

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)
	$ \quad k \bar{e}$	(constante o constructor aplicado a sus argumentos)
	$ \quad \lambda x.e$	(expresión lambda o abstracción funcional)
	$ \quad e e$	(aplicación)
	$ \quad \text{case } e \text{ of } \bar{b}$	(construcción case)
	$ \quad \text{rec } x.e$	(recursión)
$b ::=$	$k \rightarrow \bar{x} e$	(rama)

En las expresiones lambda $\lambda x.e$ la variable x queda *ligada* o *local*. Lo mismo con x en las recursiones $\text{rec } x.e$ 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.

Para facilitar la lectura y comprensión de las reglas de evaluación, definiremos un conjunto de estructuras y operaciones auxiliares indicando cual será su notación y su efecto.

Valores. Los utilizaremos para representar aquellas expresiones que no pueden o no necesitan ser reducidas. Los definimos como:

$v ::=$	$k \bar{v}$	(constantes o constructores aplicados a otros valores)
	$ \quad \lambda x.e$	(expresión lambda)

Fórmula canónica débil. Los utilizaremos para representar aquellas expresiones que no necesitan ser reducidas, para seguir evaluando una expresión. Los definimos como:

$w ::=$	$k \bar{e}$	(constantes o constructores aplicados a expresiones)
	$ \quad \lambda x.e$	(expresión lambda)

Cargar expresión. Dada una secuencia o lista de expresiones, y un expresión adicional, *cargar expresión* significa insertarlo al final de la secuencia o lista. Lo definimos con la siguiente notación: $\bar{e} <+ e$

Buscar rama. Dada una lista de ramas y un constructor, *buscar rama* se encargará de obtener las variables y la expresión asociadas a ese constructor en la lista de ramas. En caso de que haya ocurrencias repetidas del mismo constructor, se obtendrá el primero que encuentre. Utilizaremos para esta operación la siguiente notación: $\bar{b} \xrightarrow{k} (\bar{x}, e)$

Sustitución (Estructura). Una sustitución múltiple la notaremos como σ y es una *tabla* que asocia identificadores con expresiones, en una forma representable como: $[x_1, x_2, \dots, x_n := e_1, e_2, \dots, e_n]$.

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 σ_x es la expresión asociada a x en σ , o la misma x si ésta *no* se encuentra definida en σ , asimismo si la variable x se encuentra definida múltiples veces en σ , se devolverá la primer ocurrencia de la variable en la sustitución.
- Bajas. Si σ es una sustitución y \bar{x} una lista de variables, entonces $\sigma - \bar{x}$ es la sustitución (σ') que resulta de *borrar* de la tabla σ todas las entradas correspondientes a las variables en \bar{x} .

Sustitución (Efecto) o Efecto de la sustitución. 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 acepta adicionalmente la notación del efecto de una sustitución sobrecargada para ramas, listas de ramas y lista de expresiones. Se define por recursión en e :

$$\begin{aligned}
(x)\sigma &= \sigma_x \\
(k \bar{e})\sigma &= k (\bar{e}\sigma) \\
(\lambda x.e)\sigma &= \lambda x.e(\sigma - [x]) \\
(e e')\sigma &= e\sigma e'\sigma \\
(\text{case } e \text{ of } \bar{b})\sigma &= \text{case } e\sigma \text{ of } \bar{b}\sigma \\
(\text{rec } x. e)\sigma &= \text{rec } x. (e(\sigma - [x]))
\end{aligned}$$

Resta solamente estipular el efecto de una sustitución sobre una rama de $\text{case}(b)$:

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

Función semántica (Reglas de evaluación parcial).

El juicio $e \Downarrow w$, donde e es una expresión *cerrada* y w una *fórmula canónica débil*, se leerá: “ e evalúa o computa debilmente a w ”, en el sentido de que la evaluación de e da como resultado w .

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} . Este es el *efecto de la sustitución* discutido anteriormente. También se usa $\#$ para denotar la operación que calcula el *largo* de una secuencia o lista.

$$\begin{aligned}
&\text{cons} \frac{}{k \bar{e} \Downarrow k \bar{e}} \\
&\text{abs} \frac{}{\lambda x.e \Downarrow \lambda x.e} \\
&\text{ap-}\beta \frac{e \Downarrow \lambda x.e'' \quad e''[x := e'] \Downarrow w}{e e' \Downarrow w} \\
&\text{ap-cons} \frac{e \Downarrow k \bar{e}}{e e' \Downarrow k (\bar{e} \leftarrow e')} \\
&\text{case} \frac{e \Downarrow k \bar{e} \quad e'[\bar{x} := \bar{e}] \Downarrow w}{\text{case } e \text{ of } \bar{b} \Downarrow w} \left\{ \begin{array}{l} \bar{b} \xrightarrow{k} (\bar{x}, e') \\ \# \bar{x} = \# \bar{e} \end{array} \right. \\
&\frac{e[x := \text{rec } x. e] \Downarrow w}{\text{rec } x. e \Downarrow w}
\end{aligned}$$

Función semántica (Reglas de evaluación completa).

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 $\bar{e} \Downarrow \bar{v}$.)