

A semantics for Céu



www.ceu-lang.org

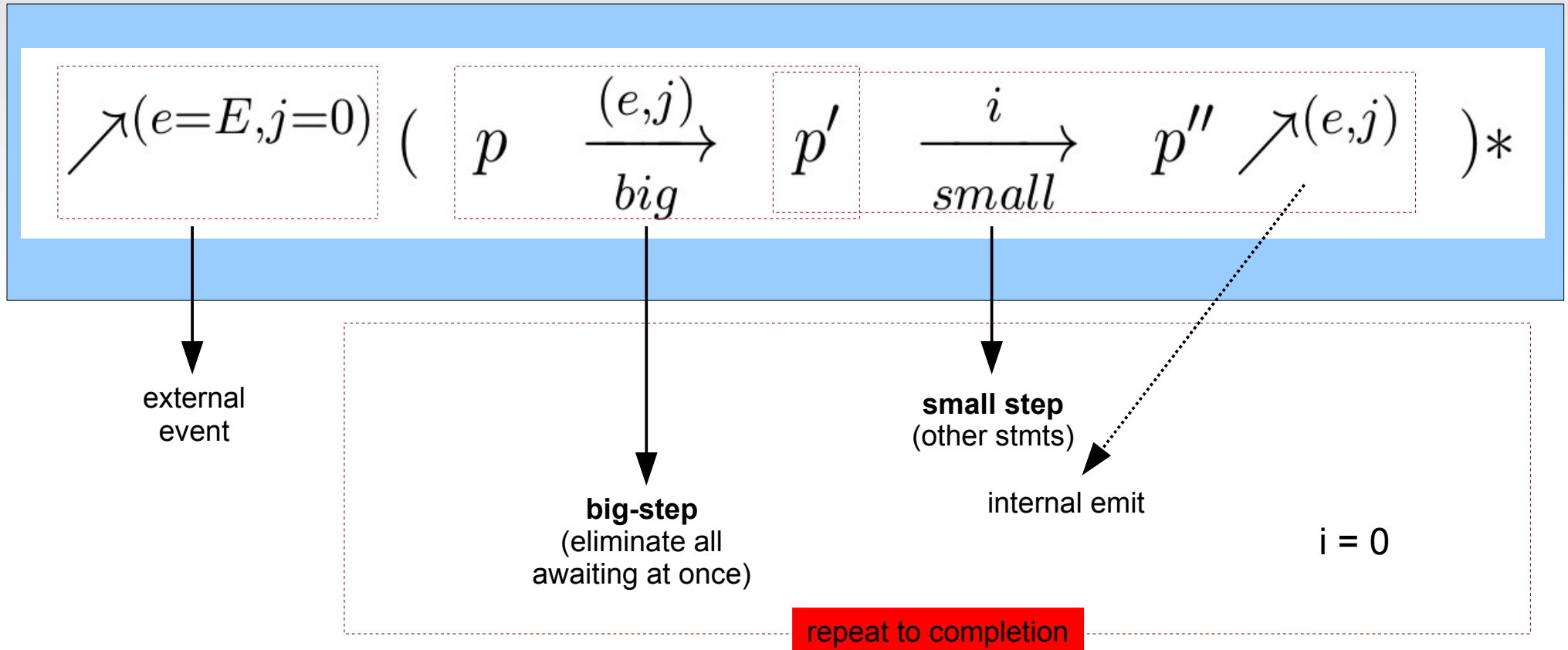
```
input int Restart;
event int v = 0;
par do
    loop do                // 1st trail
        await 1s;
        emit v(v+1);
    end
with
    loop do                // 3rd trail
        v = await Restart;
        emit v;
    end
with
    loop do                // 2nd trail
        await v;
        _printf("v = %d\n", v);
    end
end
```

Peculiarities of Céu

- Synchronous model
 - trails synchronize at each event
- Internal events
 - "stacked" execution
- Determinism
 - handled in separate


nothing	(nothing)
mem	(memory access)
await e	(event await -- int/ext)
emit e	(event emit -- int)
p ; q	(sequence)
if mem then p else q	(conditional)
loop p	(loop)
break	(break loop)
p or q	(parallel/or)
p and q	(parallel/and)

Reaction chain



Example 1

```
p = (await A ; await A)
```

 (A, 0)

$p = (\text{await } A ; \text{await } A)$

$\text{big}(A,0)$

(seq)
$$\frac{(\text{await } 1) \quad \text{await } A \xrightarrow{(A,0)} \text{mark } 0}{(\text{await } A ; \text{await } A) \xrightarrow{(A,0)} (\text{mark } 0 ; \text{await } A)}$$

$p' = (\text{mark } 0 ; \text{await } A)$
 $i=0$

$\text{small}(0)$

(seq 1)
$$\frac{(\text{mark}) \quad (\text{mark } 0) \xrightarrow{0} \text{nothing} \nearrow \emptyset}{(\text{mark } 0 ; \text{await } A) \xrightarrow{0} (\text{nothing} ; \text{await } A) \nearrow \emptyset}$$

$\text{big}(-,-)$

$\text{small}(0)$

(seq 2)
$$(\text{nothing} ; \text{await } A) \xrightarrow{i} \text{await } A \nearrow \emptyset$$

Example 2

```
p = loop
    if mem then
        break
    else
        await A
```

↖ (␣,0)

↖ (A,0)


```

p = loop
  if mem then
    break
  else
    await A

```

big(0)

(loop 1)
$$\frac{(\text{if}) \quad (if \dots) \xrightarrow{(-,0)} (if \dots)}{loop (if \dots) \xrightarrow{(-,0)} (if \text{ mem}(v) \text{ then break else await } A) @ loop (if \dots)}$$

```

p' = (if mem(v) then break else await A)
      @ loop (if ...)

```

small(0)

(loop 1)
$$\frac{(\text{if } 1) \quad (if \text{ mem}(1) \dots) \xrightarrow{0} \text{break} \nearrow \emptyset}{(if \text{ mem}(1) \dots @ loop q) \xrightarrow{0} (\text{break} @ loop (if \dots)) \nearrow \emptyset}$$

small(0)

(loop 3)
$$(break @ loop (if \dots)) \xrightarrow{0} \text{nothing} \nearrow \emptyset$$

```

p = loop
  if mem then
    break
  else
    await A

```

big(_,0)

(loop 1)
$$\frac{(\text{if}) \quad (if \dots) \xrightarrow{(-,0)} (if \dots)}{loop (if \dots) \xrightarrow{(-,0)} (if \text{ mem}(v) \text{ then break else await } A) @ loop (if \dots)}$$

```

p' = (if mem(v) then break else await A)
      @ loop (if ...)
          i = 0

```

small(0)

(loop 1)
$$\frac{(\text{if } 0) \quad (if \text{ mem}(0) \dots) \xrightarrow{0} \text{await } A \nearrow \emptyset}{(if \text{ mem}(0) \dots @ loop q) \xrightarrow{0} (\text{await } A @ loop (if \dots)) \nearrow \emptyset}$$

`p = await A`
`@ loop (if ...)`

big(A,0)

(loop 2)
$$\frac{(\text{await 1}) \quad \text{await } A \xrightarrow{(A,0)} \text{mark } 0}{\text{await } A @ \text{ loop (if ...)} \xrightarrow{(e,j)} \text{mark } 0 @ \text{ loop (if ...)}}$$

`p' = mark 0 @ loop (if ...)`

small(0)

(loop 1)
$$\frac{(\text{mark}) \quad (\text{mark } 0) \xrightarrow{0} \text{nothing} \nearrow \emptyset}{\text{mark } 0 @ \text{ loop (if ...)} \xrightarrow{0} \text{nothing} @ \text{ loop (if ...)} \nearrow \emptyset}$$

small(0)

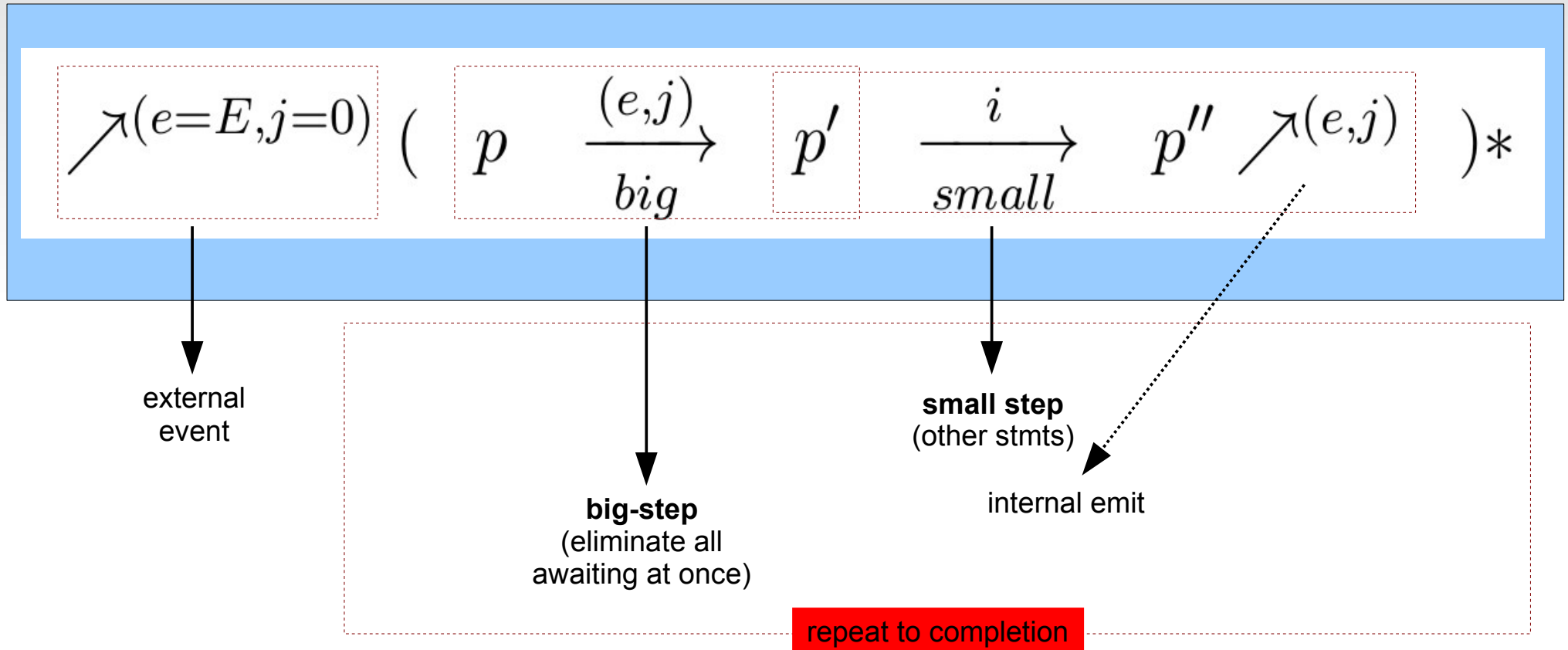
(loop 2)
$$(\text{nothing} @ \text{ loop (if ...)} \xrightarrow{0} (\text{if ...}) @ \text{ loop (if ...)} \nearrow \emptyset$$

Levels of priority

```
par/or do
  await a;
with
  await A;
  emit a;
with
  await A;
end
```

```
par/or do
  await a;
  ??? (+2)
with
  await A;
  emit a;
  ??? (+1)
with
  await A;
  ??? (-)
end
```

Reaction chain



$$i = 0 \quad ; \quad \frac{\text{prio}(p') \neq +\infty}{i \leftarrow \text{prio}(p')}$$

Example 3

```
par/or do
    await a;
    emit b;
    // no
with
    await b;
    // yes
with
    await A;
    emit a;
    // no
end
```

```
p = (await a ; emit b)
    or
    ( await b
      or
        (await A ; emit a) )
```

```

p = (await a ; emit b)
    or
    ( await b
      or
        (await A ; emit a) )

```

big(A,0)

$$\begin{array}{c}
 \text{(or)} \quad \frac{
 \begin{array}{c}
 \text{(seq)} \dots \xrightarrow{(A,0)} \dots \quad \text{(or)} \quad \frac{
 \begin{array}{c}
 \text{(await 2) } \text{await } b \xrightarrow{(A,0)} \text{await } b \quad \text{(seq)} \quad \frac{
 \begin{array}{c}
 \text{(await 1) } \text{await } A \xrightarrow{(A,0)} \text{mark } 0 \\
 \text{(await } A ; \text{emit } a) \xrightarrow{(A,0)} (\text{mark } 0 ; \text{emit } a)
 \end{array}
 }{
 \text{(await } b \text{ or (await } A ; \text{emit } a)) \xrightarrow{(A,0)} (\text{await } b \text{ or (mark } 0 ; \text{emit } a))
 }
 \end{array}
 }{
 \dots \text{ or (await } b \text{ or (await } A ; \text{emit } a)) \xrightarrow{(A,0)} \dots \text{ or (await } b \text{ or (mark } 0 ; \text{emit } a))
 }
 \end{array}
 \end{array}$$

```

(await a ; emit b) or
(await b or (mark 0; emit a))

```

(await a ; emit b) or
(await b or (mark 0; emit a))

small*(0)

$$\frac{\frac{(\text{seq 1}) \xrightarrow{(\text{mark}) (mark\ 0) \xrightarrow{0} \text{nothing}}}{(mark\ 0; emit\ a) \xrightarrow{0} \text{nothing} ; emit\ a} \quad (\text{seq 2}) (nothing ; emit\ a) \xrightarrow{0} emit\ a}{(mark\ 0 ; emit\ a) \xrightarrow{0} emit\ a} \quad (\text{emit}) emit\ a \xrightarrow{0} mark\ 1 \nearrow (a,2)}{(mark\ 0 ; emit\ a) \xrightarrow{0} mark\ 1 \nearrow (a,2)}$$

(await a ; emit b) or (await b or mark 1)

big(a,2)

(await 1)

(mark 2 ; emit b) or (await b or mark 1)

small(2)

(mark 3) or (await b or mark 1) $\nearrow (b,4)$

big(b,4)

(mark 3) or (mark 4 or mark 1)

small(4)

(or 4), (or 3)

small(4)

nothing

Nondeterminism

mem and mem

$$\text{(and 1)} \quad \frac{isReady(p, i) \quad p \xrightarrow{i} p' \nearrow (e, j)}{(p \text{ and } q) \xrightarrow{i} (p' \text{ and } q) \nearrow (e, j)}$$

$$\text{(and 2)} \quad \frac{isReady(q, i) \quad q \xrightarrow{i} q' \nearrow (e, j)}{(p \text{ and } q) \xrightarrow{i} (p \text{ and } q') \nearrow (e, j)}$$

Unbounded execution

loop mem

