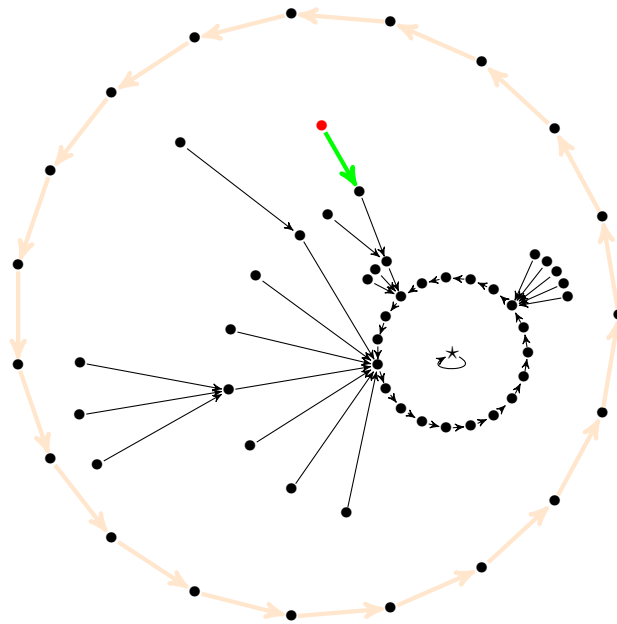


OODGraph*

Enrico Formenti
Université Côte d'Azur, CNRS, I3S, France
enrico.formenti@univ-cotedazur.fr

March 29, 2022



Abstract

This package introduces a new drawing environment which allows to draw directed graphs in which all nodes have out-degree 1 in *TikZ*.

Contents

| | |
|--|---|
| 1 Introduction | 2 |
| 2 Installing the library and starting using it | 3 |

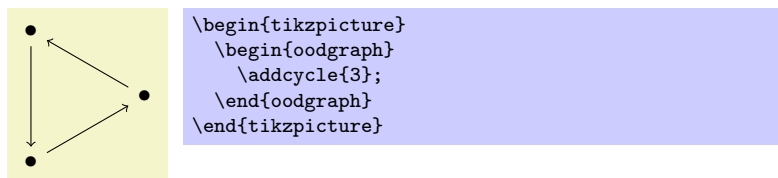
*This package has version number *v0.50* of July 15th, 2019; it is released under and subject to the [L^AT_EX Project Public License \(LPPL\)](#).

| | |
|---|----|
| 3 Commands reference guide | 4 |
| 4 How things are rendered | 5 |
| 5 Stacking the beards | 6 |
| 6 When beards have no attached cycles | 7 |
| 7 Rotating beards | 8 |
| 8 Adding some colors | 10 |
| 9 Changing the style of things | 11 |
| 10 Caution! | 11 |
| 11 More art | 12 |
| 12 Versions and what's new | 12 |
| Index | 15 |

1 Introduction

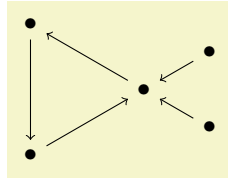
This package provides a set of primitives for drawing graphs in which all nodes have out-degree 1. These basically are graphs of functions with finite domains. They are formed by a finite number of connected components and each component is made of a cycle and of some *beards* attached to the nodes of the cycle (see the picture under the title). A *beard* is a connected directed graph in which all nodes have out-degree 1 and has exactly one node with out-degree 0, *beard tip*. Let us start illustrating the package by some examples.

A very basic example. The most basic example of OOD graph is made by a single cycle of a given length (or *period*) as follows:



Lines 1 and 3 declare a new OOD graph. This basically is a wrap up for a `scope` environment. Then, Line 2 adds to the current drawing a cycle of size 3.

A less basic example. A more sophisticated example consists in adding some ingoing edges to nodes in the cycle:



```
\begin{tikzpicture}
\begin{oodgraph}
\addcycle{3};
\addbeard{2};
\end{oodgraph}
\end{tikzpicture}
```

Stepping to a full example . The following code produces the image at the very beginning of this document.

```
\begin{tikzpicture}[every node/.style={inner sep=0,outer sep=0},>=stealth']
\begin{oodgraph}
\addcycle[nodes prefix=a,radius=4,edges style={red!50!yellow!20,ultra
thick,->}] {19};
\addcycle[xshift=1.8cm,yshift=-.5cm] {19};
\addbeard[attach node=a->11, opening angle=160, radius=2] {7};
\addbeard[attach node=a->11->1,radius=2,opening angle=60] {1};
\addbeard[attach node=a->11->4,radius=2,opening angle=60] {3};
\addbeard[attach node=a->3,radius=.75,opening angle=70] {5};
\addbeard[attach node=a->8,opening angle=60,radius=.5] {3};
\addbeard[attach node=a->8->1,opening angle=60] {2};
\addbeard[attach node=a->8->1->1,opening angle=30,nodes color=red,edges
color=green,ultra thick] {1};
\addcycle[xshift=1.8cm,yshift=-.5cm, nodes content=$\star$] {1};
\end{oodgraph}
\end{tikzpicture}
```

Since this last example is much more complex to explain, it is time to illustrate the different commands provided by the library and their options.

2 Installing the library and starting using it

The library currently consists in a single file, namely, `tikzlibraryoodgraph.tex` and shall be installed in a directory which is accessible by the lookup engine of your latex installation.

Requirements The only dependence is the *TikZ* library `calc` which comes with the standard installation.

Starting using the library. You just need to issue in the \LaTeX preamble a `\usepackage{tikz}` to load *TikZ* and `\usetikzlibrary{oodgraph}` for our library. Any OOD graph should be defined in a dedicated environment

```
\begin{oodgraph} [options]
environment contents
\end{oodgraph}
```

inside a `tikzpicture` environment:

```
\begin{tikzpicture}[\langle options \rangle]
  \langle environment contents \rangle
\end{tikzpicture}
```

available options for the environment `oodgraph` are the same as for a standard `scope` environment. However, please have a look at Section 10.

3 Commands reference guide

As we have already seen in the examples in the introduction, the basic component of an OOD graph is a cycle which is issued by the following command:

```
\addcycle{\langle num \rangle}
```

Builds a cycle of size $\langle num \rangle$. Remark that all nodes are drawn on a circle.

which has the following options:

```
/oodgraph/nodes content=\langle code \rangle (default $\bullet$)
```

Sets the nodes content to $\langle code \rangle$ (this is standard $\text{T}_{\text{E}}\text{X}$ code).

```
/oodgraph/radius=\langle value \rangle (default 1)
```

Option for setting the radius of the circle on which are drawn the nodes of the cycle.

```
/oodgraph/nodes prefix=\langle string \rangle (default a)
```

Option for setting the prefix of all nodes names of the circle. This can be anything like in the case of the name of `TikZ` nodes.

The reference name of a given node in a cycle is made using the `nodes prefix`, say `pref` and adding `->num` where `num` is the distance (in terms of edges) plus 1 from the initial node to the given node. The initial node has always name `pref->1`.

```
\addorigin{\langle string \rangle}
```

Declares a valid attachment point for beards. $\langle name \rangle$ must be the name of a `TikZ` point already declared. This command has no options.

```
\addbeard{\langle num \rangle}
```

Builds a beard of $\langle num \rangle$ nodes with tip node coinciding with the current reference node.

It has the following options:

```
/oodgraph/attach node=\langle string \rangle (default a->1)
```

Sets the name of node to which the beard tip will be attached to.

`/oodgraph/node content=<code>` (default `\bullet`)

Sets the node content to `<code>` (this is standard T_EXcode).

`/oodgraph/radius=<value>` (default 1)

Sets the length of the edges between the nodes of the beard and the tip.

`/oodgraph/opening angle=<value>` (default 120)

Sets the angle of the arc on which the nodes of the beard will be drawn. Remark that this angle is centered w.r.t. the angle made by the beard tip, the center of the cycle and the positive abscissas axis.

`/oodgraph/rotation angle=<value>` (default 0)

Sets the rotation angle to `<value>`. If `<value>` is positive then points are rotated counterclockwise, otherwise they are rotated clockwise.

4 How things are rendered

As we have already said the `\addcycle` command draws a cycle with nodes arranged on a circle of the given radius R . The angle θ between two successive nodes is given by the formula $360/n$ where n is the size of the cycle. For example, the following code

```
\begin{tikzpicture}
  \begin{oodgraph}
    \addcycle{8};
  \end{oodgraph}
\end{tikzpicture}
```

produces the cycle in Figure 1 (except for the decorations of course). Since there is no other indication, default names are given to the nodes. As explained earlier the reference node is the one making an angle of 0 degrees w.r.t. the origin and the abscissas axis. The rendering engine for beards runs as follows. In order to better understand, let us make reference to Figure 2 and assume that the beard has an odd number of nodes (the beard tip is not counted here) bigger than 1. First of all, the coordinates of the reference point (in green in the figure) are computed. The reference point is middle point of the beard. Its coordinates are set such that it will have the same angle as the beard tip w.r.t. the origin point of the coordinate system and the abscissas axis. Then, the angle θ is computed from the opening angle ω using the formula $\theta = \omega/n$, where n is the number of the nodes of the beard. Therefore, the angle α between the first and the last node of the beard is $\alpha = \theta \cdot (n - 1) = \omega \cdot \frac{n-1}{n}$. Nodes are drawn starting from the one with the lowest angle w.r.t. the reference point up to the largest and named accordingly.

When the number of nodes in a beard is even, there is a small complication since one has to take into account the offset angle $\beta = \theta/2$ (see Figure 3) when drawing the nodes and the reference point (in green in the figure) is no more part of the beard nodes.

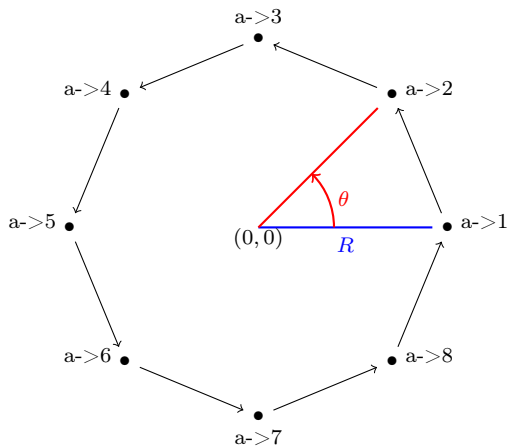
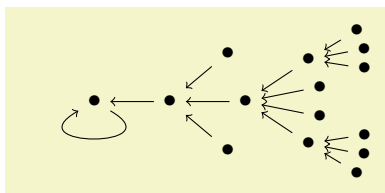


Figure 1: Drawing a cycle. Remark the default names given to the nodes.

Finally, if the beard has just a single node, then the opening angle plays no role and the reference point coincides with the beard point. The reference point will be drawn in the prolongation of the line passing through the origin of the axis and the beard tip.

5 Stacking the beards

In order to build more complex OOD graphs, it is necessary to be able to stack beards one over the other. This can be easily performed using the `attach node` option like in the following example:



```

\begin{tikzpicture}
\begin{oodgraph}
\addcycle[nodes prefix=mybeard]{1};
\addbeard[attach node=mybeard->1]{1};
\addbeard[attach node=mybeard->1->1]{3};
\addbeard[attach node=mybeard->1->1->2,opening angle=90]{4};
\addbeard[attach node=mybeard->1->1->2->1,opening angle=60,radius=.75]{3};
\addbeard[attach node=mybeard->1->1->2->4,opening angle=60,radius=.75]{3};
\end{oodgraph}
\end{tikzpicture}

```

The idea is that each node has a name which is uniquely determined. For

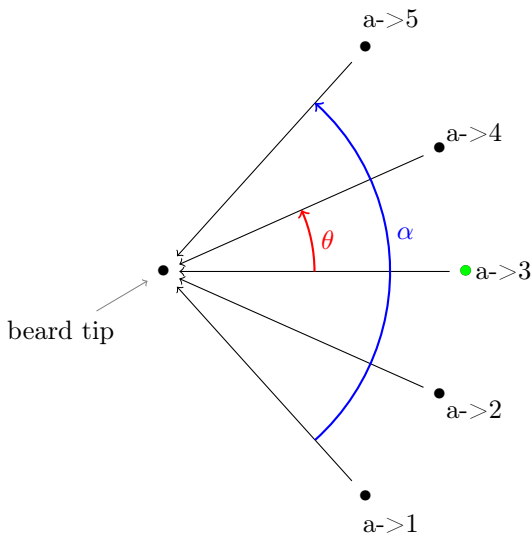


Figure 2: Rendering a beard with an odd number of nodes (not counting the beard tip). The reference point is colored in green. Remark also the default name of the nodes.

instance, in the previous example, `mybeard->1->1->2->1` means that (reading from right to left) we are looking for the node that is built on the first node of the beard built on the second node of the preceding beard which is built...

6 When beards have no attached cycles

Sometimes it is useful to draw beards which are not attached to a cycle. This can be done by using the instruction `\addorigin` before the instructions defining the beard. The purpose of the `\addorigin` instruction is to declare a possible attachment point for a beard tip. The argument of the instruction is the name of a previously declared PGF point. Below, we report the final example of the previous section with the cycle removed.

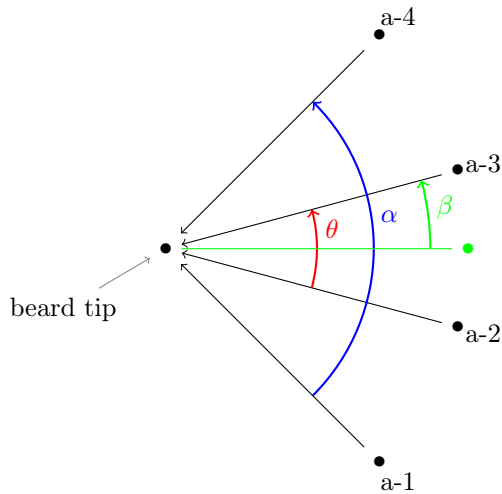
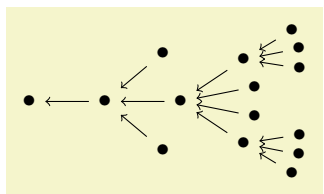


Figure 3: Rendering a beard with an even number of nodes (not counting the beard tip). The reference point is colored in green and it is now a virtual point. Remark also the default name of the nodes.



```

\begin{tikzpicture}
\begin{oodgraph}
\node(origin) at (0,0){\bullet$};
\addorigin{origin};
\addbeard[attach node=origin]{1};
\addbeard[attach node=origin->1]{3};
\addbeard[attach node=origin->1->2,opening
angle=90]{4};
\addbeard[attach node=origin->1->2->1,opening
angle=60,radius=.75]{3};
\addbeard[attach node=origin->1->2->4,opening
angle=60,radius=.75]{3};
\end{oodgraph}
\end{tikzpicture>

```

Remark the difference in the references to the attach node between the previous example and the one in the previous section. This is due to the difference between how the nodes are encoded in our library (previous section example) and in PGF (current example).

7 Rotating beards

Sometimes it is necessary to rotate the beards points in order to obtain better diagrams and, for example, to avoid intersections between two neighboring beards. This can be obtained by specifying the key `rotation angle= $\langle value \rangle$` which rotates the nodes of a beard counterclockwise if $\langle value \rangle$ is positive, clockwise otherwise.

The rotation angle θ is the angle between the abscissas axis, the beard tip and the reference point. See the Figure 4 for an illustration. Here is the code (without the decorations) for the diagram in Figure 4.

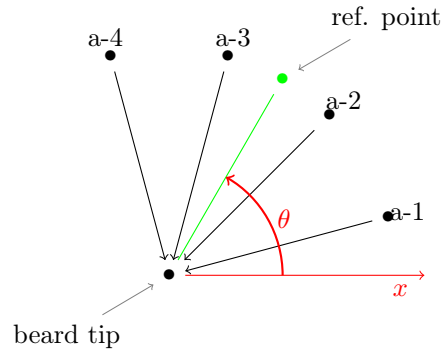


Figure 4: Illustration of how the rotation angle is computed.

```
\begin{tikzpicture}
\begin{oodgraph}
\node(origin) at (0,0){\bullet};
\addorigin{origin};
\addbeard[attach node=origin,radius=4,rotation angle=60]{4};
\end{oodgraph}
\end{tikzpicture}
```

By default the rotation angle is set to 0 (and hence the reference point is on the abscissas axis). Remark that when a beard is stacked onto another, the rotation angle is calculated w.r.t. the lowest beard tip, the current reference point and the abscissas axis. Therefore, applications of this key on consecutive paths has the effect of cumulating the rotation angle on the current beard level. See Figure 5 for an illustration of this effect. Here is the source code for the figure (without decorations as usual):

```
\begin{tikzpicture}
\begin{oodgraph}
\node[pin={pin edge={<-}225:beard tip}](origin) at (0,0){\bullet};
\addorigin{origin};
\addbeard[attach node=origin,radius=4,rotation angle=60]{5};
\addbeard[attach node=origin->3,rotation angle=30]{3};
\end{oodgraph}
\end{tikzpicture}
```

We end this section with an illustration of how the rotation angles can be exploited to produce better looking diagrams.

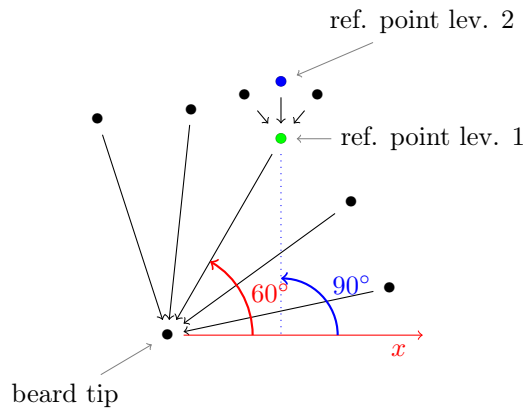
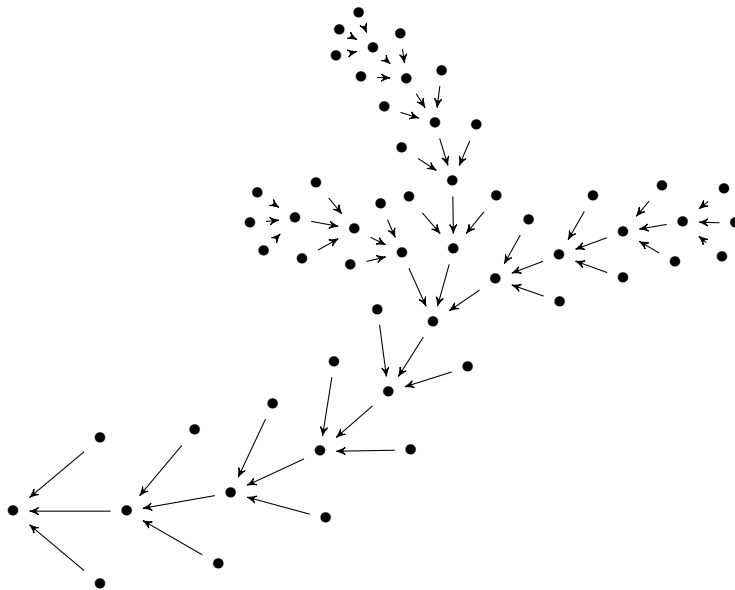


Figure 5: Effect of consecutive modifications of the rotation angle. The first beard is rotated of 60° and the second of 30° . However, the second beard is finally rotated of 90° since it cumulates with the previous rotation angle. The first reference point is colored in green, the second in blue.



8 Adding some colors

In order to have nice looking pictures some color can be added to OOD graphs drawing using the following options:

`/oodgraph/nodes color=<colorname>` (default `black`)

Nodes content will be colored with color `<colorname>`. This has to be a valid `TikZ` color.

`/oodgraph/edges color=<colorname>` (default `black`)

Arrows will be colored with color `<colorname>`. This has to be a valid `TikZ` color.

9 Changing the style of things

Several possibilities are offered if one needs to change the overall drawing aspect of a graph. The library distinguishes between options for the edges and options for the nodes.

`/oodgraph/nodes style={<options>}` (no default)

This option sets the style for all the nodes in the current `\addcycle` or `\addbeard` command. The options that can be used are the standard `TikZ` options for nodes (see the `TikZ` manual for more about them).

`/oodgraph/edges style={<options>}` (default `->`)

This option sets the style for all the edges issued by the current `\addcycle` or `\addbeard` command. The options that can be used are the standard `TikZ` options for edges (see the `TikZ` manual for more about them).

`/oodgraph/loop style={<options>}` (default `out=-30, in=-150, looseness=6`)

This option sets the style for the loop issued by the current `\addcycle` command. The options that can be used are the standard `TikZ` options for loops (see the `TikZ` manual for more about them).

There are also some quick keys to set edges thickness like in standard `TikZ`:

`/oodgraph/semi thick` (no value)

`/oodgraph/thick` (no value)

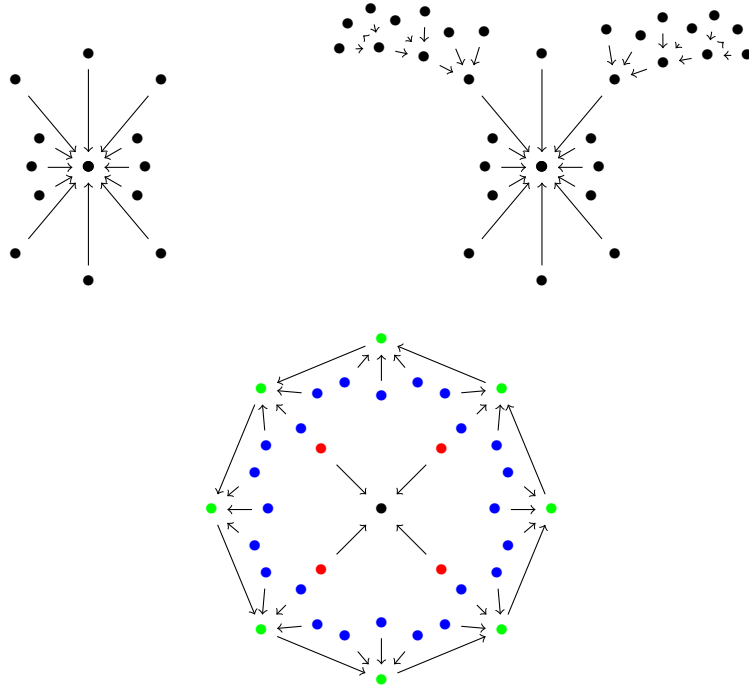
`/oodgraph/ultra thick` (no value)

10 Caution!

Only the options mentioned in the previous sections are granted to work. All other options may have unpredictable or unexpected behavior. Use them at your own risk.

11 More art

In order to better understand how things work, the reader is invited to try to understand how the following pictures have been realized. We provide them without source code to force the reader to find the solution by himself.



Acknowledgements

This library has borrowed some code from `etoolbox` package in the purpose of making it more standalone. Warm thanks hence to Philipp Lehman, Joseph Wright for having shared their very helpful package.

12 Versions and what's new

0.1 (2019/05/25)

- first public release

0.2 (2019/06/01)

- added `/.store` in key
- added `/.default` key

- cleaned up code

0.3 (2019/06/14)

- fixed drawing algorithm
- points definitions are now saved in a list
- the points list is emptied at the closing of the environment `oodgraph`

0.4 (2019/06/18)

- now commands are completely scoped
- redesigned the graph drawing algorithm to take into account that now everything is scoped

0.45 (2019/06/19)

- added the option 'rotation angle'

0.46 (2019/06/20)

- added the option 'node content'
- added bridge to tikz and pgf when a key is not known
- made the library independent of 'etoolbox' package (borrow some code from the package)
- added `oodgraph@` prefix to all commands so to avoid conflicts with other packages/libraries

0.47 (2019/06/28)

- options 'oodgraph arrow' and 'oodgraph node' renamed into 'oodgraph edges' and 'oodgraph nodes' for more consistency
- fixed a bug with edges style
- now '`->`' is the default edge style

0.48 (2019/07/14)

- added a check in `\addorigin` to verify if the point has been defined before
- some code polishing (eliminated the list `oodgraph@more@options` which is no more used, deleted `\oodgraph@resetkeys` function and integrated its code directly into the calling functions since its body contained only an `\end{scope}` command, deleted several unuseful `\expandafter` commands)
- the definition of `\oodgraph@csxdef` has been made explicit
- partially rewritten the drawing routine for beards. Now it is much faster

0.50 (2019/07/15)

- renamed style options `/oodgraph/oodgraph nodes` into the shorter and nicer `/oodgraph/nodes style`
- renamed style options `/oodgraph/oodgraph edges` into the shorter and nicer `/oodgraph/edges style`
- introduced the option `/oodgraph/loop style` which allows to style the loops
- bug fix: in the special case of cycles of order 2, the two edges were drawn one over the other (obtaining a double arrow). Thanks to François Doré for signaling this
- bug fix: the beards to be wrongly oriented when the `xshift` key was used in the `addecycle` command. Bug discovery and fix by François Doré
- bug fix: in the special case of cycles of order 2, the address of the nodes were inverted. Thanks to François Doré for signalling this

0.50.01 (2020/08/19)

- Fixed some typos in the manual.

Index

- `\addbeard`, 4
- `\addcycle`, 4
- `\addorigin`, 4
- attach node key, 4

- edges color key, 11
- edges style key, 11
- Environments
 - `oodgraph`, 3
 - `tikzpicture`, 4

- loop style key, 11

- node content key, 5
- nodes color key, 11
- nodes content key, 4
- nodes prefix key, 4
- nodes style key, 11

- `/oodgraph/`
 - attach node, 4
 - edges color, 11
 - edges style, 11
 - loop style, 11
 - node content, 5
 - nodes color, 11
 - nodes content, 4
 - nodes prefix, 4
 - nodes style, 11
 - opening angle, 5
 - radius, 4, 5
 - rotation angle, 5
 - semi thick, 11
 - thick, 11
 - ultra thick, 11
- `oodgraph` environment, 3
- opening angle key, 5

- radius key, 4, 5
- rotation angle key, 5

- semi thick key, 11

- thick key, 11
- `tikzpicture` environment, 4

- ultra thick key, 11