UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
FACULTAD DE CIENCIAS
LENGUAJES DE PROGRAMACIÓN

Semanal 03

Martínez Osorio Benjamín 312063678

Salazar Gonzalez Pedro Yamil 306037445

Ejercicio 1

(− (+ 20 3) (− − 18 (+ 50 20)))

(a)Sintaxis Abstracta:

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

(b)Evaluación Natural:

$$\dfrac{\dfrac{\dfrac{}{\text{Num(20)} \Rightarrow \text{Num(20)}} \quad \dfrac{}{\text{Num(3)} \Rightarrow \text{Num(3)}}}{\text{Add(Num(20), Num(3))} \Rightarrow \text{Num(23)}} \quad \dfrac{\dfrac{}{\text{Num(-18)} \Rightarrow \text{Num(-18)}} \quad \dfrac{\dfrac{}{\text{Num(50)} \Rightarrow \text{Num(50)}} \quad \dfrac{}{\text{Num(20)} \Rightarrow \text{Num(20)}}}{\text{Add(Num(50), Num(20))} \Rightarrow \text{Num(70)}}}{\text{Sub(Num(-18), Add(Num(50), Num(20))} \Rightarrow \text{Num(-88)}}}{\text{Sub(Add(Num(20), Num(3)), Sub(Num(-18), Add(Num(50), Num(20)))} \Rightarrow \text{Num(111)}}$$

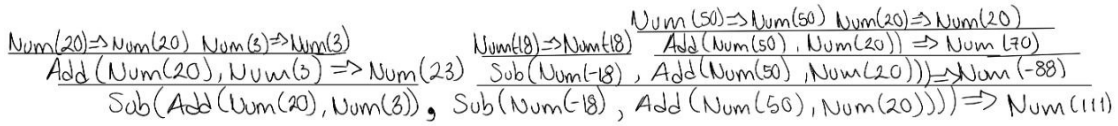

Figura 1: Evaluación Natural

(c) Evaluación Estructural

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

Resultado: **111**

Ejercicio 2:

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

(a) Sintaxis Abstracta

$$\text{Not}(\text{Add}(1, \text{Sub}(3, \text{Add}(-8, 1))))$$

(b) Evaluación usando Semántica Natural

1. $(+ - 8\ 1) \rightarrow -7$
2. $(-3\ -7) \rightarrow 10$
3. $(+1\ 10) \rightarrow 11$
4. not $11 \rightarrow$ false

Resultado: **false**

(c) Evaluación usando Semántica Estructural

1. $(+ - 8\ 1) \rightarrow -8 + 1 = -7$
2. $(-3\ (+ - 8\ 1)) \rightarrow 3 - (-7) = 10$
3. $(+1\ (-3\ (+ - 8\ 1))) \rightarrow 1 + 10 = 11$
4. not $(+1\ (-3\ (+ - 8\ 1))) \rightarrow$ not $11 =$ false

Resultado: **false**

Ejercicio 3:

```
(not (not (+ 3 5)))
```

(a) Sintaxis Abstracta

$$\text{Not}(\text{Not}(\text{Add}(3, 5)))$$

(b) Evaluación usando Semántica Natural

1. $(+3\ 5) \rightarrow 8$
2. not $8 \rightarrow$ false
3. not false $\rightarrow$ true

Resultado: **true**

(c) Evaluación usando Semántica Estructural

1. $(+3\ 5) \rightarrow 3 + 5 = 8$
2. not $8 \rightarrow$ not $8 =$ false
3. not (not 8) $\rightarrow$ not false $=$ true

Resultado: **true**

Extensión de la Batería de Operaciones en MiniLisp

Nuevos constructores:

- `*` para la multiplicación binaria de expresiones aritméticas.
- `/` para la división binaria de expresiones aritméticas (con verificación de división por cero).
- `add1` para incrementar en uno el valor de una expresión.
- `sub1` para decrementar en uno el valor de una expresión.
- `sqrt` para calcular la raíz cuadrada de una expresión (con verificación de valores negativos).

(a) Gramática en EBNF

La gramática libre de contexto en notación EBNF, incluyendo los nuevos constructores:

```
<expr> ::= <num>
         | '(' <op> <expr> <expr> ')'
         | '(' <unary-op> <expr> ')'

<num> ::= [0-9]+

<op> ::= '+' | '-' | '*' | '/'

<unary-op> ::= 'not' | 'add1' | 'sub1' | 'sqrt'
```

Se agrega unary-op dado que el not, add1, sub1 y sqrt pueden funcionar para la expresión con sólo un número

(b) Reglas de Sintaxis Abstracta

Las reglas de sintaxis abstracta se modifican para incluir los nuevos constructores:

$$\begin{aligned}\text{Expr} ::= &\text{Num}(n)\\ &|\text{Add}(\text{Expr}, \text{Expr})\\ &|\text{Sub}(\text{Expr}, \text{Expr})\\ &|\text{Mul}(\text{Expr}, \text{Expr})\\ &|\text{Div}(\text{Expr}, \text{Expr})\\ &|\text{Add1}(\text{Expr})\\ &|\text{Sub1}(\text{Expr})\\ &|\text{Sqrt}(\text{Expr})\\ &|\text{Not}(\text{Expr})\end{aligned}$$

(c) Extensión de las Reglas de Semántica

Semántica Natural

Las reglas de semántica natural para los nuevos operadores:

$$\frac{e_1 \to n_1 \quad e_2 \to n_2}{(Mul\ e_1\ e_2) \to n_1 \times n_2}$$

$$\frac{e_1 \to n_1 \quad e_2 \to n_2 \quad n_2 \neq 0}{(Div\ e_1\ e_2) \to n_1 \div n_2} \quad \text{Error si } n_2 = 0$$

$$\frac{e \to n}{(\text{add1}\ e) \to n + 1}$$

$$\frac{e \to n}{(\text{sub1}\ e) \to n - 1}$$

$$\frac{e \to n \quad n \geq 0}{(\text{sqrt } e) \to \sqrt{n}} \quad \text{Error si } n < 0$$

Semántica Estructural

Las reglas de semántica estructural para los nuevos operadores:

$$(Mul\ e_1\ e_2) \to \begin{cases} n_1 \times n_2 & \text{si } e_1 \to n_1 \text{ y } e_2 \to n_2 \\ \text{error} & \text{si alguna evaluación falla} \end{cases}$$

$$(Div\ e_1\ e_2) \to \begin{cases} n_1 \div n_2 & \text{si } e_1 \to n_1 \text{ y } e_2 \to n_2 \text{ y } n_2 \neq 0 \\ \text{error: división entre cero} & \text{si } n_2 = 0 \\ \text{error} & \text{si alguna evaluación falla} \end{cases}$$

$$(\text{add1 } e) \to \begin{cases} n+1 & \text{si } e \to n \\ \text{error} & \text{si la evaluación falla} \end{cases}$$

$$(\text{sub1 } e) \to \begin{cases} n-1 & \text{si } e \to n \\ \text{error} & \text{si la evaluación falla} \end{cases}$$

$$(\text{sqrt } e) \to \begin{cases} \sqrt{n} & \text{si } e \to n \text{ y } n \geq 0 \\ \text{error: raíz negativa} & \text{si } n < 0 \\ \text{error} & \text{si la evaluación falla} \end{cases}$$