Universidad Nacional de Rosario

Facultad de Ciencias Exactas, Ingeniería y Agrimensura

Análisis de Lenguajes de Programación

R-322

Informe Trabajo Práctico I

Alumnos

Cavigioli, Axel

Rassi, Octavio

Sferco, Martín

17 de Septiembre, 2024

1. Ejercicio 1.

1.1. Enunciado.

Extienda las sintaxis abstracta y concreta de **LIS** para incluir a los operadores $++$ y $--$ que construyen una expresión entera a partir de una variable. La expresión $v{+}{+}$ ($v{-}{-}$) tiene como valor el valor de v más (menos) uno, y además tiene como efecto secundario actualizar el valor de v luego de sumarle (restarle) uno.

Estas extensiones permiten simplificar la escritura de algunos patrones frecuentes, por ejemplo:

- La secuencia de comandos `x = x + 1; y = x` puede re-escribirse como `y = x++`.
- La secuencia de comandos `x = x - 1; if x <  0 {y = 1}` puede re-escribirse como `if x-- < 0 {y = 1}`

1.2. Resolución.

1.2.1. Extensión de la sintaxis abstracta.

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

$$
\begin{aligned}
\text{boolexp} ::=\ & \textbf{true} \mid \textbf{false} \\
& \mid \text{intexp} == \text{intexp} \\
& \mid \text{intexp} \neq \text{intexp} \\
& \mid \text{intexp} < \text{intexp} \\
& \mid \text{intexp} > \text{intexp} \\
& \mid \text{boolexp} \wedge \text{boolexp} \\
& \mid \text{boolexp} \vee \text{boolexp} \\
& \mid \neg \text{ boolexp}
\end{aligned}
$$

$$
\begin{aligned}
\text{comm} ::=\ & \textbf{skip} \\
& \mid \text{var} = \text{intexp} \\
& \mid \text{comm}; \text{comm} \\
& \mid \textbf{if} \text{ boolexp } \textbf{then} \text{ comm } \textbf{else} \text{ comm} \\
& \mid \textbf{repeat} \text{ comm } \textbf{until} \text{ boolexp}
\end{aligned}
$$

1.2.2. Extensión de la sintaxis concreta.

$$
\begin{array}{rl}
\text{digit} ::= & \texttt{'0'} \mid \texttt{'1'} \mid \cdots \mid \texttt{'9'} \\[1ex]
\text{letter} ::= & \texttt{'a'} \mid \cdots \mid \texttt{'z'} \mid \texttt{'A'} \mid \cdots \mid \texttt{'Z'} \\[1ex]
\text{nat} ::= & \text{digit} \mid \text{digit nat} \\[1ex]
\text{var} ::= & \text{letter} \mid \text{letter var} \\[1ex]
\text{intexp} ::= & \text{nat} \\
& \mid \text{var} \\
& \mid \text{var } \texttt{'++'} \\
& \mid \text{var } \texttt{'--'} \\
& \mid \texttt{'-'} \text{ intexp} \\
& \mid \text{intexp } \texttt{'+'} \text{ intexp} \\
& \mid \text{intexp } \texttt{'-'} \text{ intexp} \\
& \mid \text{intexp } \texttt{'*'} \text{ intexp} \\
& \mid \text{intexp } \texttt{'/'} \text{ intexp} \\
& \mid \texttt{'('} \text{ intexp } \texttt{')'} \\[1ex]
\text{boolexp} ::= & \texttt{'true'} \mid \texttt{'false'} \\
& \mid \text{intexp } \texttt{'=='} \text{ intexp} \\
& \mid \text{intexp } \texttt{'!='} \text{ intexp} \\
& \mid \text{intexp } \texttt{'<'} \text{ intexp} \\
& \mid \text{intexp } \texttt{'>'} \text{ intexp} \\
& \mid \text{boolexp } \texttt{'\&\&'} \text{ boolexp} \\
& \mid \text{boolexp } \texttt{'||'} \text{ boolexp} \\
& \mid \texttt{'!'} \text{ boolexp} \\
& \mid \texttt{'('} \text{ boolexp } \texttt{')'} \\[1ex]
\text{comm} ::= & \texttt{skip} \\
& \mid \text{var } \texttt{'='} \text{ intexp} \\
& \mid \text{comm}\texttt{';'}\text{comm} \\
& \mid \texttt{'if'} \text{ boolexp } \texttt{'\{'} \text{ comm } \texttt{'\}'} \\
& \mid \texttt{'if'} \text{ boolexp } \texttt{'\{'} \text{ comm } \texttt{'\}'} \texttt{ else \{} \text{ comm } \texttt{\}} \\
& \mid \texttt{'repeat'} \ \texttt{'\{'} \text{ comm } \texttt{'\}'} \ \texttt{'until'} \text{ boolexp}
\end{array}
$$

2. Ejercicio 4.

2.1. Enunciado.

Modifique la semántica *big-step* de expresiones enteras para incluir a los operadores $++$ y $--$ descriptos en el Ejercicio 2.2.

Dado que las expresiones podrán modificar el entorno de variables, utilizar la notación $[f \mid x : e]$ para denotar la función f' , tal que *dom* $f' =$ *dom* $f' \cup \{x\}$ y

$$f'(y) = \begin{cases} e & \text{si } y = x \\ f(y) & \text{si } y \neq x \end{cases}$$

2.2. Resolución.

Extenderemos la semántica *big-step* para incluír las siguientes reglas

$$\frac{x \in \text{dom } \sigma}{\langle x\texttt{++};\ \sigma\rangle \Downarrow_{\text{exp}} \langle \sigma x + 1,\ [\ \sigma \mid x : \sigma x + 1\ ]\rangle}\ \textsc{VarInc}$$

$$\frac{x \in \text{dom } \sigma}{\langle x\texttt{--};\ \sigma\rangle \Downarrow_{\text{exp}} \langle \sigma x - 1,\ [\ \sigma \mid x : \sigma x - 1\ ]\rangle}\ \textsc{VarDec}$$

3. Ejercicio 5.

3.1. Enunciado.

¿Es la relación de evaluación de un paso $\rightsquigarrow$ determinista? Si lo es, demostralo (puede suponer que la relación $\Downarrow_{\text{exp}}$ es determinista). Caso contrario, proveer un contraejemplo.

3.2. Resolución.

Demostraremos que, efectivamente, la evaluación de un paso $\rightsquigarrow$ es determinista. La idea general de la prueba se repite a lo largo de los distintos casos, y consiste en partir de $(t, t') \in \rightsquigarrow$ y analizar la última regla que se aplicó sobre $t \rightsquigarrow t'$ para luego probar que, si elegimos un t'' cualquiera y suponemos que $t \rightsquigarrow t''$, entonces la regla que se utilizó en esta ultima derivación ha de ser la misma. Finalmente, con la información que obtenemos de estas dos proposiciones, demostramos que t' y t'' son iguales, probando entonces que

$$\forall t'' \in \Gamma \quad t \rightsquigarrow t' \ \wedge \ t \rightsquigarrow t'' \Longrightarrow t' = t''$$

Demostración. Veamos que la relación de evaluación en un paso $\rightsquigarrow$ es determinista.
Es decir, queremos probar que si $t \rightsquigarrow t'$ y $t \rightsquigarrow t''$, entonces $t' = t''$.

Lo probaremos por inducción sobre la relación $\rightsquigarrow$. Queremos demostrar que la propiedad

$$P(t, t') : \forall\, t'' \in \Gamma \quad t \rightsquigarrow t'' \implies t' = t''$$

vale para todo par $(t, t') \in \rightsquigarrow$.

Separemos entonces por casos sobre el último paso de derivación en $t \rightsquigarrow t'$.

Regla Ass. Sea $t'' \in \Gamma$ cualquiera y supongamos que $t \rightsquigarrow t'$ se construyó a partir de Ass. Tenemos entonces que

- t es de la forma $\langle v = e, \sigma \rangle$.
- t' es de la forma $\langle \textbf{skip},\ [\sigma' \mid v : n] \rangle$.
- Vale $\langle e, \sigma \rangle \Downarrow_{\text{exp}} \langle n, \sigma' \rangle$.

Notemos que como t es una asignación, la única regla que es posible aplicar es Ass. Luego, si suponemos que $t \rightsquigarrow t''$, necesariamente se habrá aplicado la misma regla. Nuevamente tenemos entonces que

- t es de la forma $\langle v_1 = e_1, \sigma_1 \rangle$.
- t'' es de la forma $\langle \textbf{skip}, [\sigma_1' \mid v_1 : n_1] \rangle$.
- Vale $\langle e_1, \sigma_1 \rangle \Downarrow_{\text{exp}} \langle n_1, \sigma_1' \rangle$.

Luego, sigue de las dos definiciones de t que $v = v_1$, $e = e_1$, $\sigma = \sigma_1$. Ahora, como $\Downarrow_{\text{exp}}$ es determinista y $e = e_1, \sigma = \sigma_1$, tenemos que como $\langle e, \sigma \rangle \Downarrow_{\text{exp}} \langle n, \sigma' \rangle$ y $\langle e_1, \sigma_1 \rangle \Downarrow_{\text{exp}} \langle n_1, \sigma_1' \rangle$, deberá valer que $n = n_1$ y $\sigma' = \sigma_1'$.

Finalmente, de las igualdades anteriores sigue que $\langle \textbf{skip}, [\sigma' \mid v : n] \rangle = \langle \textbf{skip}, [\sigma_1' \mid v_1 : n_1] \rangle$. Es decir, $t' = t''$.

Regla Seq$_1$. Sea t'' cualquiera y supongamos que $t \rightsquigarrow t'$ se construyó a partir de Seq$_1$. Tenemos entonces que

- t es de la forma $\langle \textbf{skip}; c_1, \sigma \rangle$
- t' es de la forma $\langle c_1, \sigma \rangle$

Supongamos ahora que $t \rightsquigarrow t''$, luego por la forma de t la última regla con la que se construyó el par necesariamente es Seq$_1$. Esto implica que

- t es de la forma $\langle \textbf{skip}; c_1', \sigma' \rangle$
- t'' es de la forma $\langle c_1', \sigma' \rangle$

Nuevamente, sigue de comparar ambas definiciones de t que $c_1 = c_1'$ y $\sigma = \sigma'$. Luego, es claro que $t' = t''$.

Regla Seq$_2$. Sea t'' cualquiera y supongamos que $t \rightsquigarrow t'$ se construyó a partir de Seq$_2$. Tenemos entonces que

- t es de la forma $\langle c_0;\ c_1,\ \sigma \rangle$
- t' es de la forma $\langle c_0';\ c_1,\ \sigma' \rangle$
- Vale $\langle c_0, \sigma \rangle \rightsquigarrow \langle c_0', \sigma' \rangle$

Notemos que c_0 no puede tratarse de un **skip**, pues sabemos que $\langle c_0, \sigma \rangle \rightsquigarrow \langle c_0', \sigma' \rangle$ y no existe una regla tal que $\langle \textbf{skip}, \sigma \rangle \rightsquigarrow \langle c_0', \sigma' \rangle$. Entonces, al quedar descartada esa opción, si suponemos que $t \rightsquigarrow t''$, la única regla que puede haber construído al par (t, t'') es nuevamente Seq$_2$.

Ahora, tenemos que

- t es de la forma $\langle c_2;\ c_3,\ \sigma_1 \rangle$
- t'' es de la forma $\langle c_2';\ c_3,\ \sigma_1' \rangle$
- Vale $\langle c_2, \sigma_1 \rangle \rightsquigarrow \langle c_2', \sigma_1' \rangle$

Primero observemos que, comparando las definiciones de t, es claro que

$$c_0 = c_2, \quad c_1 = c_3, \quad \sigma = \sigma_1$$

Pero ademas, como $\langle c_0, \sigma\rangle \rightsquigarrow \langle c_0', \sigma'\rangle$ es una derivación previa, podemos aplicar la hipótesis inductiva sobre ella. Esto es, vale que

$$\forall t''' \in \Gamma \;\; \langle c_0, \sigma\rangle \rightsquigarrow t''' \Longrightarrow \langle c_0', \sigma'\rangle = t'''$$

Siguiendo esta idea, como vimos que vale $\langle c_2, \sigma_1\rangle \rightsquigarrow \langle c_2', \sigma_1'\rangle$ y que $c_0 = c_2$ y $\sigma = \sigma_1$, tenemos que, para $t''' = \langle c_2', \sigma_1'\rangle$,

$$\langle c_0, \sigma\rangle \rightsquigarrow \langle c_2', \sigma_1'\rangle \Longrightarrow \langle c_0', \sigma'\rangle = \langle c_2', \sigma_1'\rangle$$

Es decir,

$$c_0' = c_2', \quad \sigma' = \sigma_1'$$

Finalmente, juntando todas las igualdades anteriores, sigue que

$$t' = \langle c_0';\ c_1,\ \sigma'\rangle = \langle c_2';\ c_3,\ \sigma_1'\rangle = t''$$

como queríamos probar.

Regla If$_1$. Sea t'' cualquiera y supongamos que $t \rightsquigarrow t'$ se construyó a partir de If$_1$. Tenemos entonces que

- t es de la forma $\langle \textbf{if}\ b\ \textbf{then}\ c_0\ \textbf{else}\ c_1,\ \sigma\rangle$
- t' es de la forma $\langle c_0,\ \sigma'\rangle$
- Vale $\langle b, \sigma\rangle \Downarrow_{\text{exp}} \langle \textbf{true},\ \sigma'\rangle$

Supongamos ahora que $t \rightsquigarrow t''$. Observemos que, basándonos en la forma de t, existiría la posibilidad de que la última regla utilizada para construir $t \rightsquigarrow t''$ sea tanto If$_1$ como If$_2$.

Sin embargo, si la regla utilizada fuera If$_2$, debería valer que $\langle b, \sigma\rangle \Downarrow_{\text{exp}} \langle \textbf{false},\ \sigma_1'\rangle$, lo que no es posible pues $\langle b, \sigma\rangle \Downarrow_{\text{exp}} \langle \textbf{true},\ \sigma'\rangle$ y $\Downarrow_{\text{exp}}$ es determinista. Luego, nuevamente la última regla aplicada en ambos pasos de derivación es la misma, If$_1$. Tenemos entonces que

- t es de la forma $\langle \textbf{if}\ b_1\ \textbf{then}\ c_2\ \textbf{else}\ c_3,\ \sigma_1\rangle$
- t'' es de la forma $\langle c_2,\ \sigma_1'\rangle$
- Vale $\langle b_1, \sigma_1\rangle \Downarrow_{\text{exp}} \langle \textbf{true},\ \sigma_1'\rangle$

Sigue que

$$b = b_1, \quad c_0 = c_2, \quad c_1 = c_3, \quad \sigma = \sigma_1$$

Y por el determinismo de $\Downarrow_{\text{exp}}$, utilizando las igualdades anteriores, vale $\sigma' = \sigma_1'$.
De todo lo anterior podemos concluir que $\langle c_0,\ \sigma'\rangle = \langle c_2,\ \sigma_1'\rangle$, es decir, $t' = t''$.

Regla If$_2$. Es análoga a la regla If$_2$.

Regla Repeat. Sea t'' cualquiera y supongamos que $t \rightsquigarrow t'$ se construyó a partir de Repeat. Tenemos entonces que

- t es de la forma $\langle \textbf{repeat}\ c\ \textbf{until}\ b,\ \sigma\rangle$
- t' es de la forma $\langle c;\ \textbf{if}\ b\ \textbf{then skip else repeat}\ c\ \textbf{until}\ b,\ \sigma\rangle$

Si suponemos ahora que $t \rightsquigarrow t''$, por el argumento usual tenemos que la última regla aplicada en esta derivación ha de ser Repeat. Le sigue que

- t es de la forma $\langle \textbf{repeat}\ c_1\ \textbf{until}\ b_1,\ \sigma_1\rangle$
- t'' es de la forma $\langle c_1;\ \textbf{if}\ b_1\ \textbf{then skip else repeat}\ c_1\ \textbf{until}\ b_1,\ \sigma_1\rangle$

donde de la comparación entre las definiciones de t es claro que

$$c = c_1 \quad b = b_1 \quad \sigma = \sigma_1$$

luego t' y t'' son iguales.

4. Ejercicio 6.

4.1. Enunciado.

Pruebe que los siguientes programas son semánticamente equivalentes:

a) `x = x + 1;`
   `y = x`

b) `y = x++`

4.2. Resolución.

Evaluaremos ambos programas para demostrar que dado un estado inicial $\sigma \in \Gamma$ cualquiera, ambos llegan a la configuración final $\langle \texttt{skip},\ [[\ \sigma \mid x : \sigma x + 1\ ] \mid y : \sigma x + 1\ ]\rangle$.

Consideremos entonces $\sigma \in \Sigma$ un estado tal que $x \in \mathrm{dom}\ \sigma$, pues de otra manera no podría aplicar Var en las derivaciones. Evaluemos los programas.

4.2.1. Programa A.

Podemos escribir al programa A como:

$$\langle x = x + 1;\ y = x,\ \sigma\rangle.$$

Probemos primero que $\langle x = x + 1;\ y = x,\ \sigma\rangle \rightsquigarrow \langle \texttt{skip};\ y = x,\ [\ \sigma \mid x : \sigma x + 1\ ]\rangle$

$$\dfrac{\dfrac{\dfrac{\dfrac{x \in \mathrm{dom}\ \sigma}{\langle x, \sigma\rangle \Downarrow_{\text{exp}} \langle \sigma x,\ \sigma\rangle}\ \textsc{Var} \qquad \dfrac{}{\langle 1,\ \sigma\rangle \Downarrow_{\text{exp}} \langle 1,\ \sigma\rangle}\ \textsc{NVal}}{\langle x + 1,\ \sigma\rangle \Downarrow_{\text{exp}} \langle \sigma x + 1,\ \sigma\rangle}\ \textsc{Plus}}{\langle x = x + 1,\ \sigma\rangle \rightsquigarrow \langle \texttt{skip},\ [\ \sigma \mid x : \sigma x + 1\ ]\rangle}\ \textsc{Ass}}{\langle x = x + 1;\ y = x,\ \sigma\rangle \rightsquigarrow \langle \texttt{skip};\ y = x,\ [\ \sigma \mid x : \sigma x + 1\ ]\rangle}\ \textsc{Seq}_2$$

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

$$\dfrac{}{\langle \texttt{skip};\ y = x,\ [\ \sigma \mid x : \sigma x + 1\ ]\rangle \rightsquigarrow \langle y = x,\ [\ \sigma \mid x : \sigma x + 1\ ]\rangle}\ \textsc{Seq}_1$$

Y por último, $\langle y = x,\ [\ \sigma \mid x : \sigma x + 1\ ]\rangle \rightsquigarrow \langle \texttt{skip}, [[\ \sigma \mid x : \sigma x + 1\ ] \mid y : \sigma x + 1\ ]\rangle$

Observemos que por definición de $[\sigma \mid x : \sigma x + 1]$ y como $x \in \text{dom } \sigma$, vale $x \in \text{dom } [\sigma \mid x : \sigma x + 1]$.

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

(**Obs.** Aplicamos $[\ \sigma \mid x : \sigma x + 1\ ]\ x = \sigma x + 1$)

4.2.2. Programa B.

Podemos escribir al programa B como:

$$\langle y = x\texttt{++},\ \sigma\rangle$$

Probaremos que $\langle y = x\texttt{++},\ \sigma\rangle \rightsquigarrow \langle \texttt{skip},\ [[\ \sigma \mid x : \sigma x + 1\ ] \mid y : \sigma x + 1\ ]\rangle$

$$\dfrac{\dfrac{x \in \text{dom } \sigma}{\langle x\texttt{++},\ \sigma\rangle \Downarrow_{\text{exp}} \langle \sigma x + 1,\ [\ \sigma \mid x : \sigma x + 1\ ]\rangle}\ \textsc{VarInc}}{\langle y = x\texttt{++},\ \sigma\rangle \rightsquigarrow \langle \texttt{skip},\ [[\ \sigma \mid x : \sigma x + 1\ ] \mid y : \sigma x + 1\ ]\rangle}\ \textsc{Ass}$$