Sémantique opérationnelle de Pseudo-Pascal

François Pottier

Cette fiche récapitule la sémantique opérationnelle structurée à grands pas du langage Pseudo-Pascal.

1 Valeurs

Les **valeurs** manipulées au cours de l'exécution sont des booléens, des entiers représentables sur 32 bits, des **adresses** de tableau, ou bien la constante nil. Un tableau est représenté par son adresse en mémoire, et non pas directement par la suite de ses éléments. Cette distinction est importante : elle signifie par exemple qu'il est possible pour deux variables distinctes de type "array of τ" d'être **alias** l'une de l'autre, c'est-à-dire de contenir la même adresse, donc de pointer vers le même tableau. Elle signifie également que le contenu d'une variable de type tableau peut être stocké dans un registre, puisqu'il s'agit d'une adresse et non du tableau lui-même. Le tableau lui-même sera stocké dans le tas. La valeur nil est la valeur attribuée aux variables de type tableau non encore initialisées.

$$
\begin{array}{rcll}
v & ::= & & \textbf{valeurs} \\
& | & b & \text{constante booléenne} \\
& | & n & \text{constante entière} \\
& | & \ell & \text{adresse de tableau} \\
& | & \text{nil} & \text{adresse invalide}
\end{array}
$$

En Pseudo-Pascal, lorsqu'on déclare une variable, on donne son type τ, mais on ne spécifie pas quelle est sa valeur initiale. Dans certains langages, comme C, la valeur initiale de la variable est alors indéfinie : il est interdit d'accéder au contenu de cette variable avant de l'avoir initialisée. Dans d'autres, comme Java, la variable est considérée comme initialisée avec une valeur par défaut, laquelle dépend de τ. Pseudo-Pascal suit cette approche et définit une fonction qui à tout type τ associe une valeur par défaut de type τ :

$$
\begin{array}{rcl}
\text{default(boolean)} & = & \text{false} \\
\text{default(integer)} & = & 0 \\
\text{default(array of } \tau\text{)} & = & \text{nil}
\end{array}
$$

Le troisième cas explique pourquoi nous avons besoin de la valeur nil.

2 Opérateurs

La sémantique des opérateurs unaires est donnée par une fonction $[\![\cdot]\!]$ qui à tout opérateur uop associe une fonction partielle $[\![\text{uop}]\!]$ des valeurs dans les valeurs. Par exemple, la sémantique de l'opérateur de négation est donnée en définissant $[\![-]\!]$ comme la fonction qui à toute constante entière n associe la constante entière $-n$, normalisée à l'intervalle $[-2^{31} \ldots 2^{31} - 1]$. Il est important de noter que la fonction $[\![-]\!]$ est indéfinie en ce qui concerne les autres formes de valeurs : on ne peut l'appliquer ni à une constante booléenne, ni à une adresse, ni à la valeur nil. La sémantique d'un programme qui appliquerait l'opérateur de négation à un tableau,

par exemple, est donc indéfinie. Un tel programme est heureusement exclu par le système de types, puisque les types "integer" et "array of τ" sont incompatibles.

De même, la sémantique des opérateurs binaires est donnée par une fonction $[\![\cdot]\!]$ qui à tout opérateur bop associe une fonction partielle $[\![\text{bop}]\!]$ des paires de valeurs dans les valeurs. La définition exacte de cette fonction pour les quatre opérateurs arithmétiques et les six opérateurs de comparaison de Pseudo-Pascal est laissée en exercice au lecteur!

3 Jugements

Le contenu des variables globales est donné par un **environnement** G qui aux variables associe des valeurs. De même, le contenu des variables locales est donné par un environnement E. Enfin, le **tas** H associe aux adresses des suites finies de valeurs.

La sémantique de Pseudo-Pascal est définie par des **jugements**. L'évaluation d'une expression, d'une condition ou d'une instruction peut modifier les variables globales, les variables locales, ainsi que le tas. Les jugements qui décrivent l'évaluation mentionnent donc d'une part l'état initial G, H, E, d'autre part l'état final G', H', E'. Dans le cas des expressions et des conditions, ces jugements mentionnent également une valeur, qui représente le résultat de l'évaluation. Nos jugements sont donc de la forme

$$\begin{aligned} G, H, E/e &\longrightarrow G', H', E'/v \\ G, H, E/c &\longrightarrow G', H', E'/b \\ G, H, E/i &\longrightarrow G', H', E' \end{aligned}$$

Le premier de ces jugements se lit : "dans l'état décrit par G, H, E, l'évaluation de l'expression e mène à l'état décrit par G', H', E' et produit la valeur v". Les second et troisième jugements se lisent de façon similaire. Nous écrirons parfois S pour représenter le triplet G, H, E.

Ce style de sémantique est dit "à grands pas" car ces jugements relient directement expressions et résultats, sans mentionner explicitement les étapes intermédiaires du calcul.

La sémantique des expressions est donnée par la figure 1. Celle-ci contient un ensemble de règles qui définissent le jugement $G, H, E/e \longrightarrow G', H', E'/v$. De même, les jugements qui décrivent la sémantique des conditions et des instructions sont définis par les règles des figures 2 et 3. Par abus de notation, toutes ces règles sont implicitement paramétrées par une série de définitions de fonctions et procédures, qui sont consultées par les deux règles qui décrivent l'appel à une fonction ou procédure définie par l'utilisateur. Ces définitions de fonctions et procédures sont fixées lorsqu'on s'intéresse à un programme p précis.

La sémantique d'un programme p est définie par un dernier jugement, de la forme

$$p \longrightarrow$$

Le jugement $p \longrightarrow$ se lit "Le programme p s'exécute sans erreur et termine". Sa définition est donnée par la figure 4.

Constante
$$S/k \longrightarrow S/k$$

Variable locale
$$\frac{x \in dom(E)}{G,H,E/x \longrightarrow G,H,E/E(x)}$$

Variable globale
$$\frac{x \in dom(G) \setminus dom(E)}{G,H,E/x \longrightarrow G,H,E/G(x)}$$

Opérateur unaire
$$\frac{S/e \longrightarrow S'/v}{S/\mathrm{uop}\ e \longrightarrow S'/[\![\mathrm{uop}]\!](v)}$$

Opérateur binaire
$$\frac{S/e_1 \longrightarrow S'/v_1 \qquad S'/e_2 \longrightarrow S''/v_2}{S/e_1\ \mathrm{bop}\ e_2 \longrightarrow S''/[\![\mathrm{bop}]\!](v_1,v_2)}$$

Évaluation des arguments de fonction
$$\frac{\forall j \in \{1\ \ldots\ n\} \quad S_{j-1}/e_j \longrightarrow S_j/v_j \qquad S_n/\varphi(v_1\ \ldots\ v_n) \longrightarrow S'/v}{S_0/\varphi(e_1\ \ldots\ e_n) \longrightarrow S'/v}$$

Appel de readln
$$S/\mathrm{readln}() \longrightarrow S/n$$

Appel d'une fonction définie
$$\frac{\begin{array}{c} p \ni f(x_1:\tau_1\ \ldots\ x_n:\tau_n):\tau \quad \mathrm{var}\ x'_1:\tau'_1\ \ldots\ x'_q:\tau'_q \quad i \\ E' = (x_j \mapsto v_j)_{1\le j\le n} \cup (x'_j \mapsto \mathrm{default}(\tau'_j))_{1\le j\le q} \cup (f \mapsto \mathrm{default}(\tau)) \\ G,H,E'/i \longrightarrow G',H',E'' \qquad v = E''(f) \end{array}}{G,H,E/f(v_1\ \ldots\ v_n) \longrightarrow G',H',E/v}$$

Lecture dans un tableau
$$\frac{\begin{array}{c} S/e_1 \longrightarrow S'/\ell \qquad S'/e_2 \longrightarrow S''/n \\ S'' = G'',H'',E'' \qquad H''(\ell) = v_0 \ldots v_{p-1} \qquad 0 \le n < p \end{array}}{S/e_1[e_2] \longrightarrow S''/v_n}$$

Allocation d'un tableau
$$\frac{\begin{array}{c} S/e \longrightarrow G',H',E'/n \qquad n \ge 0 \\ \ell \# H' \qquad H'' = H'[\ell \mapsto \mathrm{default}(\tau)^n] \end{array}}{S/\mathrm{new\ array\ of}\ \tau\ [e] \longrightarrow G',H'',E'/\ell}$$

Figure 1 — Sémantique des expressions

Expression à valeur booléenne
$$\frac{S/e \longrightarrow S'/b}{S/e \longrightarrow S'/b}$$

Négation
$$\frac{S/c \longrightarrow S'/b}{S/\mathrm{not}\ c \longrightarrow S'/\neg b}$$

Conjonction (si)
$$\frac{S/c_1 \longrightarrow S'/\mathrm{false}}{S/c_1\ \mathrm{and}\ c_2 \longrightarrow S'/\mathrm{false}}$$

Conjonction (sinon)
$$\frac{S/c_1 \longrightarrow S'/\mathrm{true} \qquad S'/c_2 \longrightarrow S''/b}{S/c_1\ \mathrm{and}\ c_2 \longrightarrow S''/b}$$

Disjonction (si)
$$\frac{S/c_1 \longrightarrow S'/\mathrm{true}}{S/c_1\ \mathrm{or}\ c_2 \longrightarrow S'/\mathrm{true}}$$

Disjonction (sinon)
$$\frac{S/c_1 \longrightarrow S'/\mathrm{false} \qquad S'/c_2 \longrightarrow S''/b}{S/c_1\ \mathrm{or}\ c_2 \longrightarrow S''/b}$$

Figure 2 — Sémantique des conditions

Évaluation des arguments d'une procédure

$$\frac{\forall j \in \{1 \ldots n\} \quad S_{j-1}/e_j \longrightarrow S_j/v_j \qquad S_n/\varphi(v_1 \ldots v_n) \longrightarrow S'}{S_0/\varphi(e_1 \ldots e_n) \longrightarrow S'}$$

Appel de write

$$S/\mathsf{write}(n) \longrightarrow S$$

Appel de writeln

$$S/\mathsf{writeln}(n) \longrightarrow S$$

Appel d'une procédure définie

$$\frac{p \ni f(x_1 : \tau_1 \ \ldots \ x_n : \tau_n) \quad \mathsf{var}\ x'_1 : \tau'_1 \ \ldots \ x'_q : \tau'_q \quad i \qquad E' = (x_j \mapsto v_j)_{1 \le j \le n} \cup (x'_j \mapsto \mathrm{default}(\tau'_j))_{1 \le j \le q} \qquad G, H, E'/i \longrightarrow G', H', E''}{G, H, E/f(v_1 \ \ldots \ v_n) \longrightarrow G', H', E}$$

Affectation: variable locale

$$\frac{S/e \longrightarrow G', H', E'/v \qquad x \in \mathrm{dom}(E')}{S/x := e \longrightarrow G', H', E'[x \mapsto v]}$$

Affectation: variable globale

$$\frac{S/e \longrightarrow G', H', E'/v \qquad x \in \mathrm{dom}(G') \setminus \mathrm{dom}(E')}{S/x := e \longrightarrow G'[x \mapsto v], H', E'}$$

Écriture dans un tableau

$$\frac{S/e_1 \longrightarrow S'/\ell \qquad S'/e_2 \longrightarrow S''/n \qquad S''/e_3 \longrightarrow G''', H''', E'''/v \qquad H'''(\ell) = v_0 \ldots v_{p-1} \qquad 0 \le n < p}{S/e_1[e_2] := e_3 \longrightarrow G''', H'''[\ell \mapsto v_0 \ldots v_{n-1}\, v\, v_{n+1} \ldots v_{p-1}], E'''}$$

Séquence

$$\frac{\forall j \in \{1 \ldots n\} \quad S_{j-1}/i_j \longrightarrow S_j}{S_0/i_1 \ldots i_n \longrightarrow S_n}$$

Conditionnelle (si)

$$\frac{S/c \longrightarrow S'/\mathsf{true} \qquad S'/i_1 \longrightarrow S''}{S/\mathsf{if}\ c\ \mathsf{then}\ i_1\ \mathsf{else}\ i_2 \longrightarrow S''}$$

Conditionnelle (sinon)

$$\frac{S/c \longrightarrow S'/\mathsf{false} \qquad S'/i_2 \longrightarrow S''}{S/\mathsf{if}\ c\ \mathsf{then}\ i_1\ \mathsf{else}\ i_2 \longrightarrow S''}$$

Boucle (si)

$$\frac{S/c \longrightarrow S'/\mathsf{true} \qquad S'/i; \mathsf{while}\ c\ \mathsf{do}\ i \longrightarrow S''}{S/\mathsf{while}\ c\ \mathsf{do}\ i \longrightarrow S''}$$

Boucle (sinon)

$$\frac{S/c \longrightarrow S'/\mathsf{false}}{S/\mathsf{while}\ c\ \mathsf{do}\ i \longrightarrow S'}$$

Figure 3 — Sémantique des instructions

Programme

$$\frac{G = (x_j \mapsto \mathrm{default}(\tau_j))_{1 \le j \le n} \qquad G, \emptyset, \emptyset/i \longrightarrow S'}{\mathsf{var}\ x_1 : \tau_1 \ \ldots \ x_n : \tau_n \quad d \ \ldots \ d \quad i \longrightarrow}$$

Figure 4 — Sémantique des programmes