La especificación de χ

0. Sintaxis.
Se da una sintaxis *abstracta* en BNF. Como de costumbre, $\overline{a}$ representa una *lista* de elementos de la categoría a. Variables x y constructores c son *strings*.

$$
\begin{array}{rcll}
e ::= & & x & \text{(variable)} \\
& | & c & \text{(constructor, constante)} \\
& | & \lambda x\, e & \text{(expresión lambda o abstracción funcional)} \\
& | & e_1\ e_2 & \text{(aplicación)} \\
& | & \texttt{case}\ e\, \overline{r} & \text{(construcción case)} \\
& | & \texttt{rec}\ x\, e & \text{(recursión)}
\end{array}
$$

$r ::= c\ \overline{x}\ e$ (rama)

En las expresiones lambda $\lambda x\, e$, así como en las ramas $c\ \overline{x}\ e$, (todas las apariciones de) las variables $\overline{x}$ quedan *ligadas* o *locales*. Las (apariciones de) variables que no son ligadas en una expresión o rama se dicen *libres* o *globales*. Llámase como de costumbre expresión o rama *cerrada* a aquella que no contiene variables libres y *abierta* a la que sí. La semántica considerará inválida a toda expresión abierta.

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

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

que también se abreviará

$[\overline{x} := \overline{e}]$.

Una sustitución de éstas está pensada para *efectuarse* sobre expresiones o ramas *abiertas*. Lo que hace entonces es precisamente procesar cada aparición *libre* de una variable x en la expresión o rama 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. En el caso de χ, las expresiones e_i que forman parte de una sustitución son todas *cerradas*.
Serán necesarias las siguientes operaciones sobre estas tablas:

- *Búsqueda.* Si σ es una sustitución y x una variable, entonces $\mathsf{lkup}\ x\ \sigma$ 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 $\overline{x}$ una lista de variables, entonces $\sigma - \overline{x}$ es la sustitución que resulta de *borrar* de la tabla σ todas las entradas correspondientes a las variables en $\overline{x}$.

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. Se usa $\overline{a}\sigma$ para la distribución (*map*) de la sustitución σ sobre la lista $\overline{a}$:

$$
\begin{array}{lcl}
x\sigma & = & \mathsf{lkup}\ x\ \sigma \\
c\sigma & = & c \\
(\lambda x\, e)\sigma & = & \lambda x\, e(\sigma - x) \\
(e_1\, e_2)\sigma & = & e_1\sigma\ e_2\sigma \\
(\texttt{case}\ e\, \overline{r})\sigma & = & \texttt{case}\ e\sigma\, \overline{r}\sigma \\
(\texttt{rec}\ x\, e)\sigma & = & \texttt{rec}\ x\ e(\sigma - x)
\end{array}
$$

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

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

2. Semántica operacional.
<u>*Valores.*</u>

$v ::= t \mid \lambda x\, e$

$t ::= c \mid t\, v$

Es decir, los valores finales son, o bien *árboles* o bien *abstracciones funcionales*. Notar que cada árbol es una cadena de aplicaciones $(\ldots((c\,v_1)\,v_2)\ldots\,v_n)$, donde c es un *constructor* y los v_i son *valores*. En las reglas aquí abajo nos referimos a la forma de un tal árbol t escribiendo $t \equiv c\ \overline{v}$ pero obsérvese que esta última estructura no coincide con la de la sintaxis abstracta de t y debe ser obtenida por una función conveniente.

Reglas de reducción.

$e \to e'$, donde e y e' son expresiones cerradas, se leerá: "e reduce (en un paso) a e'".

En las reglas se usa la notación $|\_|$ para denotar la operación que calcula el *largo* de una lista. Asimismo, $c \overset{\overline{r}}{\mapsto} \overline{x}\ e$ significa que el constructor (identificador) c tiene asociada, en las ramas $\overline{r}$, a la expresión e con las variables (locales) $\overline{x}$.

$$\beta\colon \frac{}{(\lambda x\, e)\ v \to e[x := v]}$$

$$\mu\colon \frac{e_1 \to e_1'}{e_1\ e_2 \to e_1'\ e_2}$$

$$\nu\colon \frac{e \to e'}{v\ e \to v\ e'}$$

$$\gamma\colon \frac{}{\texttt{case}\ t\ \overline{b} \to e[\overline{x} := \overline{v}]} \quad \begin{cases} t \equiv c\,\overline{v} \\ c \overset{\overline{b}}{\mapsto} (\overline{x}, e) \\ |\overline{x}| = |\overline{v}| \end{cases}$$

$$\delta\colon \frac{e \to e'}{\texttt{case}\ e\ \overline{b} \to \texttt{case}\ e'\ \overline{b}}$$

$$\rho \ \frac{}{\texttt{rec}\ x\ e \to e[x := \texttt{rec}\ x\ e]}$$

$\twoheadrightarrow$ es la clausura reflexiva y transitiva de $\to$ (es decir, su aplicación sucesiva 0 o más veces).

El *valor* de una expresión e, si existe, es aquel v tal que $e \twoheadrightarrow v$.

3. Notas finales. En la definición del efecto de una sustitución 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 $c \overset{\overline{r}}{\mapsto} \overline{x}\ e$ para el caso de que el identificador c aparezca más de una vez en las ramas $\overline{r}$ ni el resultado de aplicar una expresión lambda en cuya lista de parámetros aparezcan nombres repetidos.

De la manera que ya quedó explicada, 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.