

1. What is this?

This is a `SATySFI` package for drawing a commutative diagram and its kind. An example is as follows.

$$\begin{array}{ccc} A & \xrightarrow{f} & B \\ \downarrow g & & \downarrow g' \\ C & \xrightarrow{f'} & D. \end{array}$$

The syntax is inspired by a `LATEX` package ‘`tikz-cd`’.

2. Usage

First of all, load the package by `@require: matrixcd`.

The following code produced an example in the above.

```
\eqn(${\nMatrixCD.matrixcd{\n  | A\n    \MatrixCD.arrow![MatrixCD.to `r`;MatrixCD.label ${f}]\n    \MatrixCD.arrow![MatrixCD.to `d`;MatrixCD.label ${g}]\n  | B\MatrixCD.arrow![MatrixCD.to `d`;MatrixCD.label ${g'}]\n    \MatrixCD.cr\n  | C\MatrixCD.arrow![MatrixCD.to `r`;MatrixCD.label ${f'}]\n  | D.\n|}\n});
```

Perhaps some of you think that many `MatrixCD` is annoying. You can omit this by adding open `MatrixCD` in the preamble. The simplified code is as follows.

```
\eqn(${\n  \matrixcd{\n    | A\arrow![to `r`;label ${f}]\arrow![to `d`;label ${g}]\n    | B\arrow![to `d`;label ${g'}]\cr\n  }\n});
```

```

    | C\arrow![to `r`;label ${f'}] | D.
  |}
});

```

In the following, only this simplified codes are given. (Namely open `MatrixCD` is always assumed.) In the rest, we also omit `\eqn`.

The basic command is `\matrixcd`. The objects are placed as a ‘table’ and they are separated by `|`. The command `\cr` place the next object at the next line. Here is an example. (Do not forget the last `|`.)

```

\matrixcd{
  | A | B\cr
  | C | D
|}

```

It gives the following.

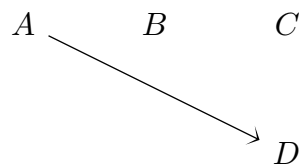
$$\begin{array}{cc}
 A & B \\
 C & D
 \end{array}$$

To draw an arrow, one can use `\arrow` command. The syntax is `\arrow![option list]`. The option list is a list of options separated with `;`. The most important option is `to` which is used to specify the target of the arrow. This can be used like `to `rrd``;. The argument is a sequence of characters `l`, `r`, `u`, `d` which stand for left, right, up and down, respectively. Here is an example of the code and the output.

```

\matrixcd{
  | A\arrow![to `rrd`] | B | C\cr
  | | D
|}

```



One can add a label to each arrow by using `label` option.

```
\matrixcd{
  | A\arrow!{to `r`; label ${f}} | B
|}
```

$$A \xrightarrow{f} B$$

One can add an option to a label by `label ?:[option list] ${<label>}`. For example, by the option `swap`, you can swap the position of the label.

```
\matrixcd{
  | A\arrow!{to `r`; label ?:[swap] ${f}} | B
|}
```

$$A \xrightarrow[f]{} B$$

3. Options for `\arrow`

```
\matrixcd{
  | A\arrow!{to `r`;twoheadrightarrow}
  | B\arrow!{to `r`;dotted}
  | C\arrow!{to `r`;dashed}
  | D\arrow!{to `r`;mapsto}
  | E\arrow!{to `r`;imply}
  | F\cr
  | G\arrow!{to `r`;dash}
  | H\arrow!{to `r`;arrow-color Color.red}
```

```
| I\arrow![to `r`;hook]
| J
| K\arrow![to `l`;hook-swap]
|}
```

$$A \longrightarrow B \cdots \cdots \cdots C \cdots \cdots \cdots D \longmapsto E \Longrightarrow F$$

$$G \longrightarrow H \longrightarrow I \hookrightarrow J \longleftarrow K$$

The option `phantom` erases the arrow and put the label between objects.

```
\matrixcd{
  a\arrow![to `r`;phantom;label ${\in}] | A
}
```

$$a \in A$$

`shift (x,y)` shifts the place of arrows, `shift-out (x,y)` (resp. `shift-in (x,y)`) shifts the source (resp. target) of the arrow.

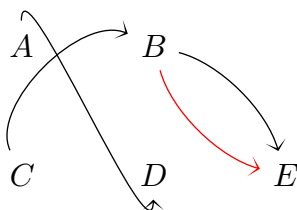
```
\matrixcd{
  | A\arrow![to `r`;shift (15pt,10pt)]
  | B\arrow![to `r`;shift-in (15pt,-15pt)]
  | C\arrow![to `r`;shift-out (-15pt,15pt)]
  | D
}
```

$$A \xrightarrow{\quad} B \begin{array}{l} \searrow \\ \nearrow \end{array} \begin{array}{l} C \nearrow \\ D \end{array}$$

Then angle at the source (resp. target) can be specified with `arrow-out` (resp. `arrow-`

in). The option `bend a`, where `a` is a float number, sets the angle at the source (resp. target) `a + <natural angle>` (resp. `180 - a + <natural angle>`). `bend-right` (resp. `bend-left`) is the same as `bend -30.0` (resp. `bend 30.0`).

```
\matrixcd{
| A\arrow![to `rd`; arrow-out 90.0;arrow-in 270.0]
| B\arrow![to `rd`;bend-right;]\arrow![to `rd`;bend-left]
| | \cr
| C
| D\arrow![to `ru`;bend 70.0]
| E
|}
```



4. Options for label

A label can be attached by `label` option with `\arrow`. The option `label` also accept options.

`label-color` changes the color of the label. `swap` swaps the position of the label. `pos f` change the position of the label. Here, `f` is a float number and `pos 0.0` (resp. `pos 1.0`) means that the label is at the source (resp. target) of the arrow. By `description`, label is placed on the arrow.

```
\matrixcd{
| A\arrow![to `r`;label ?:[label-color Color.red] ${f}]
| B\arrow![to `r`;label ?:[swap] ${g}]
| C\arrow![to `r`;label ?:[pos 0.2] ${h}]
| D\arrow![to `r`;label ?:[description] ${i}]
| E
|}
```

$$A \xrightarrow{\textcolor{red}{f}} B \xrightarrow{g} C \xrightarrow{h} D \xrightarrow{i} E$$

label-name gives a name to the label and such a name can be used to specify the position of the source or the target of an arrow. To specify via name, one can use to-by-name and from-by-name.

```
\matrixcd{
| A\arrow!{to `r`;label ?:[label-name `func-f`] ${f}} |B\cr
| C\arrow!{to `r`;label ?:[label-name `func-g`] ${g}}
| D\arrow![
  from-by-name `func-f`;to-by-name `func-g`;
  shift-out (0pt,-3pt);shift-in (0pt,3pt)]
|}
```

$$\begin{array}{ccc} A & \xrightarrow{f} & B \\ & \downarrow & \\ C & \xrightarrow{g} & D \end{array}$$

One can give a name to an empty label.

```
\matrixcd{
| A
  \arrow!{to `r`;bend-left;label [label-name `u`] ${}}
  \arrow!{to `r`;bend-right;label [label-name `d`] ${}}
  \arrow!{from-by-name `u`;to-by-name `d`;imply}
| B
|}
```

$$A \begin{array}{c} \curvearrowright \\ \Downarrow \\ \curvearrowleft \end{array} B$$