1 Preamble

- section name and its parents
- basic process header
- typed channel
- generic typed channels
- synchronisation channels
- various auxiliary declarations used

section *action_grammar_rules* **parents** *circustime_toolkit*

channel $d : \mathbb{N}$

| $outside : \mathbb{N}$

channel $c : \mathbb{N} \times \mathbb{N} \times \mathbb{N} \times \mathbb{N}$

process $CircusTimeActionTests \mathrel{\widehat=}$ **begin**

| $n1, n2 : \mathbb{N}$
| $x?, y!, z? : \mathbb{N}$

| $f : \mathbb{N} \to \mathbb{N} \times \mathbb{N}$

$S == [\, y : \mathbb{N} \,]$

$letExpr == \mathbf{let}\ x == 1 \bullet x$
$muExpr1 == (\mu\, x : \mathbb{N} \bullet x)$
$muExpr2 == (\mu\, x : \mathbb{N} \mid true)$
$condExpr == \mathbf{if}\ true\ \mathbf{then}\ 1\ \mathbf{else}\ 2$
$bindExpr == \langle\!|\, one == 1 \,|\!\rangle$
$tupleExpr == (1, \{2\}, \mathbb{P}\{3\})$

$A \mathrel{\widehat=} \mathbf{Skip}$

2 Example 1 — *CIRCWAIT*

Production rule:

`CIRCWAIT:cw expression:e`

- simple wait with expressions of various kinds and no following action
- expressions can come from within and outside the process
- expressions that look **funny like partially applied on several type expressions like theta, mu, condition, bind and tuple expressions**

$Test0 \mathrel{\widehat=} \mathbf{wait}\, 10 + outside$

$Test1 \mathrel{\widehat=} \mathbf{wait}\, n1 + n2$

$Test2 \mathrel{\widehat=} \mathbf{wait}\, f$

$Test3 \mathrel{\widehat=} \mathbf{wait}\, \theta\, S$

$Test4 \mathrel{\widehat=} \mathbf{wait}[\, x : \mathbb{N} \mid x > 10 \,]$

$Test5 \mathrel{\widehat=} \mathbf{wait}\, letExpr$

$Test6 \mathrel{\widehat=} \mathbf{wait}\, muExpr1$

$Test7 \mathrel{\widehat=} \mathbf{wait}\, muExpr2$

$Test8 \mathrel{\widehat=} \mathbf{wait}\, condExpr$

$Test9 \mathrel{\widehat=} \mathbf{wait}\, bindExpr$

$Test10 \mathrel{\widehat=} \mathbf{wait}\, tupleExpr.1$

$Test11 \mathrel{\widehat=} \mathbf{wait}\, 1 \mathinner{..} 10$

$Test12 \mathrel{\widehat=} c \longrightarrow \mathbf{wait}\, outside$

$Test13 \mathrel{\widehat=} \mathbf{wait}\, outside \;;\; \mathbf{wait}\, outside$

$Test14 \mathrel{\widehat=} c \longrightarrow \mathbf{wait}\, outside \;;\; c \longrightarrow \mathbf{wait}\, outside$

$Test15 \mathrel{\widehat=} d!x \longrightarrow \mathbf{wait}\, x + outside$

$Test16 \mathrel{\widehat=} \mathbf{vres}\, vart : \mathbb{N} \bullet \mathbf{wait}\, vart$

$Test17 \mathrel{\widehat=} \mathbf{vres}\, vart : \mathbb{N} \bullet c \longrightarrow \mathbf{wait}\, vart + outside$

The following examples presents with erroneous productions like

```
\t1 TestX1 \circdef \circwait 10 \then \Skip
\t1 TestX2 \circdef \circwait \lcirctime 10 \rcirctime
```

Production rule:

```
CIRCWAIT DECLWORD:dw COLON expression:e CIRCSPOT circusAction:ac
```

- complex wait patterns with ensuing actions
- complex wait patterns with duplicated names (i.e. already declared)
- the strokes are accepatable by the parser, but that must be avoided during type checking

$$Test18 \mathrel{\widehat=} \mathbf{wait}\ x : 10 + outside \bullet \mathbf{Skip}$$

$$Test19 \mathrel{\widehat=} \mathbf{wait}\ y : n1 + n2 \bullet \mathbf{Skip}$$

$$Test20 \mathrel{\widehat=} \mathbf{wait}\ z : f \bullet \mathbf{Skip}$$

$$Test21 \mathrel{\widehat=} \mathbf{wait}\ w : \theta\, S \bullet \mathbf{Skip}$$

$$Test22 \mathrel{\widehat=} \mathbf{wait}\ x? : S \bullet \mathbf{Skip}$$

$$Test23 \mathrel{\widehat=} \mathbf{wait}\ f : S \bullet \mathbf{Skip}$$

$$Test24 \mathrel{\widehat=} \mathbf{wait}\ m : letExpr \bullet \mathbf{Skip}$$

$$Test25 \mathrel{\widehat=} \mathbf{wait}\ m : muExpr1 \bullet \mathbf{Skip}$$

$$Test26 \mathrel{\widehat=} \mathbf{wait}\ m : muExpr2 \bullet \mathbf{Skip}$$

$$Test27 \mathrel{\widehat=} \mathbf{wait}\ m : condExpr \bullet \mathbf{Skip}$$

$$Test28 \mathrel{\widehat=} \mathbf{wait}\ m : bindExpr \bullet \mathbf{Skip}$$

$$Test29 \mathrel{\widehat=} \mathbf{wait}\ m : tupleExpr \bullet \mathbf{Skip}$$

$$Test30 \mathrel{\widehat=} \mathbf{wait}\ m : 1 \mathinner{..} 20 \bullet c \longrightarrow \mathbf{Skip}$$

$$Test31 \mathrel{\widehat=} c \longrightarrow \mathbf{wait}\ m : 1 \mathinner{..} 20 \bullet c \longrightarrow \mathbf{Skip}$$

$$Test32 \mathrel{\widehat=} \mathbf{wait}\ m : 1 \mathinner{..} 20 \bullet c \longrightarrow \mathbf{wait}\ outside$$

$$Test33 \mathrel{\widehat=} d!u \longrightarrow \mathbf{wait}\ m : 1 \mathinner{..} 20 \bullet \mathbf{wait}\ outside + u$$

$$Test34 \mathrel{\widehat=} \mathbf{wait}\ m : 1 \mathinner{..} 20 \bullet \mathbf{Skip}\ ;\ \mathbf{wait}\ n : 1 \mathinner{..} 20 \bullet \mathbf{Skip}$$

$$Test35 \mathrel{\widehat=} \mathbf{vres}\ vart : \mathbb{N} \bullet \mathbf{wait}\ m : 1 \mathinner{..} 20 \bullet \mathbf{Skip}$$

$$Test36 \mathrel{\widehat=} \mathbf{vres}\ vart : \mathbb{N} \bullet \mathbf{wait}\ m : 1 \mathinner{..} 20 \bullet c \longrightarrow \mathbf{wait}\ vart + outside$$

$$Test37 \mathrel{\widehat=} \mathbf{wait}\ m : 1 \mathinner{..} 20 \bullet \mathbf{wait}\ n : 1 \mathinner{..} 20 \bullet \mathbf{Skip}$$

3 Example 2 — *CIRCSTARTBY* and *CIRCENDBY*

Production rule:

```
LCIRCTIME expression:e RCIRCTIME CIRCSTARTBY circusAction:ca
circusAction:ca CIRCENDBY LCIRCTIME expression:e RCIRCTIME
```

- StartBy and EndBy operatrs with actions
- Expressions are acceptable by parser (similar to wait expression)

$$Test38 \mathrel{\widehat=} \langle 10 + outside \rangle \blacktriangleleft \mathbf{Skip}$$

$$Test39 \mathrel{\widehat=} \langle n1 + n2 \rangle \blacktriangleleft \mathbf{Skip}$$

$$Test40 \mathrel{\widehat=} \langle f \rangle \blacktriangleleft \mathbf{Skip}$$

$$Test41 \mathrel{\widehat=} \langle \theta\, S \rangle \blacktriangleleft \mathbf{Skip}$$

$$Test42 \mathrel{\widehat=} \langle x? \rangle \blacktriangleleft \mathbf{Skip}$$

$$Test43 \mathrel{\widehat=} \langle letExpr \rangle \blacktriangleleft \mathbf{Skip}$$

$$Test44 \mathrel{\widehat=} \langle muExpr1 \rangle \blacktriangleleft \mathbf{Skip}$$

$$Test45 \mathrel{\widehat=} \langle muExpr2 \rangle \blacktriangleleft \mathbf{Skip}$$

$$Test46 \mathrel{\widehat=} \langle condExpr \rangle \blacktriangleleft \mathbf{Skip}$$

$$Test47 \mathrel{\widehat=} \langle bindExpr \rangle \blacktriangleleft \mathbf{Skip}$$

$$Test48 \mathrel{\widehat=} \langle tupleExpr.1 \rangle \blacktriangleleft \mathbf{Skip}$$

$$Test49 \mathrel{\widehat=} \langle 1 \mathinner{..} 20 \rangle \blacktriangleleft \mathbf{Skip}$$

$$Test50 \mathrel{\widehat=} \langle outside \rangle \blacktriangleleft c \longrightarrow \mathbf{Skip}$$

$$Test51 \mathrel{\widehat=} c \longrightarrow \langle outside \rangle \blacktriangleleft \mathbf{Skip}$$

$$Test52 \mathrel{\widehat=} c \longrightarrow \langle outside \rangle \blacktriangleleft d \longrightarrow \langle outside \rangle \blacktriangleleft \mathbf{Skip}$$

$$Test53 \mathrel{\widehat=} d!u \longrightarrow \langle u \rangle \blacktriangleleft \mathbf{Skip}$$

$$Test54 \mathrel{\widehat=} \mathbf{vres}\ vart : \mathbb{N} \bullet c \longrightarrow \langle vart \rangle \blacktriangleleft \mathbf{Skip}$$

$$Test55 \mathrel{\widehat=} \mathbf{vres}\ vart : \mathbb{N} \bullet c \longrightarrow \langle vart + outside \rangle \blacktriangleleft \mathbf{Skip}$$

$$Test56 \mathrel{\widehat=} A \ ;\ \langle 10 + outside \rangle \blacktriangleleft \mathbf{Skip}$$

An erroneous example for the gievn production

```
\t1 TestX3 \circdef A \lcirctime 10 + outside \rcirctime \circstartby \Skip
\t1 TestX4 \circdef  10 \circstartby A
```

$$Test57 \mathrel{\widehat=} A \blacktriangleright \langle 10 + outside \rangle$$

$$Test58 \mathrel{\widehat=} A \blacktriangleright \langle n1 + n2 \rangle$$

$$Test59 \mathrel{\widehat=} A \blacktriangleright \langle f \rangle$$

$$Test60 \mathrel{\widehat=} A \blacktriangleright \langle \theta\, S \rangle$$

$$Test61 \mathrel{\widehat=} A \blacktriangleright \langle x? \rangle$$

$$Test62 \mathrel{\widehat=} A \blacktriangleright \langle 1 \mathinner{..} 20 \rangle$$

$$Test63 \mathrel{\widehat=} A \blacktriangleright \langle letExpr \rangle$$

$$Test64 \mathrel{\widehat=} A \blacktriangleright \langle muExpr \rangle$$

$$Test65 \mathrel{\widehat=} A \blacktriangleright \langle muExpr1 \rangle$$

$$Test66 \mathrel{\widehat=} A \blacktriangleright \langle condExpr \rangle$$

$$Test67 \mathrel{\widehat=} A \blacktriangleright \langle bindExpr \rangle$$

$$Test68 \mathrel{\widehat=} A \blacktriangleright \langle tupleExpr \rangle$$

$$Test69 \mathrel{\widehat=} c \longrightarrow A \blacktriangleright \langle outside \rangle$$

$$Test70 \mathrel{\widehat=} d!u \longrightarrow A \blacktriangleright \langle u \rangle$$

$$Test71 \mathrel{\widehat=} \mathbf{vres}\ vart : \mathbb{N} \bullet c \longrightarrow A \blacktriangleright \langle vart \rangle$$

$$Test72 \mathrel{\widehat=} \mathbf{vres}\ vart : \mathbb{N} \bullet c \longrightarrow A \blacktriangleright \langle vart + outside \rangle$$

$$Test73 \mathrel{\widehat=} A \blacktriangleright \langle outside \rangle \fatsemi A$$

An erroneous example for the gievn production

```
\t1 TestX5 \circdef A \circendby \lcirctime f \rcirctime \then A
        \t1 TestX6 \circdef A \circendby 10
```

4 Example 3 — *CIRCTIMEOUT*

Production rule:

```
circusAction:al CIRCTIMEOUT LCIRCTIME expression:e RCIRCTIME  circusAction:ar
```

- Timeout operator with actions
- Expressions are acceptable by parser (similar to previous expression patterns)

$$Test74 \mathrel{\widehat=} A \overset{\langle 10+outside\rangle}{\rhd} A$$

$$Test75 \mathrel{\widehat=} c \longrightarrow A \overset{\langle outside\rangle}{\rhd} A$$

$$Test76 \mathrel{\widehat=} c \longrightarrow A \overset{\langle outside\rangle}{\rhd} d \longrightarrow A$$

$$Test77 \mathrel{\widehat=} A \overset{\langle f\rangle}{\rhd} A \,;\; A$$

$$Test78 \mathrel{\widehat=} c \longrightarrow A \overset{\langle f\rangle}{\rhd} c \longrightarrow A \,;\; d \longrightarrow A$$

$$Test79 \mathrel{\widehat=} d!u \longrightarrow A \overset{\langle u\rangle}{\rhd} A$$

$$Test80 \mathrel{\widehat=} \mathbf{vres}\ vart : \mathbb{N} \bullet c \longrightarrow A \overset{\langle vart\rangle}{\rhd} A$$

$$Test81 \mathrel{\widehat=} \mathbf{vres}\ vart : \mathbb{N} \bullet c \longrightarrow A \overset{\langle vart+outside\rangle}{\rhd} A$$

An erroneous example for the gievn production

```
\t1 TestX7 \circdef A \circtimeout 10 A
```

5 Example 4 — *CIRCTIMEDINTERRUPT*

Production rule:

```
circusAction CIRCTIMEDINTERRUPT LCIRCTIME expression RCIRCTIME circusAction
```

- Timedinterrupt operator with actions
- Expressions are acceptable by parser (similar to previous expression patterns)

$$Test82 \mathrel{\widehat=} A \triangle_{\langle 10+outside\rangle} A$$

$$Test83 \mathrel{\widehat=} c \longrightarrow A \triangle_{\langle outside\rangle} A$$

$$Test84 \mathrel{\widehat=} c \longrightarrow A \triangle_{\langle outside\rangle} d \longrightarrow A$$

$$Test85 \mathrel{\widehat=} A \triangle_{\langle f\rangle} A \,;\; A$$

$$Test86 \mathrel{\widehat=} c \longrightarrow A \triangle_{\langle f\rangle} c \longrightarrow A \,;\; d \longrightarrow A$$

$$Test87 \mathrel{\widehat=} d!u \longrightarrow A \triangle_{\langle u\rangle} A$$

$$Test88 \mathrel{\widehat=} \mathbf{vres}\ vart : \mathbb{N} \bullet c \longrightarrow A \triangle_{\langle vart\rangle} A$$

$$Test89 \mathrel{\widehat=} \mathbf{vres}\ vart : \mathbb{N} \bullet c \longrightarrow A \triangle_{\langle vart+outside\rangle} A$$

An erroneous example for the gievn production

```
\t1 TestX8 \circdef A \circtimedinterrupt 10 A
```

6 Example 5 — timed prefix

Production rule:

```
communication PREFIXTHEN:pt LCIRCTIME expression RCIRCTIME circusAction
```

- Timed prefix with communication channel is uesed to express the complexity of timed communication using circus-time operators to test the smart scanner.
- Expressions are acceptable by parser (similar to previous expression patterns)

$$Test90 \mathrel{\widehat=} c?x?y?z \longrightarrow \langle 10 + outside \rangle \mathbf{Skip}$$

$$Test91 \mathrel{\widehat=} c?x!n1.n2 \longrightarrow \langle x? \rangle \mathbf{Skip}$$

$$Test92 \mathrel{\widehat=} c!n1?x.n2 \longrightarrow \langle f \rangle \mathbf{Skip}$$

$$Test93 \mathrel{\widehat=} c.n1!n2?x \longrightarrow \langle 20 \rangle d \longrightarrow \langle \theta\, S \rangle \mathbf{Skip}$$

$$Test94 \mathrel{\widehat=} d \longrightarrow \langle f \rangle e \longrightarrow \langle f \rangle \mathbf{Skip}$$

$$Test95 \mathrel{\widehat=} c\,?i\,!(f\ i) \longrightarrow \langle f \rangle \mathbf{Skip}$$

$$Test96 \mathrel{\widehat=} c.(S.y)?z \longrightarrow \langle 20 \rangle \mathbf{Skip}$$

$$Test97 \mathrel{\widehat=} c.(x?)!(y!)!(z?) \longrightarrow \langle 20 \rangle \mathbf{Skip}$$

$$Test98 \mathrel{\widehat=} c.(x?)!(y!)!(z?) \longrightarrow \langle 20 \rangle \langle outside \rangle \blacktriangleleft \mathbf{Skip}$$

$$Test99 \mathrel{\widehat=} c.(x?)!(y!)!(z?) \longrightarrow \langle 20 \rangle A \blacktriangleright \langle outside \rangle$$

$$Test100 \mathrel{\widehat=} c.(x?)!(y!)!(z?) \longrightarrow \langle 20 \rangle \mathbf{wait}\ outside$$

$$Test101 \mathrel{\widehat=} c.(x?)!(y!)!(z?) \longrightarrow \langle 20 \rangle \mathbf{wait}\ t : \mathbb{N} \bullet A$$

$$Test102 \mathrel{\widehat=} c.(x?)!(y!)!(z?) \longrightarrow \langle 20 \rangle d \longrightarrow A \overset{\langle outside \rangle}{\rhd} \mathbf{Skip}$$

$$Test103 \mathrel{\widehat=} c.(x?)!(y!)!(z?) \longrightarrow \langle 20 \rangle d \longrightarrow \langle 20 \rangle A \overset{\langle outside \rangle}{\rhd} \mathbf{Skip}$$

***** $S \in \mathbb{P}(\langle\!| y == \mathbb{N} |\!\rangle)$**, hence** $S.y \in \mathbb{N}$**.**

***** $x?, y!, z?$ **are decorated names; usually they appear in schemas.** If this proves too complicated, perhaps having it separate as another file is a good away to divide-and-conquer.

Description

Action	Communication pattern
$Test25$	$\mathrm{In}(x, \mathbb{N})$, $\mathrm{In}(y, \mathbb{N})$, $\mathrm{In}(z, \mathbb{N})$
$Test26$	$\mathrm{In}(x, \mathbb{N})$, $\mathrm{Out}(n1)$, $\mathrm{Dot}(n2)$
$Test27$	$\mathrm{Out}(n1)$, $\mathrm{In}(x, \mathbb{N})$, $\mathrm{Dot}(n2)$
$Test28$	$\mathrm{Dot}(n1)$, $\mathrm{Out}(n2)$, $\mathrm{In}(x, \mathbb{N})$, Synch
$Test29$	Synch, Synch, Synch
$Test30$	$\mathrm{In}(i, \mathbb{N})$, $\mathrm{Out}(\mathbb{N} \times \mathbb{N})$
$Test31$	$\mathrm{Dot}(S.y)$, $\mathrm{In}(z, \mathbb{N} \times \mathbb{N})$
$Test32$	$\mathrm{Out}(x?)$, $\mathrm{Out}(y!)$, $\mathrm{Out}(z?)$

Timed comminication prefix with restricting expression.

$$Test104 \mathrel{\widehat=} c?x : (x > 1)!(f\ x) \longrightarrow \mathbf{Skip}$$

$$Test105 \mathrel{\widehat=} c?x \longrightarrow c?z : (z > x.1).(f\ (x.2 + x.3)) \longrightarrow \mathbf{Skip}$$

$$Test106 \mathrel{\widehat=} c?x : (x > 1).(f\ (x.2 + x.3)) \longrightarrow \langle outside \rangle \mathbf{Skip}$$

$$Test107 \mathrel{\widehat=} c?x : (x > 1).(f\ (x.2 + x.3)) \longrightarrow \langle outside \rangle c?z : (z > x.1).(f\ (x.2 + x.3)) \longrightarrow \mathbf{Skip}$$

$$Test108 \mathrel{\widehat=} c?x : (x > 1).(f\ (x.2 + x.3)) \longrightarrow \langle outside \rangle c?z : (z > x.1).(f\ (x.2 + x.3)) \longrightarrow \langle 20 \rangle \mathbf{Skip}$$

$$Test109 \mathrel{\widehat=} c?x : (x > 1).(f\ (x.2 + x.3)) \longrightarrow \langle outside \rangle c?z : (z > x.1).(f\ (x.2 + x.3)) \longrightarrow \langle 20 \rangle \langle outsi$$

$$Test110 \mathrel{\widehat=} c?x : (x > 1).(f\ (x.2 + x.3)) \longrightarrow \langle outside \rangle c?z : (z > x.1).(f\ (x.2 + x.3)) \longrightarrow \langle 20 \rangle A \blacktriangleright \langle$$

Description

Action	Communication pattern
$Test33$	$\mathrm{In}(x, \{\, v : \mathbb{N} \mid v > 1 \,\})$, $\mathrm{Out}(\mathbb{N} \times \mathbb{N})$
$Test34$	$\mathrm{In}(x, \mathbb{N} \times \mathbb{N} \times \mathbb{N})$; $\mathrm{In}(z, \{\, v : \mathbb{N} \mid v > x.1 \,\})$, $\mathrm{Out}(\mathbb{N} \times \mathbb{N})$

*** type on inputs are restricted according to given predicate.**
***** $Test34$ **input on** z **is from "**$?z : (z > x.1)$**".**
Finally we have generic channels version with time

$$Test111 \mathrel{\widehat=} g[\mathbb{N} \times \mathbb{N} \times \mathbb{N}]?x!n1.n2 \longrightarrow \langle 20 \rangle \mathbf{Skip}$$

$$Test112 \mathrel{\widehat=} g.n1.(f\ n1) \longrightarrow \langle 20 \rangle \mathbf{Skip}$$

7 Example 6 — AT timed prefix

Production rule:

```
communication ATTIME DECORWORD  PREFIXTHEN  circusAction
communication ATTIME DECORWORD PREFIXTHEN LCIRCTIME expression RCIRCTIME circusAction
```

- Grammar for `@` operator with actions

- Grammer for @ operator with expression and action
- Expressions are acceptable by parser (similar to previous expression patterns)
- Comminication channel can allow previous communication patterns and restricted inputs

$$Test113 \mathrel{\widehat=} c?x?y?z@X \longrightarrow \langle 10 + outside\rangle \mathbf{Skip}$$

$$Test114 \mathrel{\widehat=} c?x!n1.n2@Z \longrightarrow \langle x?\rangle \mathbf{Skip}$$

$$Test115 \mathrel{\widehat=} c!n1?x.n2@Y \longrightarrow \langle f\rangle \mathbf{Skip}$$

$$Test116 \mathrel{\widehat=} c.n1!n2?x@x? \longrightarrow \langle 20\rangle d@f \longrightarrow \langle \theta\, S\rangle \mathbf{Skip}$$

$$Test117 \mathrel{\widehat=} d \longrightarrow \langle f\rangle e \longrightarrow \langle f\rangle \mathbf{Skip}$$

$$Test118 \mathrel{\widehat=} c\,?i\,!(f\ i)@W \longrightarrow \mathbf{Skip}$$

$$Test119 \mathrel{\widehat=} c.(S.y)?z@K \longrightarrow \mathbf{Skip}$$

$$Test120 \mathrel{\widehat=} c.(x?)!(y!)!(z?)@Q \longrightarrow \langle 20\rangle \mathbf{Skip}$$

$$Test121 \mathrel{\widehat=} c!n1?x.n2@Y \longrightarrow \langle f\rangle d \longrightarrow \mathbf{Skip}$$

$$Test122 \mathrel{\widehat=} c!n1?x.n2@Y \longrightarrow \langle f\rangle d \longrightarrow \langle outside\rangle \mathbf{Skip}$$

$$Test122 \mathrel{\widehat=} c!n1?x.n2@Y \longrightarrow \langle f\rangle d \longrightarrow \langle outside\rangle \langle 20\rangle \blacktriangleleft \mathbf{Skip}$$

$$Test123 \mathrel{\widehat=} c!n1?x.n2@Y \longrightarrow \langle f\rangle d \longrightarrow \langle outside\rangle \langle 20\rangle \blacktriangleleft \mathbf{wait}\, 20$$

$$Test124 \mathrel{\widehat=} c!n1?x.n2@Y \longrightarrow \langle f\rangle d \longrightarrow \langle outside\rangle \langle 20\rangle \blacktriangleleft \mathbf{wait}\, t : \mathbb{N} \bullet d \longrightarrow \mathbf{Skip}$$

$$Test125 \mathrel{\widehat=} c!n1?x.n2@Y \longrightarrow \langle f\rangle d@X \longrightarrow \langle outside\rangle \langle 20\rangle \blacktriangleleft \mathbf{wait}\, t : \mathbb{N} \bullet d \longrightarrow \mathbf{Skip}$$

8 (! Prolegomena) — basic process footer

It just terminates

- **Skip**

end