

The package Tikz-MultInets

Sallaccio

June 30, 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.

Contents

1	Introduction	2
2	Main extensions from original package	2
3	Macro usage	2
3.1	Global rules	2
3.2	Positioning	3
3.3	Cells	4
3.3.1	Defining cells	4
3.3.2	Then drawing its ports	7
3.3.3	Identifying ports	8
3.3.4	The case of zerocells	10
3.3.5	Free ports	11
3.3.6	Grouping ports under braces	11
3.4	Defining useful cells	12
3.5	Wires	12
3.5.1	Fake ports of cells to better draw wires	14
3.6	Boxes	14
4	Relative positionning	14
5	Global values and defaults	15
6	Miscellaneous	15
6.1	Unspecific number of ports	15

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 invisible node with port-like anchors 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 [section 5](#)).

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. Some package specific options to the given object are also available through pgf/tikz key options.

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 – $\{\langle \textit{position} \rangle\}$. 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 omitted, in order for the parser to understand the omission, the command needs to have an explicit $[\langle \textit{direction} \rangle]$ or end with a semicolon ‘;’.

Finally, the elements’ naming mechanism – normal parenthesis, $(\langle \textit{name} \rangle)$ – follows Tikz’ rules and is mandatory except if $\langle \textit{label} \rangle$ is simple, in which case the cell is implemented with the name $\langle \textit{label} \rangle$ (again, see Tikz package for details).

3.2 Positioning

An important aspect of building nets is the positioning of cells and wires. The usual way of positioning objects in tikz is by coordinates – cartesian or polar – but many other ways are available (using tikzlibraries `positioning` and `calc` included in Tikz-MultInets).

There are options to do calculations on positions: take some known position and add a vector, again cartesian or polar:

```
\node[draw] (a) at (0,0) {a};
\node[draw] (b) at ($(a)+(2,0.5)$) {b};
```



Or you can calculate a position in between two given positions at a proportion of the path:

```
\node[draw] (a) at (0,0) {a};
\node[draw] (b) at (4,0) {b};
\node[draw] (c) at ($(a)!.3!(b)$) {c};
```



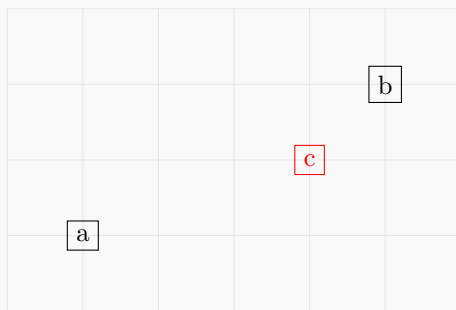
Those are composable

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

```

\node[draw] (a) at (0,0) {a};
\node[draw] (b) at (4,2) {b};
\node[draw, color=red] (c) at ($(a)!.5!(b)+(1,0)$) {c};

```

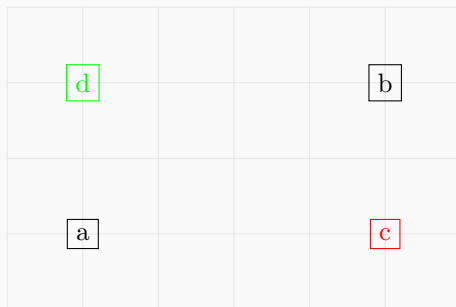


Another most interesting way is to use two elements and at the crossing of their positions: at the vertical of one of them and one the same line as the other.

```

\node[draw] (a) at (0,0) {a};
\node[draw] (b) at (4,2) {b};
\node[draw, color=red] (c) at (a -| b) {c};
\node[draw, color=green] (d) at (a |- b) {d};

```



Cells can be positioned using these techniques. Better, there positions can then be used for calculations: at the crossing of multicell *A* and multicell *B*. Actually, for more precision, even ports of cells can be used as positions.

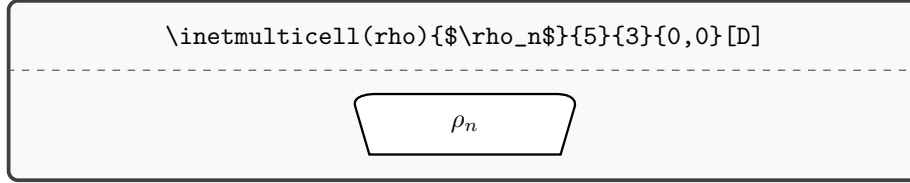
This package adds even more position capabilities connected to the specifics of cells. More on that in [section 4](#).

3.3 Cells

3.3.1 Defining cells

`\inetmulticell` [*options*] (*name*) {*label*} {*coarity*} {*arity*} {*position*} [*direction*]
options can be a tikz-style, predefined style or any options for tikz lines.

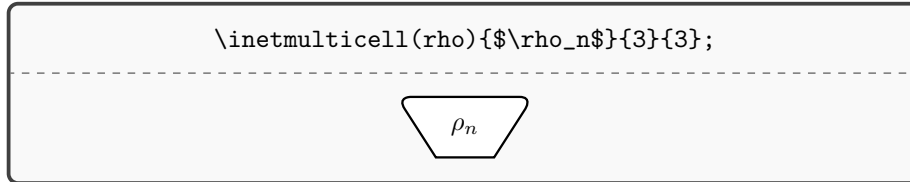
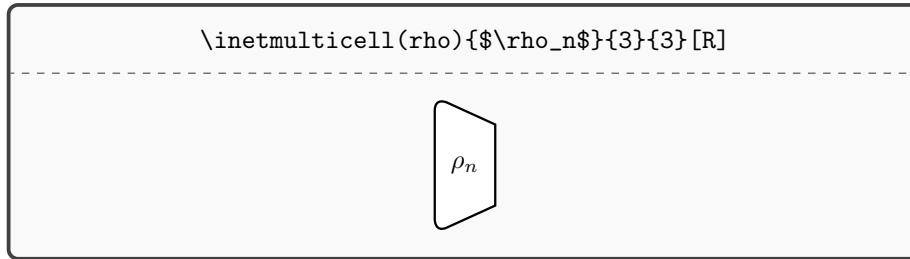
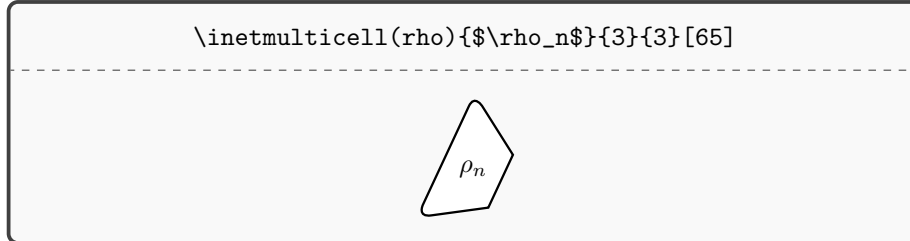
$\langle name \rangle$ is a name to reference the object when building wires and ports.
 $\langle label \rangle$ is the label of the cell. It appears inside.
 $\langle coarity \rangle$ is the number of *principal ports* of the cell.
 $\langle arity \rangle$ is the number of *auxiliary ports* of the cell.
 $\langle position \rangle$ is a position in tikz-style cartesian or polar coordinates.
 $\langle direction \rangle$ is either D (default), L, U, R or any angle where ‘0’ is down.



The length of a cell depends on the value of $\langle coarity \rangle$: it is defined as $\langle coarity \rangle \times \text{\texttt{\textbackslash portsize}}$.

If $\langle coarity \rangle=1$, then the constructed multicell will be an `\inetcell` (see below). If $\langle coarity \rangle=1$ and $\langle arity \rangle=0$, 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 $\langle position \rangle$ can be omitted (equivalent to `{0,0}` or `{0:0}`) but then the command should finish with a ‘;’ or a $\langle direction \rangle$.



`\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: defaults to `arity=4`. It is equivalent to define a simple cell of arity k or a multicell with coarity 1 and arity k . The package actually replaces the former by the latter.

```
\inetcell[arity=5]{A}{0,0};  
\inetmulticell(B){B}{1}{5}{0,-1.5};
```



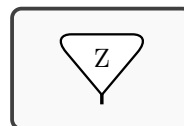
A special kind of cell is a cell with only principal ports (originally only the particular case of a cell with 1 principal port and 0 auxiliary ones was considered). Such a cell is represented by a circle.

`\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};
```



Of course, the `zerocell` is a convention. A cell with only one principal port could just be as well a triangle with no ports in the back.



3.3.2 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(alpha){$\alpha$}{5}{3};
\inetmulticell(beta){$\beta$}{3}{4}{3,0};
\end{tikzpicture}
```



or all ports of a given cell:

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



Nevertheless, the “normal” way of doing is to draw only ports that are not connected to any wire, since wires in some way contain the ports they connect. The default name of a port is of the form

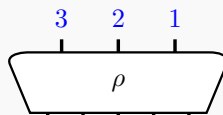
$$\langle cell\ name \rangle . \langle type\ of\ port \rangle \ \langle number \rangle$$

with a space before the number and where $\langle type\ of\ port \rangle$ is

- **pal** for *principal* ports
- **pax** for *auxiliary* ports

Ports are numbered in counter-clockwise order, separately for the two kinds:

auxiliary ports:



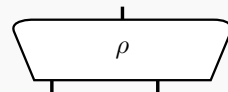
principal ports:

1 2 3 4 5

To draw a given port, use the following command

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

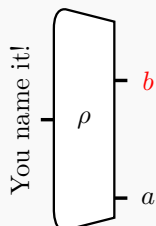


It is often the case that one needs to give and show a name of a port.

`\inetportnamed` [*options*] (*port name*) {*port label*} [*label options*]
options can be a tikz-style, predefined style or any options for tikz lines.
port name is the identification name of a port.
port label is the name you wish to show for this port.
label options are tikz-options for the label.

The label is in its own tikz-node, so all options that apply to nodes are acceptable in the *label options*.

```
\inetmulticell(rho){$\rho$}{5}{3}[R];
\inetportnamed(rho.pal 1){$a$}
\inetportnamed(rho.pal 4){$b$}[color=red]
\inetportnamed(rho.pax 2){You name it!}[rotate=90]
```



3.3.3 Identifying ports

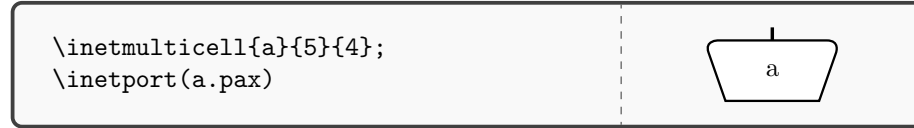
Ports need to be identified not only to draw them, but also to connect wires to them. Such identification can be done in various ways.

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 examples 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 the tikz-inet package: that port is in the middle even if the cell has even arity.



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).



pax An auxiliary port in the middle, for the case of arity 1. In the case of odd
middle pax 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.



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



right pax The last auxiliary port.



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` → `left|middle|right pax`
- `arity = 4` → `pax 1|2|3|4` and `|left|right`

3.3.4 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 differentiated 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);
```



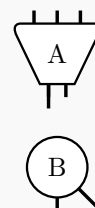
3.3.5 Free ports

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 5](#)).

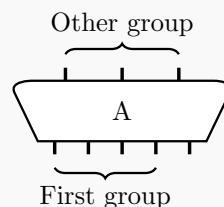
We will see later that such free ports can also be attached to other objects, like wires and invisible nodes. The definition stays the same since these objects also define a set of ports.

3.3.6 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:

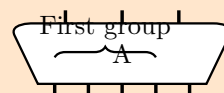
`\inetbrace` (*first port*) (*last port*) {*description*}
first port is the complete name of a port.
last port is the complete name of a port.
description is the text over the brace.

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



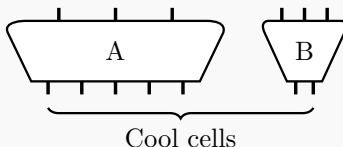
Be careful not to invert the order of the start and end ports:

```
\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 if 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}
```



3.4 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}
```

3.5 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>`)
`<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.

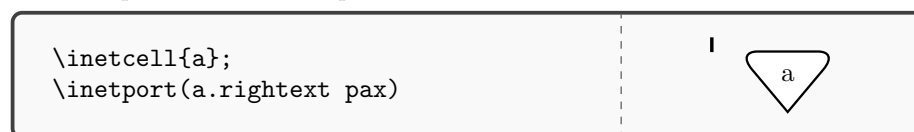
`\inetwirearoundright` [`<options>`] (`<pal>`) (`<pax>`)
`<options>` can be a tikz-style, predefined style or any options for tikz lines.
`<pal>` is the complete name of a principal port of a cell.
`<pax>` is the complete name of an auxiliary port of a (usually the same) cell.

3.5.1 Fake ports of cells to better draw wires

Simili-ports of multicells and simple cells

- lefttext pax
- righttext pax
- lefttext pal
- righttext pal

righttext pax A fake port after the last port of a cell.



External anchors

- westwest
- easteast

3.6 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*.

4 Relative positionning

There are three main features of relative positionning.

Above a port

Add a coordinate, cartesian or polar

Crossings

5 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

```

6 Miscellaneous

6.1 Unspecific number of ports

Use three dots to show repetitive ports, behaviors or connections:

```

\inetdotsbetween [<direction>] (<position>) (<position>)

```

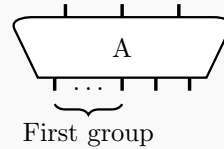
<direction> can be h or v.

<position> is any positioning tikz can understand, including cells, ports or any anchor defined by them, calculations on positions.

```

\inetmulticell{A}{5}{3};
\inetport(A.pax 1)
\inetport(A.pax 2)
\inetport(A.pax 3)
\inetport(A.pal 1)
\inetport(A.pal 3)
\inetport(A.pal 4)
\inetport(A.pal 5)
\inetdotsbetween(A.above pal 1)(A.above pal 3)
\inetbrace(A.pal 1)(A.pal 3){First group}

```



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.