

La especificación de χ

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*. La notación \bar{a} se usa para representar una *lista* de elementos de la categoría a .

$e ::=$	x	(variable)
	c	(constante o constructor)
	$\lambda \bar{x}.e$	(expresión lambda o abstracción funcional)
	$e \bar{e}$	(aplicación)
	case e of \bar{b}	(construcción case)
	rec $x.e$	(recursión)
$b ::= c \bar{x} e$		(rama)

En las expresiones lambda $\lambda \bar{x}.e$ (todas las apariciones de) las variables \bar{x} quedan *ligadas* o *locales*. Lo mismo con x en las recursiones **rec** $x.e$ y con las variables \bar{x} en las ramas de las construcciones *case*. Las (apariciones de) variables que no son ligadas en una expresión se dicen *libres* o *globales*. Llámase expresión *cerrada* a aquella que no contiene variables libres y *abierta* a la que sí.

1. Semántica operacional.

Valores.

$v ::=$	$c \bar{v}$	(árboles)
	$\lambda \bar{x}.e$	(expresión lambda)

Reglas de evaluación.

El *juicio* $e \Downarrow v$, donde e es una expresión *cerrada* y v un *valor*, se leerá: “ e evalúa o computa a v ”, en el sentido de que la evaluación de e da como resultado v . También se usa \Downarrow para relacionar *listas* \bar{e} de expresiones con sus respectivos valores \bar{v} . (En ese caso se escribe $\bar{e} \Downarrow \bar{v}$.)

En las reglas se usa la notación $e'[\bar{x} := \bar{e}]$ para la sustitución (simultánea) de las apariciones libres de las variables (identificadores) \bar{x} en e' por las expresiones \bar{e} . Esta operación se discute más abajo. También se usa $\#$ para denotar la operación que calcula el *largo* de una lista y $++$ para la de concatenación de dos listas.

Asimismo, $c \xrightarrow{\bar{b}} (\bar{x}, e')$ significa que el constructor (identificador) c tiene asociada, en las ramas \bar{b} , a la lista de parámetros \bar{x} junto con la expresión e' .

$$\begin{array}{c}
 \frac{}{c \Downarrow c[]} \\
 \\
 \frac{}{\lambda \bar{x}.e \Downarrow \lambda \bar{x}.e} \\
 \\
 \frac{e \Downarrow \lambda \bar{x}.e' \quad \bar{e} \Downarrow \bar{v} \quad e'[\bar{x} := \bar{v}] \Downarrow v}{e \bar{e} \Downarrow v} \# \bar{x} = \# \bar{e} \\
 \\
 \frac{e \Downarrow c \bar{v}_1 \quad \bar{e} \Downarrow \bar{v}_2}{e \bar{e} \Downarrow c (\bar{v}_1 ++ \bar{v}_2)} \\
 \\
 \frac{e \Downarrow c \bar{v} \quad c \xrightarrow{\bar{b}} (\bar{x}, e') \quad e'[\bar{x} := \bar{v}] \Downarrow v}{\text{case } e \text{ of } \bar{b} \Downarrow v} \# \bar{x} = \# \bar{v} \\
 \\
 \frac{e[x := (\text{rec } x. e)] \Downarrow v}{\text{rec } x. e \Downarrow v}
 \end{array}$$

Sustitución.

Una sustitución (múltiple) es una *tabla* que asocia identificadores con expresiones, en una forma representable como:

$$[x_1 := e_1, x_2 := e_2, \dots, x_n := e_n].$$

Una sustitución de éstas está pensada para *efectuarse* sobre expresiones *abiertas*. Lo que hace entonces es precisamente procesar cada aparición *libre* de una variable x en la expresión en cuestión de la siguiente forma: Si x es alguna de las x_i , la sustituye por su expresión asociada e_i y, en caso contrario (es decir, si x no está en la tabla) *no* la afecta.

Serán necesarias las siguientes operaciones sobre estas tablas:

- *Búsqueda.* Si σ es una sustitución y x una variable, entonces $\text{lkup } \sigma \ x$ es la expresión asociada a x en σ , o x misma si ésta *no* se encuentra en σ .
- *Bajas.* Si σ es una sustitución y \bar{x} una lista de variables, entonces $\sigma - \bar{x}$ es la sustitución que resulta de *borrar* de la tabla σ todas las entradas correspondientes a las variables en \bar{x} .

El *efecto* de una sustitución σ sobre una expresión e será escrito $e\sigma$ y tiene la precedencia o prioridad más alta entre todas las operaciones junto a las cuales aparece. Se define por recursión en e :

$$\begin{aligned} x\sigma &= \text{lkup } \sigma \ x \\ c\sigma &= c \\ (\lambda \bar{x}. e)\sigma &= \lambda \bar{x}. e(\sigma - \bar{x}) \\ (e \bar{e})\sigma &= e\sigma \bar{e}\sigma \\ (\text{case } e \text{ of } \bar{b})\sigma &= \text{case } e\sigma \text{ of } \bar{b}\sigma \\ (\text{rec } x. e)\sigma &= \text{rec } x. e(\sigma - x) \end{aligned}$$

Resta solamente estipular el efecto de una sustitución sobre una rama de *case*:

$$(c \ \bar{x} \ e)\sigma = c \ \bar{x} \ e(\sigma - \bar{x})$$

Notas finales. No se especifica el resultado de la operación lkup para una variable que aparece más de una vez en la sustitución en cuestión.

Similarmente, no se especifica el resultado de $c \xrightarrow{\bar{b}} (\bar{x}, e')$ para el caso de que el identificador c aparezca más de una vez en las ramas \bar{b} ni el resultado de aplicar una expresión lambda en cuya lista de parámetros aparezcan nombres repetidos.

La notación de *efecto* de una sustitución está sobrecargada al ser utilizada para ramas y listas de ramas además de para expresiones y listas de expresiones.