ {}^{ff}_{NS}]$, on a $\langle \text{while } b \text{ do } S, s\rangle \to s$ si $\mathcal{B}[\![b]\!]_s = ff$

Or la règle du $[\text{while}\ {}_{SOS}]$ nous donne

$$\langle \text{while } b \text{ do } S, s\rangle \Longrightarrow \langle\ \text{if } b \text{ then } (S; \text{while } b \text{ do } S) \text{ else skip}, s\rangle$$

D'où

$$\begin{aligned}
\langle \text{while } b \text{ do } S, s\rangle &\Longrightarrow \langle\ \text{if } b \text{ then } (S; \text{while } b \text{ do } S) \text{ else skip}, s\rangle \\
&\Longrightarrow \langle \text{skip}, s\rangle \quad \text{car } \mathcal{B}[\![b]\!]_s = ff \\
&\Longrightarrow s \quad \text{en utilisant } [\text{skip}\ {}_{SOS}]
\end{aligned}$$

Ce qui conclut l'induction.

Étape 2 : *Montrons que* $\langle S, s\rangle \Longrightarrow^k s'$ *implique* $\langle S, s\rangle \to s'$

Par récurrence sur le nombre de pas.

Initialisation : Pour $k = 1$, les seuls cas possibles sont : skip et l'assignation qui ont les mêmes règles dans les deux sémantiques donc cela justifie l'initialisation.

Hérédité : Soit $k \in \mathbb{N}^*$, supposons que la propriété est vraie pour tous les rangs $l < k$.

→ cas *comp* : Si $\langle S_1; S_2, s\rangle \Longrightarrow^k s'$, alors en utilisant le lemme 2, on a

$$\langle S_1, s\rangle \Longrightarrow^{k_1} s'' \text{ et } \langle S_2,'' \rangle \Longrightarrow^{k_2} s' \text{ avec } k_1 + k_2 = k$$

Or $k_1, k_2 \geqslant 1$, donc $k_1, k_2 < k$, on peut donc appliquer l'hypothèse de récurrence pour obtenir

$$\langle S_1, s\rangle \to s'' \text{ et } \langle S_2, s''\rangle \to s'$$

Ainsi par la règle [comp $_{NS}$], on a

$$\langle S_1; S_2, s\rangle \to s'$$

→ cas *if* : On suppose que $\langle$ if b then S_1 else $S_2, s\rangle \Longrightarrow^k s'$.
Si $\mathcal{B}[\![b]\!]_s = tt$, alors, par la règle [if $^{tt}_{SOS}$], on a

$$\langle \text{ if } b \text{ then } S_1 \text{ else } S_2, s\rangle \Longrightarrow \langle S_1, s\rangle \Longrightarrow^{k-1} s'$$

Ainsi, par hypothèse de récurrence, on a $\langle S_1, s\rangle \to s'$.
Donc par la règle [if $^{tt}_{NS}$], on a

$$\langle \text{ if } b \text{ then } S_1 \text{ else } S_2, s\rangle \to s'$$

Si $\mathcal{B}[\![b]\!]_s = ff$, alors, par la règle [if $^{ff}_{SOS}$], on a

$$\langle \text{ if } b \text{ then } S_1 \text{ else } S_2, s\rangle \Longrightarrow \langle S_2, s\rangle \Longrightarrow^{k-1} s'$$

Ainsi, par hypothèse de récurrence, on a $\langle S_2, s\rangle \to s'$.
Donc par la règle [if $^{ff}_{NS}$], on a

$$\langle \text{ if } b \text{ then } S_1 \text{ else } S_2, s\rangle \to s'$$

→ cas *while* : On suppose que $\langle$while b do $S, s\rangle \Longrightarrow^k s'$.
Par la règle [while $_{SOS}$], on a

$$\langle \text{while } b \text{ do } S, s\rangle \Longrightarrow \langle \text{ if } b \text{ then } (S; \text{while } b \text{ do } S) \text{ else skip}, s\rangle$$

Ainsi, si $\mathcal{B}[\![b]\!]_s = tt$, on a

$$\langle \text{while } b \text{ do } S, s\rangle \Longrightarrow^2 \langle S; \text{while } b \text{ do } S, s\rangle \Longrightarrow^{k-2} s'$$

On a utilisé la règle [if $^{tt}_{SOS}$].
On utilise le lemme 2, pour obtenir

$$\langle S, s\rangle \Longrightarrow^{k_1} s'' \text{ et } \langle \text{while } b \text{ do } S, s''\rangle \Longrightarrow^{k_2} s' \text{ avec } k_1 + k_2 = k - 2 < k$$

Ainsi par hypothèse de récurrence, on a

$$\langle S, s\rangle \to s'' \text{ et } \langle \text{while } b \text{ do } S, s''\rangle \to s'$$

On peut conclure en utilisant la règle [while $^{tt}_{NS}$], pour avoir

$$\langle \text{while } b \text{ do } S, s\rangle \to s'$$

Et si $\mathcal{B}[\![b]\!]_s = ff$, on a

$$\langle \text{while } b \text{ do } S, s\rangle \Longrightarrow^2 \langle \text{skip}, s\rangle \Longrightarrow s$$

On a utilisé la règle [if $^{ff}_{SOS}$]. Ainsi, on avait en fait $s' = s$.
Or la règle [while $^{ff}_{NS}$] nous donne

$$\langle \text{while } b \text{ do } S, s\rangle \to s$$

Ce qui conclut le cas ff.

Conclusion : Ainsi pour tout $k \in \mathbb{N}^*$, si $\langle S, s\rangle \Longrightarrow^k s'$, alors $\langle S, s\rangle \to s'$.

□

Remarques :

On a aussi une équivalence entre la sémantique dénotationnelle et la sémantique opérationnelle à petits pas donc grâce au théorème que l'on vient de prouver, on a une équivalence entre la sémantique dénotationnelle et la sémantique opérationnelle à grands pas.

Astuces de l'agrégatif :

Il est mieux d'écrire les lemmes dans le plan et donc de s'affranchir de cette partie lors du développement.

Il faut bien comprendre qu'on ne peut pas faire la première induction sur la structure de l'instruction mais qu'il faut le faire sur la structure de l'arbre.

Questions possibles :

- Montrer que les instructions $\mathsf{while}\ b\ \mathsf{do}\ S$ et $\mathsf{if}\ b\ \mathsf{then}\ S; \mathsf{while}\ b\ \mathsf{do}\ S\ \mathsf{else\ skip}$ sont équivalentes pour la sémantique opérationnelle à grands pas.
- Écrire des règles de sémantique à grands et petits pas pour gérer les instructions $\mathsf{repeat}\ S\ \mathsf{until}\ b$, et montrer que l'équivalence est préservée.
- Montrer que le langage IMP muni de la sémantique opérationnelle à petits pas (ou à grands pas) est déterministe, *i.e.* il y a unicité de l'état sortant s' après exécution d'une instruction S à partir d'un état s.