1 0 1 1 0 0 0 1 1 0 0 1 0 1 1 1 1 0 0 0 1 0 1 0 1 0 1 1 0 1 1 1 0 1 0 0 1 1 0 0 0 1

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) Sintaxis abstracta :
     Sub (Add (Num(20), Num(3)) , Sub (Num (-18), Add (Num(50), Num(20))))
     (b) Evaluación por semántica natural :

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

     (c) Evaluación por semántica estructurada :
     $Sub(Add(Num(20), Num(3)), Sub(Num(-18), Add(Num(50), Num(20))))$
     $\longrightarrow Sub(Num(23), Sub(Num(-18), Add(Num(50), Num(20))))$
     $\longrightarrow Sub(Num(23), Sub(Num(-18), Num(70)))$
     $\longrightarrow Sub(Num(23), Num(-88))$
     $\longrightarrow Num(111)$

   - (not (+ 1 (- 3 (+ -8 1))))
     a) $(not(add(Num(1), sub(Num(3), (Add(Num(-8), Num(1)))))))$
     b)

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{Num(11) \Rightarrow Num(11)}}{\text{Not}(11) \Rightarrow \text{Boolean(False)}}}{Num(-8) \Rightarrow Num(-8) \quad Num(1) \Rightarrow Num(1)}}{Num(3) \Rightarrow Num(3) \quad (Add(Num(-8), Num(1))) \Rightarrow (-8 + 1) \Rightarrow -7}}{(Add(Num(1), \text{Sub}(Num(3), (Add(Num(-8), Num(1)))))) \Rightarrow (1 + 10) \Rightarrow 11}}{(\text{Not}(Add(Num(1), \text{Sub}(Num(3), (Add(Num(-8), Num(1))))))) \Rightarrow (\text{Not}(11)) \Rightarrow \text{Boolean(False)}}$$

     c) $(not(add(Num(1), sub(Num(3), (Add(Num(-8), Num(1)))))))$

     $\rightarrow (not(add(Num(1), sub(Num(3), (Num(-7))))))$

     $\rightarrow (not(add(Num(1), Num(10))))$

     $\rightarrow (not(11))$

     $\rightarrow (Boolean(False))$

   - (not (not (+ 3 5)))
     a) not(not(Add(Num(3),Num(5))))

Torres Nava Hazel
Martínez Hidalgo Paola Mildred
Figueroa Barrientos Andrea Valeria

b)

$$\text{Not}(False) \Rightarrow \text{Boolean}(True)$$
$$\text{Num}(8) \Rightarrow \text{Num}(8)$$
$$\text{Not}(8) \Rightarrow \text{Boolean}(False)$$
$$\text{Num}(3) \Rightarrow \text{Num}(3)$$
$$\text{Num}(5) \Rightarrow \text{Num}(5)$$
$$\text{Add}(\text{Num}(3), \text{Num}(5)) \Rightarrow \text{Num}(8)$$
$$\text{not}(\text{not}(\text{Add}(\text{Num}(3), \text{Num}(5)))) \Rightarrow \text{Not}(8) \Rightarrow \text{Boolean}(True)$$

c) not(not(Add(Num(3), Num(5))))
$\rightarrow not(not(Num(8)))$
$\rightarrow not(not(8))$
$\rightarrow not(Boolean(False))$
$\rightarrow Boolean(True)$

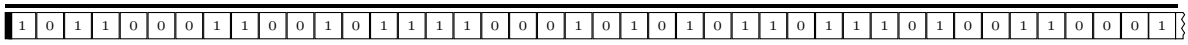
2. Como segundo ejercicio deberán extener la batería de operaciones de MiniLisp, para ello deberán (a) dar la gramática libre de contexto modificada (en notación EBNF) añadiendo las nuevas construcciones del lenguaje, (b) modificar las reglas de sintaxis abstracta para considerar los nuevos constructores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

- Veamos como es la Gramática Libre de Contexto:

```
< SinCero > ::= 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

< Expresiones > ::= < Mult > | < Div > | < Incr > | < Decr > | < Raiz >

< Mult > ::= "(" "*" < Num > < Num > ")"

< Div > ::= "(" "/" < Num > < SinCero > ")"

< Incr > := "(" "add1" < Num > ")"

< Decr > := "(" "sub1" < Num > ")"

< Raiz > ::= "(" "sqrt" < Num > ")"
```

- Especificar un nuevo constructor * para la multiplicación binaria de expresiones aritméticas. Por ejemplo:
¿ (* 20 2) 40
  - Regla de sintaxis abstracta :
    $Mult(i, d)$ es un ASA si tanto i como d son $ASAs$, entonces también :

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



– Reglas de semántica natural :

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

– Reglas de semántica estructurada:

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

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

$$\frac{}{Mult(Num(n_1),Num(n_2)) \rightarrow Num(n_1 * n_2)}$$

- 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

  – Regla de sintaxis abstracta :
  $Div(i,d)$ es un ASA si tanto i como d son $ASAs$ tal que $d \neq Num(0)ASA$, entonces también :

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

  – Reglas de semántica natural :

$$\frac{i \Rightarrow Num(n_1) \quad d \Rightarrow Num(0)}{Div(i,d) \Rightarrow Error}$$

$$\frac{i \Rightarrow Num(n_1) \quad d \Rightarrow Num(n_2)}{Div(i,d) \Rightarrow Num(i/d)}$$

  – Reglas de semántica estructurada:

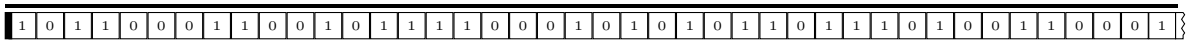
$$\frac{i \rightarrow i'}{Div(i,d) \rightarrow Div(i',d)}$$

$$\frac{d \rightarrow d'}{Div(Num(n_1),d) \rightarrow Div(Num(n_1),d')}$$

$$\frac{}{Div(Num(n_1),Num(n_2)) \rightarrow Num(n_1/n_2)}$$

- Especificar un nuevo constructor add1 que dada una expresión, incrementa en uno su valor. Por ejemplo:
  (add1 10) 11

Torres Nava Hazel
Martínez Hidalgo Paola Mildred
Figueroa Barrientos Andrea Valeria



  - Regla de sintaxis abstracta :
    $Add1(i)$ es un ASA si tanto i ASA también :

$$\frac{iASA}{Add1(i)ASA}$$

  - Reglas de semántica natural :

$$\frac{i \Rightarrow Num(n_1)}{Add1(i) \Rightarrow Num(n_1 + 1)}$$

  - Reglas de semántica estructurada:

$$\frac{i \rightarrow i'}{Add1(i) \rightarrow Add1(i')}$$

$$\frac{}{Add1(Num(n_1)) \rightarrow Num(n_1 + 1)}$$

- Especificar un nuevo constructor sub1 que dada una expresión, decrementa en uno su valor. Por ejemplo:
  ¿ (sub1 10) 9
  - Regla de sintaxis abstracta :
    $sub1(i)$ es un ASA si tanto i ASA también :

$$\frac{iASA}{sub1(i)ASA}$$

  - Reglas de semántica natural :

$$\frac{i \Rightarrow Num(n)}{sub1(i) \Rightarrow Num(n - 1)}$$

  - Reglas de semántica estructurada:

$$\frac{i \rightarrow n}{sub1(i) \rightarrow sub1(n)}$$

$$\frac{}{sub1(Num(i)) \rightarrow Num(i + 1)}$$

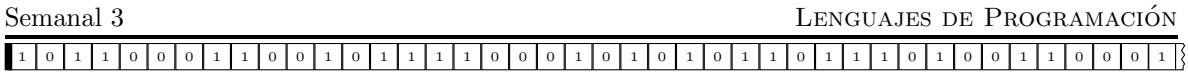
- Especificar un nuevo constructor sqrt que dada una expresión, obtiene la raíz cuadrada de dicha expresión. Consideren que no se pueden calcular raíces cuadradas de números negativos. Por ejemplo:
  ¿ (sqrt 81) 9 ¿ (sqrt -2) error: Raíz negativa
- Regla de sintaxis abstracta :
  $sqrt(i)$ es un ASA si tanto i ASA también :

$$\frac{iASA}{sqrt(i)ASA}$$

- Reglas de semántica natural :

$$\frac{i \Rightarrow Num(-n)}{sqrt(i) \Rightarrow Error}$$

Torres Nava Hazel
Martínez Hidalgo Paola Mildred
Figueroa Barrientos Andrea Valeria



$$\frac{i \Rightarrow Num(n)}{sqrt(i) \Rightarrow Num(Sqrt(n))}$$

- Reglas de semántica estructurada:

$$\frac{i \rightarrow -n}{sqrt(i) \rightarrow Error}$$

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

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

Torres Nava Hazel
Martínez Hidalgo Paola Mildred
Figueroa Barrientos Andrea Valeria