

Universidad Nacional Autónoma de México Facultad de Ciencias

lenguajes de programación

Semanal 3

Jacome Delgado Alejandro (320011704) Jimenez Sanchez Emma Alicia (320046155) 28 de agosto de 2024



1. Primera Parte

- 1. (- (+ 20 3) (- -18 (+ 50 20))):
 - $\blacksquare \ \, \text{Sintaxis Abstracta:} \ \, Sub(Add(Num(20),Num(3)),Sub(Num(-18),(Add(Num(50),Num(20)))))$
 - Evaluar usando las reglas de semántica natural:

												Num(so) →t	Jum (S	၀)	Ŋ	um (20) → Nur	1(20)
Num(20) ->1	Vum(20)		V um(3) → N	um(3)		Num	((-18)	->N	Jm(-1	٤)	Ado	(Nu	m(so)	, No	mao)) ->	Num	(70)
Add(Nor	m(20),1	Jun(3)) -	Vum(23)	_	Sub	(Nur	nL-1	8),(Add	(Nur	n(50), No	m(20))) - -	Nu	m (-8	8)
Sub(AddCN	m(20)), No	um(3	1),	Sub (Num	<u>L-18</u>),(1	Add(Num	(50)	, Nun	n(20)))) -	>	No	m(1	11)

• Evaluar usando las reglas de semántica estructural:

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

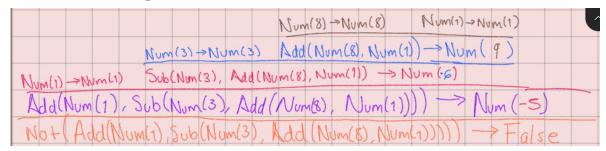
$$\longrightarrow Sub(Num(23), Sub(Num(-18), (Add(Num(50), Num(20)))))$$

$$\longrightarrow Sub(Num(23), Sub(Num(-18), (Num(70))))$$

$$\longrightarrow Sub(Num(23), Num(-88))$$

$$\longrightarrow Num(111)$$

- 2. (not (+ 1 (- 3 (+ -8 1)))):
 - Sintaxis Abstracta: not(Add(Num(1), Sub(Num(3), Add(Num(8), Num(1)))))
 - Evaluar usando las reglas de semántica natural:



• Evaluar usando las reglas de semántica estructural:

$$not(Add(Num(1), Sub(Num(3), Add(Num(8), Num(1)))))$$

$$\longrightarrow not(Add(Num(1), Sub(Num(3), Num(9)))))$$

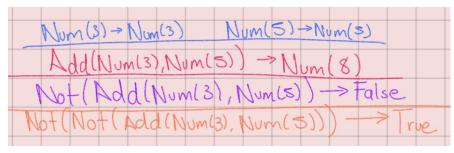
$$\longrightarrow not(Add(Num(1), Num(-6)))$$

$$\longrightarrow not(Num(-5))$$

$$\longrightarrow Boolean(False)$$

$$(2)$$

- 3. (not (not (+ 3 5))):
 - Sintaxis Abstracta: not(not(Add(Num(3), Num(5))))
 - Evaluar usando las reglas de semántica natural:



• Evaluar usando las reglas de semántica estructural:

$$not(not(Add(Num(3), Num(5))))$$

$$\longrightarrow not(not(Num(8)))$$

$$\longrightarrow not(Boolean(False))$$

$$\longrightarrow Boolean(True)$$
(3)

- 4. Añadir * (producto), / (división), add1 (sumar 1), sub1 (restar 1), sqrt (raíz cuadrada) a nuestro minilisp
 - lo primero que haremos sera definir nuestra gramática libre de contexto colocando los nuevos constructores:

$$< S > := < Expr >$$

$$< Expr > := < Int >$$

$$| < Bool >$$

$$| (+ < Expr > < Expr >)$$

$$| (- < Expr > < Expr >)$$

$$| (* < Expr > < Expr >)$$

$$| (* < Expr > < Expr >)$$

$$| (dd1 < Expr > < Expr >)$$

$$| (sub1 < Expr >)$$

$$| (sqrt < Expr >)$$

$$| (sqrt < Expr >)$$

$$| < Int > := < N >$$

$$| - < M >$$

$$< D > := 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9$$

$$< N > := 0 | < D > {< N >}$$

$$< M > := < D > {< N >}$$

$$< Bool > := #t | #f$$

 ya con la gramática actualizada, lo que vamos a hacer es crear las reglas de sintaxis abstracta de los constructores que añadimos: Prod(i, j) (es la etiqueta del producto):

$$\frac{i \text{ ASA} \qquad j \text{ ASA}}{Prod(i,j) \text{ ASA}}$$

Div(i,j) (es la etiqueta de la división):

$$\frac{i \text{ ASA} \qquad j \text{ ASA}}{Div(i,j) \text{ ASA}}$$

Add1(i) (es la etiqueta de sumar 1):

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

Sub1(i) (es la etiqueta de restar 1):

$$\frac{i \text{ ASA}}{Sub1(i) \text{ ASA}}$$

Sqrt(i) (es la etiqueta de sumar 1):

$$\frac{i \text{ ASA}}{Sqrt(i) \text{ ASA}}$$

5. ya definidas las reglas de sintaxis abstracta, se harán las reglas para definir la semántica natural (paso grande)

Prod(i, j) (es la etiqueta del producto):

$$\frac{n_1 \Rightarrow Num(i) \qquad n_2 \Rightarrow Num(j)}{Prod(i,j) \Rightarrow Num(n_1 * n_2)}$$

Div(i,j) (es la etiqueta de la división):

$$n_1 \Rightarrow Num(i) \qquad j \Rightarrow Num(0)$$

$$Div(i,j) \Rightarrow Error(Division 0)$$

$$\frac{n_1 \Rightarrow Num(i) \qquad n_2 \Rightarrow Num(j)}{Div(i,j) \Rightarrow Num(n_1/n_2)}$$

Add1(i) (es la etiqueta de sumar 1):

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

Sub1(i) (es la etiqueta de restar 1):

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

Sqrt(i) (es la etiqueta de la raíz cuadrada):

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

6. ya con la semántica natural hecha, realizaremos la semántica estructurada (paso chico):

Prod(i, j) (es la etiqueta del producto):

$$\frac{i \to i'}{Prod(i,j) \to Prod(i',j)}$$

$$\frac{j \to j'}{Prod(Num(n_1),j) \to Prod(Num(n_1),j')}$$

$$Prod(Num(n_1),Num(n_2)) \to Num(n_1*n_2)$$

Div(i, j) (es la etiqueta de la división):

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

$$\frac{j \to j'}{Div(Num(n_1),j) \to Div(Num(n_1),j')}$$

$$\overline{Div(Num(n_1),Num(0)) \to Error}$$

$$Div(Num(n_1),Num(n_2)) \to Num(n_1/n_2)$$

Add1(i) (es la etiqueta de sumar 1):

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

$$Add1(Num(n_1)) \to Num(n_1+1)$$

Sub1(i) (es la etiqueta de restar 1):

$$\frac{i \to i'}{Sub1(i) \to Sub1(i')}$$

$$Sub1(Num(n_1)) \to Num(n_1 - 1)$$

Sqrt(i) (es la etiqueta de la raíz cuadrada):

$$\frac{i \to i'}{Sqrt(i) \to Sqrt(i')}$$

$$Sqrt(Num(n_1)) \to Num(\sqrt{n_1})$$