

# Trabajo Práctico 1

## Análisis de Lenguajes de Programación

**Alumnas:**

Cipullo, Inés

Sullivan, Katherine

Universidad Nacional de Rosario

2021

## Ejercicio 1

### Gramática Abstracta

$$\begin{aligned} \text{intexp} &::= \text{nat} \mid \text{var} \mid -_u \text{intexp} \\ &\mid \text{intexp} + \text{intexp} \\ &\mid \text{intexp} -_b \text{intexp} \\ &\mid \text{intexp} \times \text{intexp} \\ &\mid \text{intexp} \div \text{intexp} \\ &\mid \text{var} = \text{intexp} \\ &\mid \text{intexp}, \text{intexp} \\ \text{boolexp} &::= \mathbf{true} \mid \mathbf{false} \\ &\mid \text{intexp} == \text{intexp} \\ &\mid \text{intexp} \neq \text{intexp} \\ &\mid \text{intexp} < \text{intexp} \\ &\mid \text{intexp} > \text{intexp} \\ &\mid \text{boolexp} \wedge \text{boolexp} \\ &\mid \text{boolexp} \vee \text{boolexp} \\ &\mid \neg \text{boolexp} \\ \text{comm} &::= \mathbf{skip} \\ &\mid \text{var} = \text{intexp} \\ &\mid \text{comm}; \text{comm} \\ &\mid \mathbf{if} \text{ boolexp} \mathbf{then} \text{ comm} \mathbf{else} \text{ comm} \\ &\mid \mathbf{repeat} \text{ comm} \mathbf{until} \text{ boolexp} \end{aligned}$$

### Gramática Concreta

Estan mal algunos simbolos

```

digit ::= '0' | '1' | ... | '9'
letter ::= 'a' | ... | 'Z'
nat ::= digit | digit nat
var ::= letter | letter var
intexp ::= nat
        | var
        | '-' intexp
        | intexp '+' intexp
        | intexp '-' intexp
        | intexp '*' intexp
        | intexp '/' intexp
        | '(' intexp ')'
        | var '=' intexp
        | intexp ',' intexp
boolexp ::= 'true' | 'false'
        | intexp '==' intexp
        | intexp '!=' intexp
        | intexp '<' intexp
        | intexp '>' intexp
        | boolexp '&&' boolexp
        | boolexp '||' boolexp
        | '!' boolexp
        | '(' boolexp ')'
comm ::= skip
        | var '=' intexp
        | comm ';' comm
        | 'if' boolexp '{' comm '}'
        | 'if' boolexp '{' comm '}' 'else' '{' comm '}'
        | 'repeat' '{' comm '}' 'until' boolexp

```

## Ejercicio 2

Realizado en archivo `src/AST.hs`.

## Ejercicio 3

Realizado en archivo `src/Parser.hs`

## Ejercicio 4

### Semántica Operacional Big-Step para Expresiones

$$\begin{array}{c}
\frac{}{\langle nv, \sigma \rangle \Downarrow_{exp} \langle \mathbf{n}v, \sigma \rangle} \text{ NVAL} \qquad \frac{}{\langle x, \sigma \rangle \Downarrow_{exp} \langle \sigma x, \sigma \rangle} \text{ VAR} \\
\\
\frac{\langle e, \sigma \rangle \Downarrow_{exp} \langle n, \sigma' \rangle}{\langle -_u e, \sigma \rangle \Downarrow_{exp} \langle -n, \sigma' \rangle} \text{ UMINUS} \qquad \frac{\langle e_0, \sigma \rangle \Downarrow_{exp} \langle n_0, \sigma' \rangle \quad \langle e_1, \sigma' \rangle \Downarrow_{exp} \langle n_1, \sigma'' \rangle}{\langle e_0 + e_1, \sigma \rangle \Downarrow_{exp} \langle n_0 + n_1, \sigma'' \rangle} \text{ PLUS} \\
\\
\frac{\langle e_0, \sigma \rangle \Downarrow_{exp} \langle n_0, \sigma' \rangle \quad \langle e_1, \sigma' \rangle \Downarrow_{exp} \langle n_1, \sigma'' \rangle}{\langle e_0 - e_1, \sigma \rangle \Downarrow_{exp} \langle n_0 - n_1, \sigma'' \rangle} \text{ BMINUS} \\
\\
\frac{\langle e_0, \sigma \rangle \Downarrow_{exp} \langle n_0, \sigma' \rangle \quad \langle e_1, \sigma' \rangle \Downarrow_{exp} \langle n_1, \sigma'' \rangle}{\langle e_0 * e_1, \sigma \rangle \Downarrow_{exp} \langle n_0 * n_1, \sigma'' \rangle} \text{ MULT} \\
\\
\frac{\langle e_0, \sigma \rangle \Downarrow_{exp} \langle n_0, \sigma' \rangle \quad \langle e_1, \sigma' \rangle \Downarrow_{exp} \langle n_1, \sigma'' \rangle \quad n_1 \neq 0}{\langle e_0 \div e_1, \sigma \rangle \Downarrow_{exp} \langle n_0 \div n_1, \sigma'' \rangle} \text{ DIV} \\
\\
\frac{\langle e_0, \sigma \rangle \Downarrow_{exp} \langle n_0, \sigma' \rangle \quad \langle e_1, \sigma' \rangle \Downarrow_{exp} \langle n_1, \sigma'' \rangle}{\langle e_0 > e_1, \sigma \rangle \Downarrow_{exp} \langle n_0 > n_1, \sigma'' \rangle} \text{ GT} \\
\\
\frac{\langle e_0, \sigma \rangle \Downarrow_{exp} \langle n_0, \sigma' \rangle \quad \langle e_1, \sigma' \rangle \Downarrow_{exp} \langle n_1, \sigma'' \rangle}{\langle e_0 < e_1, \sigma \rangle \Downarrow_{exp} \langle n_0 < n_1, \sigma'' \rangle} \text{ LT} \\
\\
\frac{\langle e_0, \sigma \rangle \Downarrow_{exp} \langle n_0, \sigma' \rangle \quad \langle e_1, \sigma' \rangle \Downarrow_{exp} \langle n_1, \sigma'' \rangle}{\langle e_0 \neq e_1, \sigma \rangle \Downarrow_{exp} \langle n_0 \neq n_1, \sigma'' \rangle} \text{ NOTEQ} \\
\\
\frac{\langle e_0, \sigma \rangle \Downarrow_{exp} \langle n_0, \sigma' \rangle \quad \langle e_1, \sigma' \rangle \Downarrow_{exp} \langle n_1, \sigma'' \rangle}{\langle e_0 == e_1, \sigma \rangle \Downarrow_{exp} \langle n_0 == n_1, \sigma'' \rangle} \text{ EQ} \qquad \frac{}{\langle bv, \sigma \rangle \Downarrow_{exp} \langle bv, \sigma \rangle} \text{ BVAL} \\
\\
\frac{\langle p, \sigma \rangle \Downarrow_{exp} \langle b, \sigma' \rangle}{\langle \neg p, \sigma \rangle \Downarrow_{exp} \langle \neg b, \sigma' \rangle} \text{ NOT} \qquad \frac{\langle p_0, \sigma \rangle \Downarrow_{exp} \langle b_0, \sigma' \rangle \quad \langle p_1, \sigma' \rangle \Downarrow_{exp} \langle b_1, \sigma'' \rangle}{\langle e_0 \vee e_1, \sigma \rangle \Downarrow_{exp} \langle b_0 \vee b_1, \sigma'' \rangle} \text{ OR} \\
\\
\frac{\langle p_0, \sigma \rangle \Downarrow_{exp} \langle b_0, \sigma' \rangle \quad \langle p_1, \sigma' \rangle \Downarrow_{exp} \langle b_1, \sigma'' \rangle}{\langle e_0 \wedge e_1, \sigma \rangle \Downarrow_{exp} \langle b_0 \wedge b_1, \sigma'' \rangle} \text{ AND}
\end{array}$$

Llamamos IASS a la asignación como expresión y ISEQ a la secuencialización de expresiones con el operador ,

$$\frac{\langle e, \sigma \rangle \Downarrow_{exp} \langle n, \sigma' \rangle}{\langle v = e, \sigma \rangle \Downarrow_{exp} \langle n, [\sigma' | v : n] \rangle} \text{ IASS}$$

$$\frac{\langle e_0, \sigma \rangle \Downarrow_{exp} \langle n_0, \sigma' \rangle \quad \langle e_1, \sigma' \rangle \Downarrow_{exp} \langle n_1, \sigma'' \rangle}{\langle e_0, e_1, \sigma \rangle \Downarrow_{exp} \langle n_1, \sigma'' \rangle} \text{ ISEQ}$$

## Ejercicio 5

### Determinismo de la relación de evaluación en un paso $\rightsquigarrow$

Queremos probar que dado un comando  $c$  y un estado  $\sigma$

$$\langle c, \sigma \rangle \rightsquigarrow \langle c', \sigma' \rangle \wedge \langle c, \sigma \rangle \rightsquigarrow \langle c'', \sigma'' \rangle \implies c' = c'' \wedge \sigma' = \sigma'' \quad (1)$$

Para probarlo haremos inducción estructural sobre  $c$

#### Caso 1:

Supongamos

$$c := \mathbf{skip}$$

Como un comando de la forma **skip** termina la ejecución, este no deriva en un paso a ningún otro comando, por lo tanto, por premisas falsas en una implicancia resulta verdadero (1).

#### Caso 2:

Supongamos

$$c := v = e$$

Por la forma de  $c$  la única regla de semántica operacional que se le puede aplicar resulta

$$\frac{\langle e, \sigma \rangle \Downarrow_{exp} \langle n, \sigma_1 \rangle}{\langle c, \sigma \rangle \rightsquigarrow \langle \mathbf{skip}, [\sigma_1 | v : n] \rangle} \text{ ASS}$$

De lo que entonces concluimos que necesariamente

$$c' = c'' = \mathbf{skip}$$

$$\sigma' = \sigma'' = \sigma_1$$

por lo que vale (1).

#### Caso 3:

Supongamos

$$c := c_1; c_2$$

donde  $c_1$  y  $c_2$  son comandos.

De aquí tenemos dos opciones: que  $c_1$  sea el comando **skip** o no.

Si  $c_1$  es **skip**, luego, como **skip** no deriva en un paso a ningún otro comando, la única regla que podemos usar dada la estructura de  $c$  es

$$\frac{}{\langle \mathbf{skip}; c_2, \sigma \rangle \rightsquigarrow \langle c_2, \sigma \rangle} \text{SEQ}_1$$

De lo que entonces concluimos que necesariamente

$$c' = c'' = c_2$$

$$\sigma' = \sigma'' = \sigma$$

por lo que vale (1).

Ahora bien, si  $c_1$  no es **skip** resulta que la única regla que podemos usar es

$$\frac{\langle c_1, \sigma \rangle \rightsquigarrow \langle c'_1, \sigma_1 \rangle}{\langle c_1; c_2, \sigma \rangle \rightsquigarrow \langle c'_1; c_2, \sigma_1 \rangle} \text{SEQ}_2$$

Por HI, sabemos que  $c'_1$  es al único comando que puede derivar en un paso  $c_1$ . Por lo tanto, sabemos que necesariamente

$$c' = c'' = c'_1; c_2$$

$$\sigma' = \sigma'' = \sigma$$

por lo que vale (1).

#### Caso 4:

Supongamos

$$c := \mathbf{if } b \mathbf{ then } c_1 \mathbf{ else } c_2$$

Tenemos dos opciones:

$$\langle b, \sigma \rangle \Downarrow_{exp} \langle \mathbf{true}, \sigma_1 \rangle \quad (2)$$

$$\langle b, \sigma \rangle \Downarrow_{exp} \langle \mathbf{false}, \sigma_1 \rangle \quad (3)$$

Como la relación  $\Downarrow_{exp}$  es determinista si vale (2) no vale (3) y viceversa.

Supongamos primero que vale (2). Luego por esto y por la forma de  $c$  la única regla de evaluación que se puede aplicar es

$$\frac{\langle b, \sigma \rangle \Downarrow_{exp} \langle \mathbf{true}, \sigma_1 \rangle}{\langle \mathbf{if } b \mathbf{ then } c_1 \mathbf{ else } c_2, \sigma \rangle \rightsquigarrow \langle c_1, \sigma_1 \rangle} \text{IF}_1$$

De lo que resulta que necesariamente

$$c' = c'' = c_1$$

$$\sigma' = \sigma'' = \sigma_1$$

por lo que vale (1).

Ahora, supongamos que en cambio vale (3), luego la única regla de la que podemos hacer uso es

$$\frac{\langle b, \sigma \rangle \Downarrow_{exp} \langle \mathbf{false}, \sigma_1 \rangle}{\langle \mathbf{if } b \mathbf{ then } c_1 \mathbf{ else } c_2, \sigma \rangle \rightsquigarrow \langle c_2, \sigma'_1 \rangle} \text{IF}_2$$

De lo que resulta que necesariamente

$$c' = c'' = c_2$$

$$\sigma' = \sigma'' = \sigma_1$$

por lo que vale (1).

### Caso 5:

Supongamos

$$c := \text{repeat } c_1 \text{ until } b$$

Por la forma de  $c$  la única regla que puede ser utilizada es

$$\frac{}{\langle \text{repeat } c_1 \text{ until } b, \sigma \rangle \rightsquigarrow \langle c_1; \text{if } b \text{ then skip else repeat } b \text{ until } c_1, \sigma \rangle} \text{ REPEAT}$$

De donde resulta necesariamente

$$c' = c'' = c_1; \text{if } b \text{ then skip else repeat } b \text{ until } c_1$$

$$\sigma' = \sigma'' = \sigma$$

por lo que vale (1).

Queda así demostrado el determinismo de la relación de evaluación en un paso  $\rightsquigarrow$ .

## Ejercicio 6

Al hablar de la clausura transitiva de  $\rightsquigarrow$  tenemos las siguientes reglas

$$\frac{\langle c, \sigma \rangle \rightsquigarrow \langle c', \sigma' \rangle}{\langle c, \sigma \rangle \rightsquigarrow^* \langle c', \sigma' \rangle} T_1 \qquad \frac{\langle c, \sigma \rangle \rightsquigarrow^* \langle c', \sigma' \rangle \quad \langle c', \sigma' \rangle \rightsquigarrow^* \langle c'', \sigma'' \rangle}{\langle c, \sigma \rangle \rightsquigarrow^* \langle c'', \sigma'' \rangle} T_2$$

**A:**

$$a := \text{repeat } x = x - y \text{ until } x == 0$$

$$\frac{\frac{}{\langle \text{skip}; \mathbf{a}, [[\sigma|x:1]|y:1] \rangle \rightsquigarrow \langle \mathbf{a}, [[\sigma|x:1]|y:1] \rangle} \text{SEQ}_1}{\langle \text{skip}; \mathbf{a}, [[\sigma|x:1]|y:1] \rangle \rightsquigarrow^* \langle \mathbf{a}, [[\sigma|x:1]|y:1] \rangle} T_1$$

**B:**

$$a := \text{repeat } x = x - y \text{ until } x == 0$$

$$\frac{\frac{}{\langle \mathbf{a}, [[\sigma|x:1]|y:1] \rangle \rightsquigarrow \langle x = x - y; \text{if } x == 0 \text{ then skip else } \mathbf{a}, [[\sigma|x:1]|y:1] \rangle} \text{REPEAT}}{\langle \mathbf{a}, [[\sigma|x:1]|y:1] \rangle \rightsquigarrow^* \langle x = x - y; \text{if } x == 0 \text{ then skip else } \mathbf{a}, [[\sigma|x:1]|y:1] \rangle} T_1$$

**C:**

$$c := \text{if } x == 0 \text{ then skip else repeat } x = x - y \text{ until } x == 0$$

$$\begin{array}{c}
\frac{}{\langle x, [[\sigma|x:1]|y:1] \rangle \Downarrow_{exp} \langle \mathbf{1}, [[\sigma|x:1]|y:1] \rangle} \text{VAR} \quad \frac{}{\langle y, [[\sigma|x:1]|y:1] \rangle \Downarrow_{exp} \langle \mathbf{1}, [[\sigma|x:1]|y:1] \rangle} \text{VAR} \\
\hline
\frac{}{\langle x-y, [[\sigma|x:1]|y:1] \rangle \Downarrow_{exp} \langle \mathbf{0}, [[\sigma|x:1]|y:1] \rangle} \text{MINUS} \\
\frac{}{\langle x=x-y, [[\sigma|x:1]|y:1] \rangle \rightsquigarrow \langle \mathbf{skip}, [[\sigma|x:0]|y:1] \rangle} \text{ASS} \\
\frac{}{\langle x=x-y; \mathbf{c}, [[\sigma|x:1]|y:1] \rangle \rightsquigarrow \langle \mathbf{skip}; \mathbf{c}, [[\sigma|x:0]|y:1] \rangle} \text{SEQ}_2 \\
\frac{}{\langle x=x-y; \mathbf{c}, [[\sigma|x:1]|y:1] \rangle \rightsquigarrow^* \langle \mathbf{skip}; \mathbf{c}, [[\sigma|x:0]|y:1] \rangle} \text{T}_1
\end{array}$$

**D:** $a := \text{repeat } x = x - y \text{ until } x == 0$ 

$$\begin{array}{c}
\frac{}{\langle \mathbf{1}, [[\sigma|x:2]|y:2] \rangle \Downarrow_{exp} \langle \mathbf{1}, [[\sigma|x:2]|y:2] \rangle} \text{NVAL} \\
\frac{}{\langle y=1, [[\sigma|x:2]|y:2] \rangle \Downarrow_{exp} \langle \mathbf{1}, [[\sigma|x:2]|y:1] \rangle} \text{IASS} \\
\frac{}{\langle x=y=1, [[\sigma|x:2]|y:2] \rangle \rightsquigarrow \langle \mathbf{skip}, [[\sigma|x:1]|y:1] \rangle} \text{ASS} \\
\frac{}{\langle x=y=1; \mathbf{a}, [[\sigma|x:2]|y:2] \rangle \rightsquigarrow \langle \mathbf{skip}; \mathbf{a}, [[\sigma|x:1]|y:1] \rangle} \text{SEQ}_2 \\
\frac{}{\langle x=y=1; \mathbf{a}, [[\sigma|x:2]|y:2] \rangle \rightsquigarrow^* \langle \mathbf{skip}; \mathbf{a}, [[\sigma|x:1]|y:1] \rangle} \text{T}_1
\end{array}$$

**E:** $c := \text{if } x == 0 \text{ then skip else repeat } x = x - y \text{ until } x == 0$ 

$$\frac{}{\langle \mathbf{skip}; \mathbf{c}, [[\sigma|x:0]|y:1] \rangle \rightsquigarrow \langle \mathbf{c}, [[\sigma|x:0]|y:1] \rangle} \text{SEQ}_1 \\
\frac{}{\langle \mathbf{skip}; \mathbf{c}, [[\sigma|x:0]|y:1] \rangle \rightsquigarrow^* \langle \mathbf{c}, [[\sigma|x:0]|y:1] \rangle} \text{T}_1$$

**F:** $c := \text{if } x == 0 \text{ then skip else repeat } x = x - y \text{ until } x == 0$ 

$$\begin{array}{c}
\frac{}{\langle x, [[\sigma|x:0]|y:1] \rangle \Downarrow_{exp} \langle \mathbf{0}, [[\sigma|x:0]|y:1] \rangle} \text{VAR} \quad \frac{}{\langle 0, [[\sigma|x:0]|y:1] \rangle \Downarrow_{exp} \langle \mathbf{0}, [[\sigma|x:0]|y:1] \rangle} \text{NVAL} \\
\hline
\frac{}{\langle x==0, [[\sigma|x:0]|y:1] \rangle \Downarrow_{exp} \langle \mathbf{true}, [[\sigma|x:0]|y:1] \rangle} \text{EQ} \\
\frac{}{\langle \mathbf{c}, [[\sigma|x:0]|y:1] \rangle \rightsquigarrow \langle \mathbf{skip}, [[\sigma|x:0]|y:1] \rangle} \text{IF}_1 \\
\frac{}{\langle \mathbf{c}, [[\sigma|x:0]|y:1] \rangle \rightsquigarrow^* \langle \mathbf{skip}, [[\sigma|x:0]|y:1] \rangle} \text{T}_1
\end{array}$$

**DEM:** $a := \text{repeat } x = x - y \text{ until } x == 0$  $c := \text{if } x == 0 \text{ then skip else repeat } x = x - y \text{ until } x == 0$ 

$$\begin{array}{c}
\frac{}{\langle x=y=1; \mathbf{a}, [[\sigma|x:2]|y:2] \rangle \rightsquigarrow^* \langle \mathbf{a}, [[\sigma|x:1]|y:1] \rangle} \text{D} \quad \frac{}{\langle x=y=1; \mathbf{a}, [[\sigma|x:2]|y:2] \rangle \rightsquigarrow^* \langle \mathbf{a}, [[\sigma|x:1]|y:1] \rangle} \text{A} \\
\frac{}{\langle x=y=1; \mathbf{a}, [[\sigma|x:2]|y:2] \rangle \rightsquigarrow^* \langle x=x-y; \mathbf{c}, [[\sigma|x:1]|y:1] \rangle} \text{T}_2 \quad \frac{}{\langle x=y=1; \mathbf{a}, [[\sigma|x:2]|y:2] \rangle \rightsquigarrow^* \langle x=x-y; \mathbf{c}, [[\sigma|x:1]|y:1] \rangle} \text{B} \\
\frac{}{\langle x=y=1; \mathbf{a}, [[\sigma|x:2]|y:2] \rangle \rightsquigarrow^* \langle \mathbf{skip}; \mathbf{c}, [[\sigma|x:0]|y:1] \rangle} \text{T}_2 \quad \frac{}{\langle x=y=1; \mathbf{a}, [[\sigma|x:2]|y:2] \rangle \rightsquigarrow^* \langle \mathbf{skip}; \mathbf{c}, [[\sigma|x:0]|y:1] \rangle} \text{C} \\
\frac{}{\langle x=y=1; \mathbf{a}, [[\sigma|x:2]|y:2] \rangle \rightsquigarrow^* \langle \mathbf{c}, [[\sigma|x:0]|y:1] \rangle} \text{T}_2 \quad \frac{}{\langle x=y=1; \mathbf{a}, [[\sigma|x:2]|y:2] \rangle \rightsquigarrow^* \langle \mathbf{c}, [[\sigma|x:0]|y:1] \rangle} \text{E} \\
\frac{}{\langle x=y=1; \mathbf{a}, [[\sigma|x:2]|y:2] \rangle \rightsquigarrow^* \langle \mathbf{skip}, [[\sigma|x:0]|y:1] \rangle} \text{T}_2 \quad \frac{}{\langle x=y=1; \mathbf{a}, [[\sigma|x:2]|y:2] \rangle \rightsquigarrow^* \langle \mathbf{skip}, [[\sigma|x:0]|y:1] \rangle} \text{F}
\end{array}$$

## Ejercicio 7

Realizado en el archivo `src/Eval1.hs`.



## Ejercicio 8

Realizado en el archivo `src/Eval2.hs`.

## Ejercicio 9

Realizado en el archivo `src/Eval3.hs`.

## Ejercicio 10

Nueva gramática abstracta

$$\begin{aligned}
 \text{intexp} &::= \text{nat} \mid \text{var} \mid \neg_u \text{intexp} \\
 &\mid \text{intexp} + \text{intexp} \\
 &\mid \text{intexp} -_b \text{intexp} \\
 &\mid \text{intexp} \times \text{intexp} \\
 &\mid \text{intexp} \div \text{intexp} \\
 &\mid \text{var} = \text{intexp} \\
 &\mid \text{intexp}, \text{intexp} \\
 \text{boolexp} &::= \mathbf{true} \mid \mathbf{false} \\
 &\mid \text{intexp} == \text{intexp} \\
 &\mid \text{intexp} \neq \text{intexp} \\
 &\mid \text{intexp} < \text{intexp} \\
 &\mid \text{intexp} > \text{intexp} \\
 &\mid \text{boolexp} \wedge \text{boolexp} \\
 &\mid \text{boolexp} \vee \text{boolexp} \\
 &\mid \neg \text{boolexp} \\
 \text{comm} &:= \mathbf{skip} \\
 &\mid \text{var} = \text{intexp} \\
 &\mid \text{comm}; \text{comm} \\
 &\mid \mathbf{if} \text{ boolexp} \mathbf{then} \text{ comm} \mathbf{else} \text{ comm} \\
 &\mid \mathbf{repeat} \text{ comm} \mathbf{until} \text{ boolexp} \\
 &\mid \mathbf{for} (\text{intexp}; \text{boolexp}; \text{intexp}) \text{ comm}
 \end{aligned}$$

Nueva semántica operacional para comandos

$$\begin{aligned}
 &\frac{\langle e, \sigma \rangle \Downarrow_{exp} \langle n, \sigma' \rangle}{\langle v = e, \sigma \rangle \rightsquigarrow \langle \mathbf{skip}, [\sigma' | v : n] \rangle} \text{ ASS} \\
 &\frac{}{\langle \mathbf{skip}; c_1, \sigma \rangle \rightsquigarrow \langle c_1, \sigma \rangle} \text{ SEQ}_1 \\
 &\frac{\langle c_0, \sigma \rangle \rightsquigarrow \langle c'_0, \sigma' \rangle}{\langle c_0; c_1, \sigma \rangle \rightsquigarrow \langle c'_0; c_1, \sigma' \rangle} \text{ SEQ}_2
 \end{aligned}$$

$$\begin{array}{c}
\frac{\langle b, \sigma \rangle \Downarrow_{exp} \langle \mathbf{true}, \sigma' \rangle}{\langle \mathbf{if } b \mathbf{ then } c_0 \mathbf{ else } c_1, \sigma \rangle \rightsquigarrow \langle c_0, \sigma' \rangle} \text{IF}_1 \qquad \frac{\langle b, \sigma \rangle \Downarrow_{exp} \langle \mathbf{false}, \sigma' \rangle}{\langle \mathbf{if } b \mathbf{ then } c_0 \mathbf{ else } c_1, \sigma \rangle \rightsquigarrow \langle c_1, \sigma' \rangle} \text{IF}_2 \\
\\
\frac{}{\langle \mathbf{repeat } c \mathbf{ until } b, \sigma \rangle \rightsquigarrow \langle c; \mathbf{if } b \mathbf{ then skip else repeat } b \mathbf{ until } c, \sigma \rangle} \text{REPEAT} \\
\\
\frac{\langle e_1, \sigma \rangle \Downarrow_{exp} \langle n_1, \sigma' \rangle}{\langle \mathbf{for } (e_1; e_2; e_3) \ c, \sigma \rangle \rightsquigarrow \langle \mathbf{if } e_2 \mathbf{ then } c; \mathbf{for } (e_3; e_2; e_3) \ c \mathbf{ else skip}, \sigma' \rangle} \text{FOR}
\end{array}$$