

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:

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

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 $\alpha = e_1 e_2 \dots e_n \in E^*$ together with a memory map $m: v \rightarrow N \in \mathcal{M}$. 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.

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

Definition 1.2. Function $clear: P \rightarrow P$ is defined inductively as follows.

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

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

Await and emit

$$\begin{aligned}
(1.1) \quad & \langle \text{await}(e), \alpha, m \rangle \xrightarrow{n} \langle @\text{awaiting}(e, n'), \alpha, m \rangle \quad \text{with } n' = n + 1 \\
(1.2) \quad & \langle @\text{awaiting}(e, n'), e\alpha, m \rangle \xrightarrow{n} \langle \text{skip}, e\alpha, m \rangle \quad \text{if } n' \leq n \\
(1.3) \quad & \langle \text{emit}(e), \alpha, m \rangle \xrightarrow{n} \langle @\text{emitting}(n'), e\alpha, m \rangle \quad \text{with } n' = |\alpha| \\
(1.4) \quad & \langle @\text{emitting}(n'), \alpha, m \rangle \xrightarrow{n} \langle \text{skip}, \alpha, m \rangle \quad \text{if } n' = |\alpha|
\end{aligned}$$

Conditionals

$$\begin{aligned}
(1.5) \quad & \langle \text{if } b \text{ then } p_1 \text{ else } p_2, \alpha, m \rangle \xrightarrow{n} \langle p_1, \alpha, m \rangle \quad \text{if } \text{eval}(b, m) = 1 \\
(1.6) \quad & \langle \text{if } b \text{ then } p_1 \text{ else } p_2, \alpha, m \rangle \xrightarrow{n} \langle p_2, \alpha, m \rangle \quad \text{if } \text{eval}(b, m) = 0
\end{aligned}$$

Sequences

$$\begin{aligned}
(1.7) \quad & \langle v := a; p, \alpha, m \rangle \xrightarrow{n} \langle p, \alpha, m' \rangle \quad \text{with } m' = m[v/\text{eval}(a)] \\
(1.8) \quad & \langle \text{break}; p, \alpha, m \rangle \xrightarrow{n} \langle \text{break}, \alpha, m \rangle \\
(1.9) \quad & \frac{\langle p_1, \alpha, m \rangle \xrightarrow{n} \langle p'_1, \alpha', m' \rangle}{\langle p_1; p_2, \alpha, m \rangle \xrightarrow{n} \langle p'_1; p_2, \alpha', m' \rangle} \quad \text{if } p_1 \neq v := a, \text{break}
\end{aligned}$$

Loops

$$(1.10) \quad \langle \text{loop } p, \alpha, m \rangle \xrightarrow{n} \langle p @ \text{loop } p, \alpha, m \rangle$$

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

$$(1.12) \quad \langle \text{break} @ \text{loop } p, \alpha, m \rangle \xrightarrow{n} \langle \text{skip}, \alpha, m \rangle$$

$$(1.13) \quad \frac{\langle p_1, \alpha, m \rangle \xrightarrow{n} \langle p'_1, \alpha', m' \rangle}{\langle p_1 @ \text{loop } p_2, \alpha, m \rangle \xrightarrow{n} \langle p'_1 @ \text{loop } p_2, \alpha', m' \rangle} \quad \text{if } p_1 \neq v := a, \text{break}$$

Par/and

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

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

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

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

$$(1.18) \quad \frac{\langle p_1, \alpha, m \rangle \xrightarrow{n} \langle p'_1, \alpha', m' \rangle}{\langle p_1 \text{ and } p_2, \alpha, m \rangle \xrightarrow{n} \langle p'_1 \text{ and } p_2, \alpha', m' \rangle} \quad \begin{array}{l} \text{if } \text{blocked}(p_1, \alpha, n) = 0 \\ \text{and } p_1 \neq v := a, \text{break} \end{array}$$

$$(1.19) \quad \frac{\langle p_2, \alpha, m \rangle \xrightarrow{n} \langle p'_2, \alpha', m' \rangle}{\langle p_1 \text{ and } p_2, \alpha, m \rangle \xrightarrow{n} \langle p_1 \text{ and } p'_2, \alpha', m' \rangle} \quad \begin{array}{l} \text{if } \text{blocked}(p_1, \alpha, n) = 1 \\ \text{and } p_2 \neq v := a, \text{break} \end{array}$$

Par/or

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

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

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

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

$$(1.24) \quad \frac{\langle p_1, \alpha, m \rangle \xrightarrow{n} \langle p'_1, \alpha', m' \rangle}{\langle p_1 \text{ or } p_2, \alpha, m \rangle \xrightarrow{n} \langle p'_1 \text{ or } p_2, \alpha', m' \rangle} \quad \begin{array}{l} \text{if } \text{blocked}(p_1, \alpha, n) = 0 \\ \text{and } p_1 \neq v := a, \text{break} \end{array}$$

$$(1.25) \quad \frac{\langle p_2, \alpha, m \rangle \xrightarrow{n} \langle p'_2, \alpha', m' \rangle}{\langle p_1 \text{ or } p_2, \alpha, m \rangle \xrightarrow{n} \langle p_1 \text{ or } p'_2, \alpha', m' \rangle} \quad \begin{array}{l} \text{if } \text{blocked}(p_1, \alpha, n) = 1 \\ \text{and } p_2 \neq v := a, \text{break} \end{array}$$

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 $p, p_1, p_2 \in P$, $\alpha, \alpha_1, \alpha_2 \in E^*$, $m, m_1, m_2 \in \mathcal{M}$, and $n \in N$,*

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

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**, $v := a$, or **break**, as there are no rules to evaluate such programs.)

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

[Case 2] $p = @awaiting(e, n')$, 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 $p_1 = p_2 = \text{skip}$, $\alpha_1 = \alpha_2 = \alpha$, and $m_1 = m_2 = m$.

[Case 3] $p = \text{emit}(e)$, for some $e \in E$. Then d_1 and d_2 are instances of axiom (1.3), and as such, $p_1 = p_2 = @emitting(n')$ with $n' = |\alpha|$, and $\alpha_1 = \alpha_2 = e\alpha$ and $m_1 = m_2 = m$.

[Case 4] $p = @emitting(e, n')$, 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 $p_1 = p_2 = \text{skip}$, $\alpha_1 = \alpha_2 = \alpha$ and $m_1 = m_2 = m$.

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

[Case 5.1] $\text{eval}(b, m) = 1$. 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] $\text{eval}(b, m) = 0$. 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] $p' = v := a$, 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, $m_1 = m_2 = m[v/\text{eval}(a)]$.

[Case 6.2] $p' = \text{break}$. 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] $p' \neq v := a, \text{break}$. 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 $d'_1 < d_1$ and $d'_2 < d_2$, 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] $p = \text{loop } p'$, for some $p' \in P$. Then d_1 and d_2 are instances of axiom (1.10), and as such, $p_1 = p_2 = p' @ \text{loop } p'$, $\alpha_1 = \alpha_2 = \alpha$, and $m_1 = m_2 = m$.

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

[Case 8.1] $p = v := a @ \text{loop } p'$, 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, $p_1 = p_2 = \text{loop } p'$, $\alpha_1 = \alpha_2 = \alpha$, and as $eval$ is a total function, $m_1 = m_2 = m[v/eval(a)]$.

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

[Case 8.3] $p = p' @ \text{loop } p''$, for some $p', p'' \in P$ such that $p' \neq v := a, \text{break}$. 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 $d'_1 < d_1$ and $d'_2 < d_2$, by induction hypothesis, $\alpha_1 = \alpha$, $m_1 = m_2$, and $p'_1 = p'_2$, which implies

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

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

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

[Case 9.3] $p = \text{break and } p'$, 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, $p_1 = p_2 = clear(p'); \text{break}$.

[Case 9.4] $p = p'$ and 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, $p_1 = p_2 = clear(p'); \text{break}$.

[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 $d'_1 < d_1$ and $d'_2 < d_2$, by induction hypothesis, $\alpha_1 = \alpha_2$, $m_1 = m_2$, and $p'_1 = p'_2$, which implies

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

If, however, $\text{blocked}(p') = 1$, 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 $d''_1 < d_1$ and $d''_2 < d_2$, by induction hypothesis, $\alpha_1 = \alpha_2$, $m_1 = m_2$, and $p''_1 = p''_2$, which implies

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

[Case 10] $p = p'$ 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 $n \in \mathbb{N}$, if $p \neq \text{skip}, v := a, \text{break}$ then either*

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

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).

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

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

Theorem 1.7 (Determinism of the outer-step relation). *For all $p, p_1, p_2 \in P$, $\alpha, \alpha_1, \alpha_2 \in E^*$, $m, m_1, m_2 \in \mathcal{M}$, and $n \in \mathbb{N}$,*

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

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 $p \neq \text{skip}, v := a, \text{break}$ then*

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

1.4 The reaction relation

From the outer-step relation we define the reaction relation, which computes a full program reaction.

[TODO: Definition (\Rightarrow^*), determinism, and termination].