

Lenguajes de Programación

MINILISP 05: Evaluación Perezosa

Manuel Soto Romero

4 de noviembre de 2024
Facultad de Ciencias UNAM

Sintaxis Concreta

```
<expr> ::= <id>
          | <num>
          | <bool>
          | (<op> <expr> {<expr>})
          | (let ([<id> <expr>] {[<id> <expr>]}) <expr>)
          | (let* ([<id> <expr>] {[<id> <expr>]}) <expr>)
          | (if <expr> <expr> <expr>)
          | (cond ([<expr> <expr>]) [else <expr>])
          | (lambda (<id> {<id>}) <expr>)
          | (<expr> <expr> {<expr>})

<num> ::= ... 2.5 | -1 | 0 | 1 | 18.35 ...

<bool> ::= #t | #f

<op> ::= + | - | * | / | add1 | sub1 | sqrt | expt
        | < | > | = | not | or | and

<id> ::= a | b | foo | ...
```

Observación: Por simplicidad se simplifican las reglas de producción de los números e identificadores. Sin embargo, es importante dejar en claro que se está suponiendo un diseño que incluye números reales. Las reglas para los identificadores serán explicadas en las clases de laboratorio.

Sintaxis Abstracta Endulzada

$$Ops = \{+, -, *, /, \text{add1}, \text{sub1}, \text{sqrt}, \text{expt}, <, >, =, \text{not}, \text{or}, \text{and}\}$$

$$\text{Binding} \subseteq \text{String} \times \text{SASA}$$

$$\frac{i : \text{String}}{\text{IdS}(i) \text{ SASA}}$$

$$\frac{n \in \mathbb{Z}}{\text{NumS}(n) \text{ SASA}}$$

$$\frac{b \in \mathbb{B}}{\text{Boolean}(b) \text{ SASA}}$$

$$\frac{f \in Ops \quad args \text{ [SASA]}}{Op(f,args) \text{ SASA}}$$

$$\frac{b : [\text{Binding}] \quad c \text{ SASA}}{Let1(b,c) \text{ SASA}}$$

$$\frac{p : [\text{String}] \quad c \text{ SASA}}{Fun(p,c) \text{ SASA}}$$

$$\frac{c \text{ SASA} \quad t \text{ SASA} \quad e \text{ SASA}}{If(c,t,e) \text{ SASA}}$$

$$\frac{f \text{ SASA} \quad a : [\text{SASA}]}{App(f,a) \text{ SASA}}$$

$$\frac{b : [\text{Binding}] \quad c \text{ SASA}}{Let(b,c) \text{ SASA}}$$

$$\frac{cs : [(\text{SASA},\text{SASA})] \quad e \text{ SASA}}{Cond(cs,e) \text{ SASA}}$$

Sintaxis Abstracta Desendulzada

$$U = \{\text{add1}, \text{sub1}, \text{sqrt}, \text{not}\}$$

$$B = \{+, -, *, /, \text{expt}, <, >, =, \text{or}, \text{and}\}$$

$$\frac{i : \text{String}}{Id(i) \text{ ASA}}$$

$$\frac{f \in U \quad arg \text{ ASA}}{Unop(f,arg) \text{ ASA}}$$

$$\frac{p : \text{String} \quad c \text{ ASA}}{Fun(p,c) \text{ ASA}}$$

$$\frac{n \in \mathbb{Z}}{Num(n) \text{ ASA}}$$

$$\frac{f \in B \quad i \text{ ASA} \quad d \text{ ASA}}{Binop(f,i,d) \text{ ASA}}$$

$$\frac{f \text{ ASA} \quad a \text{ ASA}}{App(f,a) \text{ ASA}}$$

$$\frac{b \in \mathbb{B}}{Boolean(b) \text{ ASA}}$$

$$\frac{c \text{ ASA} \quad t \text{ ASA} \quad e \text{ ASA}}{If(c,t,e) \text{ ASA}}$$

Valores Finales

$$\frac{n \in \mathbb{Z}}{NumV(n) \text{ Value}}$$

$$\frac{p : \text{String} \quad c \text{ Value} \quad \varepsilon \text{ Env}}{< p, c, \varepsilon > \text{ Value}}$$

$$\frac{b \in \mathbb{B}}{BooleanV(b) \text{ Value}}$$

$$\frac{e : \text{ASA} \quad \varepsilon \text{ Env}}{< e, \varepsilon > \text{ Value}}$$

Semántica Natural

Identificadores

El valor de un identificador se debe buscar en el ambiente de evaluación. En caso de que el ambiente no encuentre dicho identificador, se devuelve un error pues son variables libres.

$$\overline{Id(i), \varepsilon \Rightarrow \varepsilon(i)}$$

Números

Los números se reducen a sí mismos.

$$\overline{Num(n), \varepsilon \Rightarrow NumV(n)}$$

Booleanos

Los booleanos se reducen a sí mismos.

$$\overline{Boolean(b), \varepsilon \Rightarrow BooleanV(b)}$$

Operaciones unarias y binarias

Las operaciones unarias y binarias se interpretan mediante la operación correspondiente. El comportamiento de las operaciones es el habitual.

$$\frac{elige(f) = g \quad arg, \varepsilon \Rightarrow a \quad strict(v) = v' \quad g(v') = v''}{Unop(f, arg), \varepsilon \Rightarrow v''}$$

$$\frac{elige(f) = g \quad i, \varepsilon \Rightarrow i' \quad d, \varepsilon \Rightarrow d' \quad strict(i') = i'' \quad strict(d') = d'' \quad g(i'', d'') = v}{Binop(f, i, d), \varepsilon \Rightarrow v}$$

Donde:

- *elige* es una función que transforma el símbolo de operación en sintaxis abstracta en la operación correspondiente en el lenguaje anfitrión (HASKELL) para poder resolver la operación.
- $\Rightarrow_{g(x)}$ es la reducción de una lista de valores aplicando la función *g* en el elemento *x*.

Por ejemplo para la potencia:

$$\frac{elige(\mathbf{expt}) = \wedge \quad Num(2) \Rightarrow NumV(2) \quad Num(3) \Rightarrow NumV(3)}{Binop(\mathbf{expt}, Num(2), Num(3)) \Rightarrow_{2^3} NumV(8)}$$

Condicional if

$$\frac{c, \varepsilon \Rightarrow c' \quad strict(c') = Boolean(True) \quad t, \varepsilon \Rightarrow t'}{If(c, t, e), \varepsilon t'}$$

$$\frac{c, \varepsilon \Rightarrow c' \quad strict(c') = Boolean(False) \quad e, \varepsilon \Rightarrow e'}{If(c, t, e), \varepsilon e'}$$

Expresiones lambda

$$\overline{Fun(p, c), \varepsilon \Rightarrow \langle p, c, \varepsilon \rangle}$$

Aplicaciones de Función

$$\frac{f, \varepsilon \Rightarrow f' \quad strict(f') = \langle c, p, \varepsilon' \rangle \quad c, \varepsilon'[p \leftarrow \langle a, \varepsilon \rangle] \Rightarrow c_v}{App(f, a), \varepsilon \Rightarrow c_v}$$