0.1 Transition Rules

In this section some of the transition rules in SPLAD will be explained. The complete list of all the rules can be seen in appendix **??**. In the following text we use the following names to represent different syntactic categories.

- $n \in \mathbf{Num}$ - Numerals
- v - Values
- $x \in \mathbf{Var}$ - Variables
- $r \in \mathbf{Arrays}$ - Array names
- $a \in A_{exp}$ - Arithmetic expression
- $b \in B_{exp}$ - Boolean expression
- $e \in A_{exp} \cup B_{exp}$ - expressions
- $C \in \mathbf{Com}$ - Commands

0.1.1 Environment-Store Model

In our project we use the *environment-store model* to represent how a variable is bound to a storage cell (called a *location*), in the computer, and that the value of the variable is the content of the bound location. All the possible locations are denoted by **Loc** and a single location as $l \in \mathbf{Loc}$. We assume all locations are integer, and therefore $\mathbf{Loc} = \mathbb{Z}$. Since all locations are integers we can define a function to find the next location: $\mathbf{Loc} \rightarrow \mathbf{Loc}$, where $l = l + 1$.

We define the set of stores to be the mappings from locations to values $\mathbf{Sto} = \mathbf{Loc} \rightharpoonup \mathbb{Z}$, where *sto* is an single element in **Sto**.

An variable-environment is like a symbol table containing each variable and store the variables address. The store then describe which values that is on each address.

The following names represent the different environments.

- $env_V \in Env_V$ - Variable declarations
- $env_A \in Env_A$ - Array declarations
- $env_C \in Env_C$ - Constant declarations
- $env_E \in Env_E$ - Expressions declarations

0.1.2 Arithmetic Expressions

The transition rule for multiplication in SPLAD can be seen on table 0.1. The rule states, that if a_1 evaluates to v_1 and a_2 evaluates to v_2, using any of the rules from the arithmetic expressions, then $a_1 * a_2$ evaluates to v where $v = v_1 * v_2$.

0.1.3 Boolean Expressions

The transition rule for logical-or in SPLAD can be seen on table 0.2. The rules have to parts: [OR-TRUE] and [OR-FALSE]. The [OR-TRUE] rule states that either b_1 or b_2 evaluates to *true*, using any of the rules from the boolean expressions, then the expression $b_1 OR b_2$ evaluates to *true*. [OR-FALSE] states that if both b_1 and b_2 evaluates to *false* then the expression $b_1 OR b_2$ evaluates to *false*.

[OR-TRUE] $$\frac{env_E,\ sto \vdash b_1 \vee b_2 \rightarrow_b \text{true}}{env_E,\ sto \vdash b_1 \text{ OR } b_2 \rightarrow_b \text{true}}$$

[OR-FALSE] $$\frac{env_E,\ sto \vdash b_1 \wedge b_2 \rightarrow_b \text{false}}{env_E,\ sto \vdash b_1 \text{ OR } b_2 \rightarrow_b \text{false}}$$

Table 0.2: Transition rule for the boolean expression logical-or.

0.1.4 Declarations

0.1.5 Assignments

The transition rule for variable assignment in SPLAD can be seen on table 0.3. When a variable is assigned the contents of l is updated to v, where l is the location of x found in the env_V and v is the result of evaluation e.

[VAR-ASS] $$env_C, \vdash \langle x = e,\ sto\rangle \rightarrow sto[l \mapsto v]$$

where $env_C,\ sto \vdash e \rightarrow_e v$
and $env_V\ x = l$

Table 0.3: Transition rule for variable assignment.

0.1.6 Commands

The transition rule for the while statement in SPLAD can be seen on table 0.4. The rule have to parts: [WHL-TRUE] and [WHL-FALSE]. If the condition b evaluates to *true* then the [WHL-TRUE] states that C will be executed which will update the *store* (sto) and again call the expression and evaluate the new b. If the condition b evaluates to *false* then C is <u>not</u> executed and the *store* is not updated. The program exits the while statement

[WHL-TRUE] $$\frac{env_C\ \vdash \langle C,\ sto\rangle \rightarrow sto''\ env_C\ \vdash \langle \textbf{while}(b) \text{ begin } C \text{ end},\ sto''\rangle \rightarrow sto'}{env_C\ \vdash\ \langle \textbf{while}(b) \text{ begin } C \text{ end},\ sto\rangle \rightarrow sto'}$$

if $env_C,\ sto \vdash b \rightarrow_b \text{true}$

[WHL-FALSE] $$env_C\ \vdash \langle \textbf{while}(b) \text{ begin } C \text{ end},\ sto\rangle \rightarrow sto$$

if $env_C,\ sto \vdash b \rightarrow_b \text{false}$

Table 0.4: Transition rules for the while statement.

0.2 Transition Rules

[VAR-DECL] $$\frac{\langle D_V, env_V[x \mapsto l][\text{next} \mapsto \text{new } l], sto[l \mapsto v]\rangle \rightarrow_{DV} (env'_V, sto')}{\langle \textbf{var}\ x <--a; D_V, env_V, sto\rangle \rightarrow_{DV} (env'_V, sto')}$$

Continued on the next page

where $env_V, sto \vdash a \rightarrow_a v$
and $l = env_V$ next

[EMPTY-VAR-DECL] $\langle \varepsilon, env_V, sto \rangle \rightarrow_{DV} (env_V, sto)$

[FUNC-DECL] $$\frac{env_V \vdash \langle D_P, env_P[p \mapsto (S, env_V, env_P)] \rangle \rightarrow_{DP} env'_P}{env_V \vdash \langle \textbf{proc}\ p\ \textbf{is}\ S; D_P, env_P \rangle \rightarrow_{DP} env'_P}$$

[FUNC-PARA-DECL] $$\frac{env_V \vdash \langle D_P, env_P[p \mapsto (S, x, env_V, env_P)] \rangle \rightarrow_{DP} env'_P}{env_V \vdash \langle \textbf{proc}\ p(\textbf{var}\ x)\ \textbf{is}\ S; D_P, env_P \rangle \rightarrow_{DP} env'_P}$$

[EMPTY-FUNC-DECL] $env_V \vdash \langle \varepsilon, env_P \rangle \rightarrow_{DP} env_P$

Table 0.5: Declarations

[VAR-ASS] $env_C, \vdash \langle x < --e,\ sto \rangle \rightarrow sto[l \mapsto v]$

where $env_C,\ sto \vdash e \rightarrow_e v$
and $env_V\ x = l$

[ARR-ASS] $env_C \vdash \langle r[a] < --e,\ sto \rangle \rightarrow sto[l \mapsto v_2]$

where $env_C,\ sto \vdash a \rightarrow_a v_1$
and $env_C,\ sto \vdash e \rightarrow_e v_2$
and $env_A\ r[v_1] = l$

Table 0.6: Assignments

[IF-TRUE] $$\frac{env_C \vdash \langle C,\ sto \rangle \rightarrow sto'}{env_C\ \langle \textbf{if}(b)\ \text{begin}\ C\ \text{end},\ sto \rangle \rightarrow sto'}$$

if $env_C,\ sto \vdash b \rightarrow_b$ true

[IF-FALSE] $env_C \vdash \langle \textbf{if}(b)\ \text{begin}\ C\ \text{end},\ sto \rangle \rightarrow sto$

if $env_C,\ sto \vdash b \rightarrow_b$ false

[IF-ELSE-TRUE] $$\frac{env_C \vdash \langle C_1,\ sto \rangle \rightarrow sto'}{env_C \vdash \langle \textbf{if}(b)\ \text{begin}\ C_1\ \text{end},\ \textbf{else}\ \text{begin}\ C_2\ \text{end},\ sto \rangle \rightarrow sto'}$$

if $env_C,\ sto \vdash b \rightarrow_b$ true

[IF-ELSE-FALSE] $$\frac{env_C \vdash \langle C_2,\ sto \rangle \rightarrow sto'}{env_C \vdash \langle \textbf{if}(b)\ \text{begin}\ C_1\ \text{end},\ \textbf{else}\ \text{begin}\ C_2\ \text{end},\ sto \rangle \rightarrow sto'}$$

Continued on the next page

if $env_C,\ sto \vdash b \to_b$ false

[WHL-TRUE]
$$\frac{env_C\ \vdash \langle C,\ sto\rangle \to sto''\ env_C\ \vdash \langle \textbf{while}(b) \text{ begin } C \text{ end},\ sto''\rangle \to sto'}{env_C\ \vdash\ \langle \textbf{while}(b) \text{ begin } C \text{ end},\ sto\rangle \to sto'}$$

if $env_C,\ sto \vdash b \to_b$ true

[WHL-FALSE]
$$env_C\ \vdash \langle \textbf{while}(b) \text{ begin } C \text{ end},\ sto\rangle \to sto$$

if $env_C,\ sto \vdash b \to_b$ false

[FROM-TRUE]
$$\frac{}{env_V, env_P \vdash \langle \textbf{from}\ x <--n_1\ \textbf{to}\ n_2\ \textbf{step}\ n_3\ \textbf{begin}\ C\ \textbf{end}, sto\rangle} \to sto'$$

[FROM-FALSE]
$$\frac{}{env_V, env_P \vdash \langle \textbf{from}\ x <--n_1\ \textbf{to}\ n_2\ \textbf{step}\ n_3\ \textbf{begin}\ C\ \textbf{end}, sto\rangle} \to sto'$$

[CALL-BY-VAL]
$$\frac{env'_V[x \mapsto l][\text{next} \mapsto \text{new } l, env'_P \vdash \langle S, sto[l \mapsto v]\rangle \to sto'}{env_V, env_P \vdash \langle \textbf{call}\ p(a), sto\rangle \to sto'}$$

where $env_Pp = (S, x, env'_V, env'_P)$,
and $env_V, sto \vdash a \to_a v$
and $l = env_V$

[CALL-BY-REF]
$$\frac{env'_V[x \mapsto l][\text{next} \mapsto l'], env'_P \vdash \langle S, sto\rangle \to sto'}{env_V, env_P \vdash \langle \textbf{call}\ p(y), sto\rangle \to sto'}$$

where $env_Pp = (S, x, env'_V, env'_P)$,
and $l = env_V y$
and $l' = env_V$ next

Table 0.7: Commands

[EQL-TRUE]
$$\frac{env_V,\ sto \vdash a_1 \to_a v_1\ \ env_V,\ sto \vdash a_2 \to_e v_2}{env_V,\ sto \vdash a_1 = a_2 \to_b \text{true}}$$

if $v_1 = v_2$

[EQL-FALSE]
$$\frac{env_V,\ sto \vdash a_1 \to_a v_1\ \ env_V,\ sto \vdash a_2 \to_a v_2}{env_V,\ sto \vdash a_1 = a_2 \to_b \text{false}}$$

if $v_1 \neq v_2$

[NEQ-TRUE]
$$\frac{env_V,\ sto \vdash a_1 \to_a v_1\ \ env_V,\ sto \vdash a_2 \to_a v_2}{env_V,\ sto \vdash a_1! = a_2 \to_b \text{true}}$$

if $v_1 \neq v_2$

[NEQ-FALSE]
$$\frac{env_V,\ sto \vdash a_1 \to_a v_1\ \ env_V,\ sto \vdash a_2 \to_a v_2}{env_V,\ sto \vdash a_1! = a_2 \to_b \text{false}}$$

Continued on the next page

if $v_1 = v_2$

[GRT-TRUE] $$\frac{env_V,\ sto \vdash a_1 \rightarrow_a v_1 \ \ env_V,\ sto \vdash a_2 \rightarrow_a v_2}{env_V,\ sto \vdash a_1 > a_2 \rightarrow_b \text{true}}$$

if $v_1 > v_2$

[GRT-FALSE] $$\frac{env_E,\ sto \vdash e_1 \rightarrow_e v_1 \ \ env_E,\ sto \vdash e_2 \rightarrow_e v_2}{env_E,\ sto \vdash e_1 > e_2 \rightarrow_b \text{false}}$$

if $v_1 \leq v_2$

[GEQ-TRUE] $$\frac{env_V,\ sto \vdash a_1 \rightarrow_a v_1 \ \ env_V,\ sto \vdash a_2 \rightarrow_a v_2}{env_V,\ sto \vdash a_1 >= a_2 \rightarrow_b \text{true}}$$

if $v_1 \geq v_2$

[GEQ-FALSE] $$\frac{env_V,\ sto \vdash a_1 \rightarrow_a v_1 \ \ env_V,\ sto \vdash a_2 \rightarrow_a v_2}{env_V,\ sto \vdash a_1 >= a_2 \rightarrow_b \text{false}}$$

if $v_1 < v_2$

[LES-TRUE] $$\frac{env_V,\ sto \vdash a_1 \rightarrow_a v_1 \ \ env_V,\ sto \vdash a_2 \rightarrow_a v_2}{env_V,\ sto \vdash a_1 < a_2 \rightarrow_b \text{true}}$$

if $v_1 < v_2$

[LES-FALSE] $$\frac{env_V,\ sto \vdash a_1 \rightarrow_a v_1 \ \ env_V,\ sto \vdash a_2 \rightarrow_a v_2}{env_V,\ sto \vdash a_1 < a_2 \rightarrow_b \text{false}}$$

if $v_1 \geq v_2$

[LEQ-TRUE] $$\frac{env_V,\ sto \vdash a_1 \rightarrow_a v_1 \ \ env_V,\ sto \vdash a_2 \rightarrow_a v_2}{env_V,\ sto \vdash a_1 <= a_2 \rightarrow_b \text{true}}$$

if $v_1 \leq v_2$

[GEQ-FALSE] $$\frac{env_V,\ sto \vdash a_1 \rightarrow_a v_1 \ \ env_V,\ sto \vdash a_2 \rightarrow_a v_2}{env_V,\ sto \vdash a_1 <= a_2 \rightarrow_b \text{false}}$$

if $v_1 > v_2$

[NOT-TRUE] $$\frac{env_V,\ sto \vdash b \rightarrow_b \text{true}}{env_V,\ sto \vdash !(b) \rightarrow_b \text{false}}$$

[NOT-FALSE] $$\frac{env_V,\ sto \vdash b \rightarrow_b \text{false}}{env_V,\ sto \vdash !(b) \rightarrow_b \text{true}}$$

[AND-TRUE] $$\frac{env_V,\ sto \vdash b_1 \rightarrow_b \text{true}\ env_V, sto \vdash b_2 \rightarrow_b \text{true}}{env_V,\ sto \vdash b_1 \text{ AND } b_2 \rightarrow_b \text{true}}$$

Continued on the next page

[AND-FALSE]	$\dfrac{env_V,\ sto \vdash b_i \rightarrow_b \text{false}}{env_V,\ sto \vdash b_1 \text{ AND } b_2 \rightarrow_b \text{false}}$ $(i \in \{1, 2\})$
[OR-TRUE]	$\dfrac{env_E,\ sto \vdash b_1 \vee b_2 \rightarrow_b \text{true}}{env_E,\ sto \vdash b_1 \text{ OR } b_2 \rightarrow_b \text{true}}$
[OR-FALSE]	$\dfrac{env_E,\ sto \vdash b_1 \wedge b_2 \rightarrow_b \text{false}}{env_E,\ sto \vdash b_1 \text{ OR } b_2 \rightarrow_b \text{false}}$
[PAR]	$\dfrac{env_E,\ sto \vdash b_1 \rightarrow_b v}{env_E,\ sto \vdash (b_1) \rightarrow_b v}$

Table 0.8: Boolean expressions

[MUL] $$\frac{env_E,\ sto \vdash a_1 \rightarrow_a v_1 \quad env_E,\ sto \vdash a_2 \rightarrow_a v_2}{env_E,\ sto \vdash a_1 * a_2 \rightarrow_a v}$$

where $v = v_1 * v_2$

Table 0.1: The transition rule for the arithmetic multiplication expression.

[ADD] $$\frac{env_V,\ sto \vdash a_1 \rightarrow_a v_1 \quad env_V,\ sto \vdash a_2 \rightarrow_a v_2}{env_V,\ sto \vdash a_1 + a_2 \rightarrow_a v}$$

where $v = v_1 + v_2$

[SUB] $$\frac{env_V,\ sto \vdash a_1 \rightarrow_a v_1 \quad env_V,\ sto \vdash a_2 \rightarrow_a v_2}{env_V,\ sto \vdash a_1 - a_2 \rightarrow_a v}$$

where $v = v_1 - v_2$

[MUL] $$\frac{env_V,\ sto \vdash a_1 \rightarrow_a v_1 \quad env_V,\ sto \vdash a_2 \rightarrow_a v_2}{env_V,\ sto \vdash a_1 * a_2 \rightarrow_a v}$$

where $v = v_1 * v_2$

[DIV] $$\frac{env_V,\ sto \vdash a_1 \rightarrow_a v_1 \quad env_V,\ sto \vdash a_2 \rightarrow_a v_2}{env_V,\ sto \vdash \frac{a_1}{a_2} \rightarrow_a v}$$

where $v = \frac{v_1}{v_2}$

[PAR] $$\frac{env_V,\ sto \vdash a_1 \rightarrow_a v_1}{env_V,\ sto \vdash (a_1) \rightarrow_a v_1}$$

[NUM] $$env_V,\ sto \vdash n \rightarrow_a v$$

if $\mathcal{N}[n] = v$
where $\mathcal{N}$ **: Num** $\rightarrow \mathbb{R}$

[VAR] $$env_V,\ sto \vdash x \rightarrow_a v$$

if $env_V\ x = l$
and sto $l = v$

[ARR] $$env_A,\ sto \vdash r[a_1] \rightarrow_a v_2$$

if $env_A\, r[v_1] = l$ and $sto\, l = v_2$
where $a_1 \rightarrow_a v_1$

Table 0.9: Aritmethic expressions