Lenguaje LET

Este es un lenguaje simple (pero no tan simple como el aritmético) que permite la definición y referencia de variables locales.

Sintaxis

La sintaxis concreta del lenguaje se especifica con una gramática libre de contexto, las categorías sintácticas o no-terminales del lenguaje se enfatizan en itálicas, mientras que los terminales se escriben sin énfasis, o bien, en negritas para estilizar palabras reservadas. Una producción de la forma $X \to \alpha$ establece que se puede derivar la cadena de terminales y no-terminales α a partir de la categoría sintáctica X.

La sintaxis abstracta del lenguaje se especifica a partir de la estructura de las posibles expresiones, cada una de estas se denota como una lista cuyo primer elemento es el tipo de expresión, seguido de los nombres asociados a las subestructuras inmediatas, estos nombres suelen indicar el tipo particular de la subestructura. Una estructura de la forma $(x\ y_1\ y_2\ \ldots\ y_n)$ representa la existencia de: (a) un constructor x que toma n argumentos y produce un árbol de sintaxis de este tipo; (b) un predicado $x?$ que toma un argumento y determina si es de tipo x; (c) n selectores x-y_i que toman un argumento de tipo x y regresan su subestructura correspondiente a y_i.

Sintaxis concreta		**Sintaxis abstracta**
Program →	*Expression*	(a-program *exp1*)
Expression →	*Integer*	(const-exp *num*)
Expression →	**-**(*Expression* , *Expression*)	(diff-exp *exp1 exp2*)
Expression →	**zero?**(*Expression*)	(zero?-exp *exp1*)
Expression →	**if** *Expression* **then** *Expression* **else** *Expression*	(if-exp *exp1 exp2 exp3*)
Expression →	*Identifier*	(var-exp *var*)
Expression →	**let** *Identifier* = *Expression* **in** *Expression*	(let-exp *var exp1 body*)

Semántica

La interpretación de expresiones del lenguaje de acuerdo a su sintaxis abstracta se especifica utilizando semántica operacional de pasos grandes (también llamada semántica natural) que consiste de afirmaciones de la forma (value-of $e\ \rho) = v$ donde e es una expresión, ρ un entorno y v un valor expresado.

Existe un entorno vacío ρ_0, un mecanismo para extender un entorno ρ con la vinculación de una variable x a un valor v denotado como $[x : v]\rho$ y un mecanismo para aplicar entornos a variables $\rho(x)$. En particular, los entornos se modelan como funciones de variables a valores, con ρ_0 indefinida para cualquier variable.

En este lenguaje los valores expresados $ExpVal$ (resultados de expresiones) y los valores denotados $DenVal$ (vinculados a variables) son los mismos y corresponden a $Int + Bool$. Los procedimientos expval→num y expval→bool toman valores expresados y regresan los valores codificados en el lenguaje de implementación.

Reglas

(value-of-program (a-program *exp1*)) = (value-of *exp1* ρ_0)

(value-of (const-exp n) ρ) = (num-val n)

(value-of (var-exp *var*) ρ) = ρ(*var*)

(value-of (diff-exp *exp1 exp2*) ρ)
= (num-val (- (expval→num (value-of *exp1* ρ))
(expval→num (value-of *exp2* ρ))))

(value-of (zero?-exp *exp1*) ρ)
= (**let** ([*val1* (value-of *exp1* ρ)])
(bool-val (= 0 (expval→num *val1*))))

(value-of (if-exp *exp1 exp2 exp3*) ρ)
= (**if** (expval→bool (value-of *exp1* ρ))
(value-of *exp2* ρ)
(value-of *exp3* ρ))

(value-of (let-exp *var exp1 body*) ρ)
= (**let** ([*val1* (value-of *exp1* ρ)])
(value-of *body* $[var : val1]\rho$))

Reglas breves

$\mathcal{P}(exp1) = \mathcal{E}(exp1, \rho_0)$

$\mathcal{E}(n, \rho) = n$

$\mathcal{E}(x, \rho) = \rho(x)$

$\mathcal{E}(\text{-}(exp1,\ exp2), \rho) = \lceil \lfloor \mathcal{E}(exp1, \rho) \rfloor - \lfloor \mathcal{E}(exp2, \rho) \rfloor \rceil$

$$\mathcal{E}(\textbf{zero?}(exp1), \rho) = \begin{cases} \top & \text{si } \lfloor \mathcal{E}(exp1, \rho) \rfloor = 0 \\ \bot & \text{si } \lfloor \mathcal{E}(exp1, \rho) \rfloor \neq 0 \end{cases}$$

$$\mathcal{E}(\textbf{if } exp1 \textbf{ then } exp2 \textbf{ else } exp3, \rho) = \begin{cases} \mathcal{E}(exp2, \rho) & \text{si } \mathcal{E}(exp1, \rho) = \top \\ \mathcal{E}(exp3, \rho) & \text{si } \mathcal{E}(exp1, \rho) = \bot \end{cases}$$

$$\frac{\mathcal{E}(exp1, \rho) = val1}{\mathcal{E}(\textbf{let } x = exp1 \textbf{ in } body, \rho) = \mathcal{E}(body, [x : val1]\rho)}$$