

La especificación de *Imp*

0. Sintaxis.

Se da una sintaxis *abstracta* en BNF. Sin embargo, se incluyen algunos detalles de sintaxis concreta como, por ejemplo, el uso de la palabra clave of en la construcción *case* o el is en el *while*. La notación \bar{a} se usa para representar una *lista* de elementos de la categoría *a*.

Programas. $p ::=$

$\bar{x} := \bar{e}$	(Asignación múltiple)
$\text{local } \bar{x} \ p$	(Declaración de variables locales)
$p_1; p_2$	(Secuencia)
$\text{case } x \text{ of } \bar{b}$	(Selección)
$\text{while } x \text{ is } \bar{b}$	(Iteración)

Expresiones. $e ::=$

$c \ \bar{e}$	(Constructor/Constante aplicado a expresiones)
x	(Variable)

Rama. $b ::= c \ \bar{x} \rightarrow p$

1. Semántica operacional.

Valores. $v ::=$

$c \ \bar{v}$	(Constructor/Constante aplicado a valores)
null	(Ausencia de valor)

Memoria. $M ::= \overline{(x, v)}$ (Tabla no acotada, clave \rightarrow valor)

Se definen las siguientes funciones para operar sobre una Memoria:

- *Búsqueda.* Notaremos M_x a la búsqueda de la primer ocurrencia de la variable x en la memoria M .
- *Actualización.* Notaremos $M \leftarrow \overline{(x, v)}$ a actualizar la primer ocurrencia de cada variable de \bar{x} con los valores respectivos de \bar{v} en la memoria M . En caso de que una de las variables no esté presente en la memoria, la agrega.
- *Alta.* Notaremos $\bar{x} ++ M$ a definir a las variables de \bar{x} en la memoria M , asociándoles a cada una el valor *null*. Esta operación debe insertar a las nuevas variables al principio de la memoria, sin importar si estaban definidas o no anteriormente.
- *Bajas.* Notaremos $M - \bar{x}$ a sacar de M la primer ocurrencia de las variables de \bar{x} de la memoria M .

Evaluación de expresiones. El juicio $e \xRightarrow{M} v$, donde e es una expresión, M una memoria y v un valor, se leerá: “ v es el valor resultante de evaluar e bajo M ”. También se usa \Rightarrow para relacionar *listas* \bar{e} de expresiones con sus respectivos valores \bar{v} , en ese caso se escribe $\bar{e} \xRightarrow{M} \bar{v}$.

$$\text{cons} \frac{\bar{e} \xRightarrow{M} \bar{v}}{c \ \bar{e} \xRightarrow{M} c \ \bar{v}} \qquad \text{var} \frac{}{x \xRightarrow{M} M_x}$$

Ejecución de programas. El juicio $M \triangleright p \triangleright M'$, donde p es un programa y M y M' son memorias, se leerá: “ M' es la memoria resultante de ejecutar p utilizando M ”.

$$\text{ass} \frac{\bar{e} \xRightarrow{M} \bar{v}}{M \triangleright \bar{x} := \bar{e} \triangleright M \leftarrow \overline{(x, v)}} \qquad \text{loc} \frac{\bar{x} ++ M \triangleright p \triangleright M'}{M \triangleright \text{local } \bar{x} \ p \triangleright M' - \bar{x}}$$

$$\text{seq} \frac{M \triangleright p_1 \triangleright M' \quad M' \triangleright p_2 \triangleright M''}{M \triangleright p_1; p_2 \triangleright M''}$$

$$\text{case} \frac{x \xRightarrow{M} c \bar{v} \quad M \triangleright \underline{\text{local}} \bar{x} \{ \bar{x} := \bar{v}; p \} \triangleright M'}{M \triangleright \underline{\text{case}} x \text{ of } \bar{b} \triangleright M'} \left\{ \begin{array}{l} c \xrightarrow{\bar{b}} \bar{x} p \\ |\bar{x}| = |\bar{v}| \end{array} \right.$$

$$\text{while-i} \frac{x \xRightarrow{M} c \bar{v}}{M \triangleright \underline{\text{while}} x \text{ is } \bar{b} \triangleright M} c \xrightarrow{\bar{b}} \text{Nothing}$$

$$\text{while-ii} \frac{x \xRightarrow{M} c \bar{v} \quad M \triangleright \underline{\text{local}} \bar{x} \{ \bar{x} := \bar{v}; p \} \triangleright M' \quad M' \triangleright \underline{\text{while}} x \text{ is } \bar{b} \triangleright M''}{M \triangleright \underline{\text{while}} x \text{ is } \bar{b} \triangleright M''} \left\{ \begin{array}{l} c \xrightarrow{\bar{b}} \bar{x} p \\ |\bar{x}| = |\bar{v}| \end{array} \right.$$