

TP1: Lenguaje imperativo simple

Grillo, Libonati, Maiza

16/09/24

EJERCICIO 1

Sintaxis abstracta

```
intexp ::= ...
        | intexp ++
        | intexp -
```

Sintaxis concreta

```
intexp ::= ...
        | intexp '+' '+'
        | intexp '-' '-'
```

EJERCICIO 4

$$\frac{x \in \text{dom } \sigma}{\langle x ++, \sigma \rangle \Downarrow_{exp} \langle \sigma \ x ++ 1, [\sigma \mid x : \sigma \ x ++ 1] \rangle} \text{VARINC}$$
$$\frac{x \in \text{dom } \sigma}{\langle x --, \sigma \rangle \Downarrow_{exp} \langle \sigma \ x -- 1, [\sigma \mid x : \sigma \ x -- 1] \rangle} \text{VARDEC}$$

EJERCICIO 5

Queremos ver que si $t \rightsquigarrow t'$ y $t \rightsquigarrow t''$, entonces $t' = t''$.
Para ello haremos inducción sobre la derivación $t \rightsquigarrow t'$.

HI) para toda subderivación de $t \rightsquigarrow t'$ se verifica la regla.

Si la última derivación de $t \rightsquigarrow t'$ usa la regla:

- *ASS*: Tenemos que t tiene la forma $\langle v = e, \sigma \rangle$ y t' tiene la forma $\langle \text{skip}, [\sigma' \mid v : n] \rangle$. Por la forma de t , en la derivación $t \rightsquigarrow t''$ la última regla aplicada solo puede haber sido *ASS*, ya que no hay otra regla donde t pueda ser una asignación. Luego, como \Downarrow_{exp} es determinista, resulta $t' = t''$.
- *SEQ₁*: Tenemos que t tiene la forma $\langle \text{skip}; c_1, \sigma \rangle$ y t' tiene la forma $\langle c_1, \sigma \rangle$. Por la forma de t , en la derivación $t \rightsquigarrow t''$ la última regla aplicada solo puede haber sido *SEQ₁*, ya que no hay otra regla donde t pueda ser una secuenciación con un skip. Luego, $t' = t''$.
- *SEQ₂*: Tenemos que t tiene la forma $\langle c_0; c_1, \sigma \rangle$ y t' tiene la forma $\langle c'_0; c_1, \sigma' \rangle$. Por la forma de t , la última derivación en $t \rightsquigarrow t''$ debe ser aplicando la regla *SEQ₂*, donde la forma de t'' es $\langle c''_0; c_1, \sigma'' \rangle$. Nuestra HI es que si $\langle c_0, \sigma \rangle \rightsquigarrow \langle c'_0, \sigma' \rangle$ y $\langle c_0, \sigma \rangle \rightsquigarrow \langle c''_0, \sigma'' \rangle$, entonces $\langle c'_0, \sigma' \rangle = \langle c''_0, \sigma'' \rangle$. Luego por HI, $t' = \langle c'_0; c_1, \sigma' \rangle = \langle c''_0; c_1, \sigma'' \rangle = t''$.
- *IF₁*: Tenemos que t tiene la forma $\langle \text{if } b \text{ then } c_0 \text{ else } c_1, \sigma \rangle$ y t' tiene la forma $\langle c_0, \sigma' \rangle$. Por la forma de t , la última regla aplicada en la derivación $t \rightsquigarrow t''$ debe ser *IF₁*, ya que, como \Downarrow_{exp} es determinista, $\langle b, \sigma' \rangle \Downarrow_{exp} \langle \text{true}, \sigma' \rangle$. Por lo tanto, debe ser $t' = t''$.

- IF_2 : Tenemos que t tiene la forma $\langle \text{if } b \text{ then } c_0 \text{ else } c_1, \sigma \rangle$ y t' tiene la forma $\langle c_1, \sigma' \rangle$. Por la forma de t , la última regla aplicada en la derivación $t \rightsquigarrow t''$ debe ser IF_2 , ya que, como \Downarrow_{exp} es determinista, $\langle b, \sigma' \rangle \Downarrow_{exp} \langle false, \sigma' \rangle$. Por lo tanto, debe ser $t' = t''$.
- $REPEAT$: Tenemos que t tiene la forma $\langle \text{repeat } c \text{ until } b, \sigma \rangle$ y t' tiene la forma $\langle c; \text{if } b \text{ then skip else repeat } c \text{ until } b, \sigma \rangle$. Por la forma de t , la última regla aplicada en la derivación $t \rightsquigarrow t''$ debe ser $REPEAT$, y por ende $t' = t''$.

Hemos probado que para cada posible regla aplicada en la derivación $t \rightsquigarrow t'$ se verifica lo planteado, con lo cual queda probado que si $t \rightsquigarrow t'$ y $t \rightsquigarrow t''$, entonces $t' = t''$, esto es, la relación de evaluación en un paso \rightsquigarrow es determinista.

EJERCICIO 6

Construiremos los árboles de derivación de cada programa utilizando las siguientes reglas de inferencia para la relación \rightsquigarrow^* :

$$\frac{t \rightsquigarrow t'}{t \rightsquigarrow^* t'} RT_1 \quad \frac{}{t \rightsquigarrow^* t} RT_2 \quad \frac{t \rightsquigarrow^* t' \quad t' \rightsquigarrow^* t''}{t \rightsquigarrow^* t''} RT_3$$

Para el programa **a)** tenemos el siguiente árbol:

$$\frac{\frac{\frac{x \in dom \sigma}{\langle x, \sigma \rangle \Downarrow_{exp} \langle \sigma x, \sigma \rangle} VAR \quad \frac{\langle 1, \sigma \rangle \Downarrow_{exp} \langle 1, \sigma \rangle}{PLUS} NVAL}{\langle x+1, \sigma \rangle \Downarrow_{exp} \langle \sigma x+1, \sigma \rangle} ASS \quad \frac{\frac{\langle x=x+1, \sigma \rangle \rightsquigarrow \langle skip, [\sigma \mid x : \sigma x+1] \rangle}{\langle x=x+1; y=x, \sigma \rangle \rightsquigarrow \langle skip; y=x, [\sigma \mid x : \sigma x+1] \rangle} SEQ_2 \quad \frac{\langle skip; y=x, [\sigma \mid x : \sigma x+1] \rangle \rightsquigarrow \langle y=x, [\sigma \mid x : \sigma x+1] \rangle}{RT_1} SEQ_1}{\langle x=x+1; y=x, \sigma \rangle \rightsquigarrow^* \langle skip; y=x, [\sigma \mid x : \sigma x+1] \rangle \rightsquigarrow^* \langle y=x, [\sigma \mid x : \sigma x+1] \rangle} RT_1 \quad \frac{\frac{x \in dom [\sigma \mid x : \sigma x+1]}{\langle x, [\sigma \mid x : \sigma x+1] \rangle \Downarrow_{exp} \langle \sigma x, [\sigma \mid x : \sigma x+1] \rangle} VAR \quad \frac{\langle y=x, [\sigma \mid x : \sigma x+1] \rangle \rightsquigarrow \langle skip, [\sigma \mid x : \sigma x+1, y : \sigma x+1] \rangle}{RT_1} ASS}{\langle y=x, [\sigma \mid x : \sigma x+1] \rangle \rightsquigarrow^* \langle skip, [\sigma \mid x : \sigma x+1, y : \sigma x+1] \rangle} RT_3} \frac{\langle x=x+1; y=x, \sigma \rangle \rightsquigarrow^* \langle y=x, [\sigma \mid x : \sigma x+1] \rangle \rightsquigarrow^* \langle skip, [\sigma \mid x : \sigma x+1, y : \sigma x+1] \rangle}{RT_3}$$

Y para el programa **b)** tenemos:

$$\frac{\frac{\frac{x \in dom \sigma}{\langle x++, \sigma \rangle \Downarrow_{exp} \langle \sigma x+1, [\sigma \mid x : \sigma x+1] \rangle} VAR INC}{\langle y=x++, \sigma \rangle \rightsquigarrow \langle skip, [\sigma \mid x : \sigma x+1, y : \sigma x+1] \rangle} ASS}{\langle y=x++, \sigma \rangle \rightsquigarrow^* \langle skip, [\sigma \mid x : \sigma x+1, y : \sigma x+1] \rangle} RT_1$$

Luego, los programas con semánticamente equivalentes.