1 Formal Semantics

1.1 Abstract Syntax

- Simple rules

$\langle p \rangle$ ::=	mem(varid)	(memory access to '`varid`')
|	await(evtid)	(await event '`evtid`')
|	emit(evtid)	(emit event '`evtid`')
|	break	(escape)

- Compound expressions

$\langle p \rangle$ ::=	if mem(varid) then p else p	(conditional)
|	p ; p	(sequence)
|	loop p	(loop)
|	p and p	(par/and)
|	p or p	(par/or)

- Semantic rules

$\langle p \rangle$ ::=	awaiting(evtid, n)	(awaiting event '`evtid`' since sequence number '`n`')
|	emitting(n)	(emitting on stack level '`n`')
|	p @ loop p	(unwinded loop)
|	fin p	(finalization)

1.2 Operational Semantics

Relation-inner

$$< S, p > \xrightarrow{n} < S', p' >$$

where:

$S, S' \in evtid$	(sequence of event identifiers)
$p, p' \in P$	(programs following the abstract syntax)
$n \in \mathbb{N}$	(unique identifier for the reaction chain)

Rules:

$$\langle S, await(evtid) \rangle \xrightarrow{n} \langle S, awaiting(id, n+1) \rangle \qquad \text{(await)}$$

$$\langle evt : S, awaiting(evtid, m) \rangle \xrightarrow{n} \langle id : S, nop \rangle, M < n \qquad \text{(awake)}$$

$$\langle S, emit(evtid) \rangle \xrightarrow{n} \langle id : S, emitting(|S|) \rangle \qquad \text{(emit)}$$

$$\langle S, emitting(|S|) \rangle \xrightarrow{n} \langle S, nop) \rangle \qquad \text{(pop)}$$

$$\langle S, break\rangle \xrightarrow{n} \langle S, nop\rangle \qquad \text{break}$$

Compound expressions:

$$\frac{val(varid, n) \neq 0}{\langle S, \text{if mem(varid) then p else q}\rangle \xrightarrow{n} \langle S, p\rangle} \qquad \text{if-true}$$

$$\frac{val(varid, n) = 0}{\langle S, \text{if mem(varid) then p else q}\rangle \xrightarrow{n} \langle S, q\rangle} \qquad \text{if-false}$$

$$\frac{\langle S, p\rangle \xrightarrow{n} \langle S', p'\rangle}{\langle S, (p; q)\rangle \xrightarrow{n} \langle S', (p'; q)\rangle} \qquad \text{seq-adv}$$

$$\langle S, (nop; q)\rangle \xrightarrow{n} \langle S, q\rangle \qquad \text{seq-nop}$$

$$\langle S, (break; p)\rangle \xrightarrow{n} \langle S, break\rangle \qquad \text{break-seq}$$