1 Preamble

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

section *dot_field_multenv* **parents** *circus_toolkit*

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

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

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

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

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

2 Example 1 — various simple patterns

- multiple actions in one environment
- tabs and various forms of new line
- multiple field patterns: in, out, dot
- field type directly mapped to channel type
- synchronisation channel d on chained prefixing

$$Test0 \mathrel{\widehat=} c?x?y?z \longrightarrow \mathbf{Skip}$$

$$Test1 \mathrel{\widehat=} c?x!n1.n2 \longrightarrow \mathbf{Skip}$$

$$Test2 \mathrel{\widehat=} c!n1?x.n2 \longrightarrow \mathbf{Skip}$$

$$Test3 \mathrel{\widehat=} c.n1!n2?x \longrightarrow d\mathbf{Skip}$$

$$Test4 \mathrel{\widehat=} d \longrightarrow e \longrightarrow \mathbf{Skip}$$

Description

Action	Communication pattern
$Test0$	In$(x, \mathbb{N})$, In$(y, \mathbb{N})$, In$(z, \mathbb{N})$
$Test1$	In$(x, \mathbb{N})$, Out$(n1)$, Dot$(n2)$
$Test2$	Out$(n1)$, In$(x, \mathbb{N})$, Dot$(n2)$
$Test3$	Dot$(n1)$, Out$(n2)$, In$(x, \mathbb{N})$, Synch
$Test4$	Synch, Synch, Synch

LaTeX

To avoid parsing the LaTeX markup within the `\begin{verbatim}` environment we omit the slash before the begin/end environment.

```
begin{circusaction}
 \t1 Test0 \circdef c?x?y?z \then \Skip
     \also
 \t1 Test1 \circdef c?x!n1.n2 \then \Skip \\
 \t1 Test2 \circdef c!n1?x.n2 \then \Skip
     \also
 \t1 Test3 \circdef c.n1!n2?x \then d \Skip
end{circusaction}
begin{circusaction}
 \t1 Test4 \circdef d \then e \then \Skip
end{circusaction}
```

3 Example 2 — complex output expressions

- hard spaces - make no semantic difference
- application expressions on output fields
- parenthesised expressions form **one** field
- last fields get remainder type dimensions
- trickery to allow strokes on field expr — **mandatory** parenthesis
- schema binding selection as output $(S.y)$ — **mandatory** parenthesis
- function application result as output — **mandatory** parenthesis

$$Test5 \mathrel{\widehat=} c\,?i\,!(f\ i) \longrightarrow \textbf{Skip}$$
$$Test6 \mathrel{\widehat=} c.(S.y)?z \longrightarrow \textbf{Skip}$$
$$Test7 \mathrel{\widehat=} c.(x?)!(y!)!(z?) \longrightarrow \textbf{Skip}$$

Description

Action	Communication pattern
$Test5$	$\mathrm{In}(i, \mathbb{N})$, $\mathrm{Out}(\mathbb{N} \times \mathbb{N})$
$Test6$	$\mathrm{Dot}(S.y)$, $\mathrm{In}(z, \mathbb{N} \times \mathbb{N})$
$Test4$	$\mathrm{Out}(x?)$, $\mathrm{Out}(y!)$, $\mathrm{Out}(z?)$

***** $S \in \mathbb{P}\,(\langle\!| y == \mathbb{N} |\!\rangle)$**, hence** $S.y \in \mathbb{N}$**.**
***** $x?, y!, z?$ **are decorated names; usually they appear in schemas.**

LaTeX

```
begin{circusaction}
 \t1 Test5 \circdef c~?i~!(f~i) \then \Skip
     \also
 \t1 Test6 \circdef c.(x.y)?z!w \then \Skip
     \also
 \t1 Test7 \circdef c.(x?)!(y!)!(z?) \then \Skip
end{circusaction}
```

4 Example 3 — complex input with restrictions

- input prefix restrictions — **mandatory** parenthesis
- prefix restrictions and complex expressions
- fields depending on previous value input
- chained expressions depending on previous input
- tuple selection within field restriction and output
- action broken across multiple lines

$$Test8 \mathrel{\widehat=} c?x : (x > 1)!(f\ x) \longrightarrow \mathbf{Skip}$$

$$Test9 \mathrel{\widehat=} c?x \longrightarrow c?z : (z > x.1).(f\ (x.2 + x.3)) \longrightarrow \mathbf{Skip}$$

Description

Action	Communication pattern
*Test*8	$\mathrm{In}(x, \{\, v : \mathbb{N} \mid v > 1 \,\})$, $\mathrm{Out}(\mathbb{N} \times \mathbb{N})$
*Test*9	$\mathrm{In}(x, \mathbb{N} \times \mathbb{N} \times \mathbb{N})$; $\mathrm{In}(z, \{\, v : \mathbb{N} \mid v > x.1 \,\})$, $\mathrm{Out}(\mathbb{N} \times \mathbb{N})$

*** type on inputs are restricted according to given predicate.**
*** *Test*9 input on z is from "$?z : (z > x.1)$".**

LaTeX

```
  begin{circusaction}
   \t1 Test8 \circdef c?x \prefixcolon (x > 1)!(f~x) \then \Skip
\also
   \t1 Test9 \circdef c?x \then \\
                 \t4 c?z \prefixcolon (z > x.1).(f~(x.2+x.3)) \then \Skip
  end{circusaction}
```

5 Example 4 — generic channels

- explicitly given generic actuals
- implicitly inferred generic actuals (?)

$$Test10 \mathrel{\widehat=} g[\mathbb{N} \times \mathbb{N} \times \mathbb{N}]?x!n1.n2 \longrightarrow \mathbf{Skip}$$
$$Test10 \mathrel{\widehat=} g.n1.(f\ n1) \longrightarrow \mathbf{Skip}$$

LaTeX

```
begin{circusaction}
 \t1 Test10 \circdef g[\nat \cross \nat \cross \nat]?x!n1.n2 \then \Skip
     \also
 \t1 Test10 \circdef g.n1.(f~n1) \then \Skip
end{circusaction}
```

6 (! Prolegomena) — basic process footer

It just terminates

- **Skip**

end