Determinism and termination in the semantics of the Céu programming language

January 26, 2017

1 The original formulation

The semantics discussed in this section follows as much as possible the original semantics of Céu presented in [**?**]. Any deviations from original definitions are duly noted in the text.

1.1 Abstract syntax

The *abstract syntax* of Céu programs is given by the following grammar:

$$
\begin{array}{rll}
p \in P ::= & \texttt{skip} & \text{do nothing} \\
 & \mid v := a & \text{assignment} \\
 & \mid \texttt{await}(e) & \text{await event} \\
 & \mid \texttt{emit}(e) & \text{emit event} \\
 & \mid \texttt{break} & \text{break innermost loop} \\
 & \mid \texttt{if}\, b\, \texttt{then}\, p_1\, \texttt{else}\, p_2 & \text{conditional} \\
 & \mid p_1 ; p_2 & \text{sequence} \\
 & \mid \texttt{loop}\, p_1 & \text{repetition} \\
 & \mid p_1\, \texttt{and}\, p_2 & \text{par/and} \\
 & \mid p_1\, \texttt{or}\, p_2 & \text{par/or} \\
 & \mid \texttt{fin}\, p_1 & \text{finalization} \\
 & \mid \texttt{@awaiting}(e, n) & \text{awaiting } e \text{ since reaction } n \\
 & \mid \texttt{@emitting}(e, n) & \text{emitting } e \text{ on stack level } n \\
 & \mid p_1\, \texttt{@loop}\, p_2 & \text{unwinded loop}
\end{array}
$$

where $n \in N$ is an integer, $v \in V$ is a memory location (variable) identifier, $e \in E$ is an event identifier, $a \in A$ is an arithmetic expression, $b \in B$ is a boolean expression, and $p, p_1, p_2 \in P$ are programs. We assume the usual structure for arithmetic and boolean expressions, and omit their definition.

1.2 The reaction inner-step relation

The *state* of a Céu program within a reaction is represented by a stack of events $\alpha = e_1 e_2 \ldots e_n \in E^*$ together with a memory map $m\colon v \to N \in \mathcal{M}$. A *configuration* is a 4-tuple $\langle p, \alpha, m, n\rangle \in \Delta$ that represents the situation of program p waiting to be evaluated in state $\langle \alpha, m\rangle$ and reaction n. Given an initial configuration, each small-step within a program reaction is determined by the reaction-inner-step relation $\rightarrow \in \Delta \times \Delta$ such that $\langle p, \alpha, m, n\rangle \rightarrow \langle p', \alpha', m', n\rangle$ iff a reaction inner-step of program p in state $\langle \alpha, m\rangle$ and reaction number n evaluates to a modified program p' and a modified state $\langle \alpha', m'\rangle$ in the same reaction (n). Since relation $\rightarrow$ can only relate configurations with the same n, we shall write $\langle p, \alpha, m\rangle \xrightarrow{n} \langle p', \alpha', m'\rangle$ for $\langle p, \alpha, m, n\rangle \rightarrow \langle p', \alpha', m', n\rangle$.

Relation $\rightarrow$ is defined inductively with the help of the auxiliary functions *eval*, *blocked*, and *clear*. The *eval* function evaluates arithmetic or boolean expressions on a given memory; we omit its definition and assume that such evaluation is deterministic and always terminates. The *blocked* function is a predicate that determines if all trails of a program p are blocked on a given event stack and reaction number. And the *clear* function extracts the body of `fin` blocks from a given program.

Definition 1.1. Function *blocked*: $P \times E^* \times N \rightarrow \{0, 1\}$ is defined inductively as follows.

$$
\begin{aligned}
\mathit{blocked}(v := a, e\alpha, n) &= 0\\
\mathit{blocked}(\texttt{await}(e'), e\alpha, n) &= 0\\
\mathit{blocked}(\texttt{emit}(e'), e\alpha, n) &= 0\\
\mathit{blocked}(\texttt{break}, e\alpha, n) &= 0\\
\mathit{blocked}(\texttt{if}\, v\, \texttt{then}\, p_1\ \texttt{else}\, p_2, e\alpha, n) &= 0\\
\mathit{blocked}(p_1; p_2, e\alpha, n) &= \mathit{blocked}(p_1, e\alpha, n)\\
\mathit{blocked}(\texttt{loop}\, p, e\alpha, n) &= 0\\
\mathit{blocked}(p_1\ \texttt{and}\ p_2, e\alpha, n) &= \mathit{blocked}(p_1, e\alpha, n) \cdot \mathit{blocked}(p_2, e\alpha, n)\\
\mathit{blocked}(p_1\ \texttt{or}\ p_2, e\alpha, n) &= \mathit{blocked}(p_1, e\alpha, n) \cdot \mathit{blocked}(p_2, e\alpha, n)\\
\mathit{blocked}(\texttt{fin}\, p_1, e\alpha, n) &= 0\\
\mathit{blocked}(\texttt{@awaiting}(e', n'), e\alpha, n) &= \begin{cases} 1 & \text{if } e \neq e' \text{ or } n = n' \\ 0 & \text{otherwise} \end{cases}\\
\mathit{blocked}(\texttt{@emitting}(n'), e\alpha, n) &= \begin{cases} 1 & \text{if } |e\alpha| \neq n' \\ 0 & \text{otherwise} \end{cases}\\
\mathit{blocked}(p_1\ \texttt{@loop}\, p_2, e\alpha, n) &= \mathit{blocked}(p_1, e\alpha, n)
\end{aligned}
$$

Definition 1.2. Function $clear\colon P \to P$ is defined inductively as follows.

$$
\begin{aligned}
clear(v := a) &= \texttt{skip} \\
clear(\texttt{await}(e')) &= \texttt{skip} \\
clear(\texttt{emit}(e')) &= \texttt{skip} \\
clear(\texttt{break}) &= \texttt{skip} \\
clear(\texttt{if}\, v\, \texttt{then}\, p_1\, \texttt{else}\, p_2) &= \texttt{skip} \\
clear(p_1; p_2) &= clear(p_1); clear(p_2) \\
clear(\texttt{loop}\, p) &= clear(p) \\
clear(p_1\, \texttt{and}\, p_2) &= clear(p_1); clear(p_2) \\
clear(p_1\, \texttt{or}\, p_2) &= clear(p_1); clear(p_2) \\
clear(\texttt{fin}\, p) &= p \\
clear(\texttt{@awaiting}(e', n')) &= \texttt{skip} \\
clear(\texttt{@emitting}(n')) &= \texttt{skip} \\
clear(p_1\, \texttt{@loop}\, p_2) &= clear(p_1)
\end{aligned}
$$

Definition 1.3 (Reaction inner-step)**.** Relation $\to \subseteq \Delta \times \Delta$ is defined inductively as follows.

Await and emit

$$\langle \texttt{await}(e), \alpha, m\rangle \xrightarrow{n} \langle \texttt{@awaiting}(e, n'), \alpha, m\rangle \quad \text{with } n' = n + 1 \tag{1.1}$$

$$\langle \texttt{@awaiting}(e, n'), e\alpha, m\rangle \xrightarrow{n} \langle \texttt{skip}, e\alpha, m\rangle \quad \text{if } n' \leq n \tag{1.2}$$

$$\langle \texttt{emit}(e), \alpha, m\rangle \xrightarrow{n} \langle \texttt{@emitting}(n'), e\alpha, m\rangle \quad \text{with } n' = |\alpha| \tag{1.3}$$

$$\langle \texttt{@emitting}(n'), \alpha, m\rangle \xrightarrow{n} \langle \texttt{skip}, \alpha, m\rangle \quad \text{if } n' = |\alpha| \tag{1.4}$$

Conditionals

$$\langle \texttt{if}\, b\, \texttt{then}\, p_1\, \texttt{else}\, p_2, \alpha, m\rangle \xrightarrow{n} \langle p_1, \alpha, m\rangle \quad \text{if } eval(b, m) = 1 \tag{1.5}$$

$$\langle \texttt{if}\, b\, \texttt{then}\, p_1\, \texttt{else}\, p_2, \alpha, m\rangle \xrightarrow{n} \langle p_2, \alpha, m\rangle \quad \text{if } eval(b, m) = 0 \tag{1.6}$$

Sequences

$$\langle v := a; p, \alpha, m\rangle \xrightarrow{n} \langle p, \alpha, m'\rangle \quad \text{with } m' = m[v/eval(a)] \tag{1.7}$$

$$\langle \texttt{break}; p, \alpha, m\rangle \xrightarrow{n} \langle \texttt{break}, \alpha, m\rangle \tag{1.8}$$

$$\frac{\langle p_1, \alpha, m\rangle \xrightarrow{n} \langle p_1', \alpha', m'\rangle}{\langle p_1; p_2, \alpha, m\rangle \xrightarrow{n} \langle p_1'; p_2, \alpha', m'\rangle} \quad \text{if } p_1 \neq v := a, \texttt{break} \tag{1.9}$$

Loops

$$\langle \texttt{loop}\ p, \alpha, m\rangle \xrightarrow{n} \langle p\ \texttt{@loop}\ p, \alpha, m\rangle \tag{1.10}$$

$$\langle v := a\ \texttt{@loop}\ p, \alpha, m\rangle \xrightarrow{n} \langle \texttt{loop}\ p, \alpha, m'\rangle \qquad \text{with } m' = m[v/eval(a)] \tag{1.11}$$

$$\langle \texttt{break}\ \texttt{@loop}\ p, \alpha, m\rangle \xrightarrow{n} \langle \texttt{skip}, \alpha, m\rangle \tag{1.12}$$

$$\frac{\langle p_1, \alpha, m\rangle \xrightarrow{n} \langle p_1', \alpha', m'\rangle}{\langle p_1\ \texttt{@loop}\ p_2, \alpha, m\rangle \xrightarrow{n} \langle p_1'\ \texttt{@loop}\ p_2, \alpha', m'\rangle} \qquad \text{if } p_1 \neq v := a, \texttt{break} \tag{1.13}$$

Par/and

$$\langle v := a\ \texttt{and}\ p, \alpha, m\rangle \xrightarrow{n} \langle p, \alpha, m'\rangle \qquad \text{with } m' = m[v/eval(a)] \tag{1.14}$$

$$\langle p\ \texttt{and}\ v := a, \alpha, m\rangle \xrightarrow{n} \langle p, \alpha, m'\rangle \qquad \text{with } m' = m[v/eval(a)] \tag{1.15}$$

$$\langle \texttt{break}\ \texttt{and}\ p, \alpha, m\rangle \xrightarrow{n} \langle p'; \texttt{break}, \alpha, m\rangle \qquad \begin{array}{l}\text{if } p \neq v := a,\\ \quad\text{with } p' = clear(p)\end{array} \tag{1.16}$$

$$\langle p\ \texttt{and}\ \texttt{break}, \alpha, m\rangle \xrightarrow{n} \langle p'; \texttt{break}, \alpha, m\rangle \qquad \begin{array}{l}\text{if } blocked(p, \alpha, n) = 1,\\ \quad\text{with } p' = clear(p)\end{array} \tag{1.17}$$

$$\frac{\langle p_1, \alpha, m\rangle \xrightarrow{n} \langle p_1', \alpha', m'\rangle}{\langle p_1\ \texttt{and}\ p_2, \alpha, m\rangle \xrightarrow{n} \langle p_1'\ \texttt{and}\ p_2, \alpha', m'\rangle} \qquad \begin{array}{l}\text{if } blocked(p_1, \alpha, n) = 0\\ \quad\text{and } p_1 \neq v := a, \texttt{break}\end{array} \tag{1.18}$$

$$\frac{\langle p_2, \alpha, m\rangle \xrightarrow{n} \langle p_2', \alpha', m'\rangle}{\langle p_1\ \texttt{and}\ p_2, \alpha, m\rangle \xrightarrow{n} \langle p_1\ \texttt{and}\ p_2', \alpha', m'\rangle} \qquad \begin{array}{l}\text{if } blocked(p_1, \alpha, n) = 1\\ \quad\text{and } p_2 \neq v := a, \texttt{break}\end{array} \tag{1.19}$$

Par/or

$$\langle v := a\ \texttt{or}\ p, \alpha, m\rangle \xrightarrow{n} \langle p', \alpha, m'\rangle \qquad \begin{array}{l}\text{with } m' = m[v/eval(a)]\\ \quad\text{and } p' = clear(p)\end{array} \tag{1.20}$$

$$\langle p\ \texttt{or}\ v := a, \alpha, m\rangle \xrightarrow{n} \langle p', \alpha, m'\rangle \qquad \begin{array}{l}\text{if } blocked(p, \alpha, n) = 1,\\ \quad\text{with } m' = m[v/eval(a)]\\ \quad\text{and } p' = clear(p)\end{array} \tag{1.21}$$

$$\langle \texttt{break}\ \texttt{or}\ p, \alpha, m\rangle \xrightarrow{n} \langle p'; \texttt{break}, \alpha, m\rangle \qquad \text{with } p' = clear(p) \tag{1.22}$$

$$\langle p\ \texttt{or}\ \texttt{break}, \alpha, m\rangle \xrightarrow{n} \langle p'; \texttt{break}, \alpha, m\rangle \qquad \begin{array}{l}\text{if } blocked(p, \alpha, n) = 1,\\ \quad\text{with } p' = clear(p)\end{array} \tag{1.23}$$

$$\frac{\langle p_1, \alpha, m\rangle \xrightarrow{n} \langle p_1', \alpha', m'\rangle}{\langle p_1\ \texttt{or}\ p_2, \alpha, m\rangle \xrightarrow{n} \langle p_1'\ \texttt{or}\ p_2, \alpha', m'\rangle} \qquad \begin{array}{l}\text{if } blocked(p_1, \alpha, n) = 0\\ \quad\text{and } p_1 \neq v := a, \texttt{break}\end{array} \tag{1.24}$$

$$\frac{\langle p_2, \alpha, m\rangle \xrightarrow{n} \langle p_2', \alpha', m'\rangle}{\langle p_1\ \texttt{or}\ p_2, \alpha, m\rangle \xrightarrow{n} \langle p_1\ \texttt{or}\ p_2', \alpha', m'\rangle} \qquad \begin{array}{l}\text{if } blocked(p_1, \alpha, n) = 1\\ \quad\text{and } p_2 \neq v := a, \texttt{break}\end{array} \tag{1.25}$$

The next theorem establishes that the reaction inner-step relation is deterministic, i.e., that it is in fact a *partial* function.

Theorem 1.4 (Determinism of the inner-step relation)**.** *For all p, p_1, $p_2 \in P$, α, α_1, $\alpha_2 \in E^*$, m, m_1, $m_2 \in \mathcal{M}$, and $n \in N$,*

$$\begin{aligned} \textit{if} \quad \langle p, \alpha, m\rangle \xrightarrow{n} \langle p_1, \alpha_1, m_1\rangle \quad \textit{and} \quad \langle p, \alpha, m\rangle \xrightarrow{n} \langle p_2, \alpha_2, m_2\rangle, \\ \textit{then} \quad \langle p_1, \alpha_1, m_1\rangle = \langle p_2, \alpha_2, m_2\rangle. \end{aligned}$$

Proof. By induction on the structure of derivations. Suppose

$$d_1 \Vdash \langle p, \alpha, m\rangle \xrightarrow{n} \langle p_1, \alpha_1, m_1\rangle \quad \text{and} \quad d_2 \Vdash \langle p, \alpha, m\rangle \xrightarrow{n} \langle p_2, \alpha_2, m_2\rangle,$$

for some derivations d_1 and d_2. Then there are ten possibilities depending on the structure of p. (Note that p cannot be equal to `skip`, $v \coloneqq a$, or `break`, as there are no rules to evaluate such programs.)

[Case 1] $p = \mathtt{await}(e)$, for some $e \in E$. Then d_1 and d_2 are instances of axiom (1.1), and as such, $p_1 = p_2 = \mathtt{@awaiting}(e, n')$ with $n' = n + 1$, and $\alpha_1 = \alpha_2 = \alpha$ and $m_1 = m_2 = m$.

[Case 2] $p = \mathtt{@awaiting}(e, n')$, for some $e \in E$ and $n' \in N$. Then d_1 and d_2 are instances of axiom (1.2), with $n' \leq n$. Thus $p_1 = p_2 = \mathtt{skip}$, $\alpha_1 = \alpha_2 = \alpha$, and $m_1 = m_2 = m$.

[Case 3] $p = \mathtt{emit}(e)$, for some $e \in E$. Then d_1 and d_2 are instances of axiom (1.3), and as such, $p_1 = p2 = \mathtt{@emitting}(n')$ with $n' = |\alpha|$, and $\alpha_1 = \alpha_2 = e\alpha$ and $m_1 = m_2 = m$.

[Case 4] $p = \mathtt{@emitting}(e, n')$, for some $e \in E$ and $n' \in N$. Then d_1 and d_2 are instances of axiom (1.4) with $n' = |\alpha|$. Thus $p_1 = p_2 = \mathtt{skip}$, $\alpha_1 = \alpha_2 = \alpha$ and $m_1 = m_2 = m$.

[Case 5] $p = \mathtt{if}\, b\, \mathtt{then}\, p'\, \mathtt{else}\, p''$, for some $b \in B$ and p', $p'' \in P$.

[Case 5.1] $eval(b, m) = 1$. Then d_1 and d_2 are instances of axiom (1.5), and as such, $p_1 = p_2 = p'$, $\alpha_1 = \alpha_2 = \alpha$, and $m_1 = m_2 = m$.

[Case 5.2] $eval(b, m) = 0$. Then d_1 and d_2 are instances of axiom (1.6), and as such, $p_1 = p_2 = p''$, $\alpha_1 = \alpha_2 = \alpha$, and $m_1 = m_2 = m$.

[Case 6] $p = p'; p''$, for some p', $p'' \in P$.

[Case 6.1] $p' = v \coloneqq a$, for some $v \in V$ and $a \in A$. Then d_1 and d_2 are instances of axiom (1.7), and as such, $p_1 = p_2 = p''$, $\alpha_1 = \alpha_2 = \alpha$ and, as $eval$ is a total function, $m_1 = m_2 = m[v/eval(a)]$.

[Case 6.2] $p' = \mathtt{break}$. Then d_1 and d_2 are instances of axiom (1.8), and as such, $p_1 = p_2 = p'$, $\alpha_1 = \alpha_2 = \alpha$ and $m_1 = m_2 = m$.

[Case 6.3] $p' \neq v \coloneqq a, \mathtt{break}$. Then d_1 and d_2 are instances of rule (1.9). Thus there are derivations d_1' and d_2' such that

$$d_1' \Vdash \langle p', \alpha, m\rangle \xrightarrow{n} \langle p_1', \alpha_1, m_1\rangle \quad \text{and} \quad d_2' \Vdash \langle p', \alpha, m\rangle \xrightarrow{n} \langle p_2', \alpha_2, m_2\rangle,$$

for some p'_1, $p'_2 \in P$. Since $d'_1 \prec d_1$ and $d'_2 \prec d_2$, by induction hypothesis, $\alpha_1 = \alpha_2$, $m_1 = m_2$, and $p'_1 = p'_2$, which implies

$$p_1 = p'_1; p'' = p'_2; p'' = p_2.$$

[Case 7] $p = \texttt{loop}\, p'$, for some $p' \in P$. Then d_1 and d_2 are instances of axiom (1.10), and as such, $p_1 = p_2 = p' \texttt{@loop}\, p'$, $\alpha_1 = \alpha_2 = \alpha$, and $m_1 = m_2 = m$.

[Case 8] $p = p' \texttt{@loop}\, p''$, for some p', $p'' \in P$.

[Case 8.1] $p = v \coloneqq a \,\texttt{@loop}\, p'$, for some $a \in A$, $v \in V$, and $p' \in P$. Then d_1 and d_2 are instances of (1.11), and as such, $p_1 = p_2 = \texttt{loop}\, p'$, $\alpha_1 = \alpha_2 = \alpha$, and as *eval* is a total function, $m_1 = m_2 = m[v/eval(a)]$.

[Case 8.2] $p = \texttt{break}\,\texttt{@loop}\, p'$, for some $p' \in P$. Then d_1 and d_2 are instances of axiom (1.12), and as such, $p_1 = p_2 = \texttt{skip}$, $\alpha_1 = \alpha_2 = \alpha$, and $m_1 = m_2 = m$.

[Case 8.3] $p = p' \texttt{@loop}\, p''$, for some p', $p'' \in P$ such that $p' \neq v \coloneqq a, \texttt{break}$. Then d_1 and d_2 are instances of rule (1.13). Thus there are derivations d'_1 and d'_2 such that

$$d'_1 \Vdash \langle p', \alpha, m\rangle \xrightarrow{n} \langle p'_1, \alpha_1, m_1\rangle \quad \text{and} \quad d'_2 \Vdash \langle p', \alpha, m\rangle \xrightarrow{n} \langle p'_2, \alpha_2, m_2\rangle,$$

for some p'_1, $p'_2 \in P$. Since $d'_1 \prec d_1$ and $d'_2 \prec d_2$, by induction hypothesis, $\alpha_1 = \alpha$, $m_1 = m_2$, and $p'_1 = p'_2$, which implies

$$p_1 = p'_1 \,\texttt{@loop}\, p'' = p'_2 \,\texttt{@loop}\, p'' = p_2.$$

[Case 9] $p = p'$ and p'', for some p', $p'' \in P$.

[Case 9.1] $p = v \coloneqq a$ and p', for some $v \in V$, $a \in A$ and $p' \in P$. Then d_1 and d_2 are instances of axiom (1.14), and as such, $p_1 = p_2 = p'$, $\alpha_1 = \alpha_2 = \alpha$, and as such, as *eval* is a total function, $m_1 = m_2 = m[v/eval(a)]$.

[Case 9.2] $p = p'$ and $v \coloneqq a$, for some $v \in V$, $a \in A$ and $p' \in P$. Similar to Case 9.1.

[Case 9.3] $p = \texttt{break}$ andp', for some $p' \in P$. Then d_1 and d_2 are instances of axiom (1.16), and as such, $\alpha_1 = \alpha_2 = \alpha$, $m_1 = m_2 = m$, and as *clear* is a total function, $p_1 = p_2 = clear(p'); \texttt{break}$.

[Case 9.4] $p = p'$ and $\texttt{break}$, for some $p' \in P$. Then either $blocked(p') = 0$ or $blocked(p') = 1$. If $blocked(p') = 0$ then this case becomes Case 9.5. Otherwise, if $blocked(p') = 1$, then d_1 and d_2 are instances of axiom (1.17), and as such, $\alpha_1 = \alpha_2 = \alpha$, $m_1 = m_2 = m$, and as *clear* is a total function, $p_1 = p_2 = clear(p'); \texttt{break}$.

[Case 9.5] $p = p'$ and p'', for some p' and p''. Then there are two possibilities. If $blocked(p') = 0$ then d_1 and d_2 are instances of (1.18). Thus there are derivations d'_1 and d'_2 such that

$$d'_1 \Vdash \langle p', \alpha, m\rangle \xrightarrow{n} \langle p'_1, \alpha_1, m_1\rangle \quad \text{and} \quad d'_2 \Vdash \langle p', \alpha, m\rangle \xrightarrow{n} \langle p'_2, \alpha_2, m_2\rangle,$$

for some $p'_1, p'_2 \in P$. Since $d'_1 \prec d_1$ and $d'_2 \prec d_2$, by induction hypothesis, $\alpha_1 = \alpha_2$, $m_1 = m_2$, and $p'_1 = p'_2$, which implies

$$p_1 = (p'_1 \texttt{ and } p'') = (p'_2 \texttt{ and } p'') = p_2.$$

If, however, $blocked(p') = 1$, then d_1 and d_2 are instances of (1.19). Thus there are derivations d''_1 and d''_2 such that

$$d''_1 \Vdash \langle p'', \alpha, m \rangle \xrightarrow{n} \langle p''_1, \alpha_1, m_1 \rangle \quad \text{and} \quad d''_2 \Vdash \langle p'', \alpha, m \rangle \xrightarrow{n} \langle p''_2, \alpha_2, m_2 \rangle,$$

for some $p''_1, p''_2 \in P$. Since $d''_1 \prec d_1$ and $d''_2 \prec d_2$, by induction hypothesis, $\alpha_1 = \alpha_2$, $m_1 = m_2$, and $p''_1 = p''_2$, which implies

$$p_1 = (p' \texttt{ and } p''_1) = (p' \texttt{ and } p''_2) = p_2.$$

[Case 10] $p = p' \texttt{ or } p''$, for some $p', p'' \in P$. [TODO: Similar to Case 9 (we hope).]

□

The next lemma establishes that given a program either it is possible to advance it by an inner-step or all its trails are blocked, but not both.

Lemma 1.5. *For all $p \in P$, $\alpha \in E^*$, $m \in \mathcal{M}$, and $n \in N$, if $p \neq \texttt{skip}, v \coloneqq a, \texttt{break}$ then either*

$$\exists \delta \in \Delta(\langle p, \alpha, m \rangle \xrightarrow{n} \delta) \quad or \quad blocked(p, \alpha, n) = 1,$$

but not both.

Proof. [TODO: By induction on the structure of programs.] □

1.3 The reaction outer-step relation

From the previous inner-step relation we define an outer-step relation ($\Rightarrow$) that when necessary pops the event stack advances blocked programs. [TODO: Improve this description.]

Definition 1.6 (Reaction outer-step)**.**

(1.26) $$\frac{\langle p, \alpha, m \rangle \xrightarrow{n} \langle p', \alpha', m' \rangle}{\langle p, \alpha, m \rangle \overset{n}{\Rightarrow} \langle p', \alpha', m' \rangle} \quad \text{if } blocked(p, \alpha, n) = 0$$

(1.27) $$\langle p, e\alpha, m \rangle \overset{n}{\Rightarrow} \langle p, \alpha, m \rangle \quad \text{if } blocked(p, \alpha, n) = 1$$

Theorem 1.7 (Determinism of the outer-step relation)**.** *For all p, p_1, $p_2 \in P$, α, α_1, $\alpha_2 \in E^*$, m, m_1, $m_2 \in \mathcal{M}$, and $n \in N$,*

$$\begin{aligned} &if \quad \langle p, \alpha, m \rangle \overset{n}{\Rightarrow} \langle p_1, \alpha_1, m_1 \rangle \quad and \quad \langle p, \alpha, m \rangle \overset{n}{\Rightarrow} \langle p_2, \alpha_2, m_2 \rangle, \\ &then \quad \langle p_1, \alpha_1, m_1 \rangle = \langle p_2, \alpha_2, m_2 \rangle. \end{aligned}$$

Proof. [TODO: By induction on the structure of derivations.] □

Theorem 1.8 (Termination of the outer-step relation)**.** *For all $p \in P$, $\alpha \in E$, and $m \in \mathcal{M}$, if $p \neq \texttt{skip}, v := a, \texttt{break}$ then*

$$\exists \delta \in \Delta(\langle p, \alpha, m \rangle \overset{n}{\Rightarrow} \delta).$$

Proof. [TODO: Directly from Lemma 1.5.] □

1.4 The reaction relation

From the reflexive-transitive closure of the outer-step relation ($\overset{*}{\Rightarrow}$) we define the reaction relation $\vDash \subseteq \Delta \times (P \times \mathcal{M} \times N)$, which computes a full program reaction. Given an initial configuration, the reaction relation evaluates it until the event stack becomes empty.

Definition 1.9 (Reaction)**.** Let p, $p' \in P$, $\alpha \in E^*$, m, $m' \in \mathcal{M}$. Then

$$\langle p, \alpha, m \rangle \overset{n}{\vDash} \langle p', m' \rangle \quad \text{iff} \quad \langle p, \alpha, m \rangle \overset{*}{\Rightarrow} \langle p', \varepsilon, m' \rangle.$$

The next two theorems establish, respectively, that reactions are deterministic and always terminate (for the nontrivial programs $p \neq \texttt{skip}, v := a, \texttt{break}$).

Theorem 1.10 (Determinism of the reaction relation)**.** *For all p, p_1, $p_2 \in P$, $\alpha \in E^*$, m, m_1, $m_2 \in \mathcal{M}$, and $n \in N$,*

$$\begin{aligned} \textit{if} \quad \langle p, \alpha, m \rangle \overset{n}{\vDash} \langle p_1, m_1 \rangle \quad \textit{and} \quad \langle p, \alpha, m \rangle \overset{n}{\vDash} \langle p_2, m_2 \rangle, \\ \textit{then} \quad \langle p_1, m_1 \rangle = \langle p_2, m_2 \rangle. \end{aligned}$$

Proof. [TODO: ?] □

Theorem 1.11 (Termination of the reaction relation)**.** *For all $p \in P$, $\alpha \in E$, and $m \in \mathcal{M}$, if $p \neq \texttt{skip}, v := a, \texttt{break}$ then*

$$\langle p, \alpha, m \rangle \overset{n}{\vDash} \langle p', m' \rangle,$$

for some $p' \in P$ and $m' \in \mathcal{M}$.

Proof. [TODO: ?] □

2 Big-step version of the original formulation

[TODO: Minha ideia aqui é fazer uma versão big-step da formulação original. E no final comparar as duas versões, i.e., mostrar que são equivalentes.]

Definition 2.1. [TODO: Parcial e provavelmente incorreta.]
Empty program

$$\langle \varepsilon, \alpha, m, n\rangle \rightsquigarrow \langle \alpha, m, n\rangle \tag{2.1}$$

Assignment

$$\langle v := a, \alpha, m, n\rangle \rightsquigarrow \langle \alpha, m[v/eval(a)], n\rangle \tag{2.2}$$

Conditionals

$$\frac{\langle p_1, \alpha, m, n\rangle \rightsquigarrow \langle \alpha', m', n'\rangle}{\langle \texttt{if}\, b\, \texttt{then}\, p_1\, \texttt{else}\, p_2, \alpha, m, n\rangle \rightsquigarrow \langle \alpha', m', n'\rangle} \quad \text{if } eval(b, m) = 1 \tag{2.3}$$

$$\frac{\langle p_2, \alpha, m, n\rangle \rightsquigarrow \langle \alpha', m', n'\rangle}{\langle \texttt{if}\, b\, \texttt{then}\, p_1\, \texttt{else}\, p_2, \alpha, m, n\rangle \rightsquigarrow \langle \alpha', m', n'\rangle} \quad \text{if } eval(b, m) = 0 \tag{2.4}$$

Await

$$\frac{\langle \texttt{@awaiting}(e, n+1), e\alpha, m, n\rangle \rightsquigarrow \langle \alpha', m', n'\rangle}{\langle \texttt{await}(e), \alpha, m, n\rangle \rightsquigarrow \langle \alpha', m', n'\rangle} \tag{2.5}$$

$$\langle \texttt{@awaiting}(e', n'), e\alpha, m, n\rangle \rightsquigarrow \langle \alpha, m, n\rangle \quad \text{if } e' = e \text{ and } n' < n \tag{2.6}$$

$$\frac{\langle \texttt{@awaiting}(e', n'), \alpha, m, n\rangle \rightsquigarrow \langle \alpha'', m'', n''\rangle}{\langle \texttt{@awaiting}(e', n'), e\alpha, m, n\rangle \rightsquigarrow \langle \alpha'', m'', n''\rangle} \quad \text{if } e' \neq e \text{ or } n' \geq n \tag{2.7}$$

Emit

$$\frac{\langle \texttt{@emitting}(|\alpha|), e\alpha, m, n\rangle \rightsquigarrow \langle \alpha', m', n'\rangle}{\langle \texttt{emit}(e), \alpha, m, n\rangle \rightsquigarrow \langle \alpha', m', n'\rangle} \tag{2.8}$$

$$\langle \texttt{@emitting}(n'), e\alpha, m, n\rangle \rightsquigarrow \langle \alpha, m, n\rangle \quad \text{if } |e\alpha| = n' \tag{2.9}$$

$$\frac{\langle \texttt{@emitting}(n'), \alpha, m, n\rangle \rightsquigarrow \langle \alpha'', m'', n''\rangle}{\langle \texttt{@emitting}(n'), e\alpha, m, n\rangle \rightsquigarrow \langle \alpha'', m'', n''\rangle} \quad \text{if } |e\alpha| \neq n' \tag{2.10}$$