Mémoire structurée, logique de séparation

La syntaxe des commandes de IMP avec tas est définie par la grammaire

$$c ::= x \texttt{ := } a \mid \texttt{skip} \mid c_1 \texttt{ ; } c_2 \mid \texttt{if } b \texttt{ then } c_1 \texttt{ else } c_2 \mid \texttt{while } b \texttt{ do } c$$
$$x \texttt{ := } [a] \mid [x_1] \texttt{ := } a \mid x \texttt{ := alloc}(k) \mid \texttt{free}(a).$$

Les expressions arithmétiques et booléennes restent inchangées.

Ces quatre nouvelles constructions correspondent respectivement à

- l'accès à une case mémoire ;
- la modification d'une valeur d'une case mémoire ;
- l'allocation de k cases mémoires ;
- la libération de cases mémoires.

Remarque 1. C'est un langage impératif de bas niveau : on manipule de la mémoire directement. On ne s'autorise pas tout, cependant. On ne s'autorise pas, par exemple,

$$[x + i + 1] \texttt{ := } [t + i] + [t + i - 1],$$

mais on demande d'écrire

$$x \texttt{ := } [t + i] \texttt{ ; } y \texttt{ := } [t + i - 1] \texttt{ ; } [x + i + 1] \texttt{ := } x + y.$$

On raffine IMP : avant, les cases vivaient quelque part, mais on ne sait pas où, ce sont des registres ; maintenant, peut aussi allouer des

« blocs » de mémoire, et on peut donc parler de cases mémoires adjacentes.

L'allocation `alloc` est *dynamique*, similaire à `malloc` en C, où l'on alloue de la mémoire dans l'espace mémoire appelé *tas*.

1 Sémantique opérationnelle de IMP avec tas.

Définition 1 (États mémoire). Un état mémoire est la donnée de

- σ un *registre*, c-à-d un dictionnaire sur $(\mathrm{V}, \mathbb{Z})$;
- h un *tas*, c-à-d un dictionnaire sur $(\mathbb{N}, \mathbb{Z})$, c'est un gros tableau. [1]

On définit $h[k_1 \mapsto k_2]$ que si $k_1 \in \mathrm{dom}(h)$ et alors il vaut le dictionnaire où l'on assigne k_2 à k_1.

On définit $h_1 \uplus h_2$ que si $\mathrm{dom}(h_1) \cap \mathrm{dom}(h_2) = \emptyset$ et vaut l'union de dictionnaires h_1 et h_2.

On définit ainsi la sémantique dénotationnelle comme la relation $\Downarrow$ est une relation quinaire (*i.e.* avec 5 éléments) notée $c, \sigma, h \Downarrow \sigma', h'$ où c est une commande, h, h' sont deux tas, et σ, σ' sont deux registres.

$$\sigma' = \sigma[x \mapsto k] \quad \frac{a, \sigma \Downarrow k}{x := a, \sigma, h \Downarrow \sigma', h} \qquad \frac{c_1, \sigma, h \Downarrow \sigma', h' \qquad c_2, \sigma', h' \Downarrow \sigma'', h''}{c_1 \ ; \ c_2, \sigma, h \Downarrow \sigma'', h''}$$

$$\begin{matrix} k' = h(k) \\ \sigma' = \sigma[x \mapsto k'] \end{matrix} \quad \frac{a, \sigma \Downarrow k}{x := [a], \sigma, h \Downarrow \sigma' h} \qquad h' = h[k_1 \mapsto k_2]^{\,2} \quad \frac{a_1, \sigma \Downarrow k_1 \qquad a_2, \sigma \Downarrow k_2}{[a_1] := a_2, \sigma, h \Downarrow \sigma, h'}$$

$$\begin{matrix} \{k', \dots, k'+k-1\} \cap \mathrm{dom}(h) = \emptyset \\ k > 0 \\ h' = h \uplus \{k' \mapsto 0, \dots, k'+k-1 \mapsto 0\} \\ \sigma' = \sigma[x \mapsto k'] \end{matrix} \quad \frac{}{x := \texttt{alloc}(k), \sigma, h \Downarrow \sigma', h'}$$

$$h = h' \uplus \{k \mapsto k'\} \quad \frac{a, \sigma \Downarrow k}{\texttt{free}(a), \sigma, h \Downarrow \sigma, h'.}$$

1. On appelle parfois IMP avec tas, IMP avec tableau.

2 Logique de séparation.

Définition 2. On définit les assertions dans IMP avec tas comme l'enrichissement des assertions IMP avec les constructions `emp`, $\mapsto$ et $*$:

$$A ::= \cdots \mid \texttt{emp} \mid a_1 \mapsto a_2 \mid A_1 * A_2.$$

On enrichit ainsi la relation de satisfaction :

- $\sigma, h \models \texttt{emp}$ si et seulement si $h = \emptyset$;
- $\sigma, h \models a_1 \to a_2$ si et seulement si $a_1, \sigma \Downarrow k_1$, et $a_2, \sigma \Downarrow k_2$ et $h = [k_1 = k_2]$ (le tas est un singleton) ;
- $\sigma, h \models A_1 * A_2$ si et seulement si $h = h_1 \uplus h_2$ avec $\sigma, h_1 \models A_1$ et $\sigma, h_2 \models A_2$.

L'opérateur $*$ est appelé *conjonction séparante* : on découpe le tas en deux, chaque partie étant « observée » par une sous-assertion.

Note 1 (Notations).
- On note $a \mapsto \_$ pour $\exists i, a \mapsto i$.
- On note $a_1 \hookrightarrow a_2$ pour $(a_1 \mapsto a_2) * \texttt{true}$.
- On note $a \mapsto (a_1, \ldots, a_n)$ pour

$$(a \mapsto a_1) * (a + 1 \mapsto a_2) * \cdots * (a + n - 1 \mapsto a_n).$$

- On note $[A]$ pour $A \wedge \texttt{emp}$.

Ces notations signifient respectivement :

- La première formule dit qu'une case mémoire est allouée en a (ou plutôt que si a s'évalue en k, alors il y a une case mémoire allouée en k).
- La seconde formule dit que la case mémoire a_1 est allouée, et contient a_2 quelque part dans le tas.
- La troisième formule dit que les n cases mémoires suivant a contiennent respectivement a_1, puis a_2, *etc.*

▷ La quatrième formule permet de ne pas parler du tas. Intuitivement A « observe » uniquement la composante σ et alors A est une formule de la logique de Hoare.

3 Triplets de Hoare pour la logique de séparation.

On rappelle que $\{A\}\{A'\}$ est définir avec A, A' closes. La validité d'un triplet de Hoare en logique de séparation est défini par $\models \{A\}c\{A'\}$ si et seulement si dès que $\sigma, h \models A$ et $c, \sigma, h \Downarrow \sigma', h'$ alors $\sigma', h' \models A'$.

Parmi les règles d'inférences pour la logique de séparation, il y a la « *frame rule* » ou « *règle d'encadrement* » :

$$\frac{\vdash \{A\}c\{A'\}}{\vdash \{A * B\}c\{A' * B\}.} \quad \text{aucune variable } x \in \mathsf{vars}(B) \text{ n'est modifiée par } c$$

Elle permet de *zoomer*, et de ce concentrer sur un comportent local.

Les règles suivantes définissent une logique sur les commandes de IMP avec tas.

$$\frac{}{\vdash \{\texttt{emp}\}x \texttt{ := alloc}(k)\{x \mapsto (0, \ldots, 0)\}}$$

$$\frac{}{\vdash \{a \mapsto \_\}\texttt{free}(a)\{\texttt{emp}\}} \qquad \frac{}{\vdash \{a_1 \mapsto \_\}[a_1] \texttt{ := } a_2\{a_1 \mapsto a_2\}}$$

$$\frac{}{\vdash \{[x = x_0] * (a \mapsto x_1)\}x \texttt{ := } [a]\{[x = x_1] * (a[^{x_0}/_x] \mapsto x_1)\}}$$