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 $\overline{a}$ se usa para representar una *lista* de elementos de la categoría a.

$$
\begin{array}{lll}
e ::= & x & \text{(variable)} \\
 \mid & k\ \overline{e} & \text{(constante o constructor aplicado a sus argumentos)} \\
 \mid & \lambda x.e & \text{(expresión lambda o abstracción funcional)} \\
 \mid & e\ e & \text{(aplicación)} \\
 \mid & case\ e\ of\ \overline{b} & \text{(construcción case)} \\
 \mid & rec\ x.e & \text{(recursión)} \\
b ::= k \to \overline{x}\ e & & \text{(rama)}
\end{array}
$$

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

$$
\begin{array}{lll}
v ::= & k\ \overline{v} & \text{(constantes o constructores aplicados a otros valores)} \\
 \mid & \lambda x.e & \text{(expresión lambda)}
\end{array}
$$

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

$$
\begin{array}{lll}
w ::= & k\ \overline{e} & \text{(constantes o constructores aplicados a expresiones)} \\
 \mid & \lambda x.e & \text{(expresión lambda)}
\end{array}
$$

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: $\overline{e} \lessdot\!\!+ e$

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: $\overline{b} \xrightarrow{k} (\overline{x}, e)$

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

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, asímismo 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 $\overline{x}$ una lista de variables, entonces σ-$\overline{x}$ es la sustitución (σ') que resulta de *borrar* de la tabla σ todas las entradas correspondientes a las variables en $\overline{x}$.

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:

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

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

$$(k \to \overline{x}\ e)\sigma \quad = \quad \text{k} \to \overline{x}\ e(\sigma\text{-}\overline{x})$$

Función semántica (Reglas de evaluación pacial).

El *juicio* $e \Downarrow w$, donde e es una expresión *cerrada* y w una *fórma 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 $e[\overline{x} := \overline{e}]$ para la sustitución (simultánea) de las apariciones libres de las variables (identificadores) $\overline{x}$ en e por las expresiones $\overline{e}$. 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.

$$\text{cons}\ \frac{}{k\ \overline{e} \Downarrow k\ \overline{e}}$$

$$\text{abs}\ \frac{}{\lambda x.e \Downarrow \lambda x.e}$$

$$\text{ap-}\beta\ \frac{e \Downarrow \lambda x.e'' \qquad e''[x := e'] \Downarrow w}{e\ e' \Downarrow w}$$

$$\text{ap-cons}\ \frac{e \Downarrow k\ \overline{e}}{e\ e' \Downarrow k\ (\overline{e} \mathrel{<\!\!+} e')}$$

$$\text{case}\ \frac{e \Downarrow k\ \overline{e} \qquad e'[\overline{x} := \overline{e}] \Downarrow w}{case\ e\ of\ \overline{b} \Downarrow w} \begin{cases} \overline{b} \xrightarrow{k} (\overline{x}, e') \\ \#\overline{x} = \#\overline{e} \end{cases}$$

$$\text{rec}\ \frac{e[x := rec\ x.\ e] \Downarrow w}{rec\ x.\ e \Downarrow w}$$

Función semántica (Reglas de evaluación completa).

El *juicio* $e \underline{\Downarrow} v$, 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 $\underline{\Downarrow}$ para relacionar *listas* $\overline{e}$ de expresiones con sus respectivos valores $\overline{v}$. (En ese caso se escribe $\overline{e} \underline{\Downarrow} \overline{v}$.)