The package Tikz-MultInets

Sallaccio

June 20, 2017

Abstract

We describe here the use of the package Tikz-MultInets, an *extended version* of a package written by Marc De Falco to represent interaction nets. His package was meant to represent simple interaction nets, i.e. with one principal port and simple wires as well as boxes and exponential boxes for linear logic. We added the possibility of defining cells with multiple principal ports, cells with only one port at all as well as many different ways of defining new paths for wires.

1 Introduction

The package uses Tikz to create graphical representations of the most general case of interaction nets, i.e. interaction nets with *multiport cells* and *multiwires*.

The package is meant to be retro-compatible with v0.1 tikz interaction nets library by Marc de Falco which was aimed at representing interaction nets as intended by Laffont. All original commands are maintained and new commands were created in the same spirit, thus the sporadic use of french names for some of the objects.

2 Main extensions from original package

Evolutions from v0.1 tikz interaction nets library by Marc de Falco:

– apex angle of simple cells (from 60 to 100)

– added multiport cells and 1-port cells

– added multiwires

– added empty node with anchors above and above above for better building of wires

– added external auxiliary and principal ports to bypass cells

– added commands for : cellheight, size of principal port in multicells, port size, free wire size, bypass distance

3 Macro usage

3.1 Global rules

Some generic rules apply for all commands.

Arguments between square brackets – [⟨*options*⟩] – are optional and may or may not have a default value. They will be usually marked in blue. For instance, ⟨*direction*⟩ is always optional and has default value D, for *down*. This default value can be globally modified (see **??**).

The optional arguments in square brackets right after the command name are usually ⟨*tikz-options*⟩ that change or overwrite graphical options for the given entity, cell or wire. See the Tikz package for more details.

Arguments between braces – {⟨*options*⟩} – are mandatory. In general, the red colour will be used to emphasize the fact that a value is **mandatory**.

Sometimes, in some contexts, some arguments don't make much sense. Then, an otherwise mandatory option can be skipped and the default value is used. In that case, it will be shown in orange – {⟨*position*⟩}. This is mainly true of the position of a cell in the case it is the unique cell in the picture. Its value is then equivalent to {0,0}[1]. Note that if it is ommited, in order for the parser to understand the omission, the command needs to have an explicit [⟨*direction*⟩] or end with a semicolon ';'.

Finally, the elements' naming mechanism – normal parenthesis, (⟨*name*⟩) – follows Tikz' rules and is mandatory except if ⟨*label*⟩ is simple, in which case the cell is implemented with the name ⟨*label*⟩ (again, see Tikz package for details).

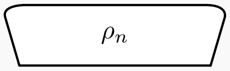
3.2 Cells

3.2.1 Defining cells

\inetmulticell [⟨*options*⟩](⟨*name*⟩){⟨*label*⟩}{⟨*coarity*⟩}{⟨*arity*⟩}{⟨*position*⟩}[⟨*direction*⟩]

⟨***options***⟩ can be a tikz-style, predefined style or any options for tikz lines.
⟨***name***⟩ is a name to reference the object when building wires and ports.
⟨***label***⟩ is the label of the cell. It appears inside.
⟨***coarity***⟩ is the number of *principal ports* of the cell.
⟨***arity***⟩ is the number of *auxiliary ports* of the cell.
⟨***position***⟩ is a position in tikz-style cartesian or polar coordinates.
⟨***direction***⟩ is either D (default),L,U,R or any angle where '0' is down.

```
\inetmulticell(rho){$\rho_n$}{5}{3}{0,0}[D]
```



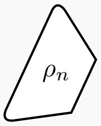
[1] To say the truth, in such a case any position value would give the same result since images are cropped.

The length of a cell depends on the value of ⟨*coarity*⟩: it is defined as ⟨*coarity*⟩×`\portsize`.

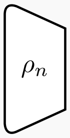
If ⟨*coarity*⟩=1, then the constructed multicell will be an `\inetcell` (see below). If ⟨*coarity*⟩=1 and ⟨*arity*⟩=0, then the constructed multicell will be an `\inetzerocell`.

A special writing is allowed for cases where positioning is not done by coordinates (e.g. in `matrix` environments). In such a case the ⟨*position*⟩ can be omitted (equivalent to `{0,0}` or `{0:0}`) but then the command should finish with a ';' or a ⟨*direction*⟩.

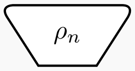
```
\inetmulticell(rho){$\rho_n$}{3}{3}[65]
```



```
\inetmulticell(rho){$\rho_n$}{3}{3}[R]
```



```
\inetmulticell(rho){$\rho_n$}{3}{3};
```



`\inetcell` `[`⟨***options***⟩`](`⟨***name***⟩`){`⟨***label***⟩`}{`⟨***position***⟩`}[`⟨***direction***⟩`]`

⟨***options***⟩ can be a tikz-style, predefined style or any options for tikz lines, or `arity=`*k* to create a cell with *k* auxiliary ports.

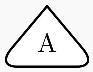
⟨***name***⟩ is a name to reference the object when building wires and ports.

⟨***label***⟩ is the label of the cell. It appears inside.

⟨***position***⟩ is a position in tikz-style Cartesian or polar coordinates.

⟨***direction***⟩ is either `D` (default)`,L,U,R` or any angle where '0' is down.

```
\inetcell(A){A}{0,0}[U];
```



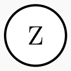
For retro-compatibility with De Falco's tikz-inet, the arity of a simple cell is defined as a tikz-option: `arity=4`. As said before, it is equivalent to define a simple cell of arity 4 or a multicell with coarity 1 and arity 4.

A special kind of cell is a cell with only principal ports (originally only the particular case of a cell with 1 principal port and 0 auxiliary ones was considered). Such a cell is represented by a circle.

`\inetzerocell` `[`⟨***options***⟩`](`⟨***name***⟩`){`⟨***label***⟩`}{`⟨***position***⟩`}`

⟨***options***⟩ can be a tikz-style, predefined style or any options for tikz lines.
⟨***name***⟩ is a name to reference the object when building wires and ports.
⟨***label***⟩ is the label of the cell. It appears inside.
⟨***position***⟩ is a position in tikz-style Cartesian or polar coordinates.

```
\inetzerocell(Z){Z}{0,0};
```

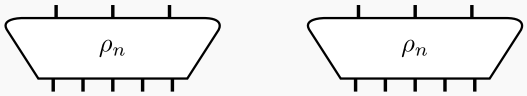


3.2.2 Then drawing its ports

Ports are not drawn by default, because they usually should not be *free* (disconnected) but rather connected through a (multi)wire to another port, and are thus drawn when the wire is created.

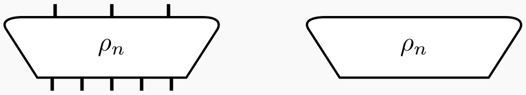
`[show ports]` It is nevertheless possible to tell the package to draw all ports of all cells in the tikzpicture:

```
\begin{tikzpicture}[show ports]
\inetmulticell(rho){$\rho_n$}{5}{3};
\inetmulticell(rho2){$\rho_n$}{5}{3}{4,0};
\end{tikzpicture}
```



or all ports of a given cell:

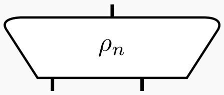
```
\begin{tikzpicture}
\inetmulticell[show ports](rho){$\rho_n$}{5}{3};
\inetmulticell(rho2){$\rho_n$}{5}{3}{4,0};
\end{tikzpicture}
```



`\inetport` [⟨***options***⟩](⟨***port name***⟩)

⟨***options***⟩ can be a tikz-style, predefined style or any options for tikz lines.
⟨***port name***⟩ is the complete name of a port.

```
\inetmulticell(rho){$\rho_n$}{5}{3};
\inetport(rho.pal 1)
\inetport(rho.pal 4)
\inetport(rho.pax 2)
```



3.2.3 Identifying ports

Ports need to be identified not only to draw them, but also to connect wires to them. Such identification can be done in various ways.

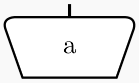
Following the naming of tikz-inet, the usual way of identifying a port of a multicell is by its type and numbering: third principal port or first auxiliary port for instance. As shown in the example above, this is done by attaching to the *name* of the cell a suffix `.pal 3` for the third principal port or `.pax 1` for the first auxiliary one.

Some ports nevertheless have special names.

Special names in multicells Multicells have few specially named ports.

`pax` A survivor from tikz-inet. That port is in the middle even if the cell has even arity.

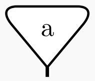
```
\inetmulticell{a}{5}{4};
\inetport(a.pax)
```



Special names in simple cells Laffont has shown that in interaction nets, all simple cells can be simulated by a small set of cells with arities 1 or 2. Such cells therefore have therefore been well treated, in particular for there special ports.

`pal` The principal port is unique, so its easy to identify (`pal 1` also works).

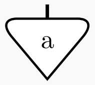
```
\inetcell{a};
\inetport(a.pal)
```



pax
middle pax An auxiliary port in the middle, for the case of arity 1. In the case of odd arity $2k + 1$, it's the same as `pax k+1`. In the case of even arity, it does not correspond to any numbered port: it's just a port in the middle.

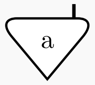
```
\inetcell{a};
\inetport(a.pax)

% \inetport(a.middle pax)
```



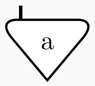
left pax The first auxiliary port (remember that ports are numbered counter-clockwise).

```
\inetcell{a};
\inetport(a.left pax)
```



right pax The last auxiliary port.

```
\inetcell{a};
\inetport(a.right pax)
```



The default simple cell has arity 4. Therefore, the default cell can be used for any arity < 4 as follows:

- arity $= 1 \rightarrow$ `pax` or `middle pax`
- arity $= 2 \rightarrow$ `left|right pax`
- arity $= 3 \rightarrow$ `left|middle|right pax`
- arity $= 4 \rightarrow$ `pax 1|2|3|4` and `|left|right`

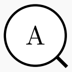
3.2.4 The case of zerocells

Zerocells were introduced to represent a cell with a unique (principal) port. To be clear, it was decided to represent them as circles as they do not have deiferentiated sets of ports. Pgf/tikz has an anchor mechanism in which circles have a natural set of positions around them that can be referred to using cardinal directions. We extended these anchors to define ports in the 8 positions of the 8-point compass: `north, south, west, east, north west, north east, south west, south east`.

We also provided some aliases for these points: the version without spaces and the initials. As a result, the possibilities are:

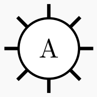
- `north, n`
- `south, s`
- `west, w`
- `east, e`
- `north west, northwest, nw`
- `north east, northeast, ne`
- `south west, southwest, sw`
- `south east, southeast, se`

```
\inetzerocell(A){A};
\inetport(A.south east);
```



Note that because of the use of anchors to define the position of the port, it is possible to have zerocells with up to 8 ports if necessary:

```
\inetzerocell(A){A}{0,0};
\inetport(A.n);
\inetport(A.ne);
\inetport(A.e);
\inetport(A.se);
\inetport(A.s);
\inetport(A.sw);
\inetport(A.w);
\inetport(A.nw);
```



3.2.5 Free ports

Ports can be *free* in the net and act as an interface of that net to the external world. In such a case, it can be useful to draw them a little longer in order to emphasize this fact.

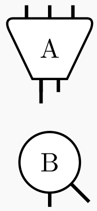
`\inetwirefree` [⟨***options***⟩](⟨***port name***⟩)

⟨***options***⟩ can be a tikz-style, predefined style or any options for tikz lines.

⟨***port name***⟩ is the complete name of a port.

```
\inetmulticell[show ports]{A}{2}{3}{0,0}
\inetwirefree(A.pal 1)

\inetzerocell{B}{0,-1.5};
\inetwirefree(B.south east)
\inetport(B.s)
```



The length of such free ports can be set globally (see section 4).

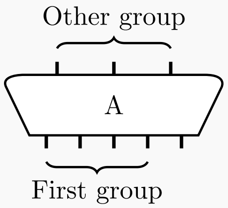
3.2.6 Grouping ports under braces

One can wish to annotate ports to comment on their value or use. The package provides a useful *curly brace* possibility to group nodes:

`\inetbrace` (⟨***first port***⟩)(⟨***last port***⟩){⟨***description***⟩}

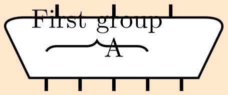
⟨***first port***⟩ is the complete name of a port.
⟨***last port***⟩ is the complete name of a port.
⟨***description***⟩ is the text over the brace.

```
\inetmulticell[show ports]{A}{5}{3};
\inetbrace(A.pal 1)(A.pal 4){First group}
\inetbrace(A.pax 1)(A.pax 3){Other group}
```



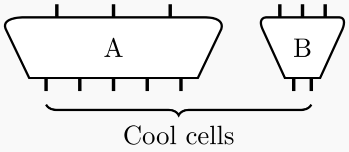
Be careful not to invert the order of the start and end ports:

```
\inetmulticell[show ports]{A}{5}{3};
\inetbrace(A.pal 4)(A.pal 1){First group}
```



This command can be hacked to annotate a group of cells instead, with rather random results. As a rule of thumb, it works of cells are well aligned.

```
\inetmulticell[show ports]{A}{5}{3};
\inetmulticell[show ports]{B}{2}{3}{2.5,0};
\inetbrace(A.pal 1)(B.pal 2){Cool cells}
```



3.3 Defining useful cells

This is a macro that enables to predefine a cell with particular label, coarity, arity and possibly direction.

`\inetmulticelltype` [⟨*options*⟩]{⟨*label*⟩}{⟨*coarity*⟩}{⟨*arity*⟩}

⟨***options***⟩ can be a tikz-style, predefined style or any options for tikz lines.
⟨***label***⟩ is the label of the cell. It appears inside.
⟨***coarity***⟩ is the number of *principal ports* of the cell.
⟨***arity***⟩ is the number of *auxiliary ports* of the cell.
⟨***direction***⟩ is either `D` (default),`L`,`U`,`R` or any angle where ‘0’ is down.

It is used inside a `\newcommand` as follows:

`\newcommand{\`⟨*specialcommand*⟩`}{\inetmulticelltype...}`

It is then possible to define such a predefined cell by just giving its name and its position:

`\`⟨***specialcommand***⟩`[`⟨*new options*⟩`](`⟨*name*⟩`){`⟨*position*⟩`}`

Notice that ⟨*new options*⟩ add up with options from the definition of the special command, and overwrite them in case of conflict.

For example:

```
\newcommand{\alphacell}{\inetmulticelltype[very thick]{$\alpha$}{3}{2}}
\alphacell[thin,minimum width=2ex](cellA){3,-1}
```

3.4 Wires

The simplest wire is between two ports:

`\inetwire` [⟨*options*⟩](⟨*port name*⟩)*<⟨*cross node*⟩>(⟨*port name*⟩)

⟨***options***⟩ can be a tikz-style, predefined style or any options for tikz lines.
⟨***cross node***⟩ is optional nodes the wire should cross (see Section)
⟨***port name***⟩ are complete names of ports.

Or between any two nodes:

`\inetwirecoords` [⟨*options*⟩](⟨*node name*⟩)(⟨*node name*⟩)

⟨***options***⟩ can be a tikz-style, predefined style or any options for tikz lines.

⟨***node name***⟩ are any node name.

For some uses, when cells are really close, it might come in handy to use more flexible wires.

\inetshortwire [⟨*options*⟩](⟨*port name*⟩)(⟨*port name*⟩)
⟨***options***⟩ can be a tikz-style, predefined style or any options for tikz lines.
⟨***port name***⟩ are complete names of ports.

\inetveryshortwire [⟨*options*⟩](⟨*port name*⟩)(⟨*port name*⟩)
⟨***options***⟩ can be a tikz-style, predefined style or any options for tikz lines.
⟨***port name***⟩ are complete names of ports.

Now the serious *connectors*:

\inetmultiwire [⟨*options*⟩](⟨*name*⟩){⟨*position*⟩}
<⟨*cross node*⟩>(⟨*port 1*⟩)
. . .
<⟨*cross node*⟩>(⟨*port n*⟩)[⟨*free port angle*⟩]
⟨***options***⟩ can be a tikz-style, predefined style or any options for tikz lines.
⟨***name***⟩ is a name for the wire. If not provided, no future use of it.
⟨***position***⟩ is a position in (cartesian or polar) coordinates for the wire.
⟨***cross node***⟩ is optional nodes the wire should cross (see Section)
⟨***port i***⟩ is a complete name of a port.
⟨***free port angle***⟩ give the angle for the freeport if provided.

At the center of a *multiwire*, there is a visible `\inetnode`:

\inetnode [⟨*option*⟩](⟨*name*⟩){⟨*position*⟩}
⟨***option***⟩ can be '`wire`' or any options for tikz circle nodes.
⟨***name***⟩ is a name for the wire.
⟨***position***⟩ is a position in (cartesian or polar) coordinates.

\inetloop [⟨*options*⟩]{⟨*position*⟩}
⟨***options***⟩ can be a tikz-style, predefined style or any options for tikz lines.
⟨***position***⟩ is a position for the center of the loop.

\inetwirearoundleft [⟨*options*⟩](⟨*pal*⟩)(⟨*pax*⟩)
⟨***pal***⟩ can be a tikz-style, predefined style or any options for tikz lines.
⟨***pal***⟩ is the complete name of a principal port of a cell.
⟨***pax***⟩ is the complete name of an auxiliary port of a (usually the same) cell.

\inetwirearoundright [⟨*options*⟩](⟨*pal*⟩)(⟨*pax*⟩)
⟨***options***⟩ can be a tikz-style, predefined style or any options for tikz lines.
⟨***pal***⟩ is the complete name of a principal port of a cell.
⟨***pax***⟩ is the complete name of an auxiliary port of a (usually the same) cell.

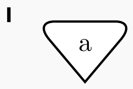
3.4.1 Positioning relative to cells

Simili-ports of multicells and simple cells

- leftext pax
- rightext pax
- leftext pal
- rightext pal

`rightext pax` A fake port after the last port of a cell.

```
\inetcell{a};
\inetport(a.rightext pax)
```



External anchors

- westwest
- easteast

3.5 Boxes

First a box around some nodes, then the promotion box of linear logic.

`\inetbox` [⟨*options*⟩]{⟨*fit*⟩}(⟨*name*⟩)
⟨***options***⟩ can be a tikz-style, predefined style or any options for tikz lines.
⟨***fit***⟩ is a space-separated list of cell names `(a) (b) ...`.
⟨***name***⟩ is a name for further reference of the box.

`\inetprombox` [⟨*options*⟩]{⟨*fit*⟩}(⟨*name*⟩)
⟨***options***⟩ can be a tikz-style, predefined style or any options for tikz lines.
⟨***fit***⟩ is a space-separated list of cell names `(a) (b) ...`.
⟨***name***⟩ is a name for the !-cell. The name of the box is `b`⟨*name*⟩.

3.6 Relative positionning

There are three main features of relative positionning.

Above a port

Add a coordinate, cartesian or polar

Crossings

4 Global values and defaults

`\inetsetportlength`	`\renewcommand{\abovedistance}{#1}`
	`\renewcommand{\aboveabovedistance}{(#1+.35cm)}`

`\inetsetportsize`	`\inetportsize`
`\inetsetcellheight`	`\inetcellheight`
`\inetsetabovedistance`	`\abovedistance`
`\inetsetaboveabovedistance`	`\aboveabovedistance`
`\inetsetbypassdistance`	`\bypassdistance`
`\inetsetwirefreelength`	`\inetwirefreelength`
`\inetsetportnamedistance`	`\inetportnamedistance`
`\inetsetbracedistance`	`\inetbracedistance`
`\inetsetbracetextdistance`	`\inetbracetextdistance`

5 Here's the code

`slurp` blabla

dadf sdf sdf

```
\begin{tikzpicture}
 bla

 blabla
\end{tikzpicture}
```

6 Creating wires

7 Known issues

Simple cells of arity 1 In the case of a simple cell with defined arity 1, the port `.pax 1` returns a *divide by 0* error. It is well defined if the multicell has coarity at least 2. This problem can arise in the case of automation of tikz-multinet creation. Otherwise:

Temporary solution: just use `.pax` instead.