

Cours MOdélisation, Vérification et Expérimentations
Exercices
Sémantique des langages de programmation
par Dominique Méry
7 mai 2025

Sémantique naturelle et sémantique SOS

Exercice 1

$$\begin{aligned} n &::= 0 \mid 1 \mid n0 \mid n1 \\ e &::= n \mid x \mid e1+e2 \mid e1-e2 \mid e1 \cdot e2 \\ b &::= tt \mid ff \mid e1 = e2 \mid e1 \neq e2 \mid e1 \leq e2 \mid e1 \geq e2 \mid e1 < e2 \mid e1 > e2 \mid \neg b \mid b1 \ \&\& \ b2 \\ S &::= x := e \mid skip \mid S1; S2 \mid (\text{if } b \text{ then } S_1 \text{ else } S_2 \text{ fi} \mid \text{while } b \text{ do } S \text{ od} \end{aligned}$$

Question 1.1 Définir une fonction sémantique pour la catégorie syntaxique des chaînes numériques NUM à valeurs dans \mathbb{Z} : $\mathcal{N} \in NUM \rightarrow \mathbb{Z}$.

Question 1.2 Evaluer les valeurs suivantes :

- $\mathcal{N}(11)$
- $\mathcal{N}(101)$
- $\mathcal{N}(0100)$

Question 1.3 Montrer que \mathcal{N} est bien définie pour toutes les expressions.

Exercice 2 On définit l'ensemble des états $States = Var \rightarrow \mathbb{Z}$ où Var est l'ensemble des variables.

Question 2.1 Une expression arithmétique $e \in Exp$ est évaluée dans un état ar la fonction sémantique $\mathcal{E} \in Exp \rightarrow (States \rightarrow \mathbb{Z})$. Définir \mathcal{E} par induction sur la syntaxe.

Question 2.2 Soit $s \in States$ tel que $s(x) = 2$ et $s(y) = 3$ où $x, y \in Var$ et $s \in States$. Evaluer les expressions suivantes en $s : x+y+101, x \cdot y$.

Question 2.3 Une expression logique $b \in Bexp$ est évaluée dans un état ar la fonction sémantique $\mathcal{B} \in Bexp \rightarrow (States \rightarrow \mathbb{B})$. Définir \mathcal{B} par induction sur la syntaxe.

Question 2.4 Soit $s \in States$ tel que $s(x) = 2$ et $s(y) = 3$ où $x, y \in Var$ et $s \in States$. Evaluer les expressions suivantes en $s : x = y, x \neq y, x \leq y, x < y \ \&\& \ x+6 \leq y$.

Question 2.5 On étend le langage des expressions logiques par les deux constructions $b1 \Rightarrow b2$ et $b1 \Leftrightarrow b2$. Ce langage est noté $Bexp1$. Montrer que pour tout expression $b \in Bexp1$, il existe une expression $b' \in Bexpt$ telle que $\mathcal{B}(b) = \mathcal{B}(b')$.

Exercice 3 Nous définissons deux opérations substitution et mise à jour. Ces deux opérations seront utilisées plus tard dans l'expression de la sémantique des instructions :

- la notation de substitution $e[x \mapsto e1]$ qui est la substitution de x par $e1$ dans e .
- la mise à jour pour un état s et on la note $s[x \mapsto v]$ qui est le nouvel état obtenu par mise à jour de la valeur de x pour s .

Question 3.1 Ecrire une définition inductive de $e[x \mapsto f]$.

Question 3.2 Définir la mise à jour pour un état s et on la note $s[x \mapsto v]$ qui est le nouvel état obtenu par mise à jour de la valeur de x pour s .

Question 3.3 Montrer que $s[x \mapsto v][y \mapsto w] = s[y \mapsto w][x \mapsto v]$ et que $s[x \mapsto v][\mapsto w] = s[x \mapsto v]$.

Question 3.4 Montrer que $\mathcal{E}(e[x \mapsto f])(s) = \mathcal{E}(e)(s[x \mapsto \mathcal{E}(f)(s)])$.

Question 3.5 Définir la substitution pour les expressions booléennes $b[x \mapsto e]$ où b est une expression booléenne de $BExp$ et e est une expression arithmétique de Exp .



Question 3.6

Montrer que $\mathcal{E}(b[x \mapsto e])(s) = \mathcal{E}(b)(s[x \mapsto \mathcal{E}(e)(s)])$.

Exercice 4

On rappelle les règles définissant la sémantique naturelle du langage de programmation \mathcal{PL}

Règles de définition selon la syntaxe

Axiome Ass $(x := e, s) \xrightarrow{\text{nat}} s[x \mapsto \mathcal{E}(e)(s)]$

Axiome Skip $(\text{skip}, s) \xrightarrow{\text{nat}} s$

Règle Comp Si $(S_1, s) \xrightarrow{\text{nat}} s'$ et $(S_2, s') \xrightarrow{\text{nat}} s''$, alors $(S_1; S_2, s) \xrightarrow{\text{nat}} s''$.

Règle Iftt Si $(S_1, s) \xrightarrow{\text{nat}} s'$ et $\mathcal{B}(b)(s) = \text{TRUE}$, alors $(\text{if } b \text{ then } S_1 \text{ else } S_2 \text{ fi}, s) \xrightarrow{\text{nat}} s'$.

Règle Ifff Si $(S_2, s) \xrightarrow{\text{nat}} s'$ et $\mathcal{B}(b)(s) = \text{FALSE}$, alors $(\text{if } b \text{ then } S_1 \text{ else } S_2 \text{ fi}, s) \xrightarrow{\text{nat}} s'$.

Règle Whilett J Si $(S, s) \xrightarrow{\text{nat}} s'$ et $(\text{while } b \text{ do } S \text{ od}, s') \xrightarrow{\text{nat}} s''$ et $\mathcal{B}(b)(s) = \text{TRUE}$, alors $(\text{while } b \text{ do } S \text{ od}, s) \xrightarrow{\text{nat}} s''$.

Règle Whieff J Si $\mathcal{B}(b)(s) = \text{FALSE}$, alors $(\text{while } b \text{ do } S \text{ od}, s) \xrightarrow{\text{nat}} s$.

Question 4.1 Soit s tel que $s(u) = 0$ et $s(v) = 1$.

— Evaluer $(u := 11; v := u + 100; u := u + v, s)$ en sémantique naturelle.

— Evaluer $(w := u; u := v; v := w, s)$ en sémantique naturelle.

Exercice 5 On dit que $S1$ est équivalent à $S2$ et on note $S1 \equiv S2$, si pour tous les états s et s' , $(S1, s) \xrightarrow{\text{nat}} s'$ si, et seulement si, $(S2, s) \xrightarrow{\text{nat}} s'$.

Question 5.1 Montrer que $\text{while } b \text{ do } S \text{ od} \equiv \text{if } b \text{ then } S; \text{while } b \text{ do } S \text{ od else skip fi}$

Question 5.2 Etendre la fonction sémantique pour l'instruction $\text{repeat } S \text{ until } b$.

Question 5.3 Montrer que $\text{repeat } S \text{ until } b \equiv S; \text{if } b \text{ then skip else repeat } S \text{ until } b \text{ fi}$

Exercice 6 On rappelle que $wp(X := E)(P(x)) = P[e(x)/x]$ et que $\{A(x)\}X := E\{B(x)\}$ est définie par $A \Rightarrow wp(X := E)(B)$. On peut assez naturellement appliquer cette définition pour

$\ell_1 : A(x)$
 $X := E(X)$
 $\ell_2 : B(x)$

Montrer la correction des triplets suivants et vérifier avec *Frama-C* en examinant les conditions de vérification engendrées :

— $\ell_1 : x = 10 \wedge y = z + x \wedge z = 2 \cdot x$
 $y := z + x$
 $\ell_2 : x = 10 \wedge y = x + 2 \cdot 10$

— On suppose que p est un nombre premier :

$$\begin{aligned} \ell_1 : x = 2^p \wedge y = 2^{p+1} \wedge x \cdot y = 2^{2 \cdot p+1} \\ x := y+x+2^x \\ \ell_2 : x = 5 \cdot 2^p \wedge y = 2^{p+1} \end{aligned}$$

—

$$\begin{aligned} \ell_1 : x = 1 \wedge y = 12 \\ x := 2 \cdot y \\ \ell_2 : x = 1 \wedge y = 24 \end{aligned}$$

—

$$\begin{aligned} \ell_1 : x = 11 \wedge y = 13 \\ z := x; x := y; y := z; \\ \ell_2 : x = 26/2 \wedge y = 33/3 \end{aligned}$$

— (1)

$$\begin{aligned} \ell_1 : x = 9 \wedge y = z+x \\ y := x+9 \\ \ell_2 : x = 9 \wedge y = x+9 \end{aligned}$$

— (2)

$$\begin{aligned} \ell_1 : x = 1 \wedge y = 3 \wedge x+y = 12 \\ x := y+x \\ \ell_2 : x = 567 \wedge y = 34 \end{aligned}$$