

The package Tikz-MultInets

Sallaccio

June 14, 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**.

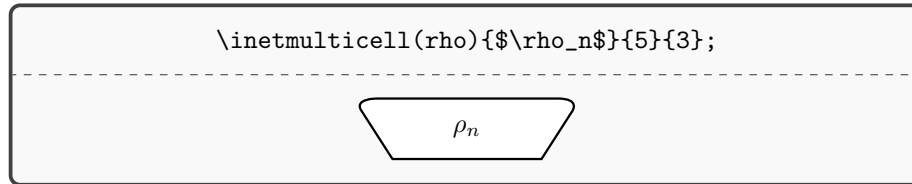
Sometimes, in some contexts, some arguments don't make much sense. Then, an otherwise mandatory option can be skipped and the default value is used. In that case, it will be shown in orange – {*position*}. This is mainly true of the position of a cell in the case it is the unique cell in the picture. Its value is then equivalent to {0,0}¹. Note that if it is omitted, 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.



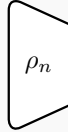
¹To say the truth, in such a case any position value would give the same result since images are cropped.

The length of a cell depends on the value of $\langle coarity \rangle$: 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 $\langle direction \rangle$.

```
\inetmulticell(rho){$\rho_n$}{3}{3}{0,0}[65];
```



```
\inetmulticell(rho){$\rho_n$}{3}{3}{0,0}[R];
```



```
\inetcell [ $\langle options \rangle$ ] ( $\langle name \rangle$ ) { $\langle label \rangle$ } { $\langle position \rangle$ } [ $\langle direction \rangle$ ]
```

$\langle options \rangle$ can be a tikz-style, predefined style or any options for tikz lines, or $arity=k$ to create a cell with k auxiliary ports.

$\langle name \rangle$ is a name to reference the object when building wires and ports.

$\langle label \rangle$ is the label of the cell. It appears inside.

$\langle position \rangle$ 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.

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



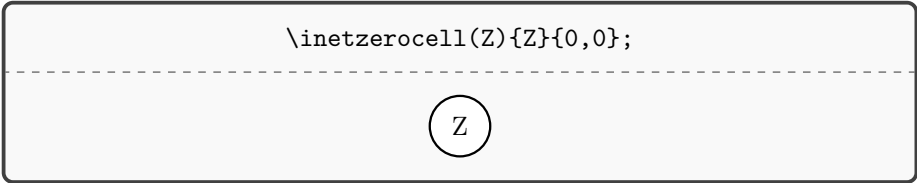
```
\inetzerocell [ $\langle options \rangle$ ] ( $\langle name \rangle$ ) { $\langle label \rangle$ } { $\langle position \rangle$ }
```

$\langle options \rangle$ 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.

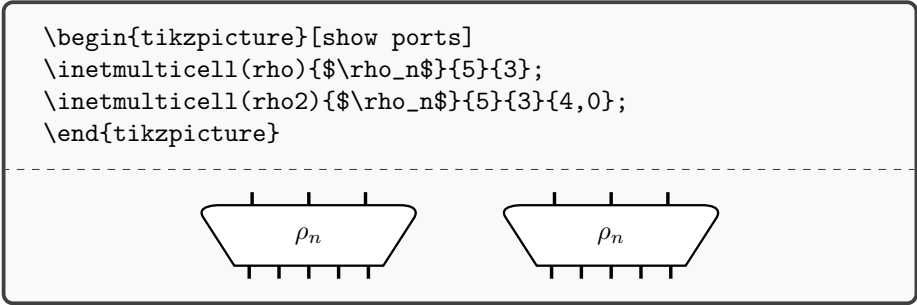
$\langle position \rangle$ is a position in tikz-style cartesian or polar coordinates.



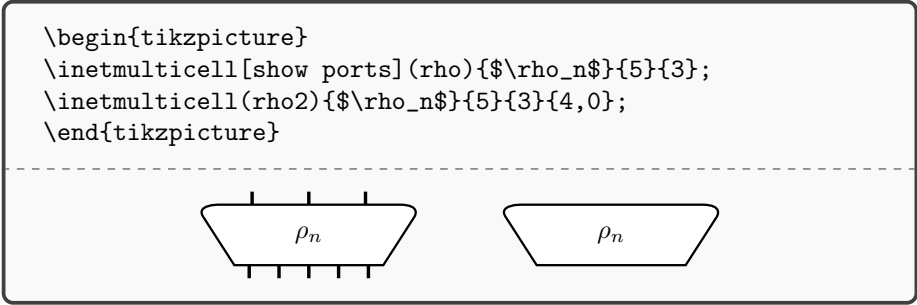
3.2.2 Then drawing its ports

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.

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



or all ports of a given cell:

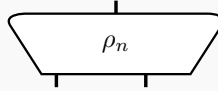


`\inetport` [*⟨options⟩*] (*⟨port name⟩*)
⟨options⟩ can be a tikz-style, predefined style or any options for tikz lines.
⟨port name⟩ is the complete name of a port.

```

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

```



3.2.3 A special port

Ports can be free in the net and act as an interface of that net to the external world. Since they are not necessarily related to a cell, a special command allows to create a free port at any given position.

```

\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.

3.2.4 Identifying the ports

The usual name of a port is its numbering as a principal or auxiliary port. Some ports have also other aliases: left, center, etc.

```

\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.

3.3 Defining usefull 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>`)

`<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 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

`\portsize`

5 Here's the code

```
slurp  blabla
      dadf sdf sdf
      \begin{tikzpicture}
      bla

      blabla
      \end{tikzpicture}
```

6 Creating wires

```
\begin{plouf}
Wasabi!
\end{plouf}
```