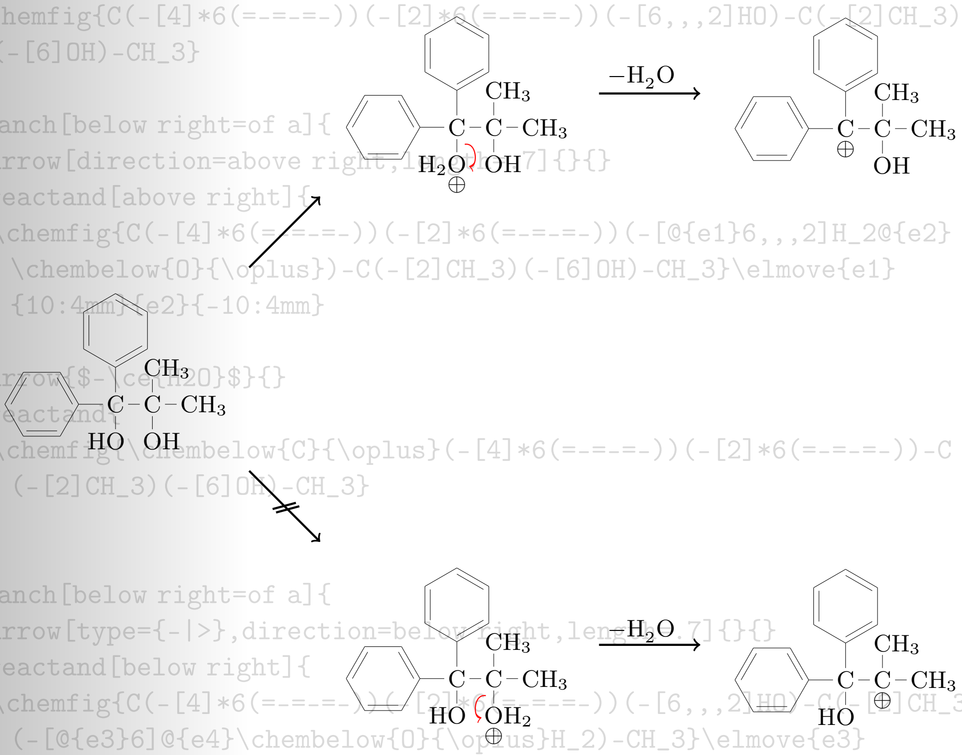
myChemistry

v1.3

April 4th 2011

Clemens Niederberger

Creating reaction schemes with LaTeX and ChemFig



```
\begin{rxn}[scale=.7]
 \setatomsep{1.5em}\footnotesize
 \reactand[,a]{
  \chemfig{C(-[4]*6(=-=-=-))(-[2]*6(=-=-=-))(-[6,,,2]HO)-C(-[2]CH_3)
   (-[6]OH)-CH_3}
 }
 \branch[below right=of a]{
  \arrow[direction=above right,length=.7]{}{}
  \reactand[above right]{
   \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{$-\ce{H2O}$}{}
  \reactand{
   \chemfig{\chembelow{C}{\oplus}(-[4]*6(=-=-=-))(-[2]*6(=-=-=-))-C
    (-[2]CH_3)(-[6]OH)-CH_3}
  }
 }
 \branch[below right=of a]{
  \arrow[type={-|>},direction=below right,length=.7]{}{}
  \reactand[below right]{
   \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{$-\ce{H2O}$}{}
  \reactand{
   \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 new features or changes of each version are marked with V followed by the version number, e. g.V1.2. The most recent changes are marked with New in V1.3.

1.1.1 Version 1.2

V1.2 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 4.1).

The environments have gained a few features, with which they can be customized, too (see section 4.13.1, section 4.14.1 and section 4.19).

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

1.1.2 Version 1.3

New in V1.3 The commands `\branch`, `\mesomeric`, `\reactand` 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[<alignment>,<anchor>,<tikz>]{}
```

Version 1.2 explicitly needed a given alignment, if one wanted to use Ti*k*Z keys, even with default alignment. Since v1.3 this isn't required any more.

```
% up to version 1.2:
\reactand{\ce{Br2}}{}\arrow{$h\nu$}{}\reactand[right,draw,inner sep=5pt]{\ce{2 \lewis{0.,Br}}}{}
% since version 1.3:
\reactand{\ce{Br2}}\arrow{$h\nu$}{}\reactand[,,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 4.15, section 4.16, section 4.17, section 4.18 and section 4.9.

There is the new command `\chemand`, which produces a +, see section 4.3.

If you're using ChemFig v0.4, myChemistry inputs the file `bondwith.tex` which provides the command `\setbondwidth{<width>}` with which you can change the line thickness of bonds.

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

1.2 Licence

myChemistry v1.3 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/Ti*k*Z pgf actually isn't just one package but a whole bundle. They are the basis layer for Ti*k*Z. 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 4.1) needs `\pgfpositionnodelater`. If you don't use this option, myChemistry should work nicely with pgf v2.00. Older versions have not been tested.

[1] `http://sourceforge.net/projects/pgf/files/`

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[1],
- 'mhchem'[2] in version 3, when ist exists,
- 'chemexec'[3], when it exists, and
- 'chemcompounds'[4], 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.

[1] by Christian Tellechea, http://www.ctan.org/tex-archive/macros/generic/chemfig/

[2] by Martin Hensel, http://www.ctan.org/tex-archive/macros/latex/contrib/mhchem/

[3] by me, http://www.ctan.org/tex-archive/macros/latex/contrib/chemexec/

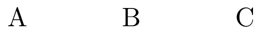
[4] by Stephan Schenk, http://www.ctan.org/tex-archive/macros/latex/contrib/chemcompounds/

2.2 Basic Principle

Within the tikzpicture reactands and arrows are placed as nodes on a chain[1].

Example 1

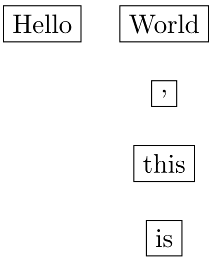
```
\begin{tikzpicture}[start chain]
\node [on chain] {A};
\node [on chain] {B};
\node [on chain] {C};
\end{tikzpicture}
```



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

Example 2

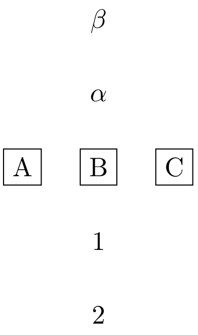
```
\begin{tikzpicture}[start chain=going right,node distance=5mm]
\node [draw,on chain] {Hello};
\node [draw,on chain] {World};
\node [draw,continue chain=going below,on chain] {,};
\node [draw,on chain] {this};
\node [draw,on chain] {is};
\end{tikzpicture}
```



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}
```



[1] Provided by the tikzlibrary ‘chains’

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 4.7). This is done so one can see, which size they have and which impact changes of the alignment have on them.

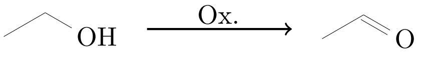
2.3 How does it work?

2.3.1 Basic Commands

Let's take a look at an example first:

Example 4

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



In this example you see the most important commands of myChemistry:

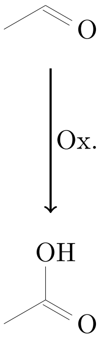
`\begin{rxn}[<keys>]` The first of two environments. It creates a centered reaction scheme between two paragraphs (see section 4.13).

`\reactand[<alignment>,<anchor>,<tikz>]{<formula>}` places a `node` on the `chain`, in which the chemical formulæ are written. The default alignment is to the `right` (see section 4.12).

`\arrow[<keys>]{<above>}{<below>}` creates an arrow, which by default has a length of 5 em pointing to the right (see section 4.1).

Example 5

```
\begin{rxn}
  \reactand{ \chemfig
     {-[::30]=_[::-60]O} }
  \arrow[direction=below]{}{Ox.}
  \reactand[below]{ \chemfig
     {-[::30](-[::60]OH)=_[::-60]O} }
\end{rxn}
```



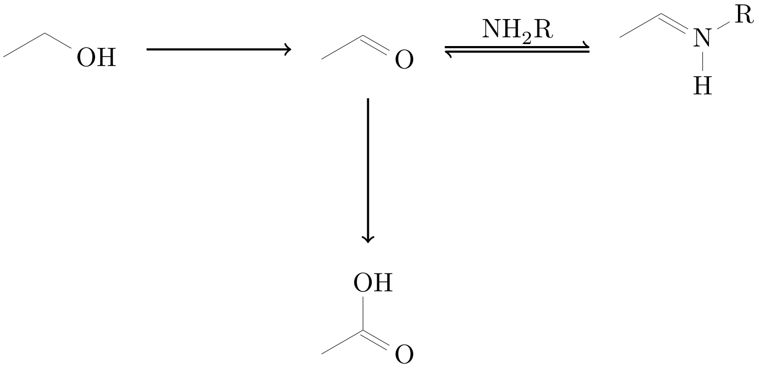
As you can see one can align the reaction scheme vertical as well. By giving \reactand the option `below`, the carbonic acid is placed below the arrow. With `direction=below` the arrow points down instead to the right.

2.3.2 Branches

With what we saw until now it is not yet clear, why one would use myChemistry. the horizontal reaction schemes can be realized with ChemFig and 'mhchem' just as well. And why would anyone need a vertical reaction scheme? But what could make myChemistry interesting is the possibility of branched reaction schemes.

Example 6

```
\begin{rxn}
 \reactand{ \chemfig{-[::30]-[::-60]OH} }
 \arrow{}{}
 \reactand[,carbonyl]{ \chemfig{-[::30]=_[::-60]O} }
 \arrow[direction=below]{}{}
 \reactand[below]{ \chemfig{-[::30](-[::60]OH)=_[::-60]O} }
 \branch[right=of carbonyl]{
   \arrow[type={<=>}]{\ce{NH2R}}{}
   \reactand{ \chemfig{-[::30]=_[::-60]N(-[6]H)-[::60]R} }
 }
\end{rxn}
```



In the last example you've got to know another important command:

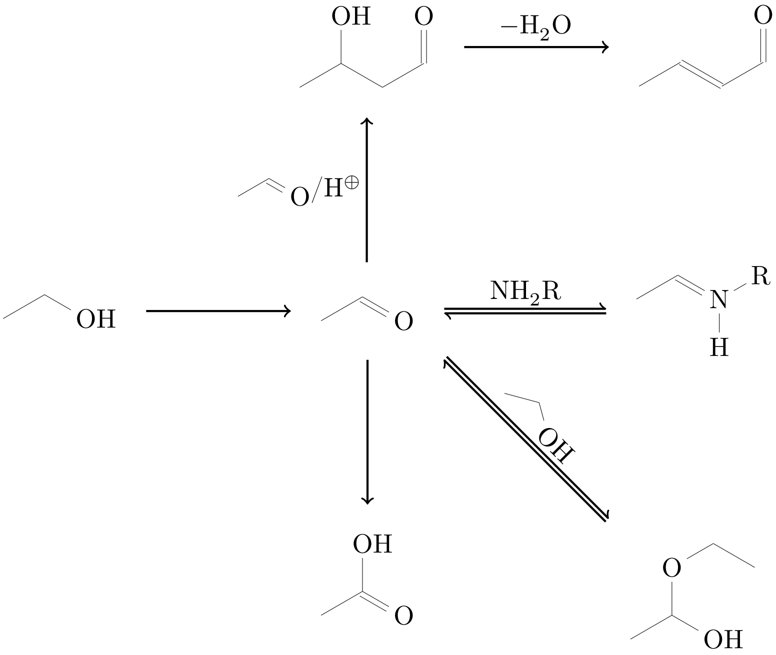
`\branch[<alignment>,<anchor>,<tikz>]{<branch commands>}` (see section 4.2)

The branch was placed right of the first reactand with the anchor `carbonyl` by using the option `right=of carbonyl`. Inside the branch we used an arrow with the key `type ={<=>}` so we would get an equilibrium arrow. Other types would for example be `->` (default), `<-` or `<->`.

Extensive reaction schemes can be realized through multiple usage of \branch:

Example 7

```
\begin{rxn}
 \reactand{ \chemfig{-[::30]-[::-60]OH} }
 \arrow{}{}
 \reactand[,carbonyl]{ \chemfig{-[::30]=_[::-60]O} }
 \arrow[direction=below]{}{}
 \reactand[below]{ \chemfig{-[::30](-[::60]OH)=_[::-60]O} }
 \branch[right=of carbonyl,imin]{
   \arrow[type={<=>},length=1.12]{\ce{NH2R}}{}
   \reactand{ \chemfig{-[::30]=_[::-60]N(-[6]H)-[::60]R} }
 }
 \branch[below right=of carbonyl,halbacetal,yshift=-2pt,xshift=3pt]{
   \arrow[type={<=>},direction=below right,length=1.12,aboveshift=3pt]{
     \chemfig{[,.75]-[::30]-[::-60]OH} }{}
   \reactand[below right]{ \chemfig{-[::30](-[::60]O-[::-60]-[::-60])
    -[::-60]OH} }
 }
 \branch[above=of carbonyl,aldol,xshift=5.2em]{
   \arrow[direction=above]{ \chemfig{[,.75]-[::30]=_[::-60]O}/\Hpl }{}
   \reactand[above]{ \chemfig{-[::30](-[::60]OH)-[::-60]-[::60]=[::60]O
    } }
   \arrow{$-\ce{H2O}$}{}
   \reactand{ \chemfig{-[::30]=[::-60]-[::60]=[::60]O} }
 }
\end{rxn}
```



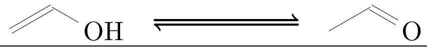
2.3.3 Numbered Schemes

The second environment provided by myChemistry works just as the first one. This time the scheme is placed within a numbered floating environment and is given a caption.

Example 8

```
\begin{rxnscheme}{Keto-enol tautomerization}
 \reactand{ \chemfig{=[::30]-[::-60]OH} }
 \arrow[type={<=>}]{}{}
 \reactand{ \chemfig{-[::30]=[::-60]O} }
\end{rxnscheme}
```

Reaction scheme 1 Keto-enol tautomerization



Here we use the environment

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

In the command reference (section 4.14) you can read how to customize the scheme.

2.4 Predefined Values

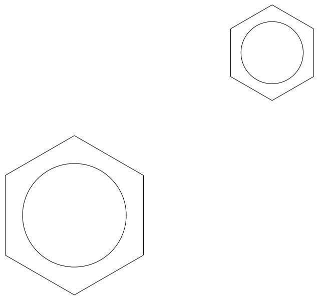
There are some predefined values, that are basically due to my personal taste. But of course you can change them according to your requirements. For ChemFig-formulæ *inside of myChemistry environments* some values are predefined as follows:

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

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

Example 9

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



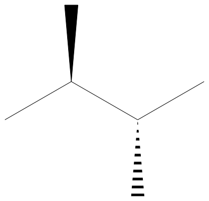
myChemistry's defaults can be changed with these commands:

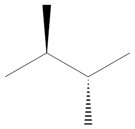
```
\setbondlength{<length>}
\setbondshape{<base length>}{<dash thickness>}{<dash spacing>}
\setatomsize{<font size>}
```

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 10

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

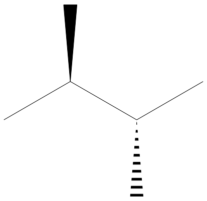


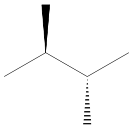


If you only want to change the parameters of a single environment you can use ChemFig's commands and LaTeX's fontsize commands *inside the environment.*

Example 11

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





The default length of an arrow is 5 em or $5\sqrt{2}$ em if it's a diagonal one. You can change that by using the following command:

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

Then the values are `<length>` or `<length>` $\cdot \sqrt{2}$ respectively.

2.5 Package options

myChemistry has a number of package options.

V1.2 `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 4.14) 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).

V1.2 `nocompounds` prevent myChemistry from loading 'chemcompounds'.

V1.2 `nomhchem` prevent myChemistry from loading 'mhchem' lädt, if 'chemexec' isn't loaded either.

V1.2 `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.

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

3 Advanced Usage, Usage with Ti*k*Z

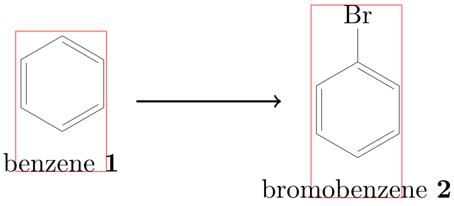
Most of the commands allow alignment with Ti*k*Z code. Using Ti*k*Z code you have much more possibilities aligning reactands relatively to one another. If you are familiar with Ti*k*Z you have even more flexibility, of course (see section 5.3).

3.1 The Alignment Question

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

Example 12

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



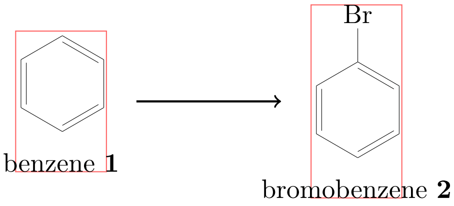
benzene **1**

bromobenzene **2**

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

Example 13

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



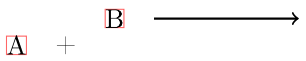
benzene **1**

bromobenzene **2**

This is, because the first reactand 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 14

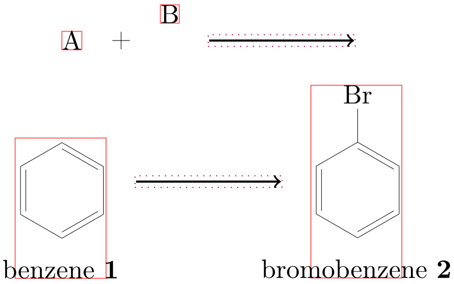
```
\makevisible
\begin{rxn}
 \reactand{A}
 \chemand
 \reactand[,,yshift=1em]{B}
 \arrow{}{}
\end{rxn}
```



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 15

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



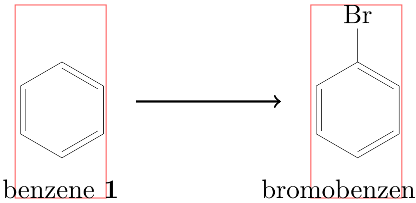
benzene **1**

bromobenzene **2**

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 16

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



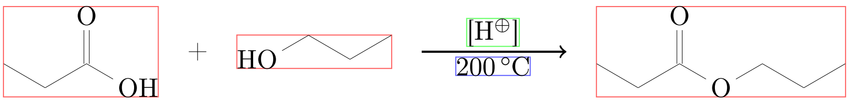
benzene **1** bromobenzene **2**

In other cases, Ti*k*Z code can successfully be used:

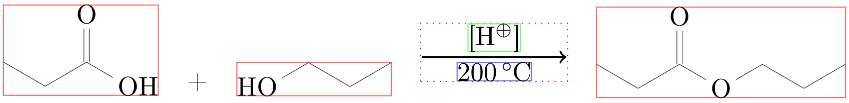
Example 17

```
\makevisible
default:
\begin{rxn}
  \reactand{\chemfig{-[:-30]-[:30](=[2]O)-[:-30]OH}}
  \chemand
  \reactand{\chemfig{HO-[:30]-[:-30]-[:30]}}
  \arrow{[\Hpl]}{\SI{200}{\celsius}}
  \reactand{\chemfig{-[:-30]-[:30](=[2]O)-[:-30]O-[:30]-[:-30]-[:30]}}
\end{rxn}
hydroxy groups at the same height:
\begin{rxn}
  \reactand{\chemfig{-[:-30]-[:30](=[2]O)-[:-30]OH}}
  \chemand[,,yshift=-1.2em]
  \reactand[,,yshift=.12em]{\chemfig{HO-[:30]-[:-30]-[:30]}}
  \branch[,,yshift=1em]{\arrow{[\Hpl]}{\SI{200}{\celsius}}}
  \reactand{\chemfig{-[:-30]-[:30](=[2]O)-[:-30]O-[:30]-[:-30]-[:30]}}
\end{rxn}
```

default:



hydroxy groups at the same height:



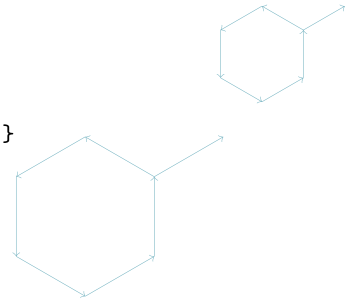
I'm afraid that in many cases you'll have to play with `xshift` and `yshift`, until the scheme looks like you want. Maybe further versions of myChemistry will provide a more user friendly alignment syntax.

3.2 Using TikZ to Achieve Other Results

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

Example 18

```
\begin{rxn}
 \reactand[,,->,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:

Example 19

```
\newcommand{\emptyreactand}{\reactand[,minimum width=5em]{ \rule[-1em]{1em}{.5pt}\;\rule[-1em]{3em}{.5pt} }{}}
\newcommand{\stoich}{\rule[-3pt]{1em}{.5pt}}
\begin{rxn}
 \reactand{\bf\Large Ionic compounds (I)}
 \reactand[below,a,yshift=1em]{Fill in the missing parts}
 \branch[below=of a,b]{ \reactand[,,minimum width=5em]{ \stoich\ Na }\chemand \emptyreactand \arrow{}{} \emptyreactand \chemand \emptyreactand }
 \branch[below=of b,,draw,inner sep=3pt]{\reactand[,,minimum width=5em]{}\chemand\reactand[,,minimum width=5em]{}\arrow{}{}\reactand[,,minimum width=5em]{Sodium chloride $+$ Hydrogen}}
\end{rxn}
```

Ionic compounds (I)

Fill in the missing parts

__ Na + __ ______ ⟶ __ ______ + __ ______

+ ⟶ Sodium chloride + Hydrogen

4 Alphabetical Command Reference

In the following section every command is explained.

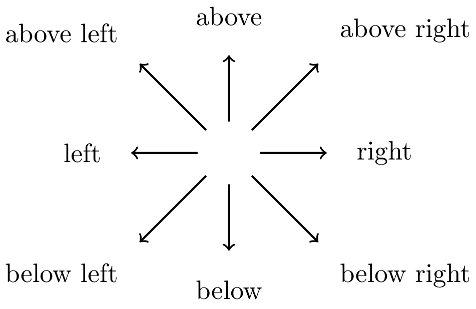
4.1 arrow

Reaction arrows are created with `\arrow`.

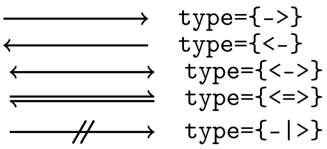
```
\arrow[<keys>]{<above>}{<below>}
```

There are several keys to customize the arrows. They are used like `key=value`.

`direction=<direction>` – possible values are:



`type=<type>` – possible values are:



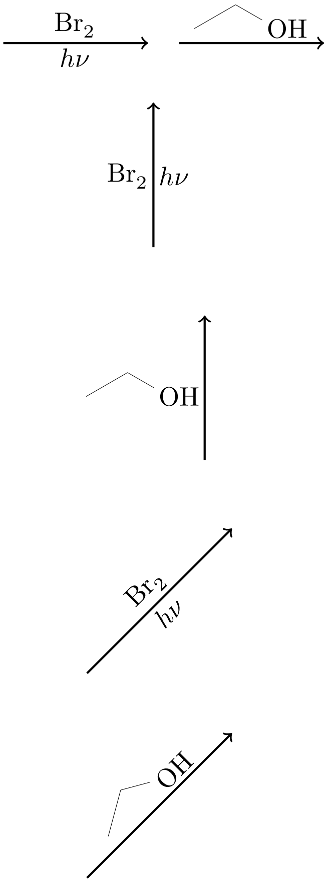
`length=<factor>` – the length (5.0 em, factor = 1.0, default) is multiplied with this factor.

`name=<anchor>` – this anchor can be used to refer to the arrow, e. g. with a branch.

V1.2 `both` – this gives both arrow argument nodes the same width and height.

Example 20

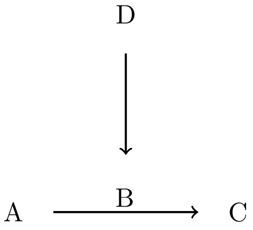
```
\begin{rxn}
 \dummy\arrow{\ce{Br2}}{$h\nu$}\arrow{\chemfig{-[::30]-[::-60]OH}}{}
\end{rxn}
\begin{rxn}
 \dummy\arrow[direction=above]{\ce{Br2}}{$h\nu$}
\end{rxn}
\begin{rxn}
 \dummy\arrow[direction=above]{\chemfig{-[::30]-[::-60]OH}}{}
\end{rxn}
\begin{rxn}
 \dummy\arrow[direction=above right]{\ce{Br2}}{$h\nu$}
\end{rxn}
\begin{rxn}
 \dummy\arrow[direction=above right]{\chemfig{-[::30]-[::-60]OH}}{}
\end{rxn}
```



Most keys in action:

Example 21

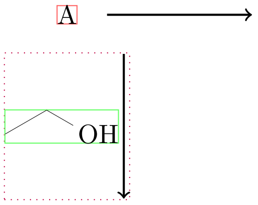
```
\begin{rxn}
 \reactand{A}
 \arrow[name=arrow_a]{B}{}
 \branch[above=of arrow_a,,yshift=-4em]{
   \arrow[type=<-,direction=above,length=.7]{}{}
   \reactand[above]{D}
 }
 \reactand{C}
 \end{rxn}
```



If an arrow is placed inside a branch (see section 4.2) the alignment of the branch possibly is determined by the width and height of the arrow arguments. If these arguments have different sizes, the alignment can go wrong.

Example 22

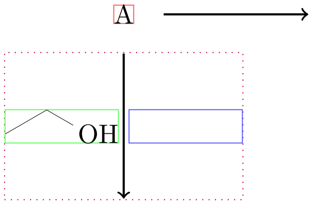
```
\makevisible
\begin{rxn}
 \reactand[,a]{A}
 \arrow{}{}
 \branch[below=of a]{
   \arrow[direction=below]{\
   chemfig{-[::30]-[::-60]OH}}{}
 }
\end{rxn}
\makeinvisible
```



By using the key `both`, both argument nodes have the same size, which can correct the alignment.

Example 23

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



There is more about the alignment problem in section 4.2.1.

4.2 branch

New in V1.3

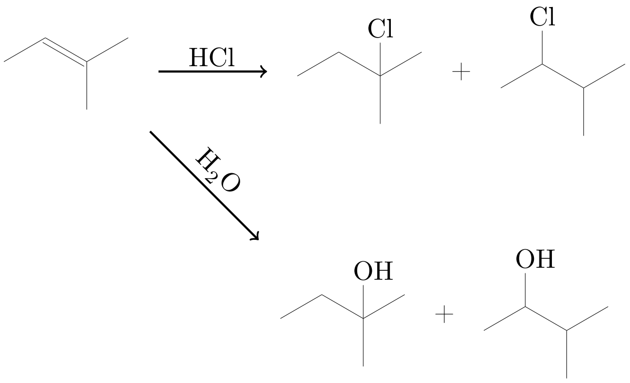
\branch is used to, well, create a branch to a reaction. **If you used earlier versions of myChemistry please be aware, that the command syntax has changed.**

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

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

Example 24

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



The first reactand got the anchor `start` (line 2, also see section 4.12).

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

`\branch` now refers to it in its alignment (line 6):

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

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

Example 25

```
\begin{rxn}
 \reactand{ \chemfig{CH_2=CH-OH} }
 \arrow[type={<=>},length=.5]{}{}
 \branch{ \reactand{ \chemfig{CH_3-CH=O} } }
\end{rxn}
```

$CH_2=CH-OH \rightleftharpoons CH_3-CH=O$

You have several options for alignment: you can either put the \branch on a chain or place it relative to a object.

chain In this case you use: on chain=going <value>.

relative In this case you use: <value>=of <anchor>.

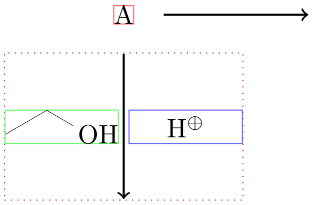
As possible <value> you have the same choices as for \reactand, see section 4.12. Default is on chain=going right.

4.2.1 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 26

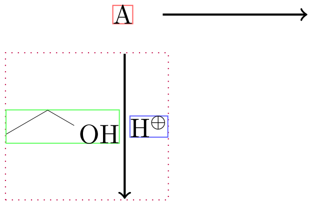
```
\makevisible
\begin{rxn}
 \reactand[,a]{A}
 \arrow{}{}
 \branch[below=of a]{
  \arrow[direction=below,both]{\
    chemfig{-[::30]-[::-60]OH}}{\Hpl}
 }
\end{rxn}
\makeinvisible
```



What you have to do is shift the branch using the TikZ keys xshift and yshift.

Example 27

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



4.3 chemand

New in V1.3

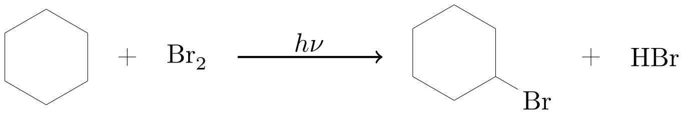
The command

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

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

Example 28

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



The optional arguments for \chemand and \reactand are the same, see section 4.12.

4.4 dummy

\dummy creates an empty node. \arrow needs to follow after a node, because \arrow internally uses \tikzchainprevious. If there is no node on the chain *before* \arrow is used, it will cause an error. But by using \dummy you can start a scheme with an arrow anyway.

Example 29

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

4.5 elmove

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

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

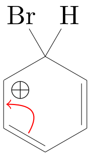
This is expanding the command

```
\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 30

```
\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}
```



4.6 makeinvisible

V1.2 `\makeinvisible` restores the normal myChemistry behaviour after `\makevisible` (see section 4.7) has been used. `\makeinvisible` only changes the looks of nodes following after it.

4.7 makevisible

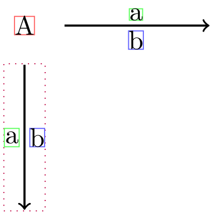
V1.2 With `\makevisible` you can visualize the nodes within which reactands, arrows and branches are set. This is useful when you're aligning branches, for example. You can see an example for `\makevisible` in section 4.1. Every kind of nodes is emphasized with a different colour:

`\reactand{}{}`, `\arrow{above}{}`, `\arrow{}{below}` und `\branch{}{}`. Also see section 4.6.

`\makevisible` only changes the looks of nodes following after it.

Example 31

```
\makevisible
\begin{rxn}
 \reactand[,a]{A}
 \arrow{a}{b}
 \branch[below=of a]{
   \arrow[direction=below,both]{
    a}{b}
 }
\end{rxn}
\makeinvisible
```



4.8 marrow

\marrow creates a double-headed arrow.

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

It is a shortcut for `\arrow[type=<->,length=.5,direction=<direction>]{}{}`. You can use `<direction>` like the alignment option of \reactand (see section 4.12 or section 4.11).

4.9 mCsetup

New in V1.3

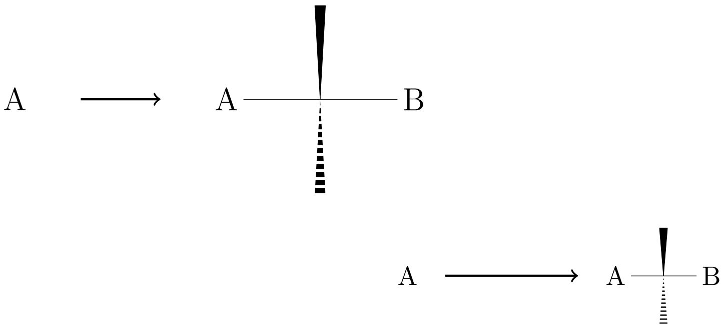
With

```
\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 32

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



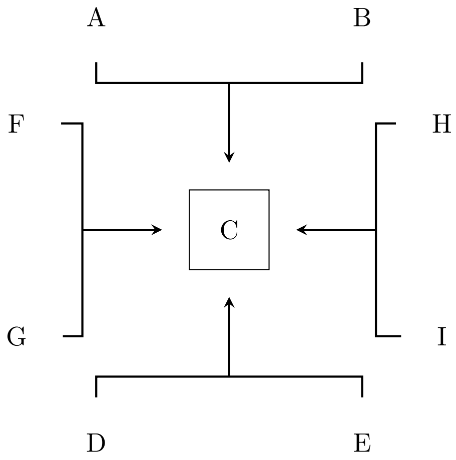
4.10 merge

\merge cannot only be used in myChemistry's environments but also in a 'tikzpicture'. With \merge you can, well, two branches into one. To be able to do that, you need to refer to anchors.

```
\merge[<key>]{<target>}{<start a>}{<start b>}
```

Example 33

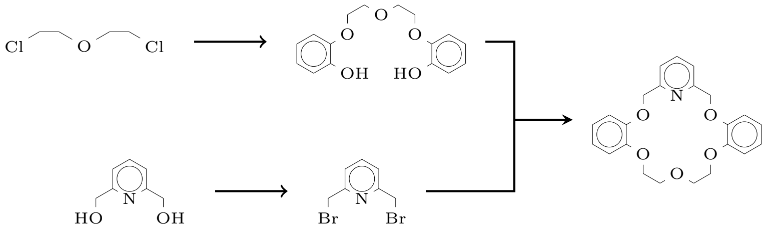
```
\begin{center}
\begin{tikzpicture}
  \node(a) at (0,0) {A};
  \node(b) at (10em,0) {B};
  \node[draw,minimum size=3em](c)
      at (5em,-8em) {C};
  \merge{c}{a}{b}
  \node(d) at (0,-16em) {D};
  \node(e) at (10em,-16em) {E};
  \merge[direction=above]{c}{d}{e
    }
  \node(f) at (-3em,-4em) {F};
  \node(g) at (-3em,-12em) {G};
  \merge[direction=right]{c}{f}{g
    }
  \node(h) at (13em,-4em) {H};
  \node(i) at (13em,-12em) {I};
  \merge[direction=left]{c}{h}{i}
\end{tikzpicture}
\end{center}
```



Usage of \merge in a myChemistry environment:

Example 34

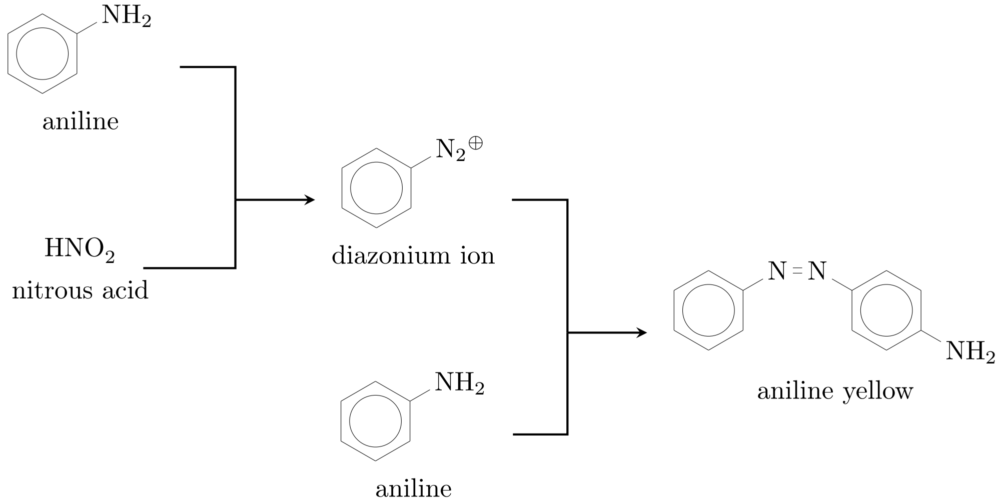
```
\begin{rxn}
 \setatomsep{1em}\tiny
 % branch 1
  \reactand[,above]{ \chemfig{Cl-[:30,1.5]--[:-30,1.5]O
    -[:30,1.5]--[:-30,1.5]Cl}{} }
  \arrow[length=.5]{}{}
  \reactand[,start_above]{ \chemfig{O(-[:-150]**6(-----(-OH)-))
    -[:90]-[:30]-[:-30]O-[:30]-[:-30]-[:-90]O-[:-30]**6(-(-HO)-----)} }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



Please note, that you should use branches to refer to, when you use \merge in a myChemistry environment. Using \merge may afford playing with xshift and yshift until you get the result you want.

Example 35

```
\begin{rxn}
 \setatomsep{1.5em}
 \reactand[,start_aa]{ \chemname{\chemfig{**6(---(-NH_2)---)}}{aniline}
     }
 \reactand[below,start_ab,yshift=-3em]{ \chemname{\ce{HNO2}}{nitrous
    acid} }
 \branch[right=of start_aa,target_a,xshift=6em,yshift=-5em]{
   \reactand{ \chemname{\chemfig{**6(---(-N|_2\op)---)}}{diazonium ion}
     }
 }% = start_ba
 \branch[below=of target_a,start_bb,yshift=-3em]{
   \reactand{ \chemname{\chemfig{**6(---(-NH_2)---)}}{aniline} }
 }
 \branch[right=of target_a,target_b,xshift=6em,yshift=-5em]{
   \reactand{ \chemname{\chemfig{N(-[:-150]**6(------))=N
    -[:-30]**6(---(-NH_2)---)}}{aniline yellow} }
 }
 \merge[direction=right]{target_a}{start_aa}{start_ab}
 \merge[direction=right]{target_b}{target_a}{start_bb}
\end{rxn}
```



4.11 mesomeric

New in V1.3

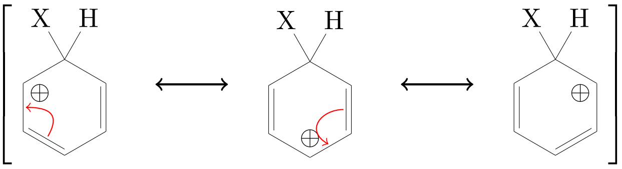
The \mesomeric command works just like \branch (see section 4.2) but places the formulæ into square brackets. **If you used earlier versions of myChemistry please be aware, that the command syntax has changed.**

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

The resonance formulæ are written into <formula>. With \marrow (see section 4.8) you create the resonance arrows. If needed you can give an anchor (<anchor>) to \mesomeric (also see section 4.2). Alignment is used the same way as with \reactand.

Example 36

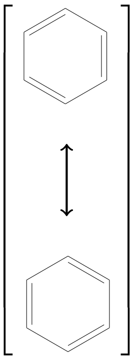
```
\begin{rxn}
 \mesomeric{
   \reactand{
     \chemfig{*6(=[@{e1}]-=-(-[:120]X)(-[:60]H)-(-[:-30,.4,,,white]\oplus)-[@{e2}])}
     \elmove{e1}{60:4mm}{e2}{0:4mm}
   }
   \marrow
   \reactand{
     \chemfig{*6(-(-[:90,.4,,,white]\oplus)-[@{e4}]=[@{e3}]-(-[:120]X)(-[:60]H)-=)}
     \elmove{e3}{180:4mm}{e4}{150:4mm}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



Or vertical, too:

Example 37

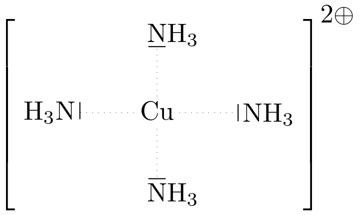
```
\begin{rxn}
 \mesomeric{
   \reactand{ \chemfig{*6(=-=-=-)} }
   \marrow[below]
   \reactand[below]{ \chemfig{*6(-=-=-=)} }
 }
\end{rxn}
```



Or maybe a coordination complex?

Example 38

```
\begin{rxn}
 \setatomsep{3em}
 \mesomeric[,a]{
   \reactand{ \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}
```



4.12 reactand

New in V1.3

The command \reactand is somehow the basic command of myChemistry. **If you used earlier versions of myChemistry please be aware, that the command syntax has changed.**

```
 \reactand[<alignment>,<anchor>,<tikz>]{<formula>}
```

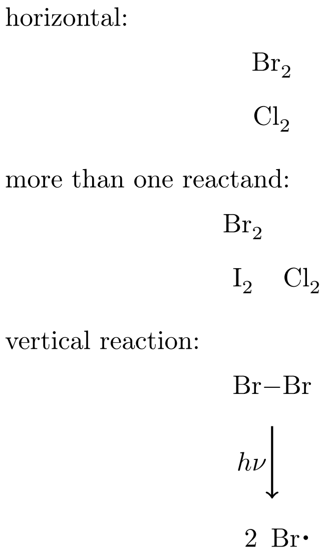
In this command the actual formulæ are written (`<formula>`). If needed, they can be given an anchor (`<anchor>`). The optional argument `<alignment>` can have 8 different values: (a) `right`, (b) `above right`, (c) `above`, (d) `above left`, (e) `left`, (f) `below left`, (g) `below`, (h) `below right` Default is `right`. You use this argument to place the reactand relatively to the reactand or arrow right before.

Example 39

```
horizontal:
\begin{rxn}
  \reactand{\ce{Br2}}
  \reactand[below]{\ce{Cl2}}
\end{rxn}

more than one reactand:
\begin{rxn}
  \reactand{\ce{Br2}}
  \reactand[below]{\ce{I2}}
  \reactand{\ce{Cl2}}
\end{rxn}

vertical reaction:
\begin{rxn}
 \reactand{\ce{Br-Br}}
 \arrow[length=.5,direction=below]{$h\nu$}{}
 \reactand[below]{\ce{2 ~\lewis{0.,Br}}}
\end{rxn}
```



4.13 rxn (environment)

rxn is a non-floating not numbered environment for reaction schemes. All schemes are centered.

```
\begin{rxn}[<keys>]
  ...
\end{rxn}
```

The optional argument `<scale factor>` has the same effect as has the key `scale=<scalefactor>` on rxnscheme. Default is 1.0.

4.13.1 Options

v1.2 rxn has two keys:

`align=<alignment>` alignment behaviour of the rxn environment; default is 'center'

`scale=<factor>` factor by which the rxn environment is scaled; default: '1.0'

Example 40

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

center ⟶ centered

right ⟶ raggedleft

left ⟶ raggedright

4.14 rxnscheme (environment)

rxnscheme is a floating environment for reaction schemes.

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

4.14.1 Options

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

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

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

scale=<factor> 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 the ChemFig formulæ.

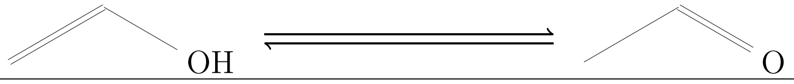
```
\begin{rxnscheme}[placement=<placement>]{<caption>}
 ...
\end{rxnscheme}
```

V1.2 align=<alignment> This key changes the alignment of the scheme. You can choose between left, center and right.

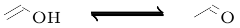
Example 41

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

4.14.2 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 42

```
\begin{rxnscheme}{ruled}
 \reactand{default style}
\end{rxnscheme}
\floatstyle{boxed}
\restylefloat{rxnfloat}
\begin{rxnscheme}{boxed}
 \reactand{framed object}
\end{rxnscheme}
\floatstyle{plain}
\restylefloat{rxnfloat}
\begin{rxnscheme}{plain}
 \reactand{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 key:

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

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

```
\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`.

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

Example 43

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

Reaction schemes

4.15 setarrowlength

New in V1.3

The default length of an reaction arrow is 5.0 em or $5.0 \cdot \sqrt{2}$ em for the diagonal ones. You can change these values using

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

into `<length>` or `<length>`$\cdot\sqrt{2}$, respectively. Mind the fact that you have to use a length unit. If you leave the argument empty, the length is reset to default.

This command replaces `\arrowlength` from earlier versions.

4.16 setatomsize

New in V1.3

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.

This command replaces `\atomsize` from earlier versions.

4.17 setbondlength

New in V1.3

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.

This command replaces `\bondlength` from earlier versions.

4.18 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.

This command replaces `\bondshape` from earlier versions.

4.19 setrcndist

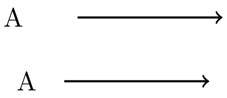
V1.2 The nodes within which the reactands an arrows are set have a certain distance between them. The default distance is 1 em. If you want to change that, you can use

```
\setrcndist{<länge>}
```

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

Example 44

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



4.20 setrxnalign/setschemealign

V1.2 With the commands

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

The default alignment behaviour of `rxn` and `rxnscheme` (see section 4.13.1 & section 4.14.1) 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 45

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

$$\mathrm{A} \longrightarrow \mathrm{B}$$

$$\mathrm{A} \longrightarrow \mathrm{B}$$

4.21 setschemename

See section 4.14.2.

4.22 transition

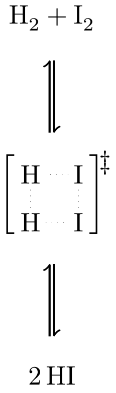
New in v1.3

\transition works exactly like \reactand (see section 4.12). **If you used earlier versions of myChemistry please be aware, that the command syntax has changed.**

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

Example 46

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

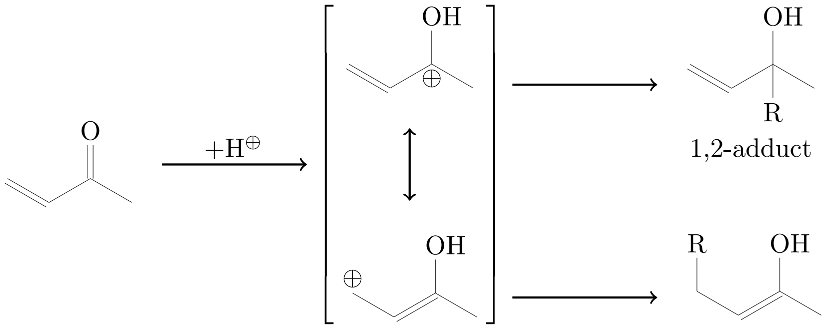


5 Examples

5.1 Addition Reaction

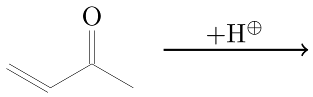
A simple reaction scheme with two different products.

Reaction scheme 10 addition reaction



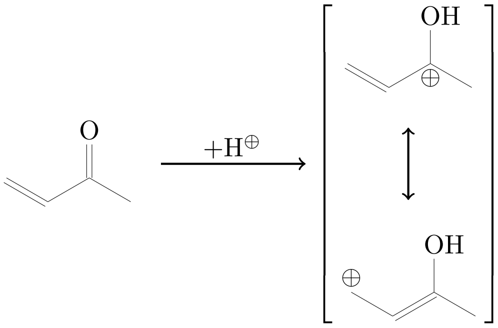
Let's take a closer look, step after step. At first we write the first reactand and the reaction arrow.

```
1 \reactand{ \chemfig{=_[::-30]-[::60](=[::60]O)-[::-60]} }
2 \arrow{ $+ \Hpl$ }{}
```



Then we write the resonance formulæ. The \mesomeric gets the anchor rf (line 7).

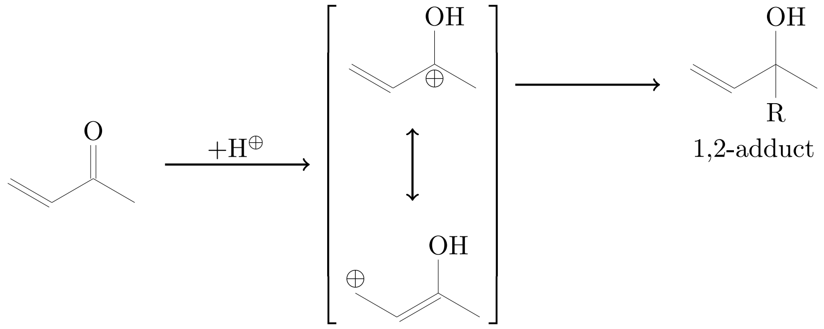
```
3 \mesomeric[,rf]{
4   \reactand{ \chemfig{=_[:-30]-[::60](-[::60]OH)
    (-[::-120,.3,,,white]\oplus)-[::-60]} }
5   \marrow[below]
6   \reactand[below]{ \chemfig{\oplus-[6,.3,,,white
    ]-[:-30]=_[::60](-[::60]OH)-[::-60]} }
7 }
```



Now comes the 1,2-adduct, placed in a branch referring rf, shifted above with yshift:

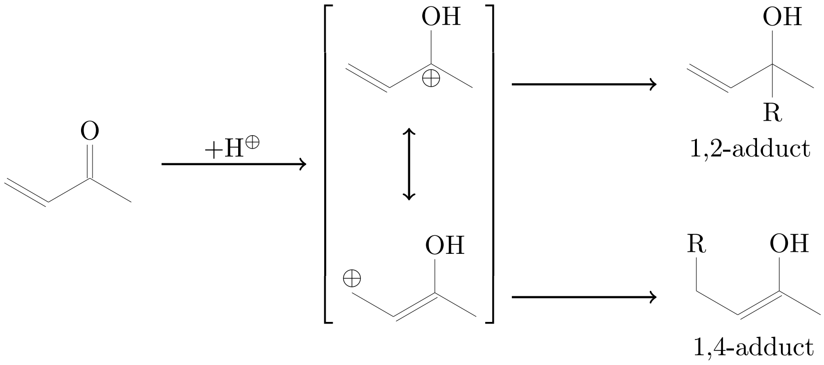
```
8 \branch[right=of rf,,yshift=3em]{
9   \arrow{}{}
```

```
  \reactand{ \chemname{\chemfig{=_[:-30]-[::60](-[::60]OH
  )(-[::-120]R)-[::-60]}}{1,2-adduct} }
}
```



At last we write the branch containing the 1,4-adduct, also referencing **rf**, shifted below with yshift:

```
\branch[right=of rf,,yshift=-5em]{
  \arrow{}{}
  \reactand{ \chemname{\chemfig{R
  -[6]-[:-30]=_[::60](-[::60]OH)-[::-60]}}{1,4-adduct} }
}
```



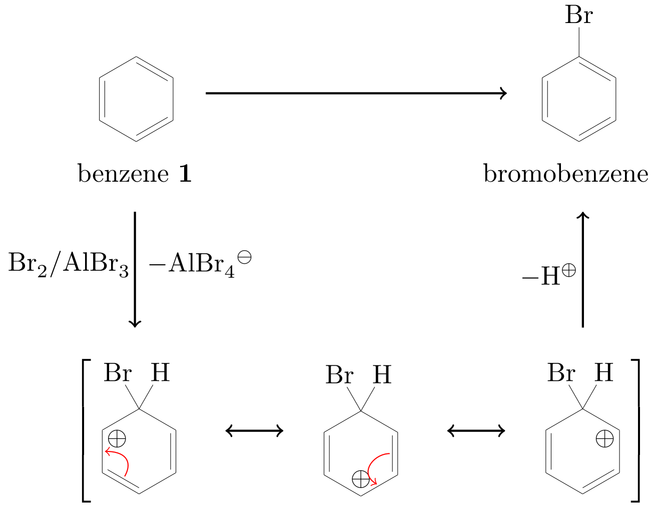
The complete code looks like this:

```
\begin{rxnscheme}{addition reaction}
 \reactand{ \chemfig{=_[::-30]-[::60](=[::60]O)-[::-60]}
  }
 \arrow{ $+ \Hpl$ }{}
 \mesomeric[,rf]{
   \reactand{ \chemfig{=_[:-30]-[::60](-[::60]OH)
  (-[::-120,.3,,,white]\oplus)-[::-60]} }
   \marrow[below]
   \reactand[below]{ \chemfig{\oplus-[6,.3,,,white
  ]-[:-30]=_[::60](-[::60]OH)-[::-60]} }
 }
 \branch[right=of rf,,yshift=3em]{
   \arrow{}{}
   \reactand{ \chemname{\chemfig{=_[:-30]-[::60](-[::60]OH)(-[::-120]R)-[::-60]}}{1,2-adduct} }
 }
 \branch[right=of rf,,yshift=-5em]{
   \arrow{}{}
   \reactand{ \chemname{\chemfig{R-[6]-[:-30]=_[::60](-[::60]OH)-[::-60]}}{1,4-adduct} }
 }
\end{rxnscheme}}
```

5.2 Mesomerism

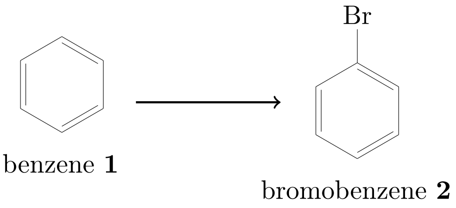
We want to display the following reaction scheme:

Reaction scheme 11 electrophilic substitution



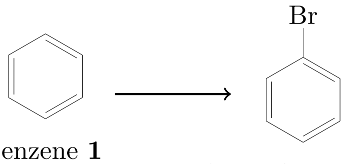
First we write the main reaction. To do so, we use the commands \reactand, \arrow and the environment \begin{rxn} ... \end{rxn}.

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



Now we make it a little bit smaller:

```
\begin{rxn}[scale=.8]
 \setatomsep{1.6em}
 \reactand{
  \chemname{\chemfig{*6(-=-=-=)}}{benzene \compound{benzene}}
    }
 \arrow{}{}
 \reactand{
    \chemname{\chemfig{*6(-=-=(-Br)-=)}}{bromobenzene \compound{bromobenzene}}
  }
\end{rxn}
```

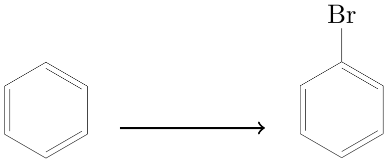


benzene **1** bromobenzene **2**

We have two possibilities to align both benzene rings at the same height. Either we shift the second one up using Ti*k*Z code:

```
\reactand[,,yshift=1em]{
 \chemname{\chemfig{*6(-=-=(-Br)-=)}}{bromobenzene \compound{bromobenzene}}
}
```

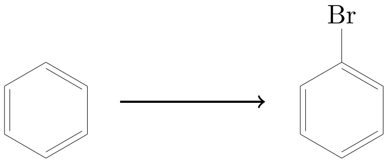
This is not the best solution, because the arrow isn't centered with respect to the rings.



benzene **1** bromobenzene **2**

The second possibility would be to shift the first ring down. We can't achieve that by using Ti*k*Z code, because the following arrow and reactand align themselves with respect to the reactand or arrow directly before. But we can write an invisible bromine to the first benzene to do the trick:

```
\reactand{
 \chemname{\chemfig{*6(-=-=(-[,,,,white]\phantom{Br})-=)}}{benzene \compound{benzene}}
}
```

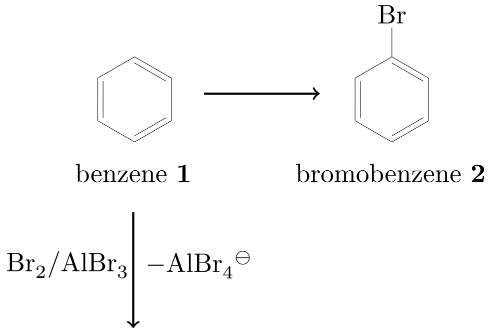


benzene **1** bromobenzene **2**

We need to give the first reactand an anchor in order to refer to it with the branch.

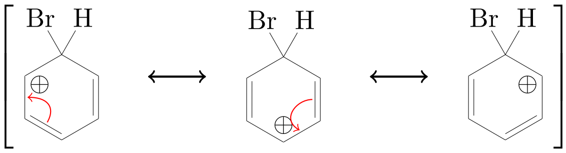
```
\begin{rxn}[scale=.8]
 \setatomsep{1.6em}
 \reactand[,start]{
  \chemname{\chemfig{*6(-=-=(-[,,,,white]\phantom{Br})-=)}}{benzene \compound{benzene}}
   }
  \branch[below=of start]{
    \arrow[direction=below,both]{ \ce{Br2 / AlBr3} }{ $-\ce{AlBr4\om}$ }
  }
 \arrow{}{}
 \reactand{
   \chemname{\chemfig{*6(-=-=(-Br)-=)}}{bromobenzene \compound{bromobenzene}}
  }
\end{rxn}
```

So the first reactand gets the anchor `start` and the branch refers to it with `below=of start`. For the reaction arrow to point below, we need to use the key `direction=below`. Now we get:

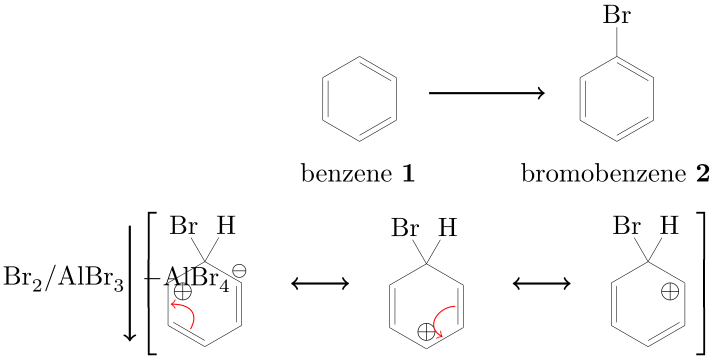


Next we write the resonance formulæ of the Wheland intermediate. To do that we use three further commands: `\mesomeric`, `\marrow` and `\elmove`.

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



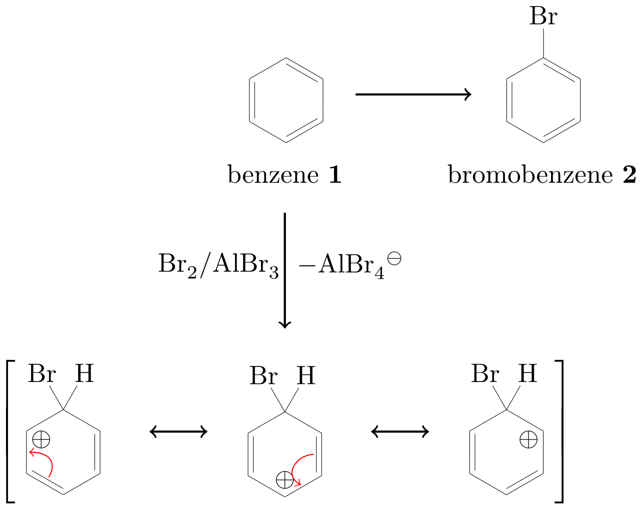
When we write the code *inside* of the branch, directly after the arrow, we get the following:



This obviously messes everything up. We can try this, though:

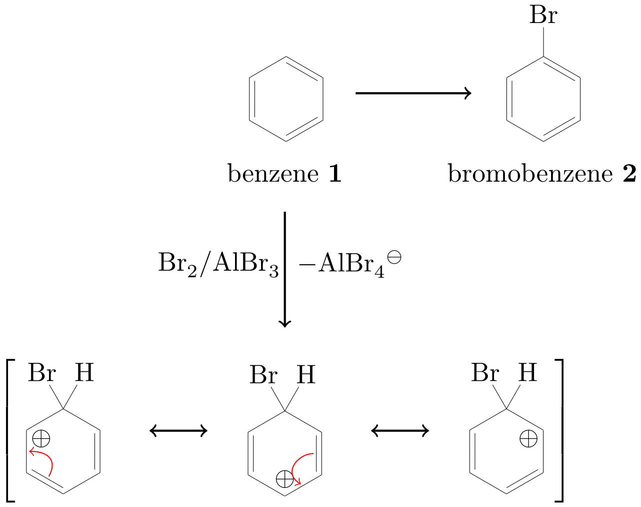
```
\begin{rxn}[scale=.8]
 \setatomsep{1.6em}
 \reactand[,start]{
  \chemname{\chemfig{*6(-=-=(-[,,,,white]\phantom{Br})-=)}}{benzene \compound{benzene}}
   }
  \branch[below=of start]{
    \arrow[direction=below,both]{\ce{Br2 / AlBr3}}{$-\ce{AlBr4\om}$}
    \mesomeric[below]{
      \reactand{
        \chemfig{*6(=[@{e1}]-=-(-[:120]Br)(-[:60]H)-(-[:-30,.4,,,white]\oplus)-[@{e2}])}
        \elmove{e1}{60:4mm}{e2}{0:4mm}
      }
      \marrow
      \reactand{
        \chemfig{*6(-(-[:90,.4,,,white]\oplus)-[@{e4}]=[@{e3}]-(-[:120]Br)(-[:60]H)-=)}
        \elmove{e3}{180:4mm}{e4}{150:4mm}
      }
      \marrow
      \reactand{
        \chemfig{*6(-=-(-[:-150,.4,,,white]\oplus)-(-[:120]Br)(-[:60]H)-=)}
      }
    }
  }
 \arrow{}{}
 \reactand{
   \chemname{\chemfig{*6(-=-=(-Br)-=)}}{bromobenzene \compound{bromobenzene}}
  }
\end{rxn}
```

The result is better:



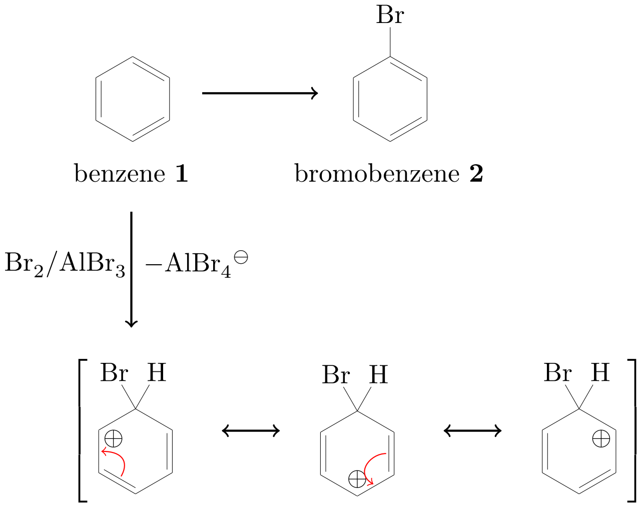
It isn't really what we want, though, because the intermediate is centered below the arrow. In order to be able to shift the whole thing, we place it into a branch of its own.

```
...
\branch[below=of start]{
  \arrow[direction=below,both,name=arrow_a]{\ce{Br2 / AlBr3}}{$-\ce{AlBr4\om}$}
}
\branch[below=of arrow_a]{
    \mesomeric{
      \reactand{
        \chemfig{*6(=[@{e1}]-=-(-[:120]Br)(-[:60]H)-(-[:-30,.4,,,white]\oplus)-[@{e2}])}
        \elmove{e1}{60:4mm}{e2}{0:4mm}
      }
      \marrow
      \reactand{
        \chemfig{*6(-(-[:90,.4,,,white]\oplus)-[@{e4}]=[@{e3}]-(-[:120]Br)(-[:60]H)-=)}
        \elmove{e3}{180:4mm}{e4}{150:4mm}
      }
      \marrow
      \reactand{
        \chemfig{*6(-=-(-[:-150,.4,,,white]\oplus)-(-[:120]Br)(-[:60]H)-=)}
      }
    }
}
...
```



In first sight this isn't better. But by shifting the branch with xshift, we get what we want:

```
...
\branch[below=of start]{
  \arrow[direction=below,both,name=arrow_a]{\ce{Br2 / AlBr3}}{$-\ce{AlBr4\om}$}
}
\branch[below=of arrow_a,,xshift=8.5em]{
  \mesomeric{
    \reactand{
      \chemfig{*6(=[@{e1}]-=-(-[:120]Br)(-[:60]H)-(-[:-30,.4,,,white]\oplus)-[@{e2}])}
      \elmove{e1}{60:4mm}{e2}{0:4mm}
    }
    \marrow
    \reactand{
      \chemfig{*6(-(-[:90,.4,,,white]\oplus)-[@{e4}]=[@{e3}]-(-[:120]Br)(-[:60]H)-=)}
      \elmove{e3}{180:4mm}{e4}{150:4mm}
    }
    \marrow
    \reactand{
      \chemfig{*6(-=-(-[:-150,.4,,,white]\oplus)-(-[:120]Br)(-[:60]H)-=)}
    }
  }
}
...
```

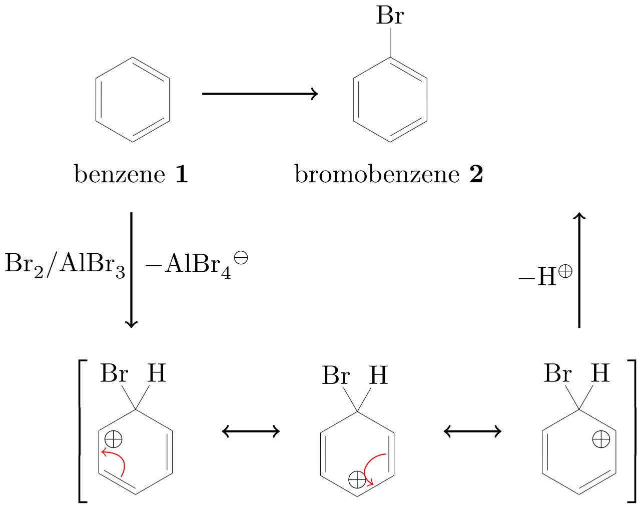


benzene **1**

bromobenzene **2**

The last arrow is also placed in its own branch, so we can shift it where we want.

```
...
\branch[below=of start]{
  \arrow[direction=below,both,name=arrow_a]{\ce{Br2 / AlBr3}}{$-\ce{AlBr4\om}$}
}
\branch[below=of arrow_a,mesomerism,xshift=8.5em]{
  \mesomeric{
    \reactand{
      \chemfig{*6(=[@{e1}]-=-(-[:120]Br)(-[:60]H)
  -(-[:-30,.4,,,white]\oplus)-[@{e2}])}
      \elmove{e1}{60:4mm}{e2}{0:4mm}
    }
    \marrow
    \reactand{
      \chemfig{*6(-(-[:90,.4,,,white]\oplus)-[@{e4}]=[@{e
  3}]-(-[:120]Br)(-[:60]H)-=)}
      \elmove{e3}{180:4mm}{e4}{150:4mm}
    }
    \marrow
    \reactand{
      \chemfig{*6(-=-(-[:-150,.4,,,white]\oplus)-(-[:120]
  Br)(-[:60]H)-=)}
    }
  }
}
\branch[above=of mesomerism,,xshift=7.25em]{
  \arrow[direction=above]{$-\Hpl$}{}
}
...
```



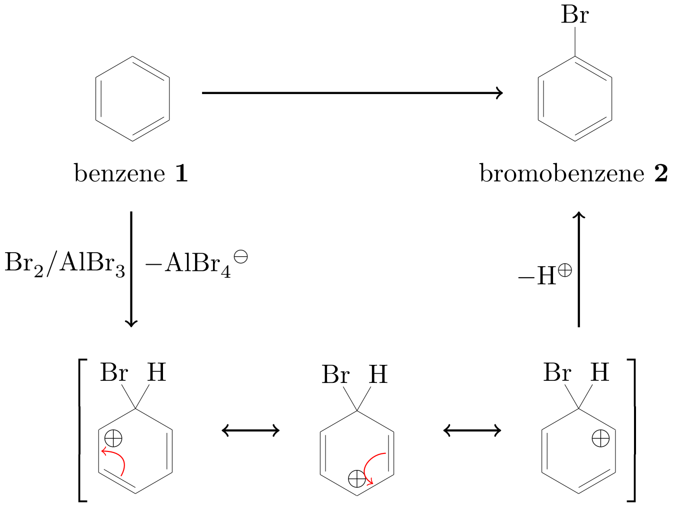
Now we're almost there: the arrow of the main reaction still is too short.

```
\begin{rxn}[scale=.8]
\setatomsep{1.6em}
\reactand[,start]{\chemname{\chemfig{*6(-=-=(-[,,,,white
  ]\phantom{Br})-=)}}{benzene \compound{benzene}} }

```

```
\branch[below=of start]{
  \arrow[direction=below,both,name=arrow_a]{\ce{Br2 / AlBr3}}{$-\ce{AlBr4\om}$}
}
\branch[below=of arrow_a,mesomerism,xshift=8.5em]{
  \mesomeric{
    \reactand{
      \chemfig{*6(=[@{e1}]-=-(-[:120]Br)(-[:60]H)-(-[:-30,.4,,,white]\oplus)-[@{e2}])}
      \elmove{e1}{60:4mm}{e2}{0:4mm}
    }
    \marrow
    \reactand{
      \chemfig{*6(-(-[:90,.4,,,white]\oplus)-[@{e4}]=[@{e3}]-(-[:120]Br)(-[:60]H)-=)}
      \elmove{e3}{180:4mm}{e4}{150:4mm}
    }
    \marrow
    \reactand{
      \chemfig{*6(-=-(-[:-150,.4,,,white]\oplus)-(-[:120]Br)(-[:60]H)-=)}
    }
  }
}
\branch[above=of mesomerism,,xshift=7.25em]{
  \arrow[direction=above]{$-\Hpl$}{}
}

\arrow[length=2.6]{}{}
\reactand{ \chemname{\chemfig{*6(-=-=(-Br)-=)}}{ bromobenzene \compound{bromobenzene}} }
\end{rxn}
```


Br
benzene 1
bromobenzene 2
$Br_2/AlBr_3$
$-AlBr_4^{\ominus}$
$-H^{\oplus}$
Br H
Br H
Br H

5.3 Creating an extensive synthesis using TikZ, myChemistry and ChemFig

Since the commands of ChemFig are functioning well in a 'tikzpicure' environment, we can create extensive syntheses, using the \merge command. The other myChemistry commands can't be used here without further attention, because they're placed on a chain. In the next example, we directly use the floating environment rxnfloat.

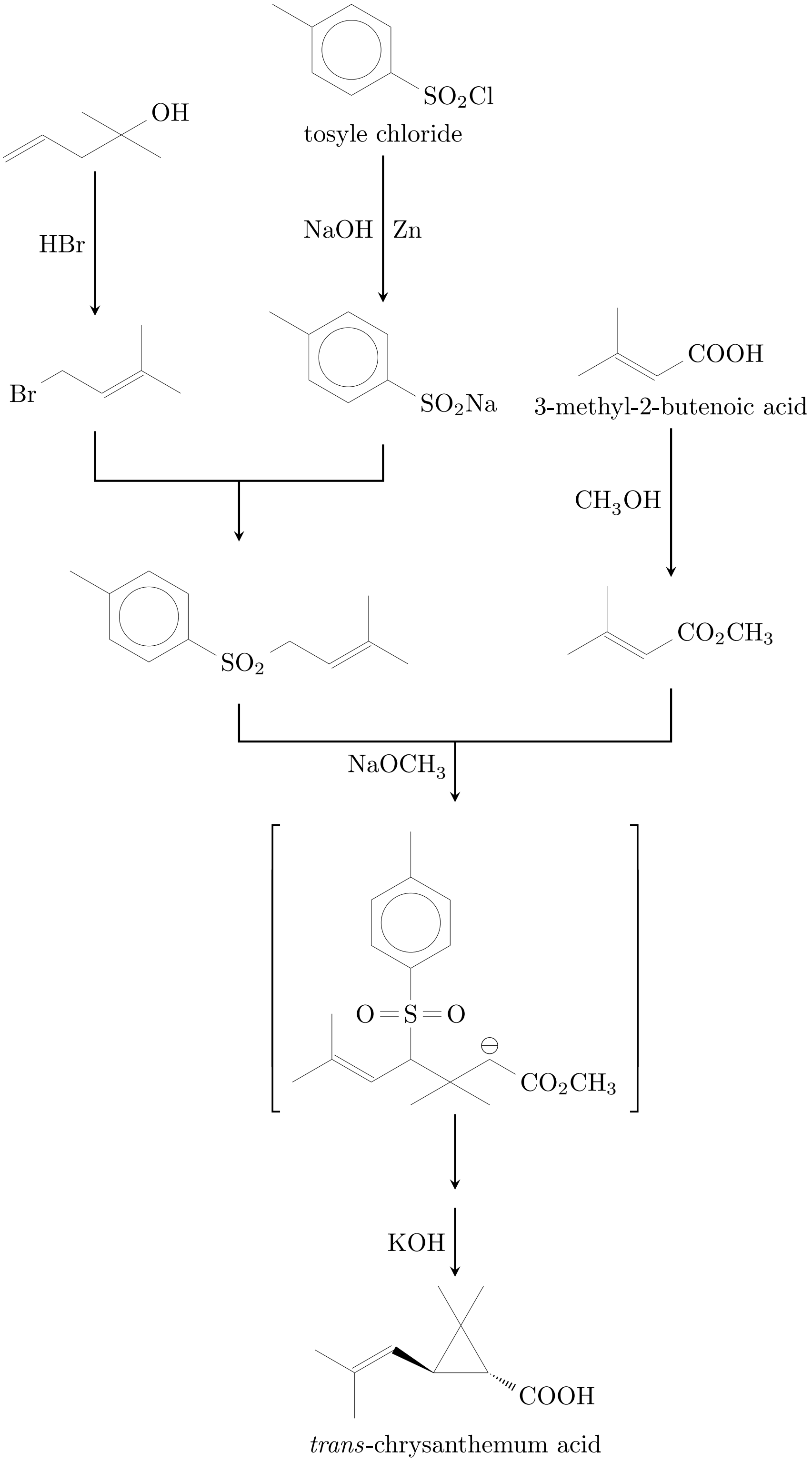
```

\begin{rxnfloat}
\setatomsep{1.8em}\setcrambond{3pt}{.5pt}{1pt}
\centering
\begin{tikzpicture}[scale=.8]
 \small
 \node(a) at (0,0) {\chemfig{=_[::30]-[::-60]-[::60](-[::-60])(-[::120])-[::0]OH}};
 \node(b) at (0,-4) {\chemfig{Br-[::30]-[::-60]=_[::60](-[::-60])-[::60]}};
 \draw[-stealth,thick] (a.south) -- node[left]{HBr} (b.north);
 \node(c) at (5,1) {\chemname{\chemfig{**6(--(-SO_2Cl)---(-)-)}}{tosyle chloride}};
 \node(d) at (5,-4) {\chemfig{**6(--(-SO_2Na)---(-)-)}};
 \draw[-stealth,thick] (c.south) -- node[left]{NaOH} node[right]{Zn} (d.north);
 \node(e) at (2.5,-8.5) {\chemfig{**6(--(-SO_2-[:30]-[::-60]=_[::60](-[::60])-[::-60])---(-)-)}};
 \node(f) at (10,-4) {\chemname{\chemfig{-[::30](-[::60])=_[::-60]-[::60]COOH}}{3-methyl-2-butenoic acid}};
 \node(g) at (10,-8.5) {\chemfig{-[::30](-[::60])=_[::-60]-[::60]CO_2CH_3}};
 \draw[-stealth,thick] (f.south) -- node[left]{\ce{CH3OH}} (g.north);
 \merge{e}{b}{d}
 \node[left delimiter={[},right delimiter={]}](h) at (6.25,-14.5) {\chemfig{-[::30](-[::60])=^[::-60]-[::60](-[::60]S(=[::90]O)(=[::-90]O)-[::0]**6(---(-)---))-[::-60](-[::0])(-[::-120])-[::60](-[::60,.5,,,white]\ominus)-[::-60]CO_2CH_3}};
 \node at (5.25,-11) {\ce{NaOCH3}};
 \merge{h}{e}{g}
 \node(i) at (6.25,-18.5) {};
```

```
  \node(j) at (6.25,-21.5) {\chemname{\chemfig{-[::-30](-[::-60])=^[::60]>[::-60](-[::90,1.2])-[::30,1.2](-[::120,1.2](-[::-60])-[::0])<:[::-30]COOH}}{\emph{trans}-chrysanthemum acid}};
  \draw[-stealth,thick] (h.south) -- (i.north);
  \draw[-stealth,thick] (i.south) -- node[left]{KOH} (j.north);
\end{tikzpicture}
\caption{synthesis of chrysanthemum acid}
\end{rxnfloat}

```

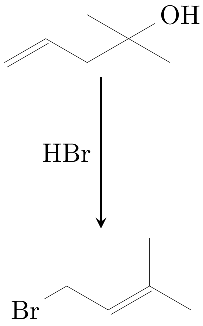
Reaction scheme 12 synthesis of chrysanthemum acid



Let's go through the code, piece by piece.

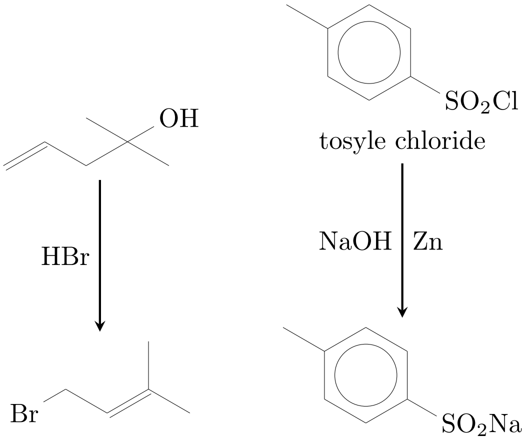
```
1
2  \begin{rxnfloat}
3  \setatomsep{1.8em}\setcrambond{3pt}{.5pt}{1pt}
4  \centering
5  \begin{tikzpicture}[scale=.8]
6   \small
7   \node(a) at (0,0) {\chemfig
     {=_[::30]-[::-60]-[::60](-[::-60])(-[::120])-[::0]OH}};
8   \node(b) at (0,-4) {\chemfig{Br
     -[::30]-[::-60]=_[::60](-[::-60])-[::60]}};
9   \draw[-stealth,thick] (a.south) -- node[left]{HBr} (b.
     north);
```

In lines 1 – 6 we begin the environment and make sure, that the formulæ don't become to big. In lines 7 – 9 the first two reactands are written (lines 7 and 8) and connected with an arrow (line 9).



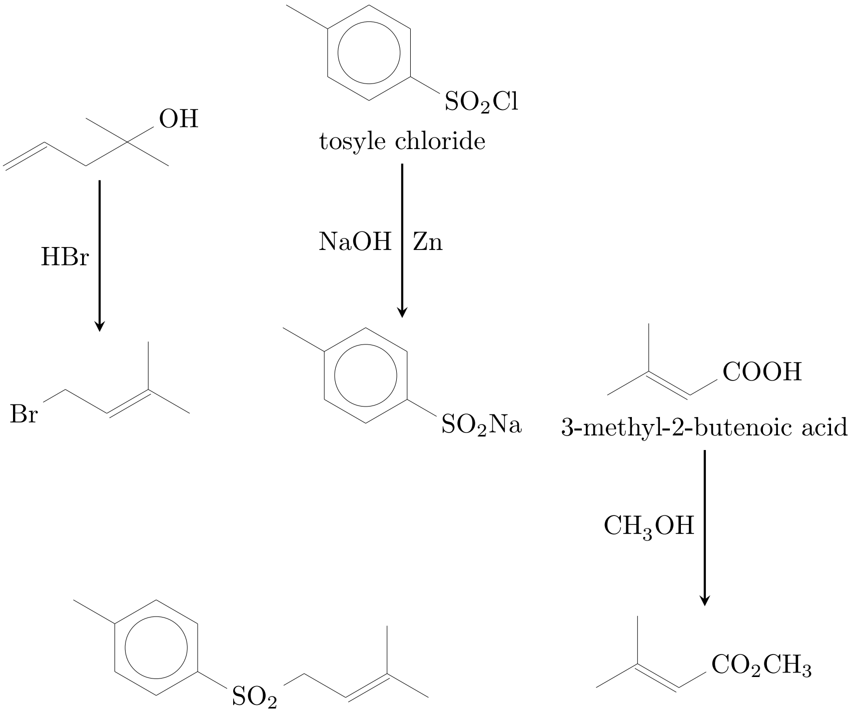
```
10  \node(c) at (5,1) {\chemname{\chemfig{**6(--(-SO_2Cl)
     ---(-)-)}}{tosyle chloride}};
11  \node(d) at (5,-4) {\chemfig{**6(--(-SO_2Na)---(-)-)}};
12  \draw[-stealth,thick] (c.south) -- node[left]{NaOH} node
     [right]{Zn} (d.north);
```

In the three following lines 10 – 12, we create the second branch of the synthesis.



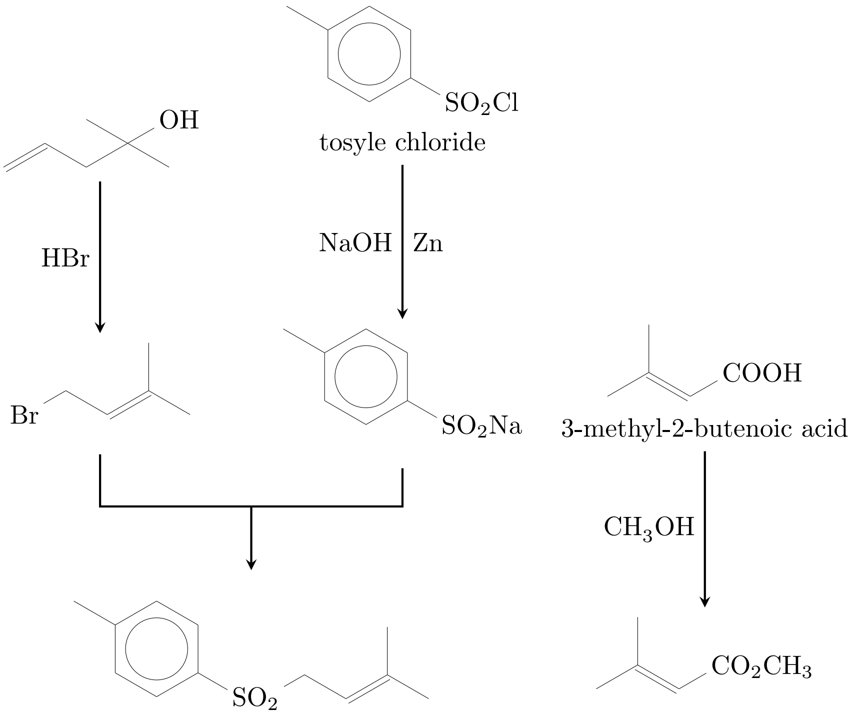
```
13  \node(e) at (2.5,-8.5) {\chemfig{**6(--(-SO
     _2-[:30]-[::-60]=_[::60](-[::60])-[::-60])---(-)-)}};
14  \node(f) at (10,-4) {\chemname{\chemfig{-[::30](-[::60])
     =_[::-60]-[::60]COOH}}{3-methyl-2-butenoic acid}};
15  \node(g) at (10,-8.5) {\chemfig{-[::30](-[::60])
     =_[::-60]-[::60]CO_2CH_3}};
16  \draw[-stealth,thick] (f.south) -- node[left]{\ce{CH3OH
     }} (g.north);
```

In lines 13 – 16 we create the third branch and the product of the first two branches.



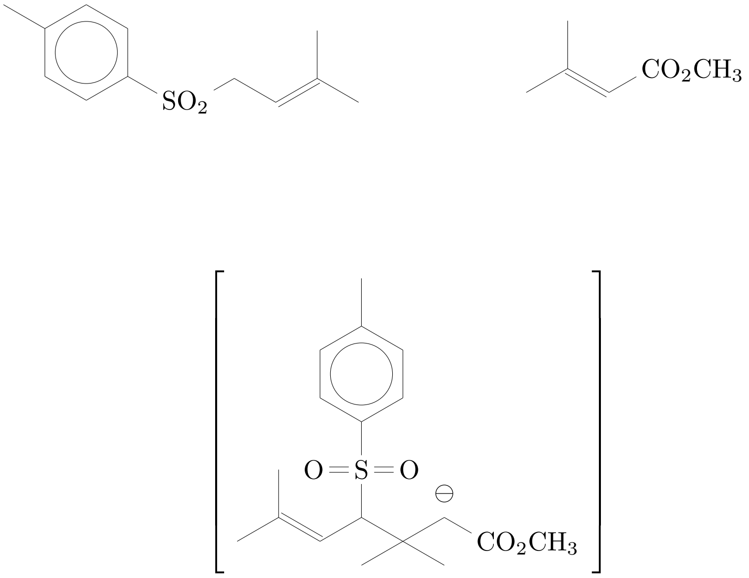
```
17 \merge{e}{b}{d}
```

In line 17 we merge the first two branches with their product.



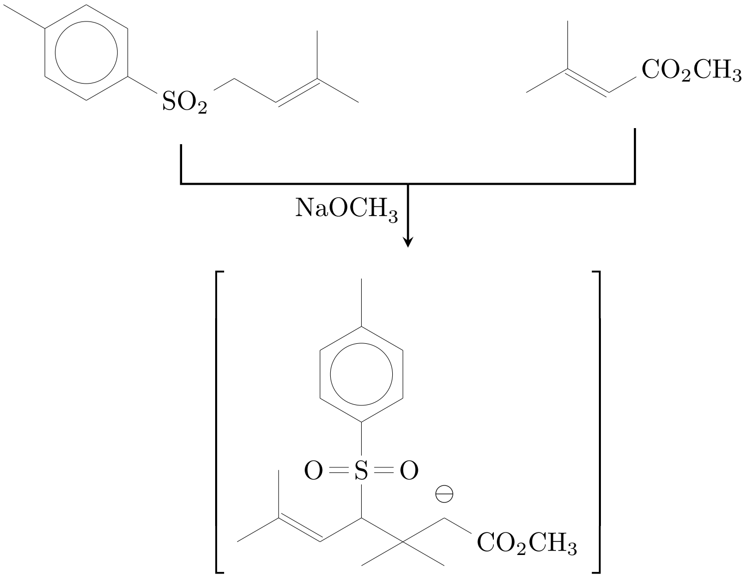
```
18 \node[left delimiter={[},right delimiter={]}](h) at
    (6.25,-14.5) {\chemfig{-[::30](-[::60])
    =^[::-60]-[::60](-[::60]S(=[::90]O)(=[::-90]O)
    -[::0]**6(---(-)---))-[::-60](-[::0])(-[::-120])
    -[::60](-[::60,.5,,,white]\ominus)-[::-60]CO_2CH_3}};
```

In line 18 we create the transition state.



```
19    \node at (5.25,-11) {\ce{NaOCH3}};
20    \merge{h}{e}{g}
```

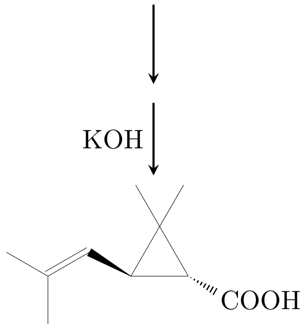
In lines 19 and 20 both branches are merged with the transition state an the merging arrow gets its reactand argument.



```
21    \node(i) at (6.25,-18.5) {};
22    \node(j) at (6.25,-21.5) {\chemname{\chemfig
       {-[::-30](-[::-60])=^[::60]>[::-60](-[::90,1.2])
       -[::30,1.2](-[::120,1.2](-[::-60])-[::0])<:[::-30]COOH}}{\
       emph{trans}-chrysanthemum acid}};
```

```
23   \draw[-stealth,thick] (h.south) -- (i.north);
24   \draw[-stealth,thick] (i.south) -- node[left]{KOH} (j.north);
25  \end{tikzpicture}
26  \caption{synthesis of chrysanthemum acid}
27  \end{rxnfloat}
28
```

In the last lines, 21 – 28, we create at first an empty node (line 21) and then the product (line 22). In lines 23 and 24 we create the last two reaction arrows. In the last four lines we end the environment.



trans-chrysanthemum acid

6 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 an 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 4th 2011