

# myChemistry

v1.2

March 20th 2011  
Clemens Niederberger

## Creating reaction schemes with L<sup>A</sup>T<sub>E</sub>X and ChemFig

```
\begin{rxn}[scale=.7]
\setatomsep{1.5em}\footnotesize
\reactand{
\chemfig{C(-[4]*6(==--))(-[2]*6(==--))(-[6,,2]HO)-C(-[2]CH_3)
(-[6]OH)-CH_3}
}{a}
\branch[below right=of a]{
\arrow[direction=above right,1H_2O]{-}{}
\reactand[above right]{
\chemfig{C(-[4]*6(==--))(-[2]*6(==--))(-[6,,2]H_2O)
\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,1H_2O]{-}{}
\reactand[below right]{
\chemfig{C(-[4]*6(==--))(-[2]*6(==--))(-[6,,2]HO)-C(-[2]CH_3)
(-[6]OH)-CH_3}
\chembelow{O}{\oplus})-C(-[2]CH_3)-[6]OH)-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}
```

The diagram illustrates the dehydration of 1-phenyl-2-phenylpropan-1-ol. The reactant is shown as a central carbon atom bonded to a phenyl ring, a methyl group, and a hydroxyl group. A curved arrow indicates the loss of water from the hydroxyl group, leading to the formation of a carbocation intermediate. The carbocation is shown with a positive charge on the carbon atom bonded to the phenyl ring. The reaction is shown in two ways: one with a curved arrow indicating the loss of water from the alcohol, and another with a straight arrow indicating the loss of water from the alcohol.

# Contents

<b>SECTION 1</b>	<b>About</b>	<b>3</b>
1.1	What's new? . . . . .	3
1.2	Licence . . . . .	3
1.3	Requirements . . . . .	3
1.4	The Idea . . . . .	4
<b>SECTION 2</b>	<b>Usage</b>	<b>4</b>
2.1	Background . . . . .	4
2.2	Basic Principle . . . . .	5
2.3	How does it work? . . . . .	6
	<i>Basic Commands, 6 • Branches, 7 • Numbered Schemes, 9</i>	
2.4	Predefined Values . . . . .	10
2.5	Package options . . . . .	11
<b>SECTION 3</b>	<b>Advanced Usage, Usage with TikZ</b>	<b>12</b>
<b>SECTION 4</b>	<b>Alphabetical Command Reference</b>	<b>13</b>
4.1	arrow . . . . .	13
4.2	arrowlength . . . . .	15
4.3	atomsizes . . . . .	15
4.4	bondlength . . . . .	16
4.5	bondshape . . . . .	16
4.6	branch . . . . .	16
	<i>Alignment problems, 17</i>	
4.7	dummy . . . . .	18
4.8	elmove . . . . .	18
4.9	makeinvisible . . . . .	19
4.10	makevisible . . . . .	19
4.11	marrow . . . . .	20
4.12	merge . . . . .	20
4.13	mesomeric . . . . .	22
4.14	reactand . . . . .	24
4.15	rxn (environment) . . . . .	25
	<i>Options, 25</i>	
4.16	rxnscheme (environment) . . . . .	25
	<i>Options, 26 • Customizing rxnscheme, 27</i>	
4.17	setrcndist . . . . .	29
4.18	setrxnalign/setschemealign . . . . .	30

4.19	setschemename . . . . .	30
4.20	transition . . . . .	30
<hr/>		
<b>SECTION 5</b>	<b>Examples</b>	<b>31</b>
5.1	Addition Reaction . . . . .	31
5.2	Mesomerism . . . . .	34
5.3	Synthesis with TikZ . . . . .	43
<hr/>		
<b>SECTION 6</b>	<b>Epilogue</b>	<b>49</b>

## 1 About

### 1.1 What's new?

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 TikZ (or pgf, actually, see section 1.3).**

There are some new "frontend" features as well. In this documentation they are marked with **new**. 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 4.15.1, section 4.16.1 and section 4.17).

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

### 1.2 Licence

myChemistry v1.2 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 09/08/2011<sup>1</sup>. More precisely: the command `\pgfpositionnodelater` must exist.

## 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**<sup>2</sup>,
- 'mhchem'<sup>3</sup> in version 3, when it exists,
- 'chemexec'<sup>4</sup>, when it exists, and
- 'chemcompounds'<sup>5</sup>, when it exists.

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

- `\ce{}` (mhchem)
- `\ox{}`, `\om{}`, `\op{}`, `\Hyd`, `\Hp1` (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 TikZ provides all the possibilities one could ask for. But since I'm a lazy guy I wrote several macros providing TikZ commands,

<sup>1</sup><http://sourceforge.net/projects/pgf/files/>

<sup>2</sup>by Christian Tellechea, <http://www.ctan.org/tex-archive/macros/latex/contrib/chemfig/>

<sup>3</sup>by Martin Hensel, <http://www.ctan.org/tex-archive/macros/latex/contrib/mhchem/>

<sup>4</sup>by me, <http://www.niederberger-berlin.net/downloads/?did=1>

<sup>5</sup>by Stephan Schenk, <http://www.ctan.org/tex-archive/macros/latex/contrib/chemcompounds/>

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 *TikZ* and stay more flexible with it, if you like.

## 2.2 Basic Principle

Within the ‘tikzpicture’ reactands and arrows are placed as nodes on a ‘chain’<sup>1</sup>.

---

### Example 1

```
1 \begin{tikzpicture}[start chain=
2 \node [on chain] {A};
3 \node [on chain] {B};
4 \node [on chain] {C};
5 \end{tikzpicture}
```

A
B
C

---

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

---

### Example 2

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

Hello

World

,

this

is

---

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

---

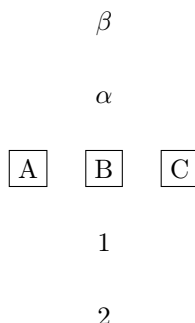
<sup>1</sup>Provided by the tikzlibrary ‘chains’

**Example 3**

```

1 \begin{tikzpicture}[start chain=
  going right,node distance=5mm]
2 \node [draw,on chain] {A};
3 \node [draw,on chain] {B};
4 { [start branch]
5 \node [on chain=going below]
  {1};
6 \node [on chain=going below]
  {2};
7 }
8 { [start branch]
9 \node [on chain=going above]
  {$\alpha$};
10 \node [on chain=going above]
  {$\beta$};
11 }
12 \node [draw,on chain] {C};
13 \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.

**2.3 How does it work?****2.3.1 Basic Commands**

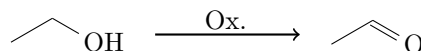
Let's take a look at an example first:

**Example 4**

```

1 \begin{rxn}
2 \reactand{ \chemfig
  {-[:30]-[: -60]OH} }{}
3 \arrow{Ox.}{}
4 \reactand{ \chemfig
  {-[:30]=_[: -60]O} }{}
5 \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 4.15).

`\reactand[<alignment>]{<formula>}{<anchor>}` places a 'node' on the 'chain', in which the chemical formulae are written. The default placement is to the **right** (see section 4.14).

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

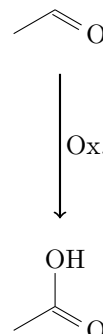
---

#### Example 5

```

1 \begin{rxn}
2 \reactand{ \chemfig
3 {-[:30]=_[::-60]O} }{}
4 \arrow[direction=below]{}{Ox.}
5 \reactand[below]{ \chemfig
6 {-[:30](-[:60]OH)=_[::-60]O}
7 }{}
8 \end{rxn}

```



As you can see one can align the reaction scheme vertical as well. By giving `\reactand` the option `below`, the carboxylic acid is placed below the arrow. With the key `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.

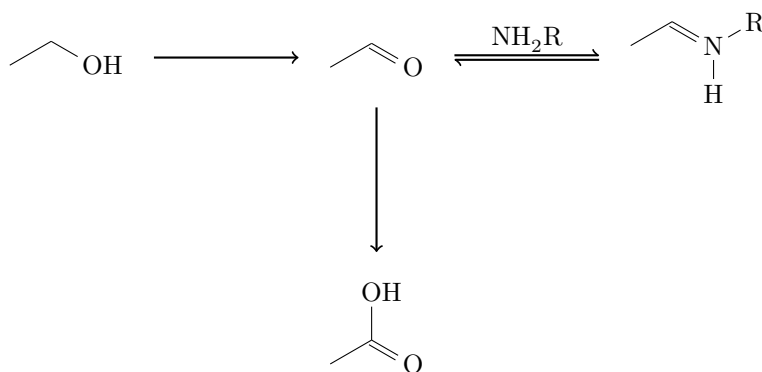
---

#### Example 6

```

1 \begin{rxn}
2 \reactand{ \chemfig{-[:30]-[::-60]OH} }{}
3 \arrow{}{}
4 \reactand{ \chemfig{-[:30]=_[::-60]O} }{carbonyl}
5 \arrow[direction=below]{}{}
6 \reactand[below]{ \chemfig{-[:30](-[:60]OH)=_[::-60]O} }{}
7 \branch[right=of carbonyl]{
8   \arrow[type={<=>}]{\ce{NH2R}}{}
9   \reactand{ \chemfig{-[:30]=_[::-60]N(-[:6]H)-[:60]R} }{}
10 }{}
11 \end{rxn}

```



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

```
\branch[<alignment>]{<branch commands>}{<anchor>}
```

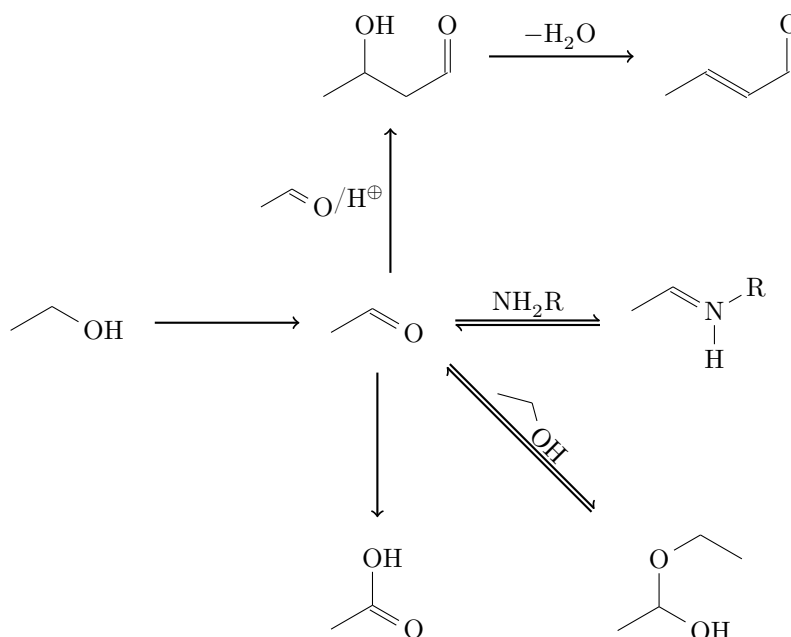
The branch was placed right of the first reactand with the anchor `carbonyl` by using the option `right=of carbonyl`. Inside the branch we used an arrow with the key `type={<=>}` so we would get an equilibrium arrow. Other types would for example be `->` (default), `<-` or `<->`.

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

#### Example 7

```
1 \begin{rxn}
2   \reactand{ \chemfig{-[:30]-[:60]OH} }{}
3   \arrow{}{}
4   \reactand{ \chemfig{-[:30]=_[:60]O} }{carbonyl}
5   \arrow[direction=below]{}{}
6   \reactand[below]{ \chemfig{-[:30](-[:60]OH)=_[:60]O} }{}
7   \branch[right=of carbonyl]{
8     \arrow[type={<=>}]{\ce{NH2R}}{}
9     \reactand{ \chemfig{-[:30]=_[:60]N(-[6]H)-[:60]R} }{}
10  }{imin}
11  \branch[below right=of carbonyl]{
12    \arrow[type={<=>},direction=below right]{ \chemfig
13      {[,.75]-[:30]-[:60]OH} }{}
14    \reactand[below right]{ \chemfig{-[:30](-[:60]O-[:60]-[:60])
15      -[:60]OH} }{}
16    }{halbacetal}
17    \branch[above=of carbonyl,xshift=5.75em]{
18      \arrow[direction=above]{ \chemfig{[,.75]-[:30]=_[:60]O}/\Hp1 }{}
19      \reactand[above]{ \chemfig{-[:30](-[:60]OH)-[:60]-[:60]=[:60]O
20        } }{}
21    }{aldol}
22  \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.

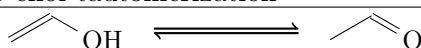
#### Example 8

```

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

```

#### Reaction scheme 1 Keto-enol tautomerization



Here we use the environment

```

1 \begin{rxnscheme}[<keys>]{<caption>}
2   ...
3 \end{rxnscheme}

```

In the command reference (section 4.16) you can read how you can customize.

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

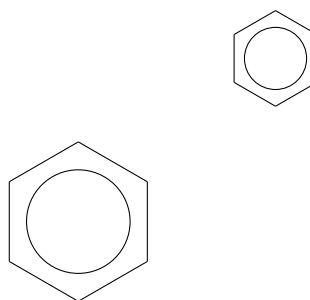
```
1 \setatomsep{1.8em}
2 \setcrambond{3pt}{0.5pt}{1pt}
```

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

---

### Example 9

```
1 \begin{rxn}
2 \reactand{\chemfig{**6(-----)}}{}
3 \end{rxn}
4 \chemfig{**6(-----)}
```



*myChemistry*'s defaults can be changed with these commands:

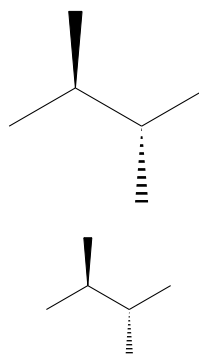
```
1 \bondlength{<length>}
2 \bondshape{<base length>}{<dash thickness>}{<dash spacing>}
3 \atomsiz{<font size>}
```

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

---

### Example 10

```
1 \bondlength{2.1em}\bondshape{5pt}
  {1pt}{2pt}\atomsiz{\Large}
2 \begin{rxn}
3 \reactand{\chemfig
  {-[:30](<[:60])
  -[:60](<[:60]) -[:60]}}{}
4 \end{rxn}
5 \bondlength{}\bondshape{}\atomsiz{}
6 \begin{rxn}
7 \reactand{\chemfig
  {-[:30](<[:60])
  -[:60](<[:60]) -[:60]}}{}
8 \end{rxn}
```



If you only want to change the parameters of a single environment you can use **ChemFig**'s commands and **L<sup>A</sup>T<sub>E</sub>X**'s fontsize commands *inside the environment*.

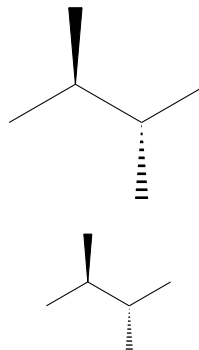
---

**Example 11**

```

1 \begin{rxn}
2 \setatomsep{2.1em}\setcrambond
  {5pt}{1pt}{2pt}\Large
3 \reactand{\chemfig
  {-[:30](<[:60])
  -[: -60](<[: -60]) -[:60]}}{}
4 \end{rxn}
5 \begin{rxn}
6 \reactand{\chemfig
  {-[:30](<[:60])
  -[: -60](<[: -60]) -[:60]}}{}
7 \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:

```
1 \arrowlength{<length>}
```

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

## 2.5 Package options

**myChemistry** has a number of package options.

**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 4.16) is changed into "Reaction scheme".

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

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

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

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

**nochemexec** prevent **myChemistry** from loading 'chemexec'.

**NEW** **nocompounds** prevent **myChemistry** from loading 'chemcompounds'.

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

**NEW** **chemstyle** load the 'chemstyle' package, without conflicts with **myChemistry**.

**new** `nopackages` prevent **myChemistry** from loading *any* package (except **ChemFig**)<sup>1</sup>.

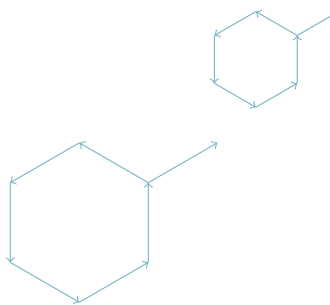
### 3 Advanced Usage, Usage with TikZ

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

---

#### Example 12

```
1 \begin{rxn}
2 \reactand[right,->,green!45!
  blue!55]{\chemfig{*6(---(-)---)} }{}
3 \end{rxn}
4 \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:

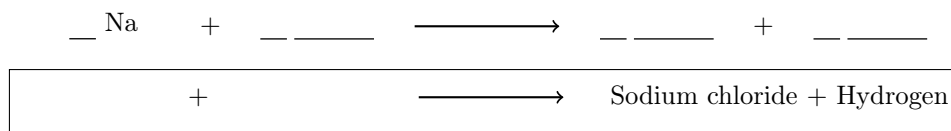
---

#### Example 13

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

#### Ionic compounds (I)

Fill in the missing parts




---

<sup>1</sup>apart from the ones **myChemistry** needs to function (like **TikZ** etc.).

## 4 Alphabetical Command Reference

In the following section every command is explained. In some of the examples the nodes (in the *TikZ* sense), within which the reactands and arrows are set, are emphasized. This is done by giving them a coloured frame through using the command `\makevisible`, see section 4.10.

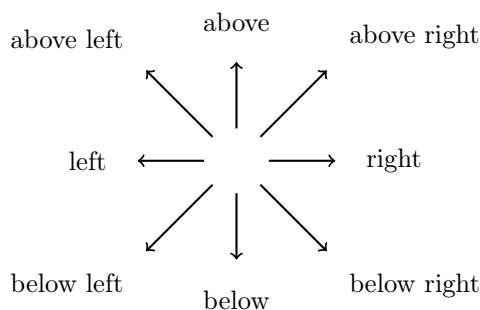
### 4.1 arrow

Reaction arrows are created with `\arrow`.

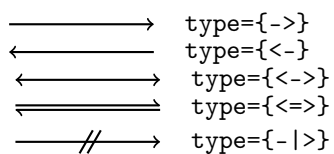
```
1 \arrow[<keys>]{<above>}{<below>}
```

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.

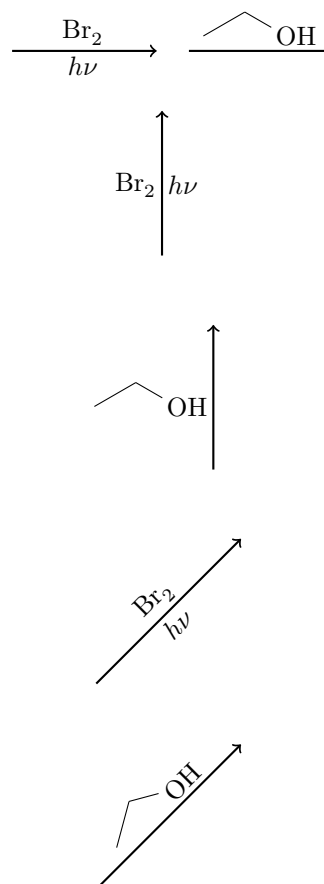
**NEW** **both** – this gives both arrow argument nodes the same width and height.

## Example 14

```

1 \begin{rxn}
2   \dummy\arrow{\ce{Br2}}{\$h\nu$}
3   \arrow{\chemfig{-[:30]-[:-60]
4     OH}}{}
5 \end{rxn}
6 \begin{rxn}
7   \dummy\arrow[direction=above]{\
8     \ce{Br2}}{\$h\nu$}
9   \end{rxn}
10 \begin{rxn}
11   \dummy\arrow[direction=above
12     right]{\ce{Br2}}{\$h\nu$}
13 \end{rxn}
14 \begin{rxn}
15   \dummy\arrow[direction=above
16     right]{\chemfig{-[:30]-[:-60]
17       OH}}{}
18 \end{rxn}

```



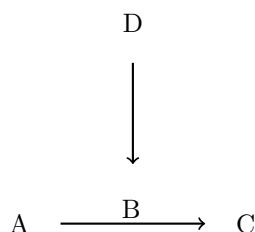
Most keys in action:

## Example 15

```

1 \begin{rxn}
2   \reactand{A}{}
3   \arrow[name=arrow_a]{B}{}
4   \branch[above=of arrow_a,yshift
5     =-4em]{
6     \arrow[type=<- ,direction=
7       above,length=.7]{}{}
8     \reactand[above]{D}{}
9   }{}
10 \reactand{C}{}
11 \end{rxn}

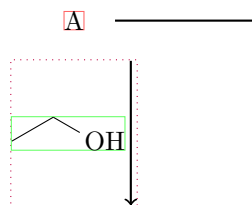
```



If an arrow is placed inside a branch (see section 4.6) 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 16

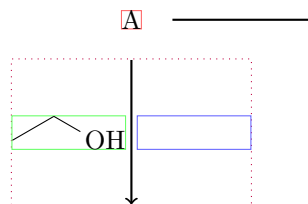
```
1 \makevisible
2 \begin{rxn}
3 \reactand{A}{a}
4 \arrow{}{}
5 \branch[below=of a]{
6 \arrow[direction=below]{\
  chemfig{-[::30]-[::-60]OH}}{}
7 }{}
8 \end{rxn}
9 \makeinvisible
```



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

#### Example 17

```
1 \makevisible
2 \begin{rxn}
3 \reactand{A}{a}
4 \arrow{}{}
5 \branch[below=of a]{
6 \arrow[direction=below,both]{\
  chemfig{-[::30]-[::-60]OH}}{}
7 }{}
8 \end{rxn}
9 \makeinvisible
```



There is more about the alignment problem in section 4.6.1.

## 4.2 arrowlength

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

```
1 \arrowlength{<length>}
```

into `<length>` or `<length> · √2`, respectively. Mind the fact that you have to use a length unit.

## 4.3 atomsize

With

```
1 \atomsiz{<font size>}
```

you can change the font size of the atom groups. Default value is `\small`.

## 4.4 bondlength

With

```
1 \bondlength{<length>}
```

you can change `\setatomsep{<length>}` for all `ChemFig` formulæ *inside* of the `myChemistry` environments. Default value is 1.8 em.

## 4.5 bondshape

With

```
1 \bondshape{<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, .5 pt and 1 pt.

## 4.6 branch

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

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

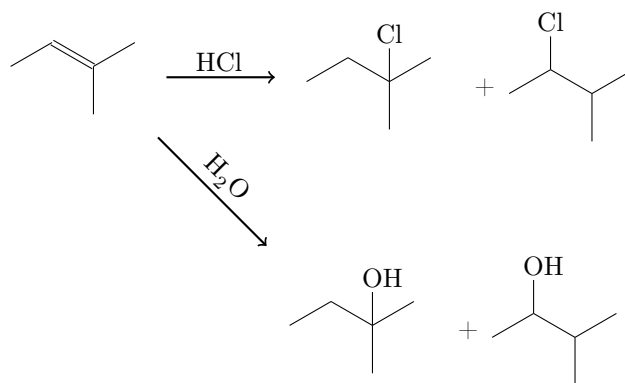
For `\branch` alignment an anchor is important. Let's take a look at an example:

---

### Example 18

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





The first reactand got the anchor **start** (line 2, also see section 4.14).

```
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 19

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

$$\text{CH}_2=\text{CH}-\text{OH} \rightleftharpoons \text{CH}_3-\text{CH}=\text{O}$$

#### 4.6.1 Alignment problems

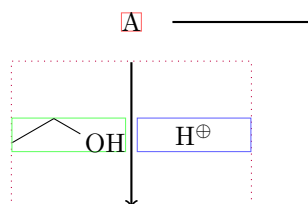
If an arrow has two arguments with different sizes and is placed inside a branch, the alignment of the branch can go wrong. In this case the `\arrow` key **both** isn't a solution, since the smaller argument then is not placed next to the arrow but is centered in its node.

**Example 20**

```

1 \makevisible
2 \begin{rxn}
3 \reactand{A}{a}
4 \arrow{}{}
5 \branch[below=of a]{
6 \arrow[direction=below,both]{\
  chemfig{-[:30]-[: -60]OH}}{\Hp1
7 }{}
8 \end{rxn}
9 \makeinvisible

```



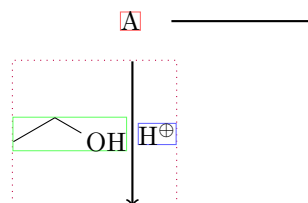
What you have to do is shift the branch using the *TikZ* keys `xshift` and `yshift`.

**Example 21**

```

1 \makevisible
2 \begin{rxn}
3 \reactand{A}{a}
4 \arrow{}{}
5 \branch[below=of a,xshift=-1.35
  em]{
6 \arrow[direction=below]{\
  chemfig{-[:30]-[: -60]OH}}{\Hp1
7 }{}
8 \end{rxn}
9 \makeinvisible

```

**4.7 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 22**

```

1 \begin{rxn}
2 \dummy\arrow{}{}
3 \end{rxn}

```

**4.8 elmove**

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

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

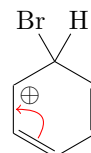
This is expanding the command

```
1 \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 23

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



### 4.9 makeinvisible

**NEW** `\makeinvisible` restores the normal **myChemistry** behaviour after `\makevisible` (see section 4.10) has been used. `\makeinvisible` only changes the looks of nodes following after it.

### 4.10 makevisible

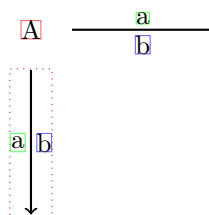
**NEW** With `\makevisible` you can visualize the nodes within which reactands, arrows and branches are set. This is useful when you're aligning branches, for example. You can see an example for `\makevisible` in section 4.1. Every kind of nodes is emphasized with a different colour:

`\reactand{}{}{}`, `\arrow{above}{}{}`, `\arrow{}{below}` und `\branch{}{}{}`. Also see section 4.9.

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

#### Example 24

```
1 \makevisible
2 \begin{rxn}
3 \reactand{A}{a}
4 \arrow{a}{b}
5 \branch[below=of a]{
6   \arrow[direction=below,both]{
7     a}{b}
8 }{}
9 \end{rxn}
10 \makeinvisible
```



### 4.11 marrow

`\marrow` creates a double-headed arrow.

```
1 \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 4.14 or section 4.13).

### 4.12 merge

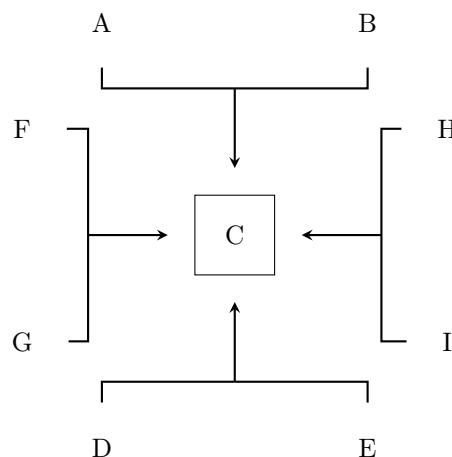
`\merge` cannot only be used in **myChemistry**'s environments but also in a 'tikzpicture'. With `\merge` you can, well, two branches into one. To be able to do that, you need to refer to anchors.

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

---

#### Example 25

```
1 \begin{center}
2 \begin{tikzpicture}
3 \node(a) at (0,0) {A};
4 \node(b) at (10em,0) {B};
5 \node[draw,minimum size=3em](c)
  at (5em,-8em) {C};
6 \merge{c}{a}{b}
7 \node(d) at (0,-16em) {D};
8 \node(e) at (10em,-16em) {E};
9 \merge[direction=above]{c}{d}{e}
10 }
11 \node(f) at (-3em,-4em) {F};
12 \node(g) at (-3em,-12em) {G};
13 \merge[direction=right]{c}{f}{g}
14 }
15 \node(h) at (13em,-4em) {H};
16 \node(i) at (13em,-12em) {I};
17 \merge[direction=left]{c}{h}{i}
18 \end{tikzpicture}
19 \end{center}
```



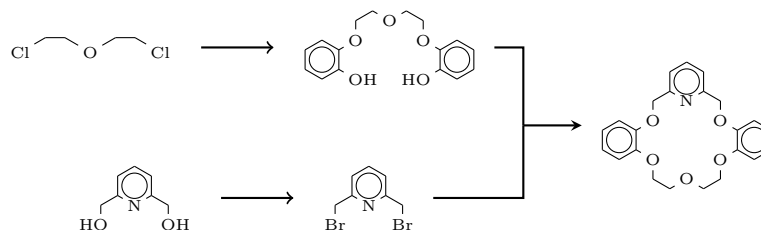
Usage of `\merge` in a **myChemistry** environment:

**Example 26**

```

1 \begin{rxn}
2 \setatomsep{1em}\tiny
3 % branch 1
4 \reactand{ \chemfig{Cl-[:30,1.5]--[: -30,1.5]O-[:30,1.5]--[: -30,1.5]Cl
5 }{ }{above}
6 \arrow{length=.5}{ }{ }
7 \reactand{ \chemfig{O(-[: -150]**6(-----(-OH)-))-[:90]-[:30]-[: -30]O
8 -[:30]-[: -30]-[: -90]O-[: -30]**6(-(-HO)-----)} }{start_above}
9 % branch 2
10 \branch[below=of above,xshift=8em,yshift=-4em]{
11 \reactand{ \chemfig{**6((-[:6,,2]HO)-N-(-[:6]OH)-----)} }{ }
12 \arrow{length=.5}{ }{ }
13 \reactand{ \chemfig{**6((-[:6]Br)-N-(-[:6]Br)-----)} }{ }
14 }{start_below}
15 % target
16 \branch[right=of start_above,xshift=5em,yshift=-4em]{
17 \reactand{ \chemfig{O(-[: -150]**6(-----(-O?) -))-[:90]-[:30]**6(-N
18 -(-[: -90]O-[: -30]**6(-(-O-[:6]-[: -150]-[:150]O-[: -150]-[:150]?)-----))
19 ----)} }{c}
20 }{target}
21 % merging:
22 \merge[direction=right]{target}{start_above}{start_below}
23 \end{rxn}

```



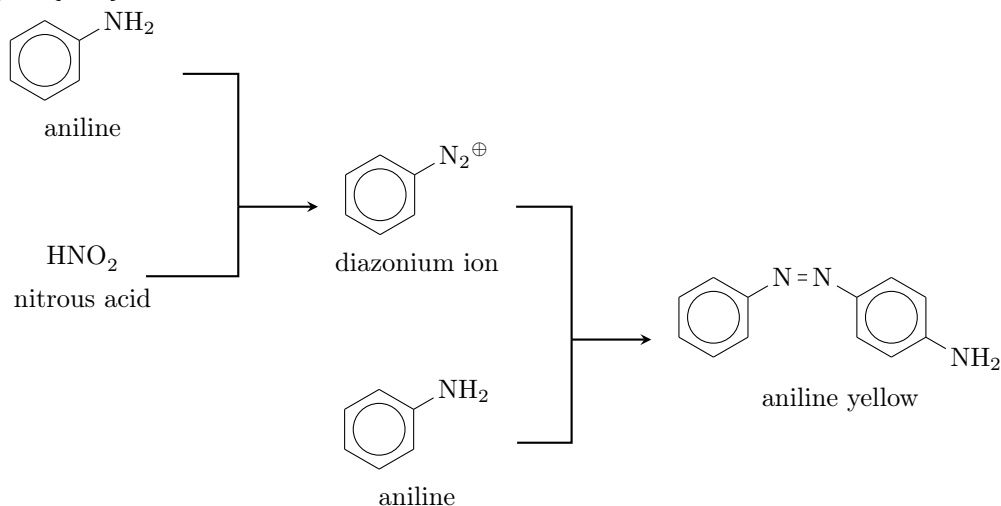
Please note, that you should use branches to refer to, when you use `\merge` in a **myChemistry** environment. Using `\merge` may afford playing with `xshift` and `yshift` until you get the result you want.

**Example 27**

```

1 \begin{rxn}
2 \setatomsep{1.5em}
3 \reactand{ \chemname{\chemfig{**6(---(-NH_2)---)}}{aniline} }{start_aa
  }
4 \reactand[below,yshift=-3em]{ \chemname{\ce{HNO2}}{nitrous acid} }{
  start_ab}
5 \branch[right=of start_aa,xshift=6em,yshift=-5em]{
6   \reactand{ \chemname{\chemfig{**6(---(-N|_2\op)---)}}{diazonium ion}
  }{}
7 }{target_a}% = start_ba
8 \branch[below=of target_a,yshift=-3em]{
9   \reactand{ \chemname{\chemfig{**6(---(-NH_2)---)}}{aniline} }{}
10 }{start_bb}
11 \branch[right=of target_a,xshift=6em,yshift=-5em]{
12   \reactand{ \chemname{\chemfig{N(-[: -150]**6(-----))=N
  -[: -30]**6(---(-NH_2)---)}}{aniline yellow} }{}
13 }{target_b}
14 \merge[direction=right]{target_a}{start_aa}{start_ab}
15 \merge[direction=right]{target_b}{target_a}{start_bb}
16 \end{rxn}

```

**4.13 mesomeric**

The `\mesomeric` command works just like `\reactand` (see section 4.14) but places the formula into square brackets.

```

1 \mesomeric[<alignment>]{<formula>}{<anchor>}

```

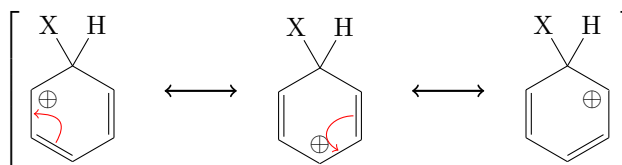
The resonance formulae are written into `<formula>`. With `\marrow` (see section 4.11) you create the resonance arrows. If needed you can give an anchor (`<anchor>`) to `\mesomeric` (also see section 4.6). Alignment is used the same way as with `\reactand`.

**Example 28**

```

1 \begin{rxn}
2 \mesomeric{
3   \reactand{
4     \chemfig{*6(=[@{e1}]--(-[:120]X)(-[:60]H)-(-[:30,.4,,,white]\
oplus)-[@{e2}]})}
5     \elmove{e1}{60:4mm}{e2}{0:4mm}
6   }{}
7   \marrow
8   \reactand{
9     \chemfig{*6(-(-[:90,.4,,,white]\oplus)-[@{e4}]=[@{e3}]-(-[:120]X)
(-[:60]H)-=)}
10    \elmove{e3}{180:4mm}{e4}{150:4mm}
11  }{}
12  \marrow
13  \reactand{
14    \chemfig{*6(-=-(-[:150,.4,,,white]\oplus)-(-[:120]X)(-[:60]H)-=)}
15  }{}
16 }{}
17 \end{rxn}

```



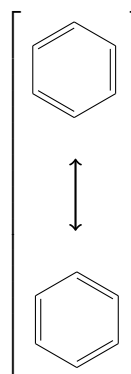
Or vertical, too:

**Example 29**

```

1 \begin{rxn}
2 \mesomeric{
3   \reactand{ \chemfig
{*6(==--==)} }{}
4   \marrow[below]
5   \reactand[below]{ \chemfig
{*6(==--==)} }{}
6 }{}
7 \end{rxn}

```



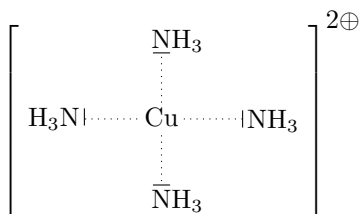
Or maybe a coordination complex?

**Example 30**

```

1 \begin{rxn}
2 \setatomsep{3em}
3 \mesomeric{
4   \reactand{ \chemfig{H_3\lewis
5     {0,N}-[,1.35,,,dotted]{Cu
6     }(-[2,,,,dotted]\lewis{6,N}H_3)
7     (-[6,,,,dotted]\lewis{2,N}H_3)
8     -[,1.2,,,dotted]\lewis{4,N}H_3}
9     }{a}
10  \node[above right=of a,yshift
11    =-1em] {$2\oplus$};
12 \end{rxn}

```

**4.14 reactand**

The command `\reactand` is somehow the basic command of **myChemistry**.

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

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 31**

```

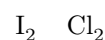
1 horizontal:
2 \begin{rxn}
3   \reactand{\ce{Br2}}{}
4   \reactand[below]{\ce{Cl2}}{}
5 \end{rxn}
6
7 more than one reactand:
8 \begin{rxn}
9   \reactand{\ce{Br2}}{}
10  \reactand[below]{\ce{I2}}{}
11  \reactand{\ce{Cl2}}{}
12 \end{rxn}
13
14 vertical reaction:
15 \begin{rxn}
16   \reactand{\ce{Br-Br}}{}
17   \arrow[length=.5,direction=
18     below]{\h\nu}{}
19   \reactand[below]{\ce{2 ~\lewis
20     {0.,Br}}}{}}
21 \end{rxn}

```

horizontal:



more than one reactand:



vertical reaction:





### 4.15 rxn (environment)

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

```
1 \begin{rxn}[<keys>]
2   ...
3 \end{rxn}
```

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

#### 4.15.1 Options

**NEW**

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 32

```
1 \begin{rxn}[align=center]
2   \reactand{center}{}\arrow{}\{}\reactand{centered}{}
3 \end{rxn}
4 \begin{rxn}[align=right]
5   \reactand{right}{}\arrow{}\{}\reactand{raggedleft}{}
6 \end{rxn}
7 \begin{rxn}[align=left]
8   \reactand{left}{}\arrow{}\{}\reactand{raggedright}{}
9 \end{rxn}
```

center                      →                      centered

right                      →                      raggedleft

left                      →                      raggedright

---

### 4.16 rxnscheme (environment)

\rxnscheme is a floating environment for reaction schemes.

```
1 \begin{rxnscheme}[<keys>]{<caption>}
2   ...
3 \end{rxnscheme}
```

### 4.16.1 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

```
1 \begin{rxnscheme}[label={rs:schema}]{<caption>}
2   ...
3 \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.

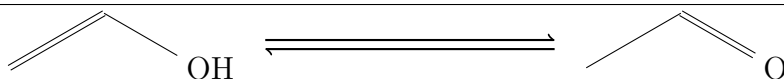
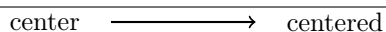
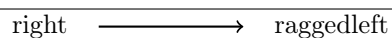
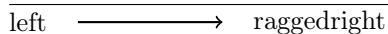
```
1 \begin{rxnscheme}[placement=<placement>]{<caption>}
2   ...
3 \end{rxnscheme}
```

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

---

#### Example 33

```
1 \begin{rxnscheme}[scale=2]{Big scheme}
2   \large\setatomsep{3.5em}
3   \reactand{ \chemfig{=[::30]-[::-60]OH} }{}
4   \arrow[type={<=>}]{}{}
5   \reactand{ \chemfig{-[::30]=[::-60]O} }{}
6 \end{rxnscheme}
7 \begin{rxnscheme}[scale=.5]{Small scheme}
8   \tiny\setatomsep{1em}
9   \reactand{ \chemfig{=[::30]-[::-60]OH} }{}
10  \arrow[type={<=>}]{}{}
11  \reactand{ \chemfig{-[::30]=[::-60]O} }{}
12 \end{rxnscheme}
13 \begin{rxnscheme}{center}
14  \reactand{center}{}\arrow{}{}\reactand{centered}{}
15 \end{rxnscheme}
16 \begin{rxnscheme}[align=right]{right}
17  \reactand{right}{}\arrow{}{}\reactand{raggedleft}{}
18 \end{rxnscheme}
19 \begin{rxnscheme}[align=left]{left}
20  \reactand{left}{}\arrow{}{}\reactand{raggedright}{}
21 \end{rxnscheme}
```

**Reaction scheme 2** Big scheme**Reaction scheme 3** Small scheme**Reaction scheme 4** center**Reaction scheme 5** right**Reaction scheme 6** left**4.16.2 Customizing rxnscheme**

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

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

There are different possible styles, provided 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 two rules, one above and one below, another rule frames the object below; default for **rxnscheme**

**Example 34**

```

1 \begin{rxnscheme}{ruled}
2 \reactand{default style}{}
3 \end{rxnscheme}
4 \floatstyle{boxed}
5 \restylefloat{rxnfloat}
6 \begin{rxnscheme}{boxed}
7 \reactand{framed object}{}
8 \end{rxnscheme}
9 \floatstyle{plain}
10 \restylefloat{rxnfloat}
11 \begin{rxnscheme}{plain}
12 \reactand{without any special formatting}{}
13 \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

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

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

```
1 \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:

```

1 \begin{rxnscheme}[placement=<placement>]{<caption>}
2 ...
3 \end{rxnscheme}

```

**Name** If you want to change the name of reaction scheme<sup>1</sup>, you can do that with

<sup>1</sup>You probably do. You reading the English documentation means probably, that you're not German.

```
1 \settschemename{<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

```
1 \makeatletter
2 \@addtoreset{rxnfloat}{section}
3 \makeatletter
4 \renewcommand{\therxnsscheme}{\arabic{section}.\arabic{rxnsscheme}}
```

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

**List of schemes** By writing

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

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

---

### Example 35

#### Reaction schemes

1	Keto-enol tautomerization . . .	9
2	Big scheme . . . . .	27
3	Small scheme . . . . .	27
4	center . . . . .	27
5	right . . . . .	27
6	left . . . . .	27
7	ruled . . . . .	28
8	boxed . . . . .	28
9	plain . . . . .	28
10	addition reaction . . . . .	31
11	electrophilic substitution . . . .	34
12	synthesis of chrysanthemum acid	44

### 4.17 setrcndist

**NEW**

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

```
1 \setrcndist{<länge>}
```

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

**Example 36**

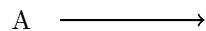
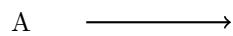

---

```

1 \setrcndist{2em}
2 \begin{rxn}
3 \reactand{A}{}\arrow{}{}
4 \end{rxn}
5 \setrcndist{}
6 \begin{rxn}
7 \reactand{A}{}\arrow{}{}
8 \end{rxn}

```

---

**4.18 setrxnalign/setschemealign****NEW**

With the commands

```

1 \setrxnalign{<alignment>}
2 \setschemealign{<alignment>}

```

The default alignment behaviour of `rxn` and `rxnscheme` (see section 4.15.1 & section 4.16.1) 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 37**

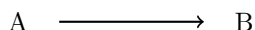
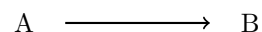

---

```

1 \setrxnalign{right}
2 \begin{rxn}
3 \reactand{A}{}\arrow{}{}\reactand{B}{}
4 \end{rxn}
5 \setrxnalign{}
6 \begin{rxn}
7 \reactand{A}{}\arrow{}{}\reactand{B}{}
8 \end{rxn}

```

---

**4.19 setschemename**

See section 4.16.2.

**4.20 transition**

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

```

1 \transition[<alignment>]{<formula>}{<anchor>}

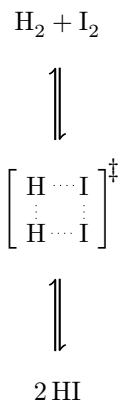
```

**Example 38**

```

1 \begin{rxn}
2 \reactand{ \ce{H2 + I2} }{}
3 \arrow[type=<=>]{length=.5,
  direction=below}{}{}
4 \transition[below]{
5 \reactand{ \chemfig[dotted
  ][] {H?-I-[2]I-[4]H?} }{}
6 }{}
7 \arrow[type=<=>]{length=.5,
  direction=below}{}{}
8 \reactand[below]{ \ce{2 HI} }{}
9 \end{rxn}

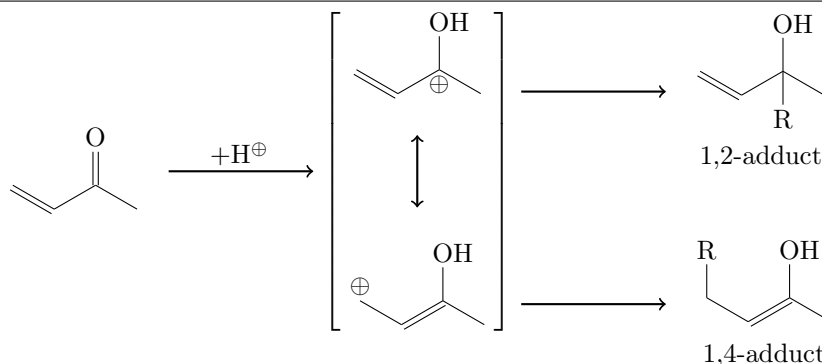
```



## 5 Examples

### 5.1 Addition Reaction

A simple reaction scheme with two different products.

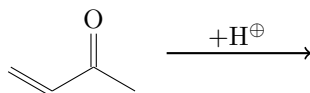
**Reaction scheme 10** addition reaction

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{ $+ \text{\Hp1$} }{}

```

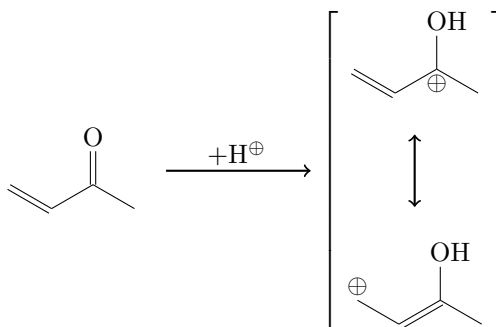


Then we write the resonance formulæ. The `\mesomeric` gets the anchor `rf` (line 7).

```

3 \mesomeric{
4   \reactand{ \chemfig{=[:-30]-[:60](-[:60]OH)(-[::-120,.3,,white
5   ]\oplus)-[::-60]} }{}
6   \marrow[below]
7   \reactand[below]{ \chemfig{\oplus-[6,.3,,white
8   ]-[:-30]=[:60](-[:60]OH)-[::-60]} }{}
9 }{rf}

```

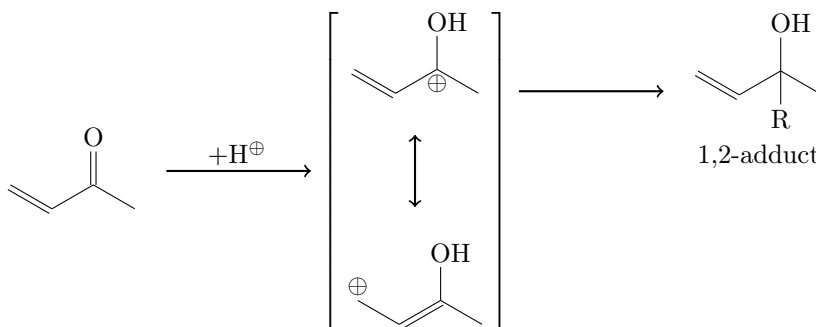


Now comes the 1,2-adduct, placed in a branch referring **rf**, shifted above with **yshift**:

```

8 \branch[right=of rf,yshift=3em]{
9   \arrow{}{}
10  \reactand{ \chemname{\chemfig{=[:-30]-[:60](-[:60]OH)(-[::-120]R
11  )-[::-60]} }{1,2-adduct} }{}

```



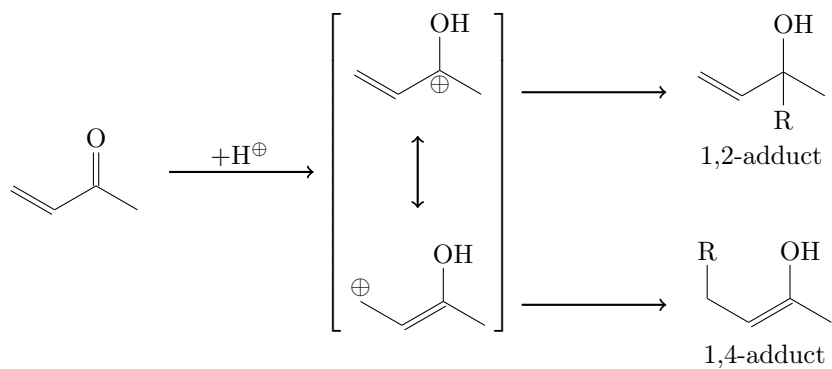
At last we write the branch containing the 1,4-adduct, also referencing **rf**, shifted below with **yshift**:

```

12 \branch[right=of rf,yshift=-5em]{
13   \arrow{}{}
14   \reactand{ \chemname{\chemfig{R-[6]-[:-30]=[:60](-[:60]OH)
15   -[::-60]} }{1,4-adduct} }{}

```





The complete code looks like this:

```

1 \begin{rxnscheme}{addition reaction}
2 \reactand{ \chemfig{=_[: -30] -[::60] (=[::60] O) -[:: -60]} }{}
3 \arrow{ $+ \Hpl$ }{}
4 \mesomeric{
5 \reactand{ \chemfig{=_[: -30] -[::60] (-[::60] OH) (-[:: -120] ,.3,, ,white
] \oplus) -[:: -60]} }{}
6 \marrow[below]
7 \reactand[below]{ \chemfig{\oplus -[6,.3,, ,white
] -[: -30] =_[::60] (-[::60] OH) -[:: -60]} }{}
8 }{rf}
9 \branch[right=of rf,yshift=3em]{
10 \arrow{}{}
11 \reactand{ \chemname{\chemfig{=_[: -30] -[::60] (-[::60] OH) (-[:: -120]
R) -[:: -60]}}{1,2-adduct} }{}
12 }{}
13 \branch[right=of rf,yshift=-5em]{
14 \arrow{}{}
15 \reactand{ \chemname{\chemfig{R -[6] -[: -30] =_[::60] (-[::60] OH)
-[:: -60]}}{1,4-adduct} }{}
16 }{}
17 \end{rxnscheme}}

```

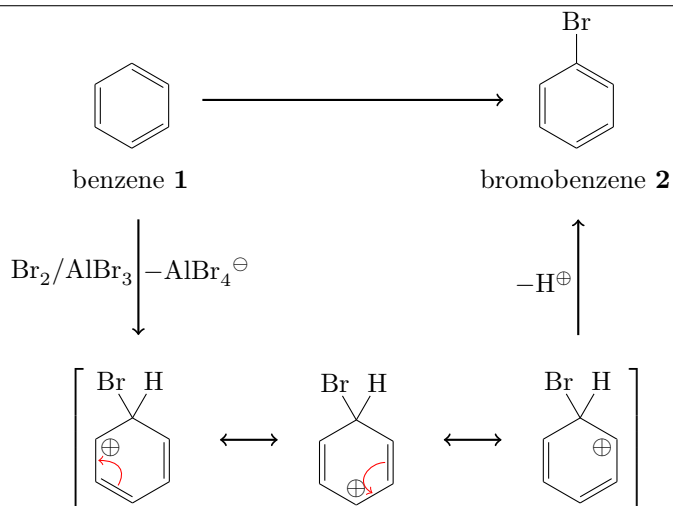
## 5.2 Mesomerism

We want to display the following reaction scheme:

---

**Reaction scheme 11** electrophilic substitution

---

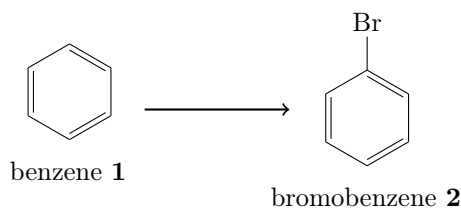


First we write the main reaction. To do so, we use the commands `\reactand`, `\arrow` and the environment `\begin{rxn} ... \end{rxn}`.

```

1 \begin{rxn}
2   \reactand{
3     \chemname{\chemfig{*6(==))==}}{benzene \compound{benzene}}
4   }{}
5   \arrow{}{}
6   \reactand{
7     \chemname{\chemfig{*6(==)=(-Br)=)}}{bromobenzene \compound{
bromobenzene}}
8   }{}
9 \end{rxn}

```



Now we make it a little bit smaller:

```

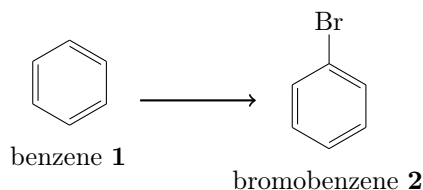
1 \begin{rxn}[scale=.8]
2   \setatomsep{1.6em}
3   \reactand{
4     \chemname{\chemfig{*6(==))==}}{benzene \compound{benzene}}
5   }{}
6   \arrow{}{}

```

```

7 \reactand{
8   \chemname{\chemfig{*6(==(-Br)-)}}{bromobenzene \compound{
  bromobenzene}}
9   }{}
10 \end{rxn}

```



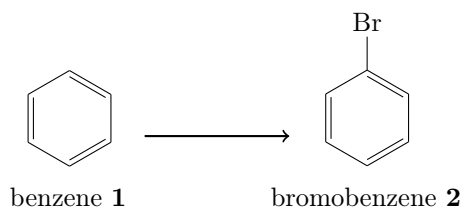
We have two possibilities to align both benzene rings at the same height. Either we shift the second one up using *TikZ* code:

```

7 \reactand[right,yshift=1em]{
8   \chemname{\chemfig{*6(==(-Br)-)}}{bromobenzene \compound{
  bromobenzene}}
9   }{}

```

Please notice, that you have to write the alignment-command of the second reactand **explicitly** in order to use *TikZ*. This is not the best solution, because the arrow isn't centered with respect to the rings.

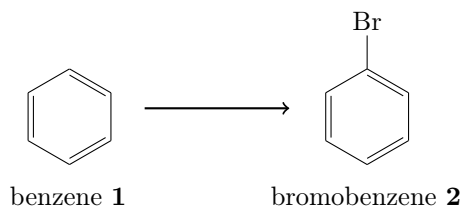


The second possibility would be to shift the first ring down. We can't achieve that by using *TikZ* code, because the following arrow and reactand align themselves with respect to the reactand or arrow directly before. But we can write an invisible bromine to the first benzene to do the trick:

```

3 \reactand{
4   \chemname{\chemfig{*6(==(-[,,,,white]\phantom{Br})-)}}{benzene \
  compound{benzene}}
5   }{}

```



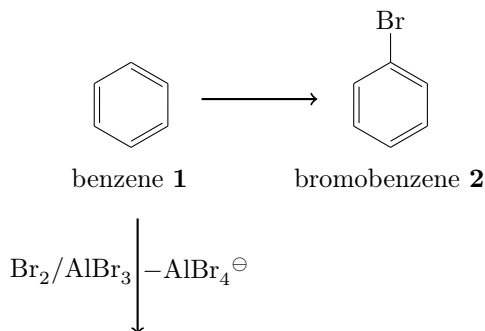
We need to give the first reactand an anchor in order to refer to it with the branch.

```

1 \begin{rxn}[scale=.8]
2 \setatomsep{1.6em}
3 \reactand{
4 \chemname{\chemfig{*6(==(-[,,,white]\phantom{Br})=)}}{benzene \
compound{benzene}}
5 }{start}
6
7 \branch[below=of start]{
8 \arrow[direction=below]{\ce{Br2 / AlBr3}}{\ce{AlBr4\om}$ }
9 }{}
10
11 \arrow{}{}
12 \reactand{
13 \chemname{\chemfig{*6(==(-Br)=)}}{bromobenzene \compound{
bromobenzene}}
14 }{}
15 \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`.

```

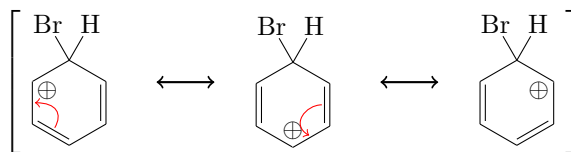
1 \mesomeric{
2 \reactand{
3 \chemfig{*6(=[@{e1}]--(-[:120]Br)(-[:60]H)-(-[:30],.4,,white)\
oplus)-[@{e2}]})}
4 \elmove{e1}{60:4mm}{e2}{0:4mm}
5 }{}
6 \marrow
7 \reactand{
8 \chemfig{*6(-(-[:90],.4,,white)\oplus)-[@{e4}]=[@{e3}]-(-[:120]Br
)(-[:60]H)=)}
9 \elmove{e3}{180:4mm}{e4}{150:4mm}
10 }{}
11 \marrow
12 \reactand{
13 \chemfig{*6(==(-[:150],.4,,white)\oplus)-(-[:120]Br)(-[:60]H)
=)}

```

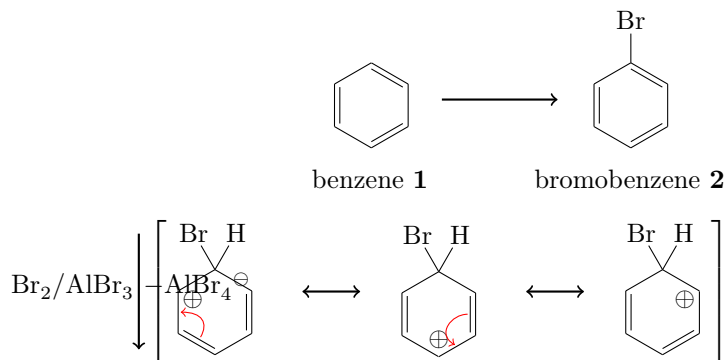
```

14   {}
15   {}

```



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:

```

1  \begin{rxn}[scale=.8]
2  \setatomsep{1.6em}
3  \reactand{
4  \chemname{\chemfig{*6(==(-[,,,,white]\phantom{Br})-=)}}{benzene \
compound{benzene}}
5  }{start}
6
7  \branch[below=of start]{
8  \arrow[direction=below]{\ce{Br2 / AlBr3}}{\$-\ce{AlBr4\om}}\$}
9  \mesomeric[below]{
10 \reactand{
11 \chemfig{*6(=[@{e1}]--(-[:120]Br)(-[:60]H)-(-[:30],.4,,,
white)\oplus)-[@{e2}]})}
12 \elmove{e1}{60:4mm}{e2}{0:4mm}
13 }{}
14 \marrow
15 \reactand{
16 \chemfig{*6(-(-[:90],.4,,,white)\oplus)-[@{e4}]=[@{e
3}]-(-[:120]Br)(-[:60]H)-=)}
17 \elmove{e3}{180:4mm}{e4}{150:4mm}
18 }{}
19 \marrow
20 \reactand{
21 \chemfig{*6(==(-[:150],.4,,,white)\oplus)-(-[:120]Br)(-[:60]
H)-=)}
22 }{}

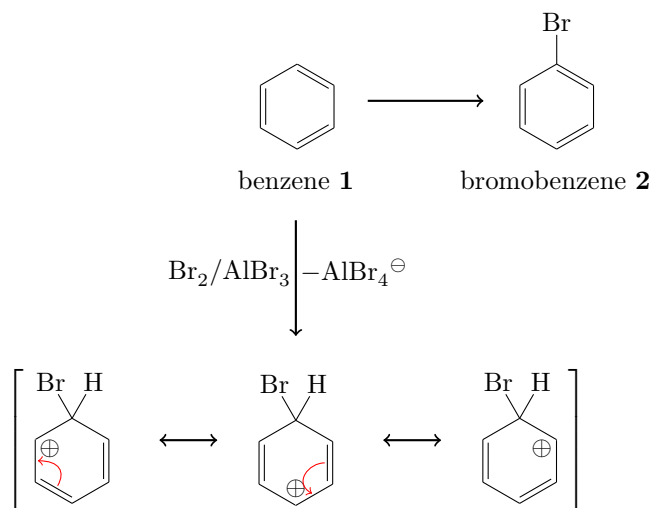
```

```

23     {}{}
24     {}{}
25
26     \arrow{}{}
27     \reactand{
28         \chemname{\chemfig{*6(==(-Br)-)}}{bromobenzene \compound{
bromobenzene}}
29     }{}
30     \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 `iot` into a branch of its own.

```

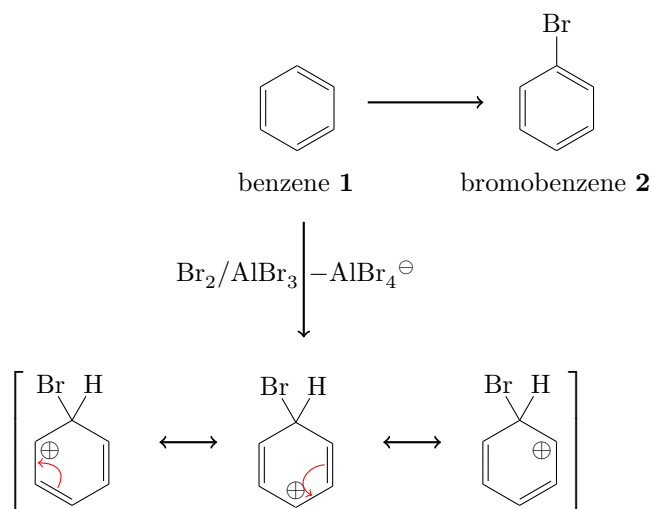
6     ...
7     \branch[below=of start]{
8         \arrow[direction=below,name=arrow_a]{\ce{Br2 / AlBr3}}{\$-\ce{AlBr4\
om}}{\$}
9     }{}
10    \branch[below=of arrow_a]{
11        \mesomeric{
12            \reactand{
13                \chemfig{*6(=[@{e1}] ==(-[:120]Br)(-[:60]H)-(-[:30],.4,,white
])\oplus)-[@{e2}]})}
14                \elmove{e1}{60:4mm}{e2}{0:4mm}
15            }{}
16            \marrow
17            \reactand{
18                \chemfig{*6(-(-[:90,.4,,white]\oplus)-[@{e4}]=[@{e3}]-(-[:120]
Br)(-[:60]H)-=)}
19                \elmove{e3}{180:4mm}{e4}{150:4mm}
20            }{}
21            \marrow
22            \reactand{
23                \chemfig{*6(==(-[:150,.4,,white]\oplus)-(-[:120]Br)(-[:60]H)
==)}

```

```

24     }{}
25     }{}
26     }{}
27     ...

```

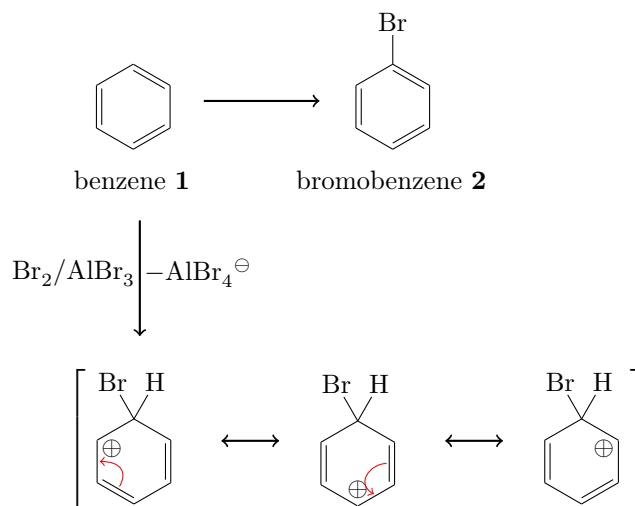


In first sight this isn't better. But by shifting the branch with xshift, we get what we want:

```

6     ...
7     \branch[below=of start]{
8         \arrow[direction=below,name=arrow_a]{\ce{Br2 / AlBr3}}{\$-\ce{AlBr4\
om}}{\$}
9     }{}
10    \branch[below=of arrow_a,xshift=8.5em]{
11        \mesomeric{
12            \reactand{
13                \chemfig{*6(=[@{e1}]--(-[:120]Br)(-[:60]H)-(-[:30,.4,,,white
14                ]\oplus)-[@{e2}])}
15            \elmove{e1}{60:4mm}{e2}{0:4mm}
16        }{}
17        \marrow
18        \reactand{
19            \chemfig{*6(-(-[:90,.4,,,white]\oplus)-[@{e4}]=[@{e3}]-(-[:120]
20            Br)(-[:60]H)-=)}
21            \elmove{e3}{180:4mm}{e4}{150:4mm}
22        }{}
23        \marrow
24        \reactand{
25            \chemfig{*6(-(-[:150,.4,,,white]\oplus)-(-[:120]Br)(-[:60]H)
26            -=)}
27        }{}
28    }{}
29    }{}
30    }{}
31    ...

```



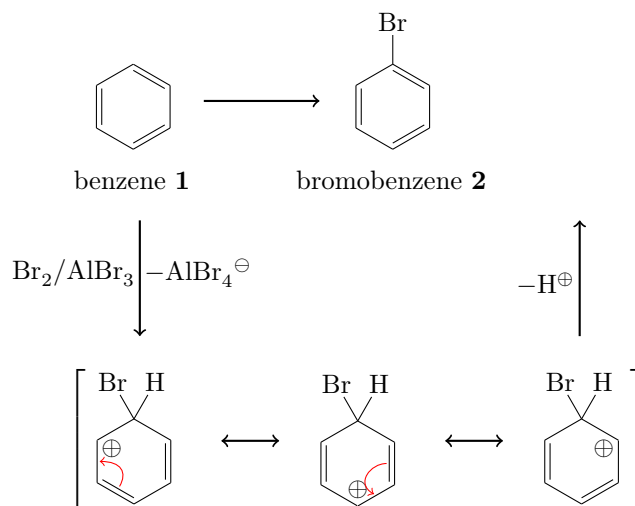
The last arrow is also placed in its own branch, so we can shift it, where we want.

```

6   ...
7   \branch[below=of start]{
8     \arrow[direction=below,name=arrow_a]{\ce{Br2 / AlBr3}}{\$-\ce{AlBr4\
om}\$}
9   }{}
10  \branch[below=of arrow_a,xshift=8.5em]{
11    \mesomeric{
12      \reactand{
13        \chemfig{*6(=[@{e1}]--(-[:120]Br)(-[:60]H)-(-[:30],.4,,,white
14        ]\oplus)-[@{e2}])}
15        \elmove{e1}{60:4mm}{e2}{0:4mm}
16      }{}
17      \marrow
18      \reactand{
19        \chemfig{*6(-(-[:90],.4,,,white)\oplus)-[@{e4}]=[@{e3}]-(-[:120]
20        Br)(-[:60]H)-=)}
21        \elmove{e3}{180:4mm}{e4}{150:4mm}
22      }{}
23      \marrow
24      \reactand{
25        \chemfig{*6(-(-[:150],.4,,,white)\oplus)-(-[:120]Br)(-[:60]H)
26        -=)}
27      }{}
28    }{}
29  }{}
30  }{}
31  \branch[above=of mesomerism,xshift=7.25em]{
32    \arrow[direction=above]{\$-\Hpl\$}{}
33  }{}
34  ...

```





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

```

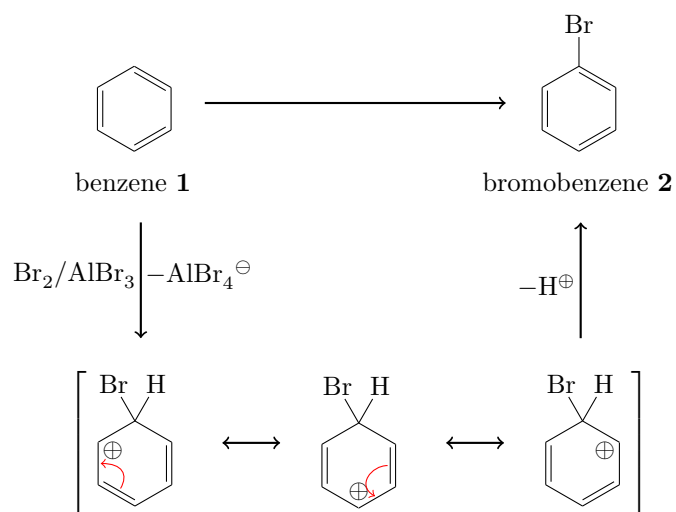
1 \begin{rxn}[scale=.8]
2 \setatomsep{1.6em}
3 \reactand{\chemname{\chemfig{*6(==(-[,,,white]\phantom{Br})=)}}{
benzene \compound{benzene}}} \start{
4
5 \branch[below=of start]{
6 \arrow[direction=below,name=arrow_a]{\ce{Br2 / AlBr3}}{\$-\ce{AlBr4\
om}}\$}
7 }{}
8 \branch[below=of arrow_a,xshift=8.5em]{
9 \mesomeric{
10 \reactand{
11 \chemfig{*6(=[@{e1}]--(-[:120]Br)(-[:60]H)-(-[:30,.4,,,white
]\oplus)-[@{e2}]})}
12 \elmove{e1}{60:4mm}{e2}{0:4mm}
13 }{}
14 \marrow
15 \reactand{
16 \chemfig{*6(-(-[:90,.4,,,white]\oplus)-[@{e4}]=[@{e3}]-(-[:120]
Br)(-[:60]H)=)}
17 \elmove{e3}{180:4mm}{e4}{150:4mm}
18 }{}
19 \marrow
20 \reactand{
21 \chemfig{*6(==(-[:150,.4,,,white]\oplus)-(-[:120]Br)(-[:60]H)
-)}
22 }{}
23 }{}
24 }{mesomerism}
25 \branch[above=of mesomerism,xshift=7.25em]{
26 \arrow[direction=above]{\$-\Hpl\$}{
27 }{}
28
29 \arrow[length=2.6]{}{}

```

```

30 \reactand{ \chemname{\chemfig{*6(==(-Br)-)}}{bromobenzene \
  compound{bromobenzene}} }{}
31 \end{rxn}

```



### 5.3 Creating an extensive synthesis using TikZ, myChemistry and ChemFig

Since the commands of **ChemFig** are functioning well in a 'tikzpicture' 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`.

```

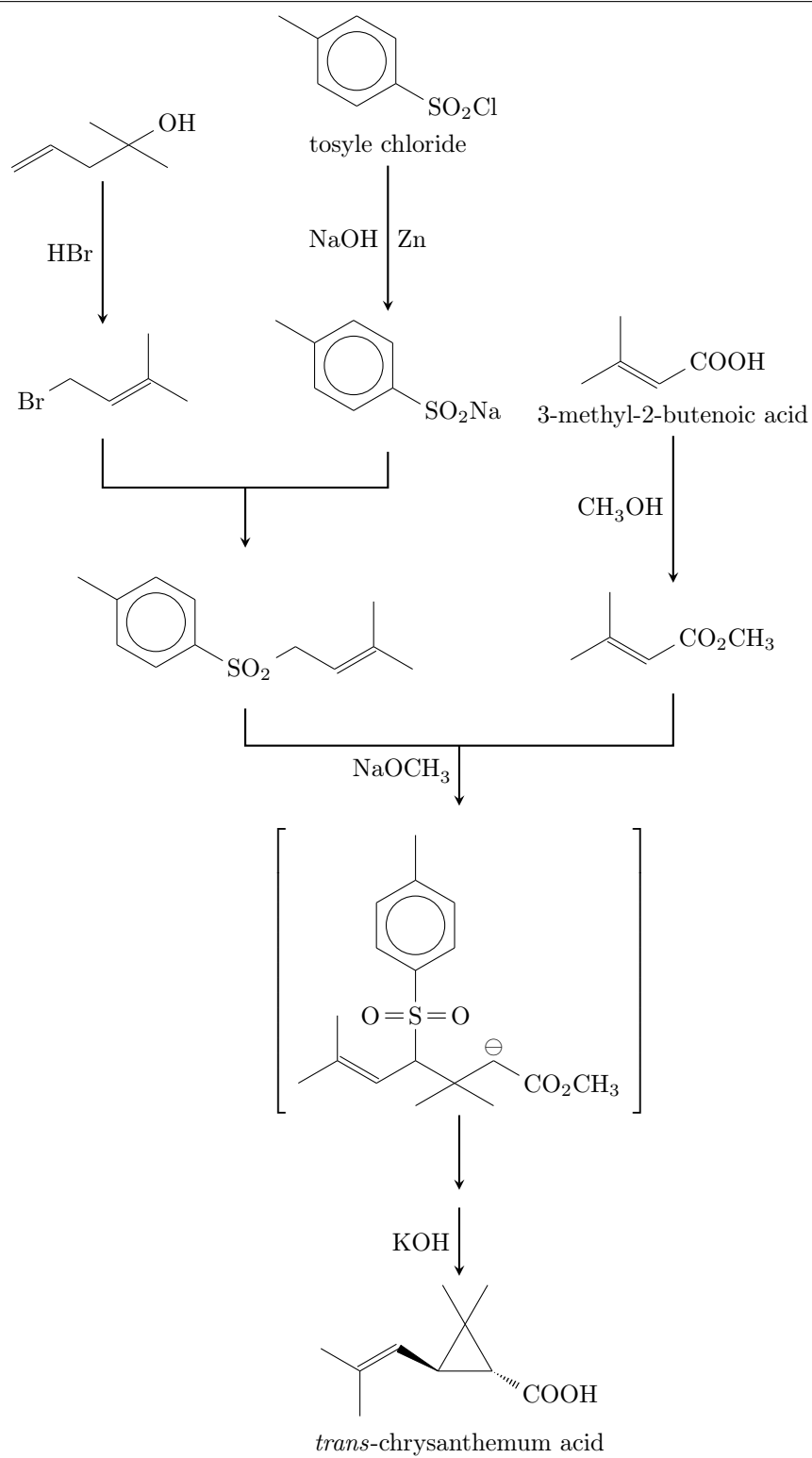
1
2 \begin{rxnfloat}
3 \setatomsep{1.8em}\setcrambond{3pt}{.5pt}{1pt}
4 \centering
5 \begin{tikzpicture}[scale=.8]
6 \small
7 \node(a) at (0,0) {\chemfig{=_{[:30]-[:60]-[:60](-[:60])
8 (-[:120])-[:0]OH}}};
9 \node(b) at (0,-4) {\chemfig{Br-[:30]-[:60]=_{[:60](-[:60])
10 -[:60]}}};
11 \draw[-stealth,thick] (a.south) -- node[left]{HBr} (b.north);
12 \node(c) at (5,1) {\chemname{\chemfig{**6(--(-SO_2Cl)---(-)-)}}{
13 tostyle chloride}};
14 \node(d) at (5,-4) {\chemfig{**6(--(-SO_2Na)---(-)-)}};
15 \draw[-stealth,thick] (c.south) -- node[left]{NaOH} node[right]{Zn}
16 (d.north);
17 \node(e) at (2.5,-8.5) {\chemfig{**6(--(-SO
18 _2-[:30]-[:60]=_{[:60](-[:60])---(-)-)}}};
19 \node(f) at (10,-4) {\chemname{\chemfig{-[:30](-[:60])
20 =_{[:60]-[:60]COOH}}}{3-methyl-2-butenic acid}};
21 \node(g) at (10,-8.5) {\chemfig{-[:30](-[:60])=_{[:60]-[:60]CO_2
22 CH_3}}};
23 \draw[-stealth,thick] (f.south) -- node[left]{\ce{CH3OH}} (g.north);
24 \merge{e}{b}{d}
25 \node[left delimiter={},right delimiter={}](h) at (6.25,-14.5) {\
26 chemfig{-[:30](-[:60])=^[:60]-[:60](-[:60]S(=[:90]O)(=[:90]O)
27 -[:0]**6(---(-)---)-[:60](-[:0])(-[:120])-[:60](-[:60,.5,,
28 white)\ominus)-[:60]CO_2CH_3}};
29 \node at (5.25,-11) {\ce{NaOCH3}};
30 \merge{h}{e}{g}
31 \node(i) at (6.25,-18.5) {};
32 \node(j) at (6.25,-21.5) {\chemname{\chemfig{-[:30](-[:60])
33 =^[:60]>[:60](-[:90,1.2])-[:30,1.2](-[:120,1.2](-[:60])-[[:0])
34 <[:30]COOH}}}{\emph{trans}-chrysanthemum acid}};
35 \draw[-stealth,thick] (h.south) -- (i.north);
36 \draw[-stealth,thick] (i.south) -- node[left]{KOH} (j.north);
37 \end{tikzpicture}
38 \caption{synthesis of chrysanthemum acid}
39 \end{rxnfloat}

```

---

**Reaction scheme 12** synthesis of chrysanthemum acid

---



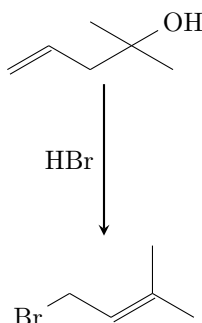
Let's go through the code, piece by piece.

```

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 too big. In lines 7 – 9 the first two reactands are written (lines 7 and 8) and connected with an arrow (line 9).

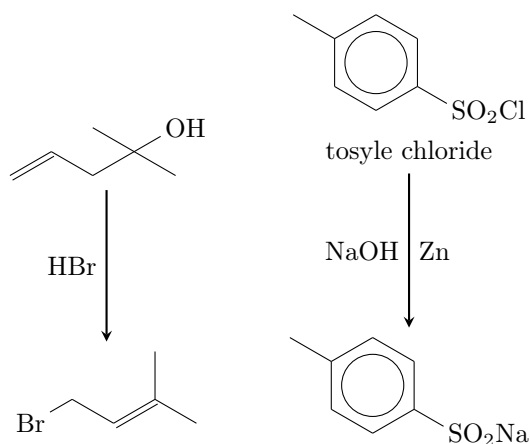


```

10 \node(c) at (5,1) {\chemname{\chemfig{**6(--(-SO_2Cl)---(-)-)}}{
  tosyl 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.

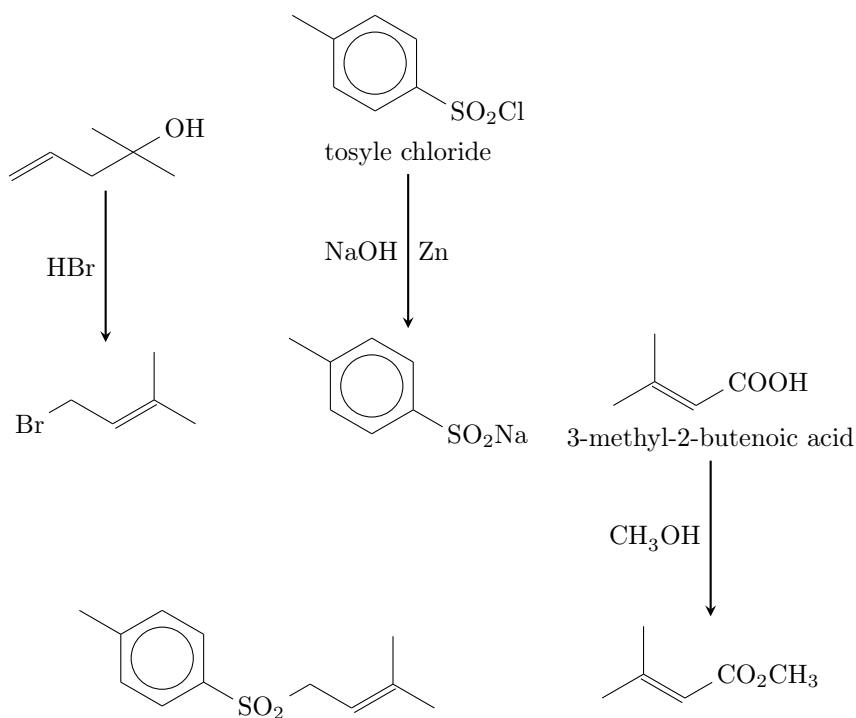


```

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-butenic acid}};
15 \node(g) at (10,-8.5) {\chemfig{-[::30](-[::60])=_[::-60]-[::60]CO_2
   CH_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.

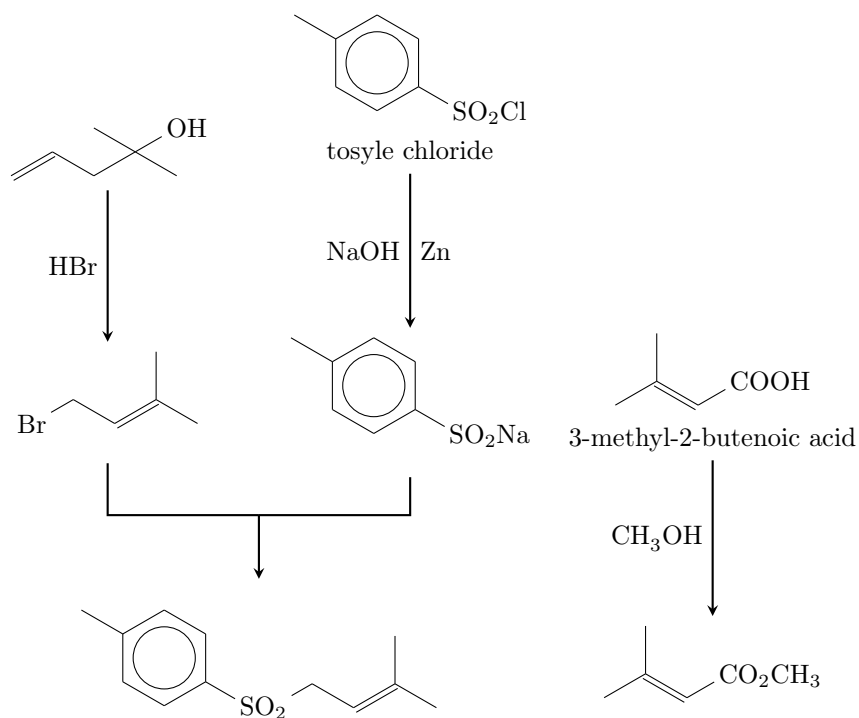


```

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

```

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

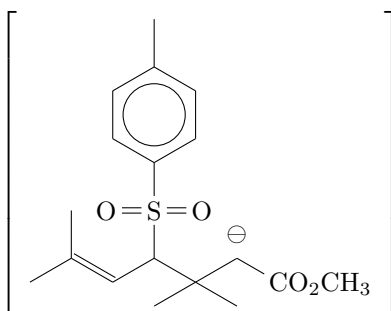
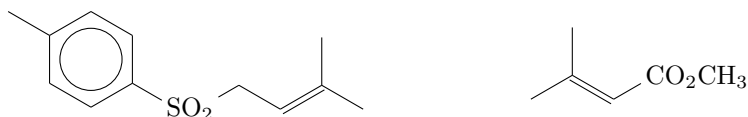


```

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.

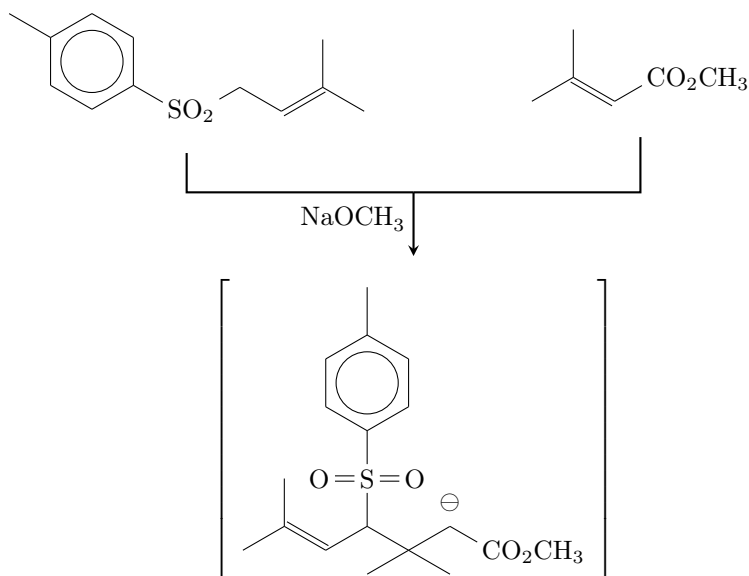


```

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 and the merging arrow gets its reactant argument.

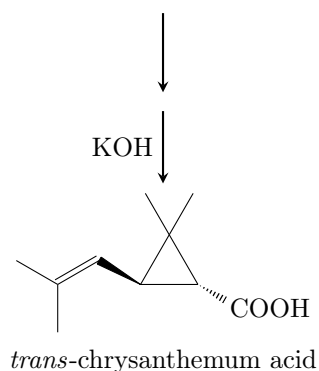


```

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.





## 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 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 documentation anyway.

Have fun with myChemistry!

Clemens Niederberger, Berlin, March 20th 2011