

Semanal 03

Ana Lilia Carballido Camacateco - 315314601

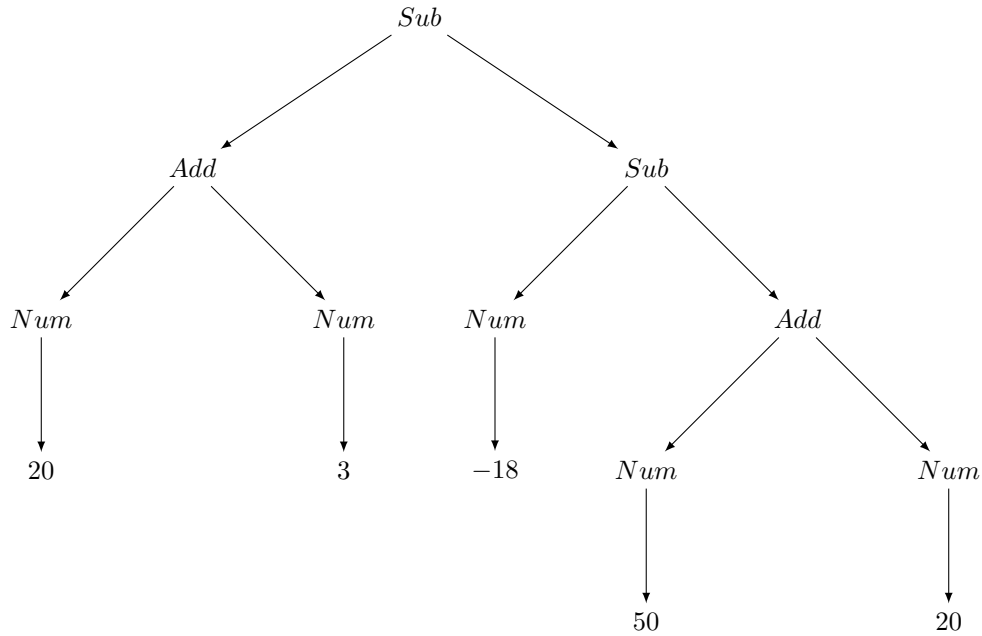
Miguel Angel Vargas Campos - 423114223

Oscar Fernando Frias Dominguez - 314255662

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) Árbol de Sintaxis Abstracta



(b) Semántica Natural

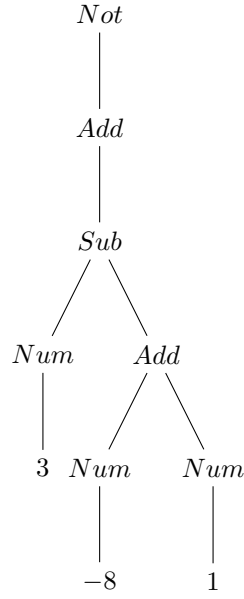
$$\begin{array}{c}
 \frac{Num(-18) \Rightarrow Num(-18), Num(70) \Rightarrow Num(70), Num(-88) \Rightarrow Num(-88)}{Sub((Num(-18)), (Num(70))) \Rightarrow Num(-88)} \\
 \frac{sub((Num(-18))(Add(Num(50)), (Num(20)))) \Rightarrow Num(70)}{Num(20) \Rightarrow Num(20), Num(3) \Rightarrow Num(3), Num(23)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

(c) Semántica Estructural

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

- (not (+ 1 (- 3 (+ -8 1))))

(a) Árbol de Sintaxis Abstracta



(b) Semántica Natural

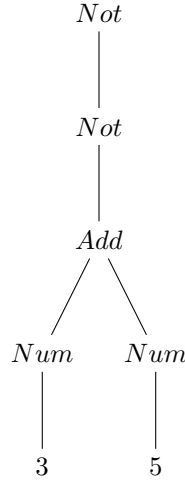
$$\begin{array}{c}
\frac{Num(-8 \Rightarrow Num(-8)) \quad Num(1) \Rightarrow Num(1)}{Add(Num(-8), Num(1)) \Rightarrow Num(-7)} \\
\frac{Sub(Num(3), Add(Num(-8), Num(1))) \Rightarrow Num(-4)}{Add(Num(1), Sub(Num(3), Add(Num(-8), Num(1)))) \Rightarrow Num(-2)} \\
\hline
Not(+1(-3(+ - 81))) \Rightarrow Bool(False)
\end{array}$$

(c) Semántica Estructural

$$\begin{array}{l}
Not(Add(Num(1), Sub(Num(3), Add(Num(-8), Num(1))))) \\
\rightarrow Not(Add(Num(1), Sub(Num(3), Num(-7)))) \\
\rightarrow Not(Add(Num(1), Num(-3))) \\
\rightarrow Not(-2) \\
\rightarrow Bool(False)
\end{array}$$

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

(a) Árbol de Sintaxis Abstracta



(b) Semántica Natural

$$\frac{\frac{\frac{Num(3) \Rightarrow Num(3) \quad Num(5) \Rightarrow Num(5)}{Add(Num(3), Num(5)) \Rightarrow Num(8)}}{Not(Add(Num(3), Num(5))) \Rightarrow Bool(False)}}{Not(Not(Add(Num(3), Num(5)))) \Rightarrow Bool(True)}$$

(c) Semántica Estructural

$$\begin{aligned}
& Not(Not(Add(Num(3), Num(5)))) \\
& \rightarrow Not(Not(Num(8))) \\
& \rightarrow Not(Bool(False)) \\
& \rightarrow Bool(True)
\end{aligned}$$

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. Por ejemplo:

> (* 20 2)
40

(a) Gramática Libre de Contexto A la gramatica previa le añadimos:

$$\langle Expr \rangle ::= (\text{mult } \langle Expr \rangle \langle Expr \rangle)$$

(b) Sintaxis Abstracta

$$\frac{e, e' \quad \mathbf{ASA}}{Mult(e, e') \quad \mathbf{ASA}} \quad (1)$$

(c) Semántica Natural y Estructural

i. Semántica Natural

$$\frac{i \Rightarrow Num(i') \quad j \Rightarrow Num(j')}{mult(i, j) \Rightarrow Num(i' * j')}$$

ii. Semántica Estructural

Caso 1:

$$\frac{i \rightarrow i'}{Mult(i, j) \rightarrow Mult(i', j)}$$

Caso 2:

$$\frac{j \rightarrow j'}{Mult(Num(i), j) \rightarrow Mult(Num(i), j')}$$

Caso 3:

$$\frac{}{Mult(Num(i), Num(j)) \rightarrow Num(i * j)}$$

- Especificar un nuevo constructor / para la división binaria de expresiones aritméticas. Consideren que no se pueden realizar divisiones entre cero. Por ejemplo:

```
> (/ 20 2)
10
> (/ 10 0)
error: División entre cero
```

- (a) Gramática Libre de Contexto
A la gramática previa le añadimos:

$\langle Expr \rangle ::= (Div \langle Expr \rangle \langle Expr \rangle)$

- (b) Sintaxis Abstracta

$$\frac{e, e' \quad \mathbf{ASA}}{Div(e, e') \quad \mathbf{ASA}} \quad (1)$$

- (c) Semántica Natural y Estructural

- i. Semántica Natural

$$\frac{i \Rightarrow Num(i') \quad j \Rightarrow Num(0)}{Div(i, j) \Rightarrow \text{Error: división entre cero}}$$

$$\frac{i \Rightarrow Num(i') \quad j \Rightarrow Num(j')}{Div(i, j) \Rightarrow Num(i'/j')}$$

- ii. Semántica Estructural

Caso 1:

$$\frac{i \rightarrow i'}{Div(i, Num(0)) \Rightarrow \mathbf{Error: División entre cero}}$$

Caso 2:

$$\frac{i \rightarrow i'}{Div(i, j) \rightarrow Div(i', j)}$$

Caso 3:

$$\frac{j \rightarrow j'}{Div(Num(i), j) \rightarrow Div(Num(i), j')}$$

Caso 4:

$$\frac{}{Div(Num(i), Num(j)) \rightarrow Num(i/j)}$$

- Especificar un nuevo constructor *add1* que dada una expresión, incrementa en uno su valor. Por ejemplo:

```
> (add1 10)
11
```

- (a) Gramática Libre de Contexto
A la gramática previa le añadimos:

$\langle Expr \rangle ::= (Add1 \langle Expr \rangle)$

- (b) Sintaxis Abstracta

$$\frac{e \quad \mathbf{ASA}}{Add1(e) \quad \mathbf{ASA}} \quad (1)$$

- (c) Semántica Natural y Estructural

i. Semántica Natural

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

ii. Semántica Estructural

$$\frac{n \rightarrow m}{Add1(n) \rightarrow Add1(m)} \quad (1)$$

$$\frac{}{Add1(Num(n)) \rightarrow Num(n+1)} \quad (2)$$

- Especificar un nuevo constructor *sub1* que dada una expresión, decrementa en uno su valor. Por ejemplo:

```
> (sub1 10)
9
```

(a) Gramática Libre de Contexto

A la gramatica previa le añadimos:

$\langle Expr \rangle ::= (\text{sub1 } \langle Expr \rangle)$

(b) Sintaxis Abstracta

$$\frac{e \quad \mathbf{ASA}}{Sub1(e) \quad \mathbf{ASA}} \quad (1)$$

(c) Semántica Natural y Estructural

i. Semántica Natural

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

ii. Semántica Estructural

$$\frac{n \rightarrow n'}{Sub1(n) \rightarrow Sub1(n')} \quad (1)$$

$$\frac{}{Sub1(Num(n)) \rightarrow Num(n-1)} \quad (2)$$

- Especificar un nuevo constructor *sqr* que dada una expresión, obtiene la raíz cuadrada de dicha expresión. Consideren que no se pueden calcular raíces cuadradas de números negativos. Por ejemplo:

```
> (sqr 81)
9
> (sqr -2)
error: Raíz negativa
```

(a) Gramática Libre de Contexto

A la gramatica previa le añadimos:

$\langle Expr \rangle ::= (\text{sqr } \langle Expr \rangle)$

(b) Sintaxis Abstracta

$$\frac{e \quad \mathbf{ASA}}{Sqr(e) \quad \mathbf{ASA}} \quad (1)$$

(c) Semántica Natural y Estructural

i. Semántica Natural

$$\frac{n \Rightarrow Num(n') \text{ con } n' < 0}{Sqr(n) \Rightarrow \text{error: Raíz negativa}} \quad (1)$$

$$\frac{n \Rightarrow Num(n')}{Sqr(n) \Rightarrow Num(\sqrt{n'})} \quad (2)$$

ii. Semántica Estructural

$$\frac{n \rightarrow n'}{Sqrt(n) \rightarrow Sqrt(n')} \quad (1)$$

$$\frac{n < 0}{Sqrt(Num(n)) \rightarrow \text{error: Raíz negativa}} \quad (2)$$

$$\frac{}{Sqrt(Num(n)) \rightarrow Num(\sqrt{n})} \quad (3)$$