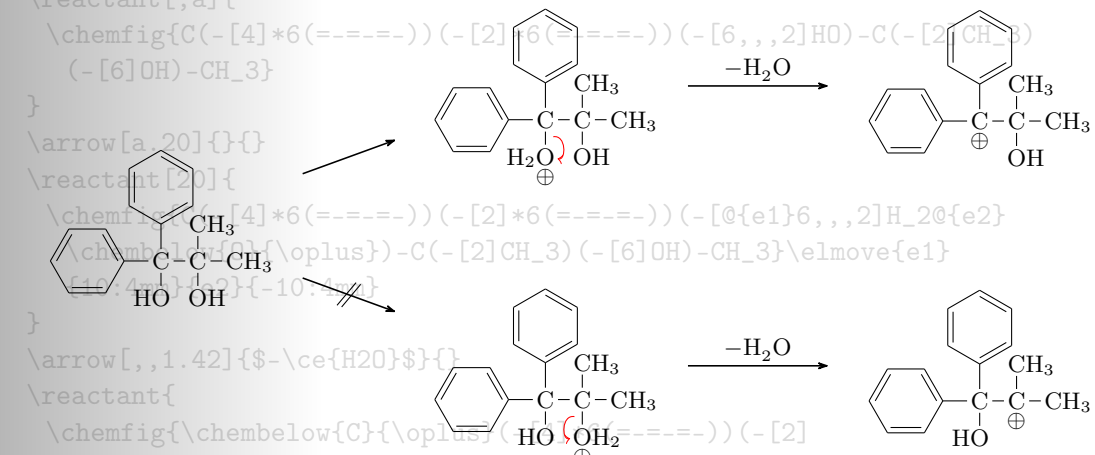
myChemistry

v1.5.1

April 27th 2011

Clemens Niederberger

Creating reaction schemes with LaTeX and ChemFig




```
\begin{rxn}[,.7]
 \setatomsep{1.5em}\footnotesize
 \reactant[,a]{
  \chemfig{C(-[4]*6(=-=-=-))(-[2]*6(=-=-=-))(-[6,,,2]HO)-C(-[2]CH_3)
   (-[6]OH)-CH_3}
 }
 \arrow[a,20]{}{}
 \reactant[20]{
  \chemfig{C(-[4]*6(=-=-=-))(-[2]*6(=-=-=-))(-[@{e1}6,,,2]H_2@{e2}
  \chembelow{O}{\oplus})-C(-[2]CH_3)(-[6]OH)-CH_3}\elmove{e1}
  {10:4mm}{e2}{-10:4mm}
 }
 \arrow[,,1.42]{$-\ce{H2O}$}{}
 \reactant{
  \chemfig{\chembelow{C}{\oplus}(-[2]*6(=-=-=-))(-[2]
  *6(=-=-=-))-C(-[2]CH_3)(-[6]OH)-CH_3}
 }
 \arrow[a.-20,-|>]{}{}
 \reactant[-20]{
  \chemfig{C(-[4]*6(=-=-=-))(-[2]*6(=-=-=-))(-[6,,,2]HO)-C(-[2]CH_3)
   (-[@{e3}6]@{e4}\chembelow{O}{\oplus}H_2)-CH_3}\elmove{e3}
   {170:4mm}{e4}{-170:4mm}
 }
 \arrow[,,1.42]{$-\ce{H2O}$}{}
 \reactant{
  \chemfig{C(-[4]*6(=-=-=-))(-[2]*6(=-=-=-))(-[6,,,2]HO)-
   \chembelow{C}{\oplus}(-[2]CH_3)-CH_3}
 }
\end{rxn}
```

Contents

1 About

1.1 Changes

The most recent changes are marked with new.

1.1.1 Version 1.2

Since v1.2 there are some new features. The main part plays behind the scenes: the wrong arrow- and branch-alignments have been corrected. **Due to these changes myChemistry now needs Version 2.10 of Ti*k*Z (or pgf, actually**, see section 1.3).

There are some new "frontend" features as well. For one thing there are some new package options (see section 2.5). Additionally the arrows have got two new keys (see section 4.2).

The environments have gained a few features, with which they can be customized, too (see section 4.14.1, section 4.15.1 and section 4.23).

Last but not least since v1.2 myChemistry underlies LPPL version 1.3 or later.

1.1.2 Version 1.3

The commands `\branch`, `\mesomeric`, `\reactant` and `\transition` now cannot only accept alignment as optional argument but Ti*k*Z keys as well. Also the second mandatory

Argument <anchor> now is an optional one. So the first argument still is alignment, the second is the anchor name and in the third you can use arbitrary Ti*k*Z keys.

```
\command[<pos>,<anchor>,<tikz>]{}
```

Version 1.2 explicitly needed a given alignment, if one wanted to use Ti*k*Z keys, even with default alignment. Since v1.5.1 this isn't required any more.

```
% up to version 1.2:
\reactant{\ce{Br2}}{}\arrow{$h\nu$}{}\reactant[right,
   draw,inner sep=5pt]{\ce{2 \lewis{0.,Br}}}{}
% since version 1.3:
\reactant{\ce{Br2}}\arrow{$h\nu$}{}\reactant[,,draw,
   inner sep=5pt]{\ce{2 \lewis{0.,Br}}}
```

The commands to customize myChemistry have been renamed and the command \mCsetup has been added. Now you can customize myChemistry using only one command. See section 4.17, section 4.18, section 4.19, section 4.21 and section 4.10.

There is the new command \chemand, which produces a +, see section 4.4.

And you might find this one nice: every myChemistry command in a listing in this documentation is a hyperlink referring to the corresponding entry in the command reference.

1.1.3 Version 1.4

The command \merge has been rewritten. Now the arrow can be labeled.

The main purpose of the command \dummy now is obsolete. The command still exists, though.

Both environments rxn and rxnscheme have changed regarding the usage of their options, see section 4.14.1 and section 4.15.1.

The keys of the \arrow command now are options, so the syntax of the command now is consistent with the syntax of the other commands. Then there are three new arrow types, see section 4.2. Also new is the appearance of the arrows and the possibility of customizing the line thickness of the arrows, see section 4.20.

There is another new command \anywhere (section 4.1), with which text or formulæ can be placed off the chain.

1.1.4 Version 1.5.1

NEW Much happened in the background. This also provides some new features in the foreground. Most importantly is a new handling of the positioning of objects. All objects (reactants, arrows, branches ...) now cannot only be positioned with key words like `right below` but by using an actual angle.

There are the new arrow types <=>> and <<=> which indicate an imbalanced equilibrium.

The default placement identifier of rxnscheme has been changed from `H` to `htp`.

The example section has been moved from this manual to the file `examples.tex` or `examples.pdf`, resp.

The commands \arrow, \reactant, \mesomeric, \transition, \anywhere, \dummy, \branch and \chemand are only defined inside the rxn and rxnscheme environments.

1.2 Licence

myChemistry v1.5.1 underlies the The LaTeX project public license (http://www.latex-project.org/lppl.txt).

1.3 Requirements

In order to function properly myChemistry needs some packages to be available.

ChemFig without it why would you use myChemistry?

ifthen for internal queries;

calc for internal calculations;

xkeyval package options and command keys are created with this package;

float the rxnscheme environment is defined with this package;

pgf/TikZ pgf actually isn't just one package but a whole bundle. They are the basis layer for TikZ. myChemistry needs at least the version from 09/08/2010[1]. More precisely: the command \pgfpositionnodelater must exist. Even more precisely: only the \arrow key `both` (see section 4.2) needs \pgfpositionnodelater. If you don't use this option, myChemistry should work nicely with pgf v2.00. Older versions have not been tested.

1.4 The Idea

Since ChemFig was published August 2010, there is a flexible solution for creating organic structures. With ChemFig and 'mhchem' one is able to create nearly all structural and molecular formulars that a chemist needs. There is one thing, though, in which 'ochem' still beats ChemFig: creating reaction mechanisms. This is where myChemistry comes in.

myChemistry loads the packages

- ChemFig[2],
- 'mhchem'[3] in version 3, when ist exists,
- 'chemexec'[4], when it exists, and

[1] http://sourceforge.net/projects/pgf/files/
[2] by Christian Tellechea, http://www.ctan.org/tex-archive/macros/generic/chemfig/
[3] by Martin Hensel, http://www.ctan.org/tex-archive/macros/latex/contrib/mhchem/
[4] by me, http://www.ctan.org/tex-archive/macros/latex/contrib/chemexec/

- 'chemcompounds'[1], when ist exists.

How these packages work, you can read in their documentations. Commands provided by these packages are for example

- `\ce{}` (mhchem)
- `\ox{}{}`, `\om[]`, `\op[]`, `\Hyd`, `\Hpl` (chemexec)
- `\chemfig[][]{}`, `\chemrel[]{}`, `\chemsign[]{}`, `\lewis{}` (ChemFig)
- `\declarecompound[]{}`, `\compound{}` (chemcompounds).

Some of these commands are used in the examples of this documentation *without being mentioned explicitly.*

Above all, myChemistry provides commands for creating reaction mechanisms.

2 Usage

2.1 Background

myChemistry provides two environments within which the mechanisms are created. Both environments basically are `tikzpicture` environments. One might ask oneself: why? You can do loads of stuff with ChemFig already. And Ti*k*Z provides all the possibilities one could ask for. But since I'm a lazy guy I wrote several macros providing Ti*k*Z commands, I often used. They got so many and got more and more possibilities to adjust, that I bundled them into this package. Of course you can still use Ti*k*Z and stay more flexible with it, if you like.

2.2 Basic Principle

Within the `tikzpicture` reactants and arrows are placed as nodes on a `chain`[2].

Example 1

```
\begin{tikzpicture}[start chain]
\node [on chain] {A};
\node [on chain] {B};
\node [on chain] {C};
\end{tikzpicture}
```

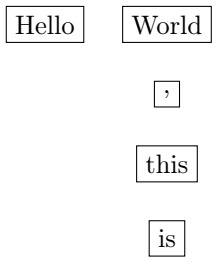
A B C

This way there are several possibilities to place the nodes relative to the others.

[1] by Stephan Schenk, http://www.ctan.org/tex-archive/macros/latex/contrib/chemcompounds/
[2] Provided by the tikzlibrary 'chains'

Example 2

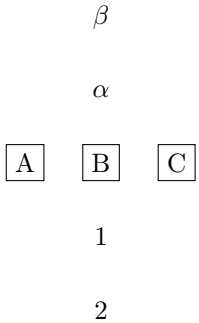
```
\begin{tikzpicture}[start chain=
    going right,node distance=5mm]
\node [draw,on chain] {Hello};
\node [draw,on chain] {World};
\node [draw,continue chain=going
     below,on chain] {,};
\node [draw,on chain] {this};
\node [draw,on chain] {is};
\end{tikzpicture}
```



Above all myChemistry uses the possibility of creating branches to the chain.

Example 3

```
\begin{tikzpicture}[start chain=
    going right,node distance=5mm]
 \node [draw,on chain] {A};
 \node [draw,on chain] {B};
{ [start branch]
 \node [on chain=going below]
    {1};
 \node [on chain=going below]
    {2};
}
{ [start branch]
 \node [on chain=going above]
    {$\alpha$};
 \node [on chain=going above]
    {$\beta$};
}
 \node [draw,on chain] {C};
\end{tikzpicture}
```



You don't have to understand that mechanism in detail, but you should remember the placement commands in the last example, because myChemistry uses them in the same way.

In some of the examples in this documentation the nodes are boxed with a coloured frame (see section 4.8). This is done so one can see, which size they have and which impact changes of the alignment have on them.

2.3 How does it work?

2.3.1 Basic Commands

There are three basic commands:

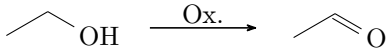
```
\begin{rxn}[<alignment>,<scale factor>]
 \reactant[<pos>,<name>,<tikz>]{<formula>}
```

```
\arrow[<pos>,<type>,<length factor>,<name>,both,<tikz
    >]{<above>}{<below>}
\end{rxn}
```

Schemes are created within the `rxn` environment. There you place reactants and arrows.

Example 4

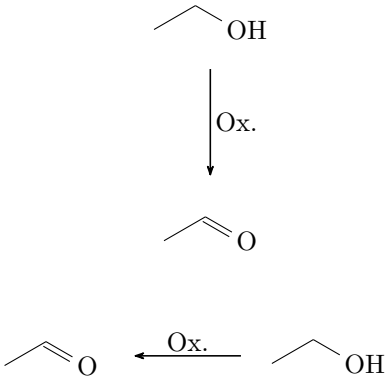
```
\begin{rxn}
 \reactant{ \chemfig
    {-[::30]-[::-60]OH} }
 \arrow{Ox.}{}
 \reactant{ \chemfig
    {-[::30]=_[::-60]O} }
\end{rxn}
```



When you don't use options reactants and arrows always are put to the right of the last object. With the option `<pos>` you can change this behaviour.

Example 5

```
% positioning using key words
\begin{rxn}
 \reactant{ \chemfig
    {-[::30]-[::-60]OH} }
 \arrow[below]{Ox.}{}
 \reactant[below]{ \chemfig
    {-[::30]=_[::-60]O} }
\end{rxn}
% positioning using an angle
\begin{rxn}
 \reactant{ \chemfig
    {-[::30]-[::-60]OH} }
 \arrow[180]{}{Ox.}
 \reactant[180]{ \chemfig
    {-[::30]=_[::-60]O} }
\end{rxn}
```



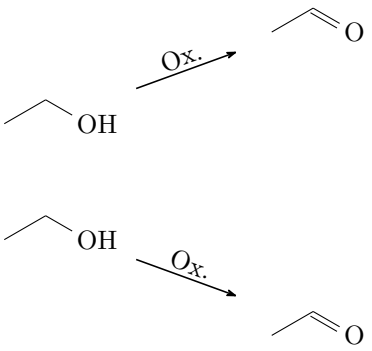
You can see in the last example, that positioning can be realized through key words (see table 1) like `below` or by using the angle relative to the horizontal line. Every angle from the interval $[-360°; 360°]$. $0°$ corresponds to `right`, which is the default value. *Positive angles* mean a turn *counter clockwise*, negative ones a turn clockwise, like you're used to in mathematics.

key word	pos. angle	neg. angle
`right`	0	±360
`right above`	45	−315
`above`	90	−270
`above left`	135	−225
`left`	180	−180
`below left`	225	−135
`below`	270	−90
`below right`	315	−45

Table 1: key words for positioning

Example 6

```
\begin{rxn}
 \reactant{ \chemfig
    {-[::30]-[::-60]OH} }
 \arrow[20]{Ox.}{}
 \reactant[20]{ \chemfig
    {-[::30]=_[::-60]O} }
\end{rxn}
\begin{rxn}
 \reactant{ \chemfig
    {-[::30]-[::-60]OH} }
 \arrow[-20]{Ox.}{}
 \reactant[-20]{ \chemfig
    {-[::30]=_[::-60]O} }
\end{rxn}
```

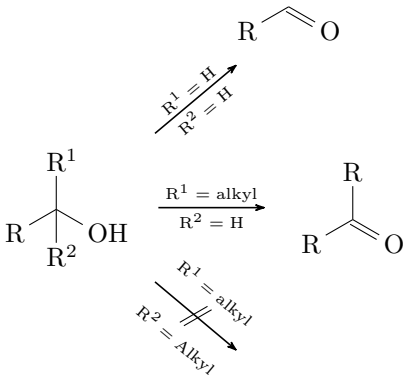


2.3.2 Positioning

Reactants and arrows cannot only be positioned through key words and angles. They can refer to another reactant or arrow.

Example 7

```
\begin{rxn}
 \reactant[,start]{ \chemfig{R-[::30](-[::60]R|^1) (-[::-120]R|^2)
    -[::-60]OH} }
 \arrow[40]{\tiny$\text{R}^1=\text{H}$} {\tiny$\text{R}^2=\text{H}$}
 \reactant[40]{ \chemfig{R-[::30]=_[::-60]O} }
 \arrow[start.0]{\tiny$\text{R}^1=\text{alkyl}$} {\tiny$\text{R}^2=\
    text{H}$}
 \reactant{ \chemfig{R-[::30](-[::60]R)=_[::-60]O} }
 \arrow[start.-40,-|>]{\tiny$\text{R}^1=\text{alkyl}$} {\tiny$\text{R
    }^2=\text{Alkyl}$}
\end{rxn}
```



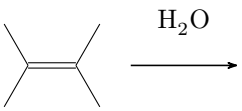
In the last example the first reactant got the `<name> start`. The arrows in lines 5 and 7 now could refer to it. A *previously given name* can act as an anchor for later reactants or arrows, if the positioning is written like

```
<anchor>.<angle>
```

Arrows can be given names, too. The anchor point of an arrow always is in the middle of the arrow line and has *no* size.

Example 8

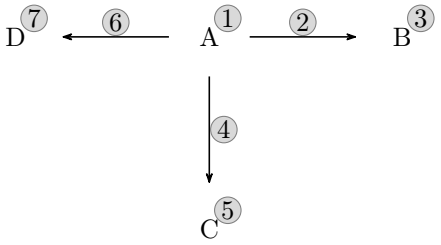
```
\begin{rxn}
 \reactant{\chemfig
         {[:60]-(-[::60])=[::-60](-[::-60])-}}
 \arrow[,,,arrow]{}{}
 \reactant[arrow.90]{\ce{H2O}}
\end{rxn}
```



Using this kind of positioning does *not* break the chain.

```
\begin{rxn}
 \reactant[,a]{A}
 \arrow{}{}
 \reactant{B}
 \arrow[a.-90]{}{}
 \reactant[-90]{C}
 \arrow[a.180]{}{}
 \reactant[180]{D}
\end{rxn}
```

All seven objects of this example are logically speaking part of the same chain. The next obect is placed to the right of the last one, if no positioning is used.



2.3.3 Branches

To break a chain you use the command

```
\branch[<pos>,<name>,<tikz>]{<formulae>}
```

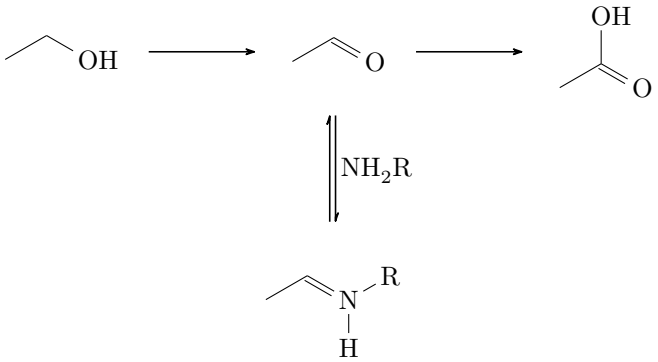
The positioning of a branch is slightly different from earlier objects, although the syntax is similar. A branch has two additional ways of positioning. Every positioning that refers to an anchor cause that the branch is not a part of the chain but is a real branch.

```
<angle>                 % on the chain
<key>                   % on the chain
<anchor>.<angle>        % not on the chain
on chain=going <key>    % on the chain
<key>=of <anchor>       % not on the chain
```

As `<key>` you again use the values from table 1.

Example 9

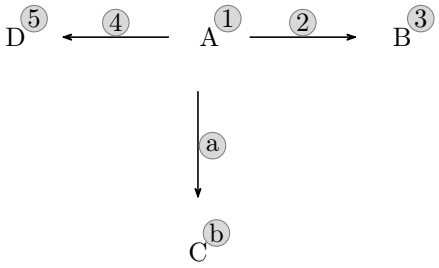
```
\begin{rxn}
 \reactant{ \chemfig{-[::30]-[::-60]OH} }
 \arrow{}{}
 \reactant[,carbonyl]{ \chemfig{-[::30]=_[::-60]O} }
 \branch[carbonyl.-90]{
   \arrow[-90,<=>]{\ce{NH2R}}{}
   \reactant[-90]{ \chemfig{-[::30]=_[::-60]N(-[6]H)-[::60]R} }
 }
 \arrow{}{}
 \reactant{ \chemfig{-[::30](-[::60]OH)=_[::-60]O} }
\end{rxn}
```



Please note that in the last example the arrow and the reactant placed after branch continue the original chain.

```
\begin{rxn}
 \reactant[,a]{A}
 \arrow{}{}
 \reactant{B}
 \branch[a.-90]{
   \arrow[-90]{}{}
   \reactant[-90]{C}
 }
 \arrow[a.180]{}{}
 \reactant[180]{D}
\end{rxn}
```

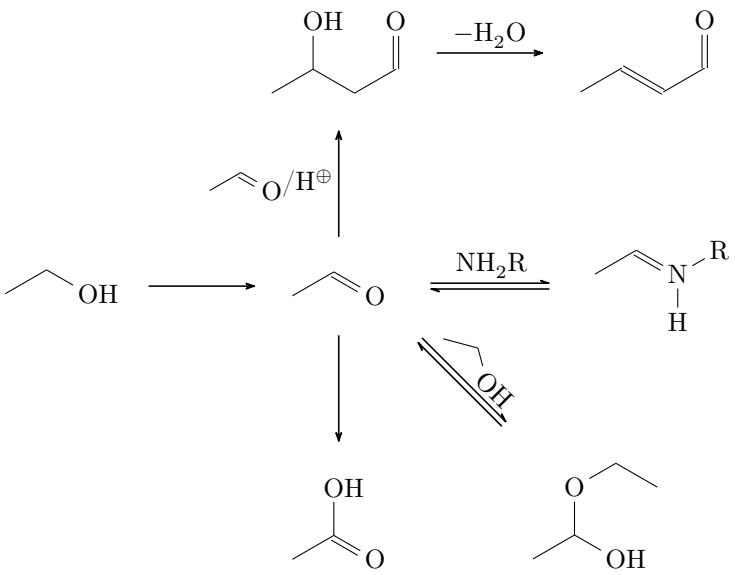
The chain is broken by the branch which starts a new chain itself.



Using \branch allows to create schemes with several branches:

Example 10

```
\begin{rxn}
 \reactant{ \chemfig{-[::30]-[::-60]OH} }
 \arrow{}{}
 \reactant[,carbonyl]{ \chemfig{-[::30]=_[::-60]O} }
 \arrow[-90]{}{}
 \reactant[-90]{ \chemfig{-[::30](-[::60]OH)=_[::-60]O} }
 \branch[right=of carbonyl]{
   \arrow[,<=>,1.12]{\ce{NH2R}}{}
   \reactant{ \chemfig{-[::30]=_[::-60]N(-[6]H)-[::60]R} }
 }
 \branch[below right=of carbonyl]{
   \arrow[-45,<=>,1.12]{ \chemfig{[,.75]-[::30]-[::-60]OH} }{}
   \reactant[-45]{ \chemfig{-[::30](-[::60]O-[::-60]-[::-60])-[::-60]OH
    } }
 }
 \arrow[carbonyl.90]{ \chemfig{[,.75]-[::30]=_[::-60]O}/\Hpl }{}
 \reactant[90]{ \chemfig{-[::30](-[::60]OH)-[::-60]-[::60]=[::60]O} }
 \arrow{$-\ce{H2O}$}{}
 \reactant{ \chemfig{-[::30]=[::-60]-[::60]=[::60]O} }
\end{rxn}
```



2.3.4 Numbered Schemes

There is another environment to create schemes.

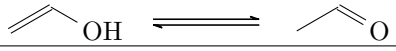
```
\begin{rxnscheme}[<label>,<placement>,<alignment>,<skale
   factor>,<title>]{<caption>}
  ...
\end{rxnscheme}
```

This is a floating environment for schemes, which can be given a `<caption>`, a `<label>` and the common placement identifiers like `htp` (default).

Example 11

```
\begin{rxnscheme}[,H]{Keto-Enol-tautomerism}
\reactant{ \chemfig{=[::30]-[::-60]OH} }
\arrow[,<=>]{}{}
\reactant{ \chemfig{-[::30]=[::-60]O} }
\end{rxnscheme}
```

Reaction scheme 1 Keto-Enol-tautomerism



2.4 Predefined Values

There are some predefined values, that are basically due to my personal taste. But of course you can change them according to your requirements. For ChemFig-formulæ

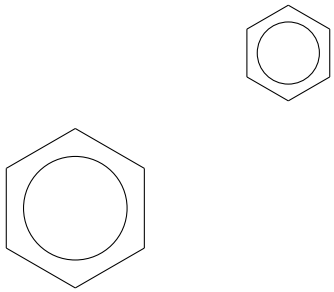
inside of myChemistry *environments* some values are predefined as follows:

```
\setatomsep{1.8em}
\setcrambond{3pt}{0.5pt}{1pt}
```

Outside the myChemistry environments the defaults of ChemFig still are set.

Example 12

```
\begin{rxn}
 \reactant{\chemfig{**6(------)
          }}
\end{rxn}
\chemfig{**6(------)}
```



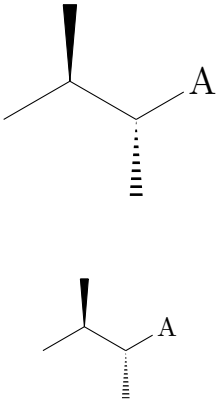
myChemistry's defaults can be changed with these commands:

```
\setbondlength{<length>}
\setbondshape{<base length>}{<dash thickness>}{<dash
   spacing>}
\setatomsize{<font size>}
```

With these commands, the parameters are changed *for all following* myChemistry environments. If you leave the arguments empty, default values are restored. Default for \setatomsize is \small.

Example 13

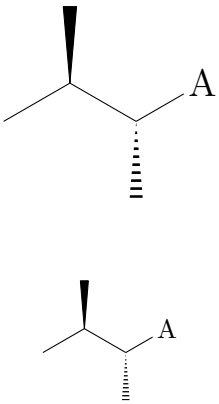
```
\setbondlength{2.1em}\setbondshape{5pt}{1pt}{2pt}\setatomsize{\Large}
\begin{rxn}
 \reactant{\chemfig{-[::30](<[::60])-[::-60](<:[::-60])-[::60]A}}
\end{rxn}
\setbondlength{}\setbondshape{}{}{}\setatomsize{}
\begin{rxn}
 \reactant{\chemfig{-[::30](<[::60])-[::-60](<:[::-60])-[::60]A}}
\end{rxn}
```



If you only want to change the parameters of a single environment you can use ChemFig's commands and LaTeX's fontsize commands *inside the environment*.

Example 14

```
\begin{rxn}
 \setatomsep{2.1em}\setcrambond{5pt}{1pt}{2pt}\Large
 \reactant{\chemfig{-[::30](<[::60])-[::-60](<:[::-60])-[::60]A}}
\end{rxn}
\begin{rxn}
 \reactant{\chemfig{-[::30](<[::60])-[::-60](<:[::-60])-[::60]A}}
\end{rxn}
```



The default length of reaction arrows is 4 em, the default line width is `semithick` and the default label distance is 0.2 em. You can change these default values with the commands

```
\setarrowlength{<length>}
\setarrowline{<line width>}
\setarrowlabel{<label distance>}
```

or with

```
\mCsetup{arrowlength=<length>,
         arrowline=<line width>,
         arrowlabel=<label distance>}
```

2.5 Package options

myChemistry has a number of package options.

`chemstyle` load the 'chemstyle' package, without conflicts with myChemistry.

`color=<colour>` This loads 'chemexec' with the options `color<colour>` & `shade=true`.

`english` With this option, myChemistry loads the english version of 'chemexec', if the package isn't loaded separately before. The name of the `rxnscheme` environment (see section 4.15) is changed into "Reaction scheme".

`nochemexec` prevent myChemistry from loading 'chemexec'.

`nocolor` 'chemexec' is loaded without colour and with the option `shade=false` (default behaviour of myChemistry).

`nocompounds` prevent myChemistry from loading 'chemcompounds'.

`nomhchem` prevent myChemistry from loading 'mhchem' lädt, if 'chemexec' isn't loaded either.

`nopackages` prevent myChemistry from loading *any* package (except ChemFig)[1].

`placement=<position>` The default placement behaviour of the `rxnscheme` environment is changed to `<position>`.

`shade` loads 'chemexec' with the `shade=true` option.

3 Advanced Usage, Usage with TikZ

The biggest problem with myChemistry usually is the correct positioning of reactants and arrows. section 3.1 looks a little bit into this topic.

Some of the commands can be given TikZ code as third optional argument. More precisely you can use the same TikZ keys there as you would with a `\node` inside a `tikzpicture`. If a node is placed with `\node[<tikz>](<placement>){<anything>};`, then `<tikz>` is about the same in e. g. `\reactant[,,<tikz>]{}`. With this you can customize your scheme in many ways.

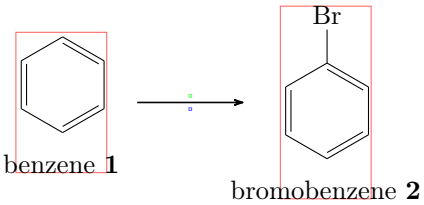
[1] apart from the ones myChemistry needs to function (like TikZ etc.).

3.1 The Alignment Question

Since reactants, arrows and branches are aligned centered to the referred object, the default alignment not always produces nice results.

Example 15

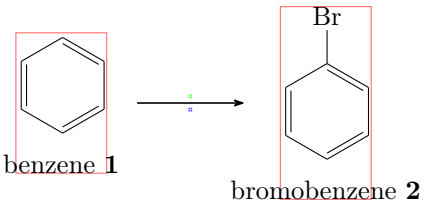
```
\makevisible
\begin{rxn}
 \reactant{ \chemname{\chemfig{*6(-=-=-=)}}{benzene \compound{benzene}}
   }
 \arrow{}{}
 \reactant{ \chemname{\chemfig{*6(-=-=(-Br)-=)}}{bromobenzene \compound
  {bromobenzene}} }
\end{rxn}
```



As you can see, both reactants are not aligned equally to the arrow, as far as the benzene ring is concerned. The first reactant seems to be shifted up. Trying to solve this with Ti*k*Z code fails:

Example 16

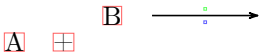
```
\makevisible
\begin{rxn}
 \reactant[,,yshift=-1em]{ \chemname{\chemfig{*6(-=-=-=)}}{benzene \
  compound{benzene}} }
 \arrow{}{}
 \reactant{ \chemname{\chemfig{*6(-=-=(-Br)-=)}}{bromobenzene \compound
  {bromobenzene}} }
\end{rxn}
```



This is, because the first reactant is shifted with the respect to the object it refers to. Since it is the first object on the chain itself, it isn't shifted at all. The following arrow always is centered to the object before.

Example 17

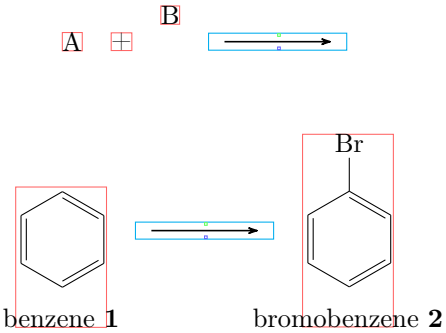
```
\makevisible
\begin{rxn}
 \reactant{A}
 \chemand
 \reactant[,,yshift=1em]{B}
 \arrow{}{}
\end{rxn}
```



Since there is no possibility to change the alignment of the arrow itself (yet), what you can do is put it inside a branch.

Example 18

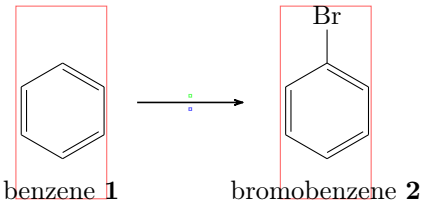
```
\makevisible
\begin{rxn}
 \reactant{A}
 \chemand
 \reactant[,,yshift=1em]{B}
 \branch[,,yshift=-1em]{\arrow{}{}}
\end{rxn}
\begin{rxn}
 \reactant{ \chemname{\chemfig{*6(-=-=-=)}}{benzene \compound{benzene}}
   }
 \branch[,,yshift=1em]{\arrow{}{}}
 \reactant{ \chemname{\chemfig{*6(-=-=(-Br)-=)}}{bromobenzene \compound
  {bromobenzene}} }
\end{rxn}
```



For the last example this isn't the best solution, though, because exact alignment needs lots of tries until you get the required result. There is another solution: an invisible bromine to the first benzene.

Example 19

```
\makevisible
\begin{rxn}
 \reactant{ \chemname{\chemfig{*6(-=-=(-[,,,,draw=none]\phantom{Br})-=)
  }}{benzene \compound{benzene}} }
 \arrow{}{}
 \reactant{ \chemname{\chemfig{*6(-=-=(-Br)-=)}}{bromobenzene \compound
  {bromobenzene}} }
\end{rxn}
```

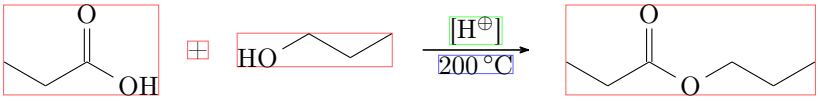


In other cases, too, an invisible substituent should be preferred over Ti*k*Z code, since it's easier and more precise:

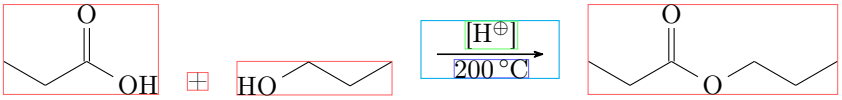
Example 20

```
\makevisible
default:
\begin{rxn}
 \reactant{\chemfig{-[:-30]-[:30](=[2]O)-[:-30]OH}}
 \chemand
 \reactant{\chemfig{HO-[:30]-[:-30]-[:30]}}
 \arrow{[\Hpl]}{\SI{200}{\celsius}}
 \reactant{\chemfig{-[:-30]-[:30](=[2]O)-[:-30]O-[:30]-[:-30]-[:30]}}
\end{rxn}
hydroxy groups at the same height through TikZ:
\begin{rxn}
 \reactant{\chemfig{-[:-30]-[:30](=[2]O)-[:-30]OH}}
 \chemand[,,yshift=-1.2em]
 \reactant[,,yshift=.12em]{\chemfig{HO-[:30]-[:-30]-[:30]}}
 \branch[,,yshift=1.08em]{\arrow{[\Hpl]}{\SI{200}{\celsius}}}
 \reactant{\chemfig{-[:-30]-[:30](=[2]O)-[:-30]O-[:30]-[:-30]-[:30]}}
\end{rxn}
hydroxy groups at the same height through an invisible substituent:
\begin{rxn}
 \reactant{\chemfig{-[:-30]-[:30](=[2]O)-[:-30]OH}}
 \chemand
 \reactant{\chemfig{HO-[:30](=[2,,,,draw=none]\phantom{O})
  -[:-30]-[:30]}}
 \arrow{[\Hpl]}{\SI{200}{\celsius}}
 \reactant{\chemfig{-[:-30]-[:30](=[2]O)-[:-30]O-[:30]-[:-30]-[:30]}}
\end{rxn}
```

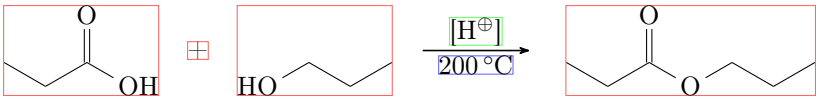
default:



hydroxy groups at the same height through TikZ:



hydroxy groups at the same height through an invisible substituent:



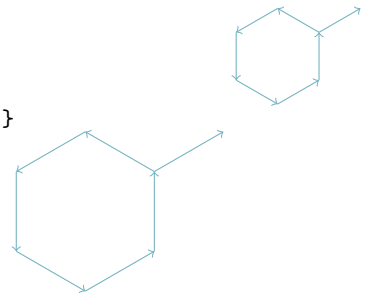
I'm afraid that in many other cases you'll have to play with `xshift` and `yshift`, though, until the scheme looks the way you want. Maybe further versions of myChemistry will provide a more user friendly alignment syntax.

3.2 Using TikZ to Achieve Other Results

You could, just for fun?, change the looks of a molecule with TikZ.

Example 21

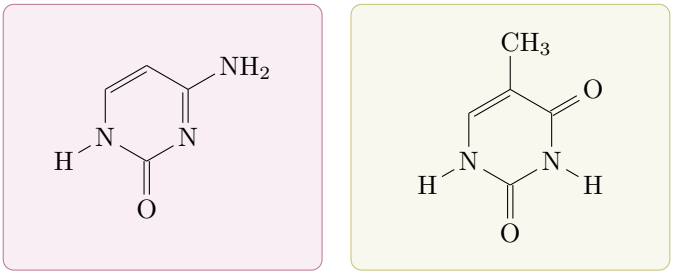
```
\begin{rxn}
 \reactant[,,->,green!45!blue
     !55]{ \chemfig{*6(---(-)---)} }
\end{rxn}
\chemfig[->,green!45!blue
     !55]{*6(---(-)---)}
```



The last example is not very good, of course, since you can achieve the same result using ChemFig's own possibilities. But other cases are imaginable: for example one could define a style with which reactants are shown:

Example 22

```
\colorlet{mCgreen}{green!50!gray}
\colorlet{mCblue}{cyan!50!gray}
\colorlet{mCred}{magenta!50!gray}
\colorlet{mCyellow}{yellow!50!gray}
\begin{rxn}
 \tikzset{reactant/.style={draw=#1,fill=#1!10,inner sep=1em,minimum
    height=10em,minimum width=12em,rounded corners}}
 \reactant[,cytosine,reactant=mCred]{\chemfig{H-[:30]N*6(-(=O)-N=(-NH
    _2)-=-)}}
 \anywhere{cytosine.-90,,yshift=-2mm}{Cytosine}
 \reactant[,thymine,reactant=mCyellow]{\chemfig{H-[:30]N*6(-(=O)-N(-H)
    -(=O)-(-CH_3)=-)}}
 \anywhere{thymine.-90,,yshift=-2mm}{Thymine}
 \reactant[cytosine.-90,adenine,yshift=-2em,reactant=mCblue]{\chemfig
    {[:-36]*5(-N(-H)-*6(-N=-N=(-NH_2)--)--N=)}}
 \anywhere{adenine.-90,Guanin,yshift=-2mm}{Adenine}
 \reactant[,guanine,reactant=mCgreen]{\chemfig{[:-36]*5(-N(-H)-*6(-N=(-
    NH_2)-N(-H)-(=O)--)--N=)}}
 \anywhere{guanine.-90,,yshift=-2mm}{Guanine}
\end{rxn}
```



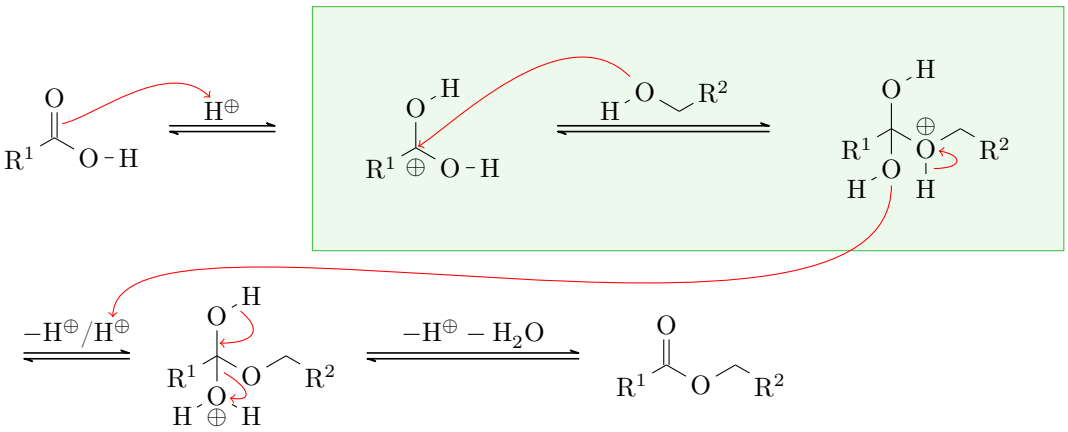
Cytosine Thymine

Adenine Guanine

This way certain parts of a scheme could be emphasized:

Example 23

```
\begin{rxn}
 \setatomsep{1.5em}
 \colorlet{mCgreen}{green!50!gray}
 \tikzset{emph/.style={draw=mCgreen,fill=mCgreen!10,inner sep=1em}}
 \reactant[,lineOne]{\chemfig{R^1-[:30](=[@{b1}2]O)-[:-30]O-H}}
 \arrow[,<=>]{\chemfig{@{Hpl1}\Hpl}}{}
 \branch[,,emph]{
   \reactant{\chemfig{R^1-[:30]@{C1}(-[2]O-[:30]H)(-[6,.5,,,draw=none]\
    oplus)-[:-30]O-H}}
   \arrow[,<=>,2]{\chemfig{[:30]H-@{O1}O-[::-60]-R^2}}{}
   \reactant{\chemfig{R^1-[:30](-[2]O-[:30]H)(-[6]@{O2}O-[:-150]H)
    -[:-30]@{O3}\chemabove{O}{\oplus}(-[@{b2}6]H)-[:30]-[:-30]R^2}}
 }
 \anywhere{lineOne.-90,lineTwo,xshift=-3em,yshift=-7em}{}
 \arrow[lineTwo.0,<=>]{$-$\Hpl/\chemfig{@{Hpl2}\Hpl}}{}
 \reactant{\chemfig{R^1-[:30](-[@{b3}2]O-[@{b4}:30]H)(-[@{b5}6]@{O4}\
    chembelow{O}{\oplus}(-[:-30]H)-[:-150]H)-[:-30]O-[:30]-[:-30]R^2}}
 \arrow[,<=>,2]{\ce{- \Hpl - H2O}}{}
 \reactant{\chemfig{R^1-[:30](=[2]O)-[:-30]O-[:30]-[:-30]R^2}}
 \anywhere{lineTwo.-90}{
   \elmove{b1}{10:5mm}{Hpl1}{135:1cm}
   \elmove{O1}{135:1.5cm}{C1}{30:5mm}
   \elmove{O2}{-90:3cm}{Hpl2}{90:2cm}
   \elmove{b2}{0:5mm}{O3}{-10:5mm}
   \elmove{b4}{-40:5mm}{b3}{0:5mm}
   \elmove{b5}{-30:5mm}{O4}{-10:5mm}
 }
\end{rxn}
```



4 Alphabetical Command Reference

In the following section every command is explained.

4.1 anywhere

new Sometimes it might be useful to be able to place a reactant or anything else off the chain.

```
\anywhere{<pos>,<name>,<tikz>}{<something>}
```

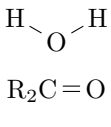
For this case there's \anywhere. It is positioned with <pos> similar to \branch.

```
<anchor>.<angle>      % not on the chain
on chain=going <key> % on the chain
<key>=of <anchor>     % not on the chain
```

Please be aware that <pos> cannot be left empty.

Example 24

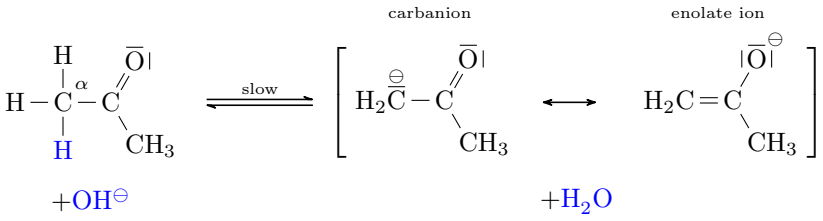
```
\begin{rxn}
 \reactant[,carbonyl_A]{\chemfig
   {R_2C=O}}
 \anywhere{above=of carbonyl_A
   }{\chemfig{H-[:-30]O-[:30]H}};
\end{rxn}
```



You can use this command e. g. for placing labels naxt to reactants.

Example 25

```
\begin{rxn}
 \reactant[,ketone]{\chemfig{H-\chemabove{C}{\hspace*{5mm}\scriptstyle\
   alpha}(-[2]H)(-[6,,,2]{}|{\textcolor{blue}H})-C(=[:60]\lewis{02,O})
   -[:-60]C|H_3}}
 \anywhere{below=of ketone}{$+$\color{blue}\Hyd}
 \arrow[,<=>]{\tiny slow}{}
 \mesomeric[,mesomer]{
   \reactant[,carbanion]{\chemfig{H_2|\chemabove[3pt]{\lewis{2,C}}{\
   scriptstyle\ominus}-C(=[:60]\lewis{02,O})-[:-60]C|H_3}}
   \marrow
   \reactant[,enolate]{\chemfig{H_2C=C(-[:60]\chemabove{\lewis{024,O
   }}{\hspace*{5mm}\scriptstyle\ominus})-[:-60]C|H_3}}
 }
 \anywhere{above=of enolate}{\tiny enolate ion}
 \anywhere{above=of carbanion}{\tiny carbanion}
 \anywhere{below=of mesomer}{$+$\color{blue}\ce{H2O}}
\end{rxn}
```



You can find many further examples for the usage of `\anywhere` in `examples.tex` or `examples.pdf`, resp.

4.2 arrow

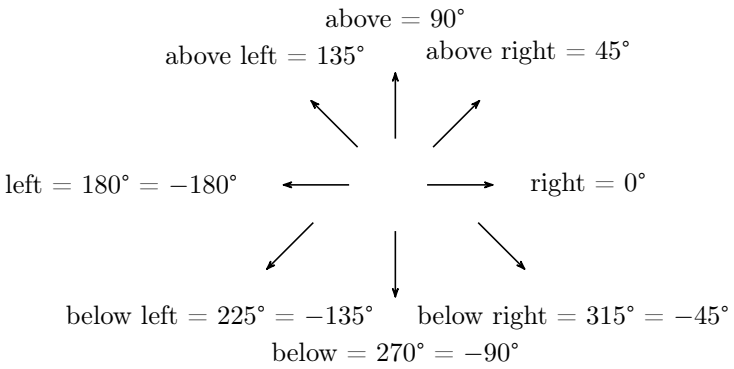
Reaction arrows are created with `\arrow`.

```
\arrow[<pos>,<type>,<length factor>,<name>,both,<tikz>]{<
    above>}{<below>}
```

4.2.1 Options

There are a number of options to customize the arrows. The options must be used in the right order, seperated with commas.

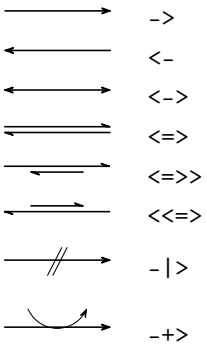
1. `<pos>` – possible values:



NEW Additionally you can use every angle from the interval $[-360°; 360°]$.

You can also use an angle with respect to an object that's been named with `<name>`: `<name>.<angle>`. This means the arrow is placed in an angle of `<angle>` next to `<name>`. You can find lots of examples in `examples.tex` or`examples.pdf`, resp.

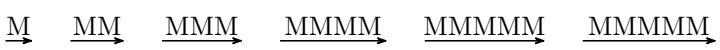
2. `<type>` – possible values:



3. `<length factor>` – this factor multiplies with the arrow length (4 em with factor = 1.0, default)
4. `<name>` – can be used to give the arrow an anchor, to which another object can refer.
5. `both`– this gives both nodes within which the labels are written the same size.
6. `<tikz>` – you can modify the arrow with Ti*k*Z keys. Not every Ti*k*Z keys have an effect, though. You can't shift arrows with `shift=<coord>`, for example.

Example 26

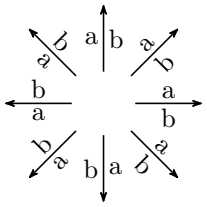
```
\begin{rxn}
\arrow[,,.25]{M}{}\arrow[,,.5]{MM}{}\arrow[,,.75]{MMM}{}\arrow{MMMM}{}\arrow[,,1.125]{MMMMM}{}\arrow[,,1.25]{MMMMM}{}
\end{rxn}
```



Please note, that the labels `<above>` and `<below>` are rotated with the arrow. At an angle of 180° `<above>` actually is *below* the arrow.

Example 27

```
\begin{rxn}
\setarrowlength{2.5em}
\dummy[a]
\arrow{a}{b}
\arrow[a.45]{a}{b}
\arrow[a.90]{a}{b}
\arrow[a.135]{a}{b}
\arrow[a.180]{a}{b}
\arrow[a.225]{a}{b}
\arrow[a.270]{a}{b}
\arrow[a.315]{a}{b}
\end{rxn}
```



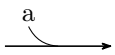
NEW You need to be aware of one or two points, when using the arrow type -+>: if you leave the labels empty, the arrow is the same as ->. The first label is the one added to the reaction, the second the one subtracted. If you use only one of the labels only the corresponding arrow part is drawn.

Example 28

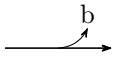
```
corresponds \verb"->":
\begin{rxn}
 \arrow[,-+>]{}{}
\end{rxn}
add:
\begin{rxn}
 \arrow[,-+>]{a}{}
\end{rxn}
subtract:
\begin{rxn}
 \arrow[,-+>]{}{b}
\end{rxn}
both add and subtract:
\begin{rxn}
 \arrow[,-+>]{a}{b}
\end{rxn}
Blanks are \emph{not} an empty
    label:
\begin{rxn}
 \arrow[,-+>]{ }{ }
\end{rxn}
```

corresponds ->:

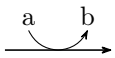
add:



subtract:



both add and subtract:

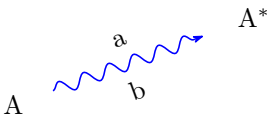


Blanks are *not* an empty label:

NEW You can change the appearance of an arrow with Ti*k*Z keys:

Example 29

```
\begin{rxn}
 \mCsetup{arrowlabel=.7em,arrowlength=6em}
 \reactant{A}
 \arrow[20,,,,,decorate,decoration=snake,blue]{a}{b}
 \reactant[20]{A$^*$}
\end{rxn}
```



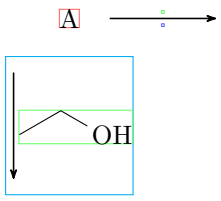
4.2.2 Alignment

If an arrow is placed inside a branch, the alignment of the branch may be determined by the size of the label nodes. If the two labels have a different size, alignment can go

wrong.

Example 30

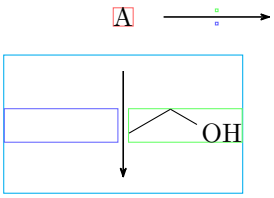
```
\makevisible
\begin{rxn}
 \reactant[,a]{A}
 \arrow{}{}
 \branch[below=of a]{
   \arrow[-90]{\chemfig
    {-[::30]-[::-60]OH}}{}
 }
\end{rxn}
\makeinvisible
```



By using the option `both` both nodes get the same size. This can correct the alignment.

Example 31

```
\makevisible
\begin{rxn}
 \reactant[,a]{A}
 \arrow{}{}
 \branch[below=of a]{
  \arrow[-90,,,,both]{\chemfig
    {-[::30]-[::-60]OH}}{}
 }
\end{rxn}
\makeinvisible
```



There's more to the question of alignment in section 4.3.2.

4.2.3 Appearance

You can change the general appearance of arrows with `\setarrowlength` (section 4.17) and `\setarrowline` (section 4.20).

4.3 branch

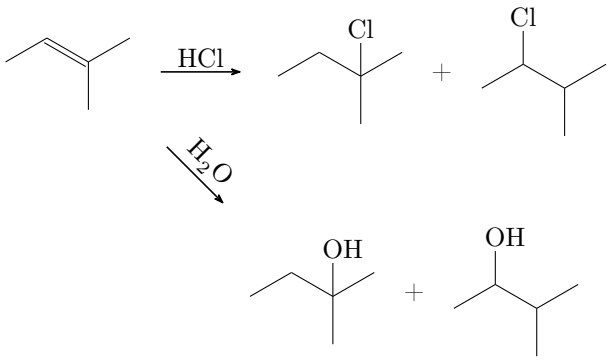
`\branch` is used to, well, create a branch to a reaction.

```
\branch[<pos>,<anchor>,<tikz>]{<branch code>}
```

For `\branch` positioning an anchor is important. Let's take a look at an example:

Example 32

```
1 \begin{rxn}
2  \reactant[,start]{\chemfig{-[::30]=_[::-60](-[::-60])-[::60]}}
3  \arrow[,,.75]{\ce{HCl}}{}
4  \reactant{\chemfig{-[::30]-[::-60](-[::120]Cl)(-[::-60])-[::60]}}
5  \chemand
6  \reactant{\chemfig{-[::30](-[::60]Cl)-[::-60](-[::-60])-[::60]}}
7  \branch[below right=of start]{
8    \arrow[-45,,.75]{\ce{H2O}}{}
9    \reactant[-45]{\chemfig{-[::30]-[::-60](-[::120]OH)(-[::-60])
      -[::60]}}
10   \chemand
11   \reactant{\chemfig{-[::30](-[::60]OH)-[::-60](-[::-60])-[::60]}}
12 }
13 \end{rxn}
```



The first reactant got the anchor `start` (line 2, also see section 4.13).

```
2   \reactant[,start]{ ... }
```

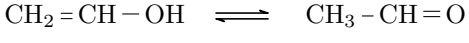
`\branch` now refers to it in its alignment (line 7):

```
7   \branch[below right=of start]{ ... }
```

If you don't use the alignment reference to an anchor, you automatically refer to the last `\reactant` or `\arrow`. If you don't use alignment at all, then the branch is aligned to the right of the last `\reactant` or `\arrow`.

Example 33

```
1 \begin{rxn}
2  \reactant{ \chemfig{CH_2=CH-OH}
      }
3  \arrow[,<=>,.5]{}{}
4  \branch{ \reactant{ \chemfig{CH
     _3-CH=O} } }
5 \end{rxn}
```



4.3.1 Positioning

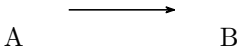
There are several ways to position a branch. It can either be part of the chain or be a real branch.

```
<angle>                % on the chain
<key>                  % on the chain
<anchor>.<angle>       % not on the chain
on chain=going <key>   % on the chain
<key>=of <anchor>      % not on the chain
```

Default behaviour is equal to `\branch[0]{}`. In different situations different ways can be favoured. For example if you want to use `\branch` to shift an arrow, it can still be part of the chain.

Example 34

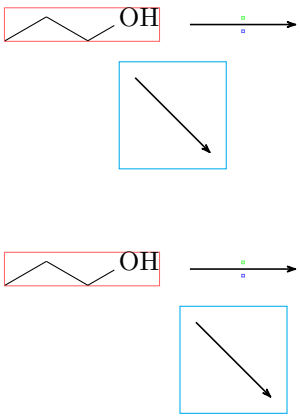
```
\begin{rxn}
 \reactant{A}
 \branch[,,yshift=1em]{\arrow
    {}{}}
 \reactant[,,yshift=-1em]{B}
\end{rxn}
```



If you like to start a real branch, it *does* matter, which way you use.

Example 35

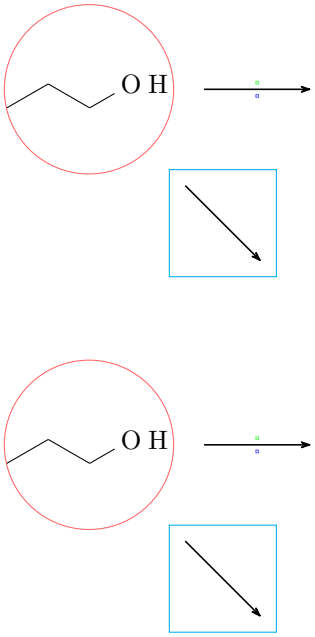
```
\makevisible
\begin{rxn}
 \reactant[,a]{\chemfig
    {[:30]--[::-60]-OH}}
 \arrow{}{}
 \branch[a.-45]{\arrow[-45]{}{}}
\end{rxn}
\begin{rxn}
 \reactant[,a]{\chemfig
    {[:30]--[::-60]-OH}}
 \arrow{}{}
 \branch[below right=of a]{\
    arrow[-45]{}{}}
\end{rxn}
```



If a reactant isn't squared or circled −45° does not mean the same as "below right".

Example 36

```
\makevisible
\begin{rxn}
 \reactant[,a,circle]{\chemfig
    {[:30]--[::-60]-OH}}
 \arrow{}{}
 \branch[a.-45]{\arrow[-45]{}{}}
\end{rxn}
\begin{rxn}
 \reactant[,a,circle]{\chemfig
    {[:30]--[::-60]-OH}}
 \arrow{}{}
 \branch[below right=of a]{\
    arrow[-45]{}{}}
\end{rxn}
```

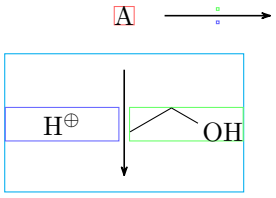


4.3.2 Alignment problems

If an arrow has two arguments with different sizes and is placed inside a branch, the alignment of the branch can go wrong. In this case the `\arrow` key `both` isn't a solution, since the smaller argument then is not placed next to the arrow but is centered in its node.

Example 37

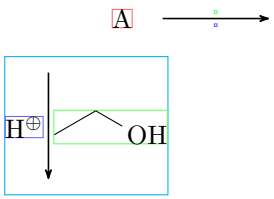
```
\makevisible
\begin{rxn}
 \reactant[,a]{A}
 \arrow{}{}
 \branch[below=of a]{
  \arrow[below,,,,both]{\chemfig
    {-[::30]-[::-60]OH}}{\Hpl}
 }
\end{rxn}
\makeinvisible
```



What you have to do is shift the branch using the Ti*k*Z keys `xshift` and `yshift`.

Example 38

```
\makevisible
\begin{rxn}
 \reactant[,a]{A}
 \arrow{}{}
 \branch[below=of a,,xshift
    =-1.35em]{
  \arrow[below]{\chemfig
    {-[::30]-[::-60]OH}}{\Hpl}
 }
\end{rxn}
\makeinvisible
```



4.4 chemand

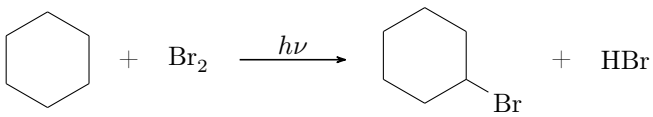
The command

```
\chemand[<alignment>,<anchor>,<tikz>]
```

produces and places a + in the same way \reactant places arbitrary text.

Example 39

```
\begin{rxn}
 \reactant{\chemfig{*6(------)}}
 \chemand
 \reactant{\ce{Br2}}
 \arrow{$h\nu$}{}
 \reactant{\chemfig{*6(--(-Br)----)}}
 \chemand
 \reactant{\ce{HBr}}
\end{rxn}
```



The optional arguments for \chemand and \reactant are the same, see section 4.13 for \reactant's options.

4.5 dummy

NEW \dummy creates an empty node. Up to v1.3 \arrow needed to follow after a node, because \arrow internally uses \tikzchainprevious. If there was no node on the chain *before* \arrow was used, it caused an error. With \branch it was similar. By using \dummy you could start a scheme with an arrow anyway.

```
\begin{rxn}
```

```
  \dummy\arrow{}{}
\end{rxn}
```

This is *not* necessary any more. In some (alignment) cases, an empty node still might be useful, so the command still exists.

4.6 elmove

\elmove just is a shortcut for ChemFig's \chemmove.

```
\elmove[<tikz>]{<start>}{<start direction>}{<end>}{<end
    direction>}
```

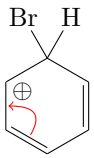
This is expanding the command

```
\chemmove{\draw[<tikz>](<start>).. controls +(<start
    direction>) and +(<end direction>)..(<end>);}
```

using `[->,red,shorten <=3pt,shorten >=1pt]` as default for `<tikz>`. How you use \chemmove is described in the documentation for ChemFig.

Example 40

```
\begin{center}
 \setatomsep{1.8em}
 \chemfig{*6(=[@{e1}]-=-(-[:120]Br)(-[:60]H)-(-[:-30,.4,,,white]\oplus)
    -[@{e2}])}
 \elmove{e1}{60:4mm}{e2}{0:4mm}
\end{center}
```



You should use \elmove *only inside* of \anywhere, \reactant or \transition. Otherwise, you might get alignment errors.

4.7 makeinvisible

\makeinvisible restores the normal myChemistry behaviour after \makevisible (see section 4.8) has been used. \makeinvisible only changes the looks of nodes following after it.

4.8 makevisible

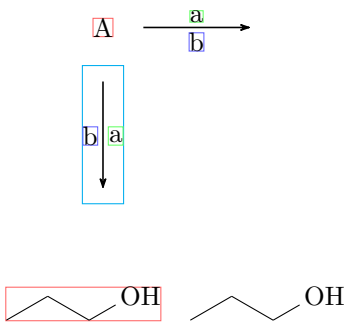
With \makevisible you can visualize the nodes within which reactants, arrows and branches are set. This is useful when you're aligning branches, for example. You can see an example for \makevisible in section 4.2. Every kind of nodes is emphasized with a different colour:

`\reactant{}`, `\arrow{above}{}`, `\arrow{}{below}` und `\branch{}`. Also see section 4.7.

`\makevisible` only changes the looks of nodes following after it ond only effects the group whithin which it is used.

Example 41

```
\begin{rxn}
 \makevisible
 \reactant[,a]{A}
 \arrow{a}{b}
 \branch[below=of a]{
   \arrow[-90,,,,both]{a}{b}
 }
\end{rxn}
\begin{rxn}
 {\makevisible
 \reactant{\chemfig
    {[:30]--[::-60]-OH}}
 }
 \reactant{\chemfig
    {[:30]--[::-60]-OH}}
\end{rxn}
```



4.9 marrow

`\marrow` creates a double-headed arrow.

```
\marrow[<direction>]
```

It is a shortcut for `\arrow[<direction>,<->,.5]{}{}`.

4.10 mCsetup

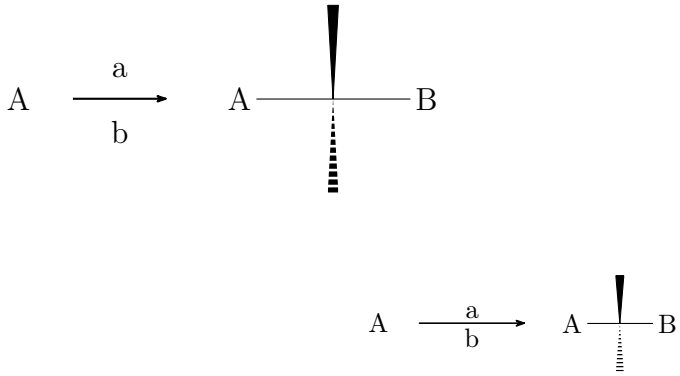
With

```
\mCsetup{<keys>}
```

you can fully customize myChemistry. For each of myChemistry's commands going like `\set<command>` except `\setbondshape` there is a key `<command>=<value>`. Additionally there is the key `align=<value>`, with which you can change the alignment behaviour of both `rxn` and `rxnscheme`, and the key `reset`, with whith all values are reset to default.

Example 42

```
\mCsetup{
 align=left,
 arrowlabel=.7em,
 arrowlength=3em,
 arrowline=thick,
 atomsize=\large,
 bondlength=3em,
 mergelength=4em,
 rcndist=2em
 %rxnalign=right,
 %schemealign=left
}
\setbondshape{4pt}{2pt}{1pt}
\begin{rxn}
 \reactant{A}\arrow{a}{b}\reactant{\chemfig{A-(<[2])(<:[6])-B}}
\end{rxn}
\mCsetup{reset}
\begin{rxn}
 \reactant{A}\arrow{a}{b}\reactant{\chemfig{A-(<[2])(<:[6])-B}}
\end{rxn}
```



\mCsetup as well as for every other setup command only effects the group whithin which it is used.

4.11 merge

NEW

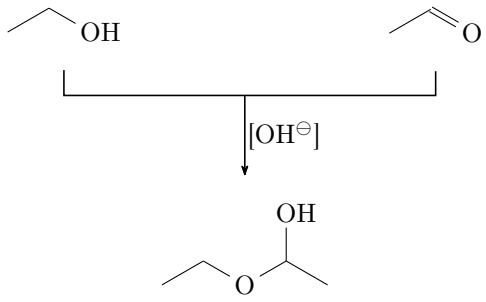
The merge command is used to merge different reaction chains. In order to do that, the reactants that are to be merged must have an anchor name (\reactant[,<anchor>]{}, similar with branches, see section 4.13 and section 4.3) and be placed *before* \merge.

```
\merge[<label>,<direction>,<length>]{<target>}{<start a
   >}{<start b>}
```

\merge has three optional and three mandatory arguments. The latter are the anchor names of the reactants, that are to be merged.

Example 43

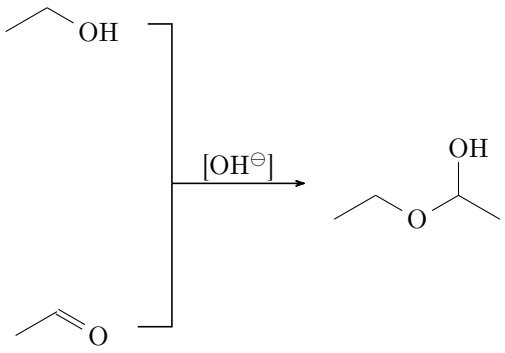
```
\begin{rxn}
 \branch[,first]{
  \reactant[,start_a]{\chemfig{-[:30]-[:-30]OH}}
  \reactant[,start_b,xshift=9em]{\chemfig{-[:30]=[:-30]O}}
 }
 \branch[below=of first,target,yshift=-5em]{
  \reactant{\chemfig{-[:30]-[:-30]O-[:30](-[2]OH)-[:-30]}}
 }
 \merge[\ce{[\Hyd]}]{target}{start_a}{start_b}
\end{rxn}
```



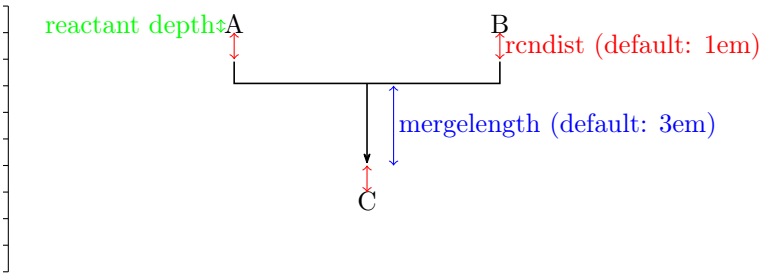
The default <direction> is below, other possible values are right, left or above . With <length> the length of the arrow from the point of merging to the tip can be changed. The default length is 3 em. The default length can be changed with \setmergelength or \mCsetup.

Example 44

```
\begin{rxn}
 \reactant[,start_a]{\chemfig{-[:30]-[:-30]OH}}
 \branch[below=of start_a,start_b,yshift=-9em]{
  \reactant{\chemfig{-[:30]=[:-30]O}}
 }
 \branch[right=of start_a,target,xshift=7em,yshift=-6em]{
  \reactant{\chemfig{-[:30]-[:-30]O-[:30](-[2]OH)-[:-30]}}
 }
 \merge[\ce{[\Hyd]},right,5em]{target}{start_a}{start_b}
\end{rxn}
```



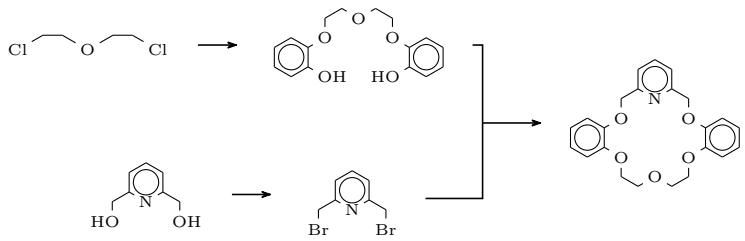
Since you have to place the reactants first, it might be useful to know a little bit about involved lengths. There are three values, which influence the needed distance between start points and the target reactant. There is the depth of the involved reactants, the distance of the "reaction nodes" rcndist (see section 4.23) and the length of the `\merge` arrow mergelength (also see section 4.22).



At last an example where two chains are merged.

Example 45

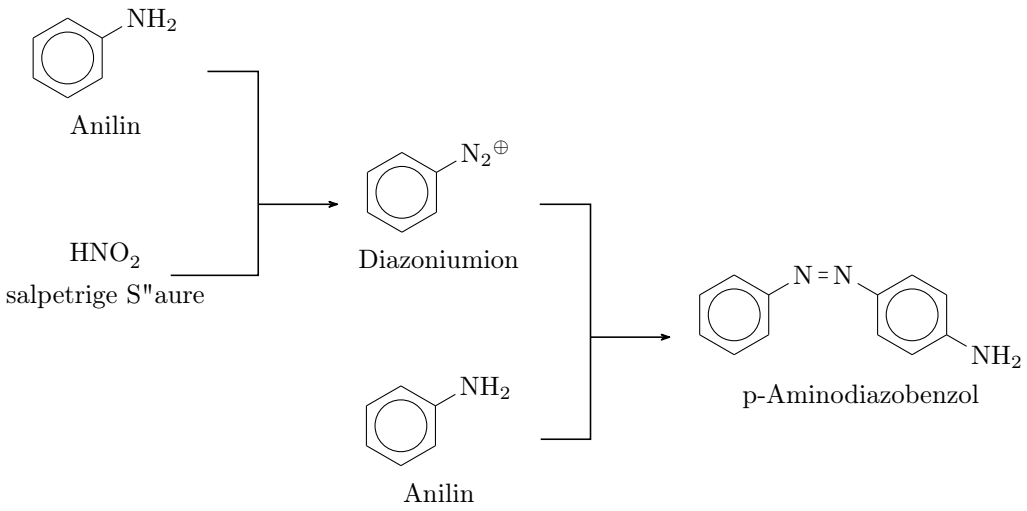
```
\begin{rxn}
 \setatomsep{1em}\tiny
 % chain 1
  \reactant[,first]{ \chemfig{Cl-[:30,1.5]--[:-30,1.5]O
    -[:30,1.5]--[:-30,1.5]Cl}{} }
  \arrow[,,.5]{}{}
  \reactant[,start_above]{ \chemfig{O(-[:-150]**6(-----(-OH)-))
    -[:90]-[:30]-[:-30]O-[:30]-[:-30]-[:-90]O-[:-30]**6(-(-HO)-----)} }
 % chain 2
 \branch[below=of first,start_below,xshift=8em,yshift=-4em]{
  \reactant{ \chemfig{**6((--[6,,,2]HO)-N-(--[6]OH)----)} }
  \arrow[,,.5]{}{}
  \reactant{ \chemfig{**6((--[6]Br)-N-(--[6]Br)----)} }
 }
 % target
 \branch[right=of start_above,target,xshift=5em,yshift=-4em]{
  \reactant[,c]{ \chemfig{O(-[:-150]**6(-----(-O?)-))-[:90]-[:30]**6(-N
    -(--[:-90]O-[:-30]**6(-(-O-[6]-[:-150]-[:150]O-[:-150]-[:150]?)-----))
    ----)} }
 }
 % merging:
 \merge[,right]{target}{start_above}{start_below}
\end{rxn}
```



The usage of \merge may require some playing with branches, `xshift` and `yshift`, until you get the desired result.

Example 46

```
\begin{rxn}
 \setatomsep{1.5em}
 \reactant[,start_aa]{ \chemname{\chemfig{**6(---(-NH_2)---)}}{Anilin}
    }
 \reactant[below,start_ab,yshift=-3em]{ \chemname{\ce{HNO2}}{salpetrige
     S"aure} }
 \branch[right=of start_aa,target_a,xshift=6em,yshift=-5em]{
   \reactant{ \chemname{\chemfig{**6(---(-N|_2\op)---)}}{Diazoniumion}
    }
 }% = start_ba
 \branch[below=of target_a,start_bb,yshift=-3em]{
   \reactant{ \chemname{\chemfig{**6(---(-NH_2)---)}}{Anilin} }
 }
 \branch[right=of target_a,target_b,xshift=6em,yshift=-5em]{
   \reactant{ \chemname{\chemfig{N(-[:-150]**6(------))=N
    -[:-30]**6(---(-NH_2)---)}}{p-Aminodiazobenzol} }
 }
 \merge[,right]{target_a}{start_aa}{start_ab}
 \merge[,right]{target_b}{target_a}{start_bb}
\end{rxn}
```



4.12 mesomeric

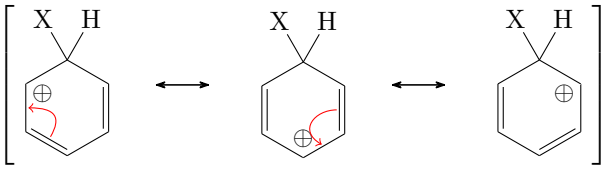
The \mesomeric command works just like \branch (see section 4.3) but places the formulæ into square brackets.

```
\mesomeric[<pos>,<anchor>,<tikz>]{<formula>}
```

The resonance formulæ are written into <formula>. With \marrow (see section 4.9) you create the resonance arrows. If needed you can give an anchor (<anchor>) to \mesomeric (also see section 4.3). Alignment is used the same way as with \reactant.

Example 47

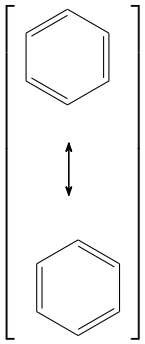
```
\begin{rxn}
 \mesomeric{
   \reactant{
     \chemfig{*6(=[@{e1}]-=-(-[:120]X)(-[:60]H)-(-[:-30,.4,,,white]\
    oplus)-[@{e2}])}
     \elmove{e1}{60:4mm}{e2}{0:4mm}
   }
   \marrow
   \reactant{
     \chemfig{*6(-(-[:90,.4,,,white]\oplus)-[@{e4}]=[@{e3}]-(-[:120]X)
    (-[:60]H)-=)}
     \elmove{e3}{180:4mm}{e4}{150:4mm}
   }
   \marrow
   \reactant{
     \chemfig{*6(-=-(-[:-150,.4,,,white]\oplus)-(-[:120]X)(-[:60]H)-=)}
   }
  }
\end{rxn}
```



Or vertical, too:

Example 48

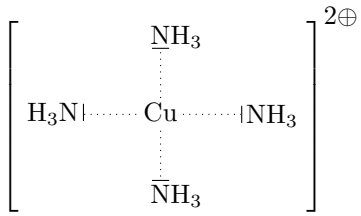
```
\begin{rxn}
 \mesomeric{
   \reactant{ \chemfig
    {*6(=-=-=-)} }
   \marrow[below]
   \reactant[below]{ \chemfig
    {*6(-=-=-=)} }
 }
\end{rxn}
```



Or maybe a coordination complex?

Example 49

```
\begin{rxn}
 \setatomsep{3em}
 \mesomeric[,a]{
   \reactant{ \chemfig{H_3\lewis{0,N}-[,1.35,,,dotted]{Cu}(-[2,,,,
    dotted]\lewis{6,N}H_3)(-[6,,,,dotted]\lewis{2,N}H_3)-[,1.2,,,dotted]\
    lewis{4,N}H_3} }
 }
 \node[above right=of a,yshift=-1em] {$2\oplus$};
\end{rxn}
```



4.13 reactant

\reactant[1] is something like the basic command.

```
\reactant[<pos>,<name>,<tikz>]{<formula>}
```

Formulæ (<formula>) are written inside this command and can be named (<name>) for other objects to refer to. <pos> can either be one of these key words

a) right,

b) above right,

c) above,

d) above left,

e) left,

f) below left,

g) below,

h) below right

NEW or be an angle from the intervall $[-360°; 360°]$. You also can position it relativ to another object via <anchor>.<angle>. The default equals \reactant[0]{}.

[1] In earlier versions it was called \reactand. You can still use it.

Example 50

```
below:
\begin{rxn}
  \reactant{\ce{Br2}}
  \reactant[-90]{\ce{Cl2}}
\end{rxn}

more than one reactant:
\begin{rxn}
  \reactant{\ce{Br2}}
  \reactant[-90]{\ce{I2}}
  \reactant{\ce{Cl2}}
\end{rxn}

going down:
\begin{rxn}
 \reactant{\ce{Br-Br}}
 \arrow[-90,,.5]{$h\nu$}{}
 \reactant[-90]{\ce{2 ~\lewis
   {0.,Br}}}
\end{rxn}
```

below:

$$\mathrm{Br_2}$$

$$\mathrm{Cl_2}$$

more than one reactant:

$$\mathrm{Br_2}$$

$$\mathrm{I_2} \quad \mathrm{Cl_2}$$

going down:

$$\mathrm{Br{-}Br} \downarrow h\nu \; 2\ \mathrm{Br^{\bullet}}$$

You can find many more examples in the file `examples.tex` or `examples.pdf`, resp.

4.14 rxn (environment)

`rxn` is a non-floating not numbered environment for reaction schemes. Per default all schemes are centered.

```
\begin{rxn}[<alignment>,<scalefactor>]
  ...
\end{rxn}
```

4.14.1 Options

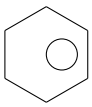
NEW `rxn` has two options, which are to be used in the following order, separated by a comma:

1. `<alignment>` alignment behaviour of the `rxn` environment; default is 'center'.
2. `<scalefactor>` factor by which the `rxn` environment is scaled; default: '1.0'. Same behaviour as with `rxnscheme`, see section 4.15.

If you use the `scalefactor` option, you might see strange effects on ChemFig formulæ.

Example 51

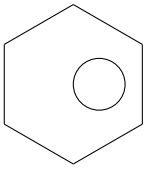
```
\begin{rxn}[,.5]
 \reactant{\chemfig{**6(------)
    }}
\end{rxn}
```



Scaling does in general not affect the size of ChemFig formulæ, but does scale the aromaticity ring of benzene and similar molecules. This is due to a possible bug in ChemFig itself.

Example 52

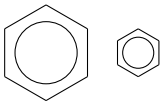
```
\chemfig[scale=.5]{**6(------)}
```



This can be solved either by using the *first* optional argument of \chemfig to undo the scaling or by using the *second* optional argument to scale the rest of the molecule.

Example 53

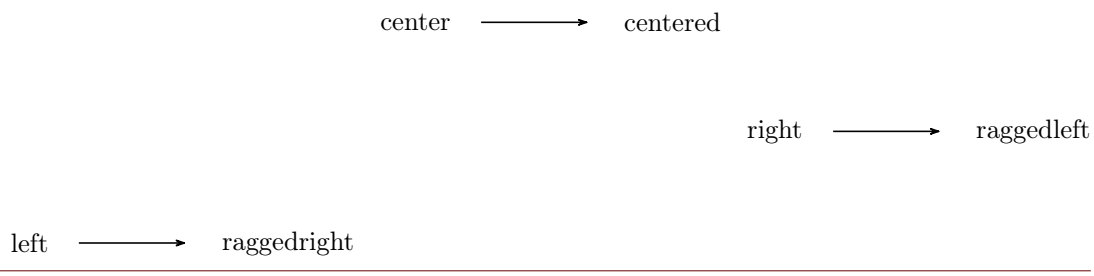
```
\begin{rxn}[,.5]
 \reactant{\chemfig[scale=2]{**6(------)}}
 \reactant{\chemfig[][scale=.5]{**6(------)}}
\end{rxn}
\chemfig[scale=.5][scale=.5]{**6(------)}
```



Alignment examples:

Example 54

```
\begin{rxn}[center]
 \reactant{center}\arrow{}{}\reactant{centered}
\end{rxn}
\begin{rxn}[right]
 \reactant{right}\arrow{}{}\reactant{raggedleft}
\end{rxn}
\begin{rxn}[left]
 \reactant{left}\arrow{}{}\reactant{raggedright}
\end{rxn}
```



4.15 rxnscheme (environment)

rxnscheme is a floating environment for reaction schemes.

```
\begin{rxnscheme}[<label>,<placement>,<alignment>,<
   scalefactor>,<name>]{<caption>}
 ...
\end{rxnscheme}
```

4.15.1 Options

NEW rxnscheme has five options, which are to be used in the following order, separated by commas:

1. <label> Like every other floating environment rxnscheme can be given a label. To do that, you need to use the option <label>. For example if you use

   ```
   \begin{rxnscheme}[rs:schema]{<caption>}
    ...
   \end{rxnscheme}
   ```

   you can refer to it by using \ref{rs:schema} as usual.

2. <placement> With this option you can change the placement of the float, e. g. with htp. The default value is htp.

3. <alignment>This option changes the alignment of the scheme. You can choose between `left`, `center` and `right`.

4. <scalefactor> `rxnscheme` has another key with which the scheme can be scaled. Please keep in mind that it doesn't affect the font size and the size of ChemFig formulæ. You can have strange effects on ChemFig formulæ if you use this key, though. See section 4.14.1 for more information.

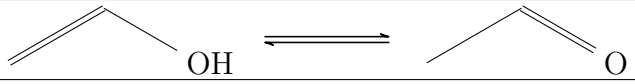
5. <name> This option changes the name of the actual scheme from "Reaktionschema" or "Reaction scheme" into <name>.

```
\begin{rxnscheme}[,<placement>]{<caption>}
  ...
\end{rxnscheme}
```

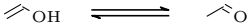
Example 55

```
\begin{rxnscheme}[,H,,2]{Big scheme}
 \large\setatomsep{3.5em}
 \reactant{ \chemfig{=[::30]-[::-60]OH} }
 \arrow[,<=>]{}{}
 \reactant{ \chemfig{-[::30]=[::-60]O} }
\end{rxnscheme}
\begin{rxnscheme}[,H,,.5]{Small scheme}
 \tiny\setatomsep{1em}
 \reactant{ \chemfig{=[::30]-[::-60]OH} }
 \arrow[,<=>]{}{}
 \reactant{ \chemfig{-[::30]=[::-60]O} }
\end{rxnscheme}
\begin{rxnscheme}[,H]{center}
 \reactant{center}\arrow{}{}\reactant{centered}
\end{rxnscheme}
\begin{rxnscheme}[,H,right]{right}
 \reactant{right}\arrow{}{}\reactant{raggedleft}
\end{rxnscheme}
\begin{rxnscheme}[,H,left]{left}
 \reactant{left}\arrow{}{}\reactant{raggedright}
\end{rxnscheme}
```

Reaction scheme 2 Big scheme



Reaction scheme 3 Small scheme



Reaction scheme 4 center

center ⟶ centered

Reaction scheme 5 right

right ⟶ raggedleft

Reaction scheme 6 left

left ⟶ raggedright

4.15.2 Customizing rxnscheme

Style If you don't like the style of `rxnscheme` you can change it by using

```
\floatstyle{<new style>}
\restylefloat{rxnfloat}
```

There are different possible styles, privided by the 'float' package:

`plain` without any special formatting, the caption is below the object

`plaintop` like `plain`, but the caption is placed above the object

`boxed` the object is boxed, the caption placed below

`ruled` the caption is placed above the object framed by to rules, one above and one below, another rule frames the object below; default for `rxnscheme`

Example 56

```
\begin{rxnscheme}[,H]{ruled}
 \reactant{default style}
\end{rxnscheme}
\floatstyle{boxed}
\restylefloat{rxnfloat}
\begin{rxnscheme}[,H]{boxed}
 \reactant{framed object}
\end{rxnscheme}
\floatstyle{plain}
\restylefloat{rxnfloat}
\begin{rxnscheme}[,H]{plain}
 \reactant{without any special formatting}
\end{rxnscheme}
```

Reaction scheme 7 ruled

default style

framed object

Reaction scheme 8: boxed

without any special formatting

Reaction scheme 9: plain

Placement Usually floating environments have an optional argument for their placement. `rxnscheme`'s default placement is `H` which means, it is placed *exactly here*. If you want to change it into `htp` or something, you can use

```
\floatplacement{rxnfloat}{<placement>}
```

It's easier, though, loading myChemistry with the 'placement' option:

```
\usepackage[placement=<placement>]{mychemistry}
```

This will change the default placement behaviour from `H` to `<placement>`. You can also change the placement behaviour of just one `rxnscheme` environment by using the placement option:

```
\begin{rxnscheme}[,<placement>]{<caption>}
  ...
\end{rxnscheme}
```

Name If you want to change the name of reaction scheme[1], you can do that with

[1] You probably do. You reading the English documention means probably, that you're not German.

```
\setschemename{<new name>}
```

The default name is "Reaktionschema" or, with package option ‘english’, "Reaction scheme".

Counter The counter can be changed just as usual. For example by using

```
\makeatletter
\@addtoreset{rxnfloat}{section}
\makeatletter
\renewcommand{\therxnscheme}{\arabic{section}.\arabic{
   rxnscheme}}
```

the counter is reset with every new section an looks like `section.rxnscheme`. Please be aware, that you have to write `\@addtoreset` between `\makeatletter` and `\makeatother` because of the @.

List of schemes By writing

```
\listof{rxnfloat}{<title>}
```

you can create a list of all schemes created with `rxnscheme`.

Example 57

```
\listof{rxnfloat}{Reaction
    schemes}
```

Reaction schemes

4.16 setarrowlabel

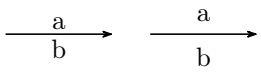
NEW The distance of arrow labels to the arrow has a default value of 0.2 em. You can change this with

```
\setarrowlabel{<distance>}
```

into `<distance>`.

Example 58

```
\begin{rxn}
 \arrow{a}{b}
 \setarrowlabel{.5em}
 \arrow{a}{b}
\end{rxn}
```



4.17 setarrowlength

The default length of an reaction arrow is 4 em. You can change this value with

```
\setarrowlength{<length>}
```

into `<length>`. Mind the fact that you have to use a length unit. If you leave the argument empty, the length is reset to default.

4.18 setatomsize

With

```
\setatomsize{<font size>}
```

you can change the font size of the atom groups. Default value is `\small`. If you leave the argument empty, the size is reset to default.

4.19 setbondlength

With

```
\setbondlength{<length>}
```

you can change `\setatomsep{<length>}` for all ChemFig formulæ *inside* of the myChemistry environments. Default value is 1.8 em. If you leave the argument empty, the length is reset to default.

4.20 setarrowline

NEW With the command

```
\setarrowline{<value>}
```

the thickness of the arrows can be customized. Possible values are

- `ultra thin`
- `very thin`
- `thin`
- `semithick` (default)
- `thick`
- `very thick`
- `ultra thick`

The values `very thick` and `ultra thick` should not be used.
This command also applies to \merge.

4.21 setbondshape

With

```
\setbondshape{<base length>}{<dash thickness>}{<dash spacing>}
```

you can change \setcrambond{<base length>}{<dash thickness>}{<dash spacing>} for all ChemFig formulæ *inside* of the myChemistry environments. Default values are (in this order) 3 pt, 0.5 pt and 1 pt. If you leave an argument empty, the value is reset to default.

4.22 setmergelength

NEW With

```
\setmergelength{<länge>}
```

you can change the length of the \merge arrow. More precisely you can change the length of the arrow from the point of line crossing to the arrow tip (see section 4.11). If you leave an argument empty, the value is reset to default (3 em).

4.23 setrcndist

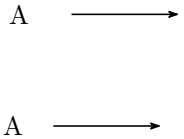
The nodes within which the reactants an arrows are set have a certain distance between them. The default distance is 1 em. If you want to change that, you can use

```
\setrcndist{<länge>}
```

If you leave the argument empty, the distance is reset to 1 em.

Example 59

```
\setrcndist{2em}
\begin{rxn}
 \reactant{A}\arrow{}{}
\end{rxn}
\setrcndist{}
\begin{rxn}
 \reactant{A}\arrow{}{}
\end{rxn}
```



4.24 setrxnalign/setschemealign

With the commands

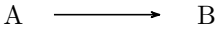
```
\setrxnalign{<alignment>}
\setschemealign{<alignment>}
```

The default alignment behaviour of `rxn` and `rxnscheme` (see section 4.14.1 & section 4.15.1) can be set. You can choose between `left`, `center` and `right`.

If you leave the argument empty, myChemistry's default behaviour (`center`) is restored.

Example 60

```
\setrxnalign{right}
\begin{rxn}
 \reactant{A}\arrow{}{}\reactant{B}
\end{rxn}
\setrxnalign{}
\begin{rxn}
 \reactant{A}\arrow{}{}\reactant{B}
\end{rxn}
```



4.25 setschemename

See section 4.15.2.

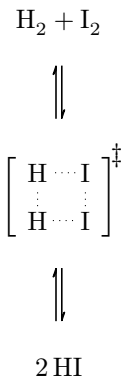
4.26 transition

`\transition` works exactly like `\reactant` (see section 4.13).

```
\transition[<pos>,<anchor>,<tikz>]{<formula>}
```

Example 61

```
\begin{rxn}
 \reactant{ \ce{H2 + I2} }
 \arrow[below,<=>,.5]{}{}
 \transition[below]{ \chemfig[
        dotted][]{H?-I-[2]I-[4]H?} }
 \arrow[below,<=>,.5]{}{}
 \reactant[below]{ \ce{2 HI} }
\end{rxn}
```



5 Epilogue

myChemistry is still very new. This means there are probably a number of bugs I haven't discovered yet. There also might be missing one or two features, that would be useful. Since I only can test and work on myChemistry in my spare time, I'd be very glad about *every* kind of feedback. If you like myChemistry, why don't you help me improve it by telling me your experiences?

I tried using real chemical reactions but I didn't make sure, that they all make sense chemically. So you shouldn't trust the examples in respect to chemistry but rather take a look into a real chemistry teaching book.

I apologize for any bad or wrong English. I hope you understood the documention anyway.

Have fun with myChemistry!

Clemens Niederberger, Berlin, April 27th 2011

6 Thanks

I owe thanks for bug reports and suggestions to:

F. Chervet, Ferghun, V. Garibal and C. Tellechea (who pointed out some essential issues).

Index