

Evaluación Semanal 3 - Oaxacanda

Ricardo Iván Martínez Cano Ángeles Sánchez Aldo Javier Jurado Guadalupe Aldo Emilio

August 2024

1. Dadas las siguientes expresiones en sintaxis concreta de nuestro lenguaje MiniLisp (a) obtener su sintaxis abstracta, (b) evaluarlas usando las reglas de semántica natural y (c) evaluarlas usando las reglas de semántica estructural. Todas las reglas las podrán consultar en la Nota de Clase 6 y la Nota de Clase 7.

- (- (+ 20 3) (- -18 (+ 50 20)))

(a) Sub (Add (Num(20), Num(3)), Sub (Num(-18), Add (Num(50), Num(20))))

(b)

$$\begin{array}{c}
 \frac{\text{Num}(20) \Rightarrow \text{Num}(20) \quad \text{Num}(3) \Rightarrow \text{Num}(3)}{\text{Add}(\text{Num}(20), \text{Num}(3)) \Rightarrow \text{Num}(23)} \quad \frac{\text{Num}(-18) \Rightarrow \text{Num}(-18) \quad \frac{\text{Num}(50) \Rightarrow \text{Num}(50)}{\text{Add}(\text{Num}(50), \text{Num}(20)) \Rightarrow \text{Num}(70)}}{\text{Sub}(\text{Num}(-18), \text{Num}(70)) \Rightarrow \text{Num}(-88)}} \\
 \text{Sub}(\text{Num}(23), \text{Num}(-88)) \Rightarrow \text{Num}(111)
 \end{array}$$

Lo que no se alcanza a ver es $\text{Add}(\text{Num}(50), \text{Num}(20)) \Rightarrow \text{Num}(70)$ y $\text{Num}(20) \Rightarrow \text{Num}(20)$

(c)

Semántica estructural:

$$\begin{aligned}
 & \text{Sub}(\text{Add}(\text{Num}(20), \text{Num}(3)), \text{Sub}(\text{Num}(-18), \text{Add}(\text{Num}(50), \text{Num}(20)))) \\
 & \rightarrow \text{Sub}(\text{Num}(23), \text{Sub}(\text{Num}(-18), \text{Add}(\text{Num}(50), \text{Num}(20)))) \\
 & \rightarrow \text{Sub}(\text{Num}(23), \text{Sub}(\text{Num}(-18), \text{Num}(70))) \\
 & \rightarrow \text{Sub}(\text{Num}(23), \text{Num}(-88)) \\
 & \rightarrow \text{Num}(-111)
 \end{aligned}$$

(a) Not (Add (Num(1), Sub(Num(3), Add(Num(-8), Num(1)))))

(b) Num(1) \Rightarrow Num(1) Num(3) \Rightarrow Num(3) Num(-8) \Rightarrow Num(-8) Num(1) \Rightarrow Num(1)

$$\text{Add}(\text{Num}(-8), \text{Num}(1)) \Rightarrow \text{Num}(-7)$$

$$\text{Sub}(\text{Num}(3), \text{Add}(\text{Num}(-8), \text{Num}(1))) \Rightarrow \text{Num}(10)$$

$$\text{Add}(\text{Num}(1), \text{Sub}(\text{Num}(3), \text{Add}(\text{Num}(-8), \text{Num}(1)))) \Rightarrow \text{Num}(11)$$

Not (Add (Num(1), Sub(Num(3), Add(Num(-8), Num(1))))) \Rightarrow Boolean(False)

(c) Not (Add (Num(1), Sub(Num(3), Add(Num(-8), Num(1)))))

- Not (Add (Num(1), Sub (Num(3), Num(-7))))
- Not (Add (Num(1), Num(10)))
- Not (Num(11))
- Boolean(False)

- (not (not (+ 3 5)))

(a) Not (Not (Add (Num(3), Num(5))))

(b) Num(3) \Rightarrow Num(3) Num(5) \Rightarrow Num(5)

Add(Num(3), Num(5)) \Rightarrow Num(8)

Not(Add (Num(3), Num(5))) \Rightarrow Boolean(False)

Not(Not(Add(Num(3), Num(5)))) \Rightarrow Boolean(True)

(c) Not (Not (Add (Num(3), Num(5))))

- Not (Not (Num(8)))
- Not (Boolean(False))
- Boolean(True)

2. Como segundo ejercicio deberán extender la batería de operaciones de MiniLisp, para ello deberán (a) dar la gramática libre de contexto modificada (en notación EBNF) añadiendo las nuevas construcciones del lenguaje, (b) modificar las reglas de sintaxis abstracta para considerar los nuevos constructores y finalmente (c) extender las reglas de semántica natural y estructural. En los tres casos, deberás usar la notación formal que vimos en clase.

- Especificar un nuevo constructor $*$ para la multiplicación binaria de expresiones aritméticas.

(b)

$$\frac{i \quad ASA \quad d \quad ASA}{Mult(i, d) \quad ASA}$$

(c)

- Especificar un nuevo constructor $/$ para la división binaria de expresiones aritméticas. Consideren que no se pueden realizar divisiones entre cero.
- Especificar un nuevo constructor add1 que dada una expresión, incrementa en uno su valor.
- Especificar un nuevo constructor sub1 que dada una expresión, decrementa en uno su valor.

El constructor lo dejamos en la gramática que pusimos al final

(a)

```

< S > := < Expr >
< Expr > ::= < Int >
           | ( + < Expr > < Expr > )
           | ( - < Expr > < Expr > )
           | ( * < Expr > < Expr > )
           | ( / < Expr > < Expr > )
           | (add1 < Expr >)
           | (sub1 < Expr >)
           | (sqrt < Expr >)
           | (not < Expr >)
< Int > := < N >
           | -< M >
< D > := 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
< N > := 0 | < D >{ < N >}
< M > := < D >{ < N >}
< Bool > := #f | #t

```

(b)

Multiplicación

$$\frac{i \quad ASA \quad d \quad ASA}{Mult(i,d) \quad ASA}$$

División

$$\frac{i \quad ASA \quad d \quad ASA}{Div(i,d) \quad ASA}$$

Add1

$$\frac{d \quad ASA}{Add1(d) \quad ASA}$$

Sub1

$$\frac{d \quad ASA}{Sub1(d) \quad ASA}$$

Raíz cuadrada

$$\frac{d \quad ASA}{Sqrt(d) \quad ASA}$$

(c)

Multiplicación Semantica Natural

$$\frac{i \Rightarrow Num(n_1) \quad d \Rightarrow Num(n_2)}{Mult(i,d) \Rightarrow Num(n_1 * n_2)}$$

Semantica Estructural

$$\frac{i \Rightarrow i'}{Mult(i, d) \Rightarrow Mult(i', d)}$$

$$\frac{d \Rightarrow d'}{Mult(Num(n_1, d)) \Rightarrow Mult(Num(n_1, d'))}$$

$$\frac{}{Mult(Num(i), Num(d)) \Rightarrow Num(i * d)}$$

División

a) dar la gramática libre de contexto modificada (en notación EBNF) añadiendo las nuevas construcciones del lenguaje.

$$\begin{aligned} \langle S \rangle &::= \langle \text{Expr} \rangle \\ \langle \text{Expr} \rangle &::= \langle \text{Int} \rangle \\ &| \langle \text{Bool} \rangle \\ &| (+ \langle \text{Expr} \rangle \langle \text{Expr} \rangle) \\ &| (- \langle \text{Expr} \rangle \langle \text{Expr} \rangle) \\ &| (/ \langle \text{Expr} \rangle \langle \text{Expr} \rangle) \quad (\text{Nueva regla para división}) \\ &| (\text{not } \langle \text{Expr} \rangle) \end{aligned}$$

$$\begin{aligned} \langle \text{Int} \rangle &::= \langle N \rangle | -\langle M \rangle \\ \langle N \rangle &::= \langle 0 \rangle | \langle D \rangle \{ \langle N \rangle \} \\ \langle D \rangle &::= 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 \\ \langle M \rangle &::= \langle D \rangle \{ \langle N \rangle \} \\ \langle \text{Bool} \rangle &::= \#t | \#f \end{aligned}$$

b) modificar las reglas de sintaxis abstracta para considerar los nuevos constructores
 $e1 \in ASA, e2 \in ASA$

$$\frac{e1 \text{ ASA} \quad e2 \text{ ASA}}{Div(e1, e2) \text{ ASA}}$$

(c) extender las reglas de semántica natural y estructural

Natural

$$\frac{e1 \Rightarrow Num(n_1) \quad e2 \Rightarrow Num(n_2)}{Div(e1, e2) \Rightarrow Num(n_1/n_2)} \quad n_2 \neq 0$$

$$\frac{e1 \Rightarrow Num(n_1) \quad e2 \Rightarrow Num(0)}{Div(e1, Num(0)) \Rightarrow \text{error: indefinido}}$$

Estructural

$$\begin{array}{c}
\frac{e1 \rightarrow e1'}{Div(e1, e2) \rightarrow Div(e1', e2)} \\
\frac{e2 \rightarrow e2'}{Div(Num(n1), e2) \rightarrow Div(Num(n1), e2')} \\
\frac{n2 \neq 0}{Div(Num(n1), Num(n2)) \rightarrow Num(n1/n2)} \\
\frac{}{Div(Num(n1), Num(0)) \rightarrow \text{error: indefinido}}
\end{array}$$

Add1

a) dar la gramática libre de contexto modificada (en notación EBNF) añadiendo las nuevas construcciones del lenguaje.

$$\begin{array}{l}
\langle S \rangle ::= \langle \text{Expr} \rangle \\
\langle \text{Expr} \rangle ::= \langle \text{Int} \rangle \\
\quad | \langle \text{Bool} \rangle \\
\quad | (+ \langle \text{Expr} \rangle \langle \text{Expr} \rangle) \\
\quad | (- \langle \text{Expr} \rangle \langle \text{Expr} \rangle) \\
\quad | (add1 \langle \text{Expr} \rangle) \quad (\text{Nueva regla para add1}) \\
\quad | (not \langle \text{Expr} \rangle)
\end{array}$$

$$\begin{array}{l}
\langle \text{Int} \rangle ::= \langle N \rangle | -\langle M \rangle \\
\langle N \rangle ::= \langle 0 \rangle | \langle D \rangle \{ \langle N \rangle \} \\
\langle D \rangle ::= 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 \\
\langle M \rangle ::= \langle D \rangle \{ \langle N \rangle \} \\
\langle \text{Bool} \rangle ::= \#t | \#f
\end{array}$$

b) modificar las reglas de sintaxis abstracta para considerar los nuevos constructores
 $e \in ASA$

$$\frac{e \in ASA}{Add1(e) \in ASA}$$

(c) extender las reglas de semántica natural y estructural

Natural

$$\frac{e \Rightarrow Num(n)}{Add1(e) \Rightarrow Num(n+1)}$$

Estructural

$$\frac{\frac{e \rightarrow e'}{Add1(e) \rightarrow Add1(e')}}{-}{Add1(Num(n)) \rightarrow Num(n+1)}$$

Sub1

Semántica Natural

$$\frac{d \Rightarrow Num(d')}{Sub1(d) \Rightarrow Num(d' - 1)}$$

Semántica Estructural

$$\frac{}{Sub1(Num(n)) \rightarrow Num(n - 1)}$$

$$\frac{d \rightarrow d'}{Sub1(d) \rightarrow Sub1(d')}$$

Sqrt

Semántica Natural

$$\frac{d \Rightarrow Num(-d')}{Sqrt(d) \Rightarrow \text{error: Raíz negativa}}$$

$$\frac{d \Rightarrow Num(d')}{Sqrt(d) \Rightarrow Num(\sqrt{d'})}$$

Semántica Estructural

$$\frac{}{Sqrt(Num(-n)) \rightarrow \text{error: Raíz negativa}}$$

$$\frac{}{Sqrt(Num(n)) \rightarrow Num(\sqrt{n})}$$

$$\frac{d \rightarrow d'}{Sqrt(d) \rightarrow Sqrt(d')}$$