Trabajo Práctico
Lenguaje Imperativo Simple

Análisis de Lenguajes de Programación
Facultad de Ciencias Exactas, Ingeniería y Agrimensura
Escuela de Ciencias Exactas y Naturales

Martín Binner
Agustín Merino

18 de septiembre de 2024

1. Ejercicio 1: v++ y v–

Modificación de la sintaxis abstracta

$$
\begin{aligned}
intexp ::= {} & nat \mid var \mid -_u \; intexp \\
& \mid intexp + intexp \\
& \mid intexp -_b \; intexp \\
& \mid intexp \times intexp \\
& \mid intexp \div intexp \\
& \mid intexp ++ \\
& \mid intexp --
\end{aligned}
$$

Modificación de la sintaxis concreta

$$
\begin{aligned}
intexp ::= {} & nat \mid var \mid' -_u' \; intexp \\
& \mid intexp \; ' + ' \; intexp \\
& \mid intexp \; ' -_b ' \; intexp \\
& \mid intexp \; ' \times ' \; intexp \\
& \mid intexp \; ' \div ' \; intexp \\
& \mid intexp \; ' ++' \\
& \mid intexp \; ' --'
\end{aligned}
$$

Desanbigüamos la gramática

$$intexp ::= intexp \; ' + ' \; iterm \mid intexp \; ' -_b ' \; iterm \mid iterm$$

$$iterm ::= iterm \; ' * ' \; minus \mid iterm \; '/' \; minus \mid minus$$

$$minus ::= \; ' -_u ' \; mm \mid mm$$

$$mm ::= nat \mid var \mid var \; ' ++' \mid var \; ' --' \mid '(' \; intexp \; ')'$$

2. Ejercicio 4: Nueva regla de big-step

Nuestra nueva regla de construcción solo requiere tener una variable en el dominio de los estados y la semantica del operador '++' es sumarle uno al valor de la variable y asignar el nuevo valor en la variable, de esta forma se modifica su estado

$$\frac{x \in \mathrm{dom}(\sigma)}{\langle x{+}{+}, \sigma\rangle \Downarrow_{exp} \langle \sigma x + 1, [\sigma \mid x : \sigma x + 1]\rangle}\text{PlusPlus}$$

Lo mismo sucede para el operador '–' pero en vez de sumarle uno a la variable, se le resta 1 y se asigna el valor obtenido

$$\frac{x \in \mathrm{dom}(\sigma)}{\langle x{-}{-}, \sigma\rangle \Downarrow_{exp} \langle \sigma x - 1, [\sigma \mid x : \sigma x - 1]\rangle}\text{MinusMinus}$$

3. Ejercicio 5: Demostración de determinismo

Vamos a probar que la relación de evaluación $\rightsquigarrow$ es determinista. Esto quiere decir que dados $t,\ t',\ t'' \in\ \Gamma,\ t \rightsquigarrow t'\ y\ t \rightsquigarrow t'', entonces\ t' = t''$

Demostración. Por inducción en la derivación $t\ \rightsquigarrow\ t'$.
Supongamos que la última regla de derivación fue **Ass**, entonces

- $< e, \sigma > \Downarrow_{exp} < n, \sigma' >$
- t es de la forma $< v = e,\ \sigma >$
- t' es de la forma $< skip,\ [\sigma' \mid v : n >]$

Ahora, veamos cual podría ser la última regla utilizada en la derivación $t \rightsquigarrow t''$

1. **Seq1** No puede ser ya que en la configuración $< skip;\ C1,\ \sigma >$ difiere de la expresión $v\ =\ e$
2. **Seq2** No puede ser ya que en en $< C_0, \sigma > \Downarrow_{exp} < C_0', \sigma >$ C_0 deriva a C_0' pero $v\ =\ e$ no deriva a una expresión.
3. **IF1** y **IF2** No puede ser ya que $v\ =\ e$ difiere de $if\ b\ then\ C_0\ else\ C_1$.
4. **repeat** No puede ser ya que $v\ =\ e$ difiere de $repeat\ C_0\ until\ b$

Por lo tanto, la única regla aplicable es **Ass** y como $\Downarrow_{exp}$ es determinista $t' = t''$.

Supongamos que la última regla de derivación fue **Seq1**, entonces

- t es de la forma $< skip;\ C1,\ \sigma >$
- t' es de la forma $< C1,\ \sigma >$

Ahora, veamos cual podría ser la última regla utilizada en la derivación $t \rightsquigarrow t''$

1. **Ass** No puede ser ya que en la configuración $v\ =\ e$ difiere de la expresión $< skip;\ C1,\ \sigma >$.
2. **Seq2** No puede ser ya que en $< C_0, \sigma > \Downarrow_{exp} < C_0', \sigma >$ C_0 deriva a C_0' pero C_0 no puede ser $skip$ ya que no deriva a nada.
3. **IF1** y **IF2** No puede ser ya que $skip$ difiere de $if\ b\ then\ C_0\ else\ C_1$.
4. **repeat** No puede ser ya que $skip$ difiere de $repeat\ C_0\ until\ b$

Por lo tanto, la única regla aplicable es **Seq1** y como $\Downarrow_{exp}$ es determinista $t' = t''$.

Supongamos que la última regla de derivación fue **Seq2**, entonces

- $< C, \sigma > \rightsquigarrow < C_0', \sigma' >$
- t es de la forma $< C_0;\ C1,\ \sigma >$
- t' es de la forma $< C_0';\ C1,\ \sigma' >$

Ahora, veamos cual podría ser la última regla utilizada en la derivación $t \rightsquigarrow t''$

1. **Ass** No puede ser ya que en la configuración $v = e$ difiere de la expresión $C_0;\ C1$.
2. **Seq1** No puede ser ya que en $< skip, \sigma >$ difiere de $C_0;\ C_1$
3. **IF1** y **IF2** No puede ser ya que $C_0;\ C_1$ difiere de $if\ b\ then\ C_0\ else\ C_1$.
4. **repeat** No puede ser ya que $C_0; C_1$ difiere de $repeat\ C_0\ until\ b$

Por lo tanto, la única regla aplicable es **Seq2** y, por hipotesis inductiva, la derivación $< C, \sigma > \rightsquigarrow < C_0', \sigma' >$ es determinista, resulta $t' = t''$.

Supongamos que la última regla de derivación fue **repeat**, entonces

- t es de la forma $< repeat\ C\ until\ b,\ \sigma >$
- t' es de la forma $< C;\ if\ b\ then\ skip\ else\ repeat\ C\ until\ b,\ \sigma >$.

Ahora, veamos cual podría ser la última regla utilizada en la derivación $t \rightsquigarrow t''$

1. **Ass** No puede ser ya que en la configuración $v = e$ difiere de la expresión $repeat\ C\ until\ b$.
2. **Seq1** y **Seq2** No puede ser ya que en $< skip, \sigma >$ difiere de $repeat\ C\ until\ b$
3. **IF1** y **IF2** No puede ser ya que $repeat\ C\ until\ b$ difiere de $if\ b\ then\ C_0\ else\ C_1$.

Por lo tanto, la única regla aplicable es **repeat**, resulta $t' = t''$.

Supongamos que la última regla de derivación fue **IF1**, entonces

- $< b, \sigma > \Downarrow_{exp} < true, \sigma' >$
- t es de la forma $if\ b\ then\ C_0\ else\ C_1$
- t' es de la forma $< C_0,\ \sigma' >$

Ahora, veamos cual podría ser la última regla utilizada en la derivación $t \rightsquigarrow t''$

1. **Ass** No puede ser ya que en la configuración $v = e$ difiere de la expresión $if\ b\ then\ C_0\ else\ C_1$.
2. **Seq1** No puede ser ya que en $< skip, \sigma >$ difiere de $if\ b\ then\ C_0\ else\ C_1$.
3. **IF1** y **IF2** No puede ser ya que por hipotesis b deriva a $true$ y en **IF2** b deriva a $false$ lo cual es absurdo.
4. **repeat** No puede ser ya que $repeat\ C_0\ until\ b$ difiere de $if\ b\ then\ C_0\ else\ C_1$

Por lo tanto, la única regla aplicable es **IF1** y como $\Downarrow_{exp}$ es determinista $t' = t''$.

Supongamos que la última regla de derivación fue **IF2**, entonces

- $< b, \sigma > \Downarrow_{exp} < false, \sigma' >$
- t es de la forma $if\ b\ then\ C_0\ else\ C_1$
- t' es de la forma $< C_1,\ \sigma' >$

Ahora, veamos cual podría ser la última regla utilizada en la derivación $t \rightsquigarrow t''$

1. **Ass** No puede ser ya que en la configuración $v = e$ difiere de la expresión $if\ b\ then\ C_0\ else\ C_1$.
2. **Seq1** No puede ser ya que en $< skip, \sigma >$ difiere de $if\ b\ then\ C_0\ else\ C_1$.
3. **IF1** No puede ser ya que por hipótesis b deriva a $false$ y en **IF2** b deriva a $true$ lo cual es absurdo.
4. **repeat** No puede ser ya que $repeat\ C_0\ until\ b$ difiere de $if\ b\ then\ C_0\ else\ C_1$

Por lo tanto, la única regla aplicable es **IF2** y como $\Downarrow_{exp}$ es determinista $t' = t''$.

$\square$

Queda demostrado, por inducción sobre las reglas de derivación de $\rightsquigarrow$, que $t \rightsquigarrow t'\ y\ t \rightsquigarrow t'', entonces\ t' = t''$

4. Ejercicio 6: Programas equivalentes

Demostración. Vamos a probar que los siguientes programas son equivalentes mostrando sus arboles de derivación, utilizando las reglas de inferencia dadas.

Programa 1
$x = x + 1;$
$y = x;$

Programa 2
$y = x + +;$

4.1. Árbol de derivación para el programa 1.

4.1.1. Árbol 1 (A1)

$$\cfrac{\cfrac{\cfrac{\cfrac{x \in dom\sigma}{\langle x, \sigma\rangle \Downarrow_{exp} \langle \sigma x, \sigma\rangle}\text{Var} \quad \cfrac{}{\langle 1, \sigma\rangle \Downarrow_{exp} \langle \mathbf{1}, \sigma\rangle}\text{NVal}}{\langle x + 1, \sigma\rangle \Downarrow_{exp} \langle \sigma x {+} 1, \sigma\rangle}\text{Plus}}{\langle x = x + 1, \sigma\rangle \rightsquigarrow \langle \mathbf{skip}, [\sigma' \mid x : \sigma x + 1]\rangle}\text{Ass}}{\langle x = x + 1; y = x, \sigma\rangle \rightsquigarrow \langle \mathbf{skip}; y = x, [\sigma' \mid x : \sigma x + 1]\rangle}\text{Seq2}$$

4.1.2. Árbol 2 (A2)

$$\cfrac{}{\langle \mathbf{skip}; y = x, [\sigma' \mid x : \sigma x + 1]\rangle \rightsquigarrow \langle y = x, [\sigma' \mid x : \sigma x + 1]\rangle}\text{Seq1}$$

4.1.3. Árbol 3 (A3)

$$\cfrac{\cfrac{x \in dom\sigma}{\langle x, [\sigma' \mid x : \sigma x + 1]\rangle \Downarrow_{exp} \langle \sigma x, [\sigma' \mid x : \sigma x + 1]\rangle}\text{Var}}{\langle y = x, [\sigma' \mid x : \sigma x + 1]\rangle \rightsquigarrow \langle \mathbf{skip}, [[\sigma' \mid x : \sigma x + 1] \mid y : \sigma' x]\rangle}\text{Ass}$$

Luego, vemos que con A1 concluimos que:

$$\langle x = x + 1; y = x, \sigma\rangle \rightsquigarrow \langle \mathbf{skip}; y = x, [\sigma' \mid x : \sigma x + 1]\rangle \tag{1}$$

y por la regla de $\rightsquigarrow^*$

$$\langle x = x + 1; y = x, \sigma\rangle \rightsquigarrow^* \langle \mathbf{skip}; y = x, [\sigma' \mid x : \sigma x + 1]\rangle \tag{2}$$

de la misma manera, concluimos en A2 que:

$$\langle \mathbf{skip}; y = x, [\sigma' \mid x : \sigma x + 1]\rangle \rightsquigarrow^* \langle y = x, [\sigma' \mid x : \sigma x + 1]\rangle \tag{3}$$

y de igual forma en A3:

$$\langle y = x, [\sigma' \mid x : \sigma x + 1]\rangle \rightsquigarrow^* \langle \mathbf{skip}, [[\sigma' \mid x : \sigma x + 1] \mid y : \sigma' x]\rangle \tag{4}$$

Finalmente vemos que por (2) y (3) podemos justificar, mediante las reglas de $\rightsquigarrow^*$ que

$$\langle x = x + 1; y = x, \sigma\rangle \rightsquigarrow^* \langle y = x, [\sigma' \mid x : \sigma x + 1]\rangle \tag{5}$$

y luego por (5) y (4) concluimos que:

$$\langle x = x + 1; y = x, \sigma\rangle \rightsquigarrow^{*} \langle \textbf{skip}, [[\sigma' \mid x : \sigma x + 1] \mid y : \sigma' x]\rangle \tag{6}$$

llegando así al final de la evaluación semántica del **programa 1**.

4.2. Árbol de derivación para el programa 2.

$$\dfrac{\dfrac{x \in dom\sigma}{\langle x++, \sigma\rangle \Downarrow_{exp} \langle \sigma x + 1, [\sigma' \mid x : \sigma x + 1]\rangle}\,\text{Var}}{\langle y = x++, \sigma\rangle \rightsquigarrow \langle \textbf{skip}, [[\sigma' \mid x : \sigma x + 1] \mid y : \sigma' x]\rangle}\,\text{PlusPlus}$$

Luego vemos que, por la regla de $\rightsquigarrow^{*}$, podemos deducir que

$$\langle y = x++, \sigma\rangle \rightsquigarrow^{*} \langle \textbf{skip}, [[\sigma' \mid x : \sigma x + 1] \mid y : \sigma' x]\rangle \tag{7}$$

Finalmente, por (7) y (6), los programas 1 y 2 son semánticamente equivalentes. □