

myChemistry

v1.3

04. April 2011
Clemens Niederberger

Reaktionsschemata mit L^AT_EX und ChemFig erstellen

```
\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=1cm]{\ce{H2O}}
\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[below right]{
\chemfig{C(-[4]*6(==---))(-[2]*6(==---))(-[6,,,2]HO)-C(-[2]CH_3)
(-[6]OH)-CH_3}
}
}
\branch[below right=of a]{
\arrow[type=-|>,direction=below right,length=1cm]{\ce{H2O}}
\reactand[below right]{
\chemfig{C(-[4]*6(==---))(-[2]*6(==---))(-[6,,,2]HO)-C(-[2]CH_3)
(-[6]OH)-CH_3}
\chembelow{O}{\oplus})-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})-C(-[2]CH_3)-CH_3}
}
}
\end{rxn}
```

The image displays two chemical reaction schemes. The first scheme shows the dehydration of a dihydroxy ketone to a carbocation. The reactant is a dihydroxy ketone with a central carbon atom bonded to two phenyl rings, a methyl group, and two hydroxyl groups. The reaction arrow is labeled $-\text{H}_2\text{O}$. The product is a carbocation where one of the hydroxyl groups has been removed, leaving a positive charge on the central carbon atom. The second scheme shows the dehydration of a dihydroxy ketone to a carbocation. The reactant is a dihydroxy ketone with a central carbon atom bonded to two phenyl rings, a methyl group, and two hydroxyl groups. The reaction arrow is labeled $-\text{H}_2\text{O}$. The product is a carbocation where one of the hydroxyl groups has been removed, leaving a positive charge on the central carbon atom.

Inhaltsverzeichnis

ABSCHNITT 1 Über	3
1.1 Änderungen	3
<i>Version 1.2, 3 • Version 1.3, 3</i>	
1.2 Lizenz	4
1.3 Voraussetzungen	4
1.4 Die Idee	5
ABSCHNITT 2 Verwendung	6
2.1 Hintergrund	6
2.2 Das Grundprinzip	6
2.3 Wie funktioniert's?	7
<i>Basisbefehle, 7 • Verzweigungen, 8 • Nummerierte Schemata, 10</i>	
2.4 Voreinstellungen	11
2.5 Paket-Optionen	12
ABSCHNITT 3 Fortgeschrittene Anwendung, Verwendung von TikZ	13
3.1 Die Ausrichtungsfrage	13
3.2 Mit TikZ andere Ziele erreichen	17
ABSCHNITT 4 Alphabetische Befehlsreferenz	18
4.1 arrow	18
4.2 branch	20
<i>Ausrichtungsprobleme, 22</i>	
4.3 chemand	22
4.4 dummy	23
4.5 elmove	23
4.6 makeinvisible	24
4.7 makevisible	24
4.8 marrow	25
4.9 mCsetup	25
4.10 merge	26
4.11 mesomeric	28
4.12 reactand	30
4.13 rxn (Umgebung)	31
<i>Optionen, 31</i>	
4.14 rxnscheme (Umgebung)	32
<i>Optionen, 32 • rxnscheme anpassen, 33</i>	
4.15 setarrowlength	36

4.16	setatomsizes	36
4.17	setbondlength	36
4.18	setbondshape	36
4.19	setrendist	37
4.20	setrxnalign/setschemealign	37
4.21	setschemename	38
4.22	transition	38
<hr/>		
ABSCHNITT 5 Beispiele		38
5.1	Addition	38
5.2	Mesomerie	41
5.3	Synthese mit TikZ	51
<hr/>		
ABSCHNITT 6 Nachwort		58

1 Über

1.1 Änderungen

Neue Befehle oder Veränderungen wurden mit **seit V** gefolgt von der Versionsnummer markiert, z. B. **seit V1.2**. Die neuesten Änderungen sind mit **Neu in V1.3** gekennzeichnet.

1.1.1 Version 1.2

seit V1.2

Neben einigen Bugfixes gibt es in Version v1.3 ein paar Neuerungen. Insbesondere wurde das fehlerhafte Verhalten bei der Ausrichtung von Branches sowie das seltsame Verhalten von Pfeilbeschriftungen, wenn man die Pfeillänge änderte, korrigiert. **Diese Veränderung hat zur Folge, dass myChemistry nun TikZ oder vielmehr pgf in der Version 2.10 benötigt** (siehe [Abschnitt 1.3](#)).

Während sich diese Neuerungen im Hintergrund abspielen, gibt es auch ein paar Neuerungen für die Bedienung. Zum Beispiel gibt es nun einige Paketoptionen, um die automatisch eingebundenen Pakete besser zu handhaben ([Abschnitt 2.5](#)). Außerdem haben die Pfeile zwei neue Keys bekommen (siehe [Abschnitt 4.1](#)).

Auch die Umgebungen haben nun ein paar Möglichkeiten mehr, den persönlichen Vorstellungen angepasst zu werden (siehe [Abschnitt 4.13.1](#), [Abschnitt 4.14.1](#) und [Abschnitt 4.19](#)).

Nicht zuletzt steht **myChemistry** ab v1.2 nun unter der LPPL Version 1.3 oder später.

1.1.2 Version 1.3

Neu in V1.3

Die Befehle `\branch`, `\mesomeric`, `\reactand` und `\transition` können als optionales Argument neben der Ausrichtung auch TikZ-Keys erhalten. Das zweite Argument ist nun ebenfalls als Option einzusetzen. Damit ist das erste optionale Argument immer

noch die Ausrichtung, das zweite der Anker und in das dritte können beliebige `TikZ`-Keys eingesetzt werden.

```
1 \befehl [<ausrichtung>,<anker>,<tikz>] {}
```

Bis Version 1.2 musste die Ausrichtung explizit angegeben werden, auch wenn die Default-einstellung verwendet werden sollte, falls man `TikZ`-Keys verwenden wollte. Das ist seit v1.3 nicht mehr nötig.

```
1 % bisher:
2 \reactand{\ce{Br2}}{\arrow{$h\nu$}}\reactand[right,
3   draw,inner sep=5pt]{\ce{2 \lewis{0.,Br}}}{
4 % jetzt:
5 \reactand{\ce{Br2}}\arrow{$h\nu$}}\reactand[, ,draw,
6   inner sep=5pt]{\ce{2 \lewis{0.,Br}}}
```

Die Voreinstellungsbefehle wurden umbenannt und der Befehl `\mCsetup` hinzugefügt, mit dem die Voreinstellungen mit einer Schnittstelle gehandhabt werden können. Siehe [Abschnitt 4.15](#), [Abschnitt 4.16](#), [Abschnitt 4.17](#), [Abschnitt 4.18](#) und [Abschnitt 4.9](#).

Es gibt den neuen Befehl `chemand`, der ein `+` erzeugt, siehe [Abschnitt 4.3](#).

Wenn man `ChemFig` v0.4 verwendet, bindet `myChemistry` die Datei `bondwidth.tex` ein, die den Befehl `\setbondwidth{<dicke>}` bereitstellt. Damit kann die Liniendicke der Bindungen von `ChemFig`-Formeln verändert werden.

Und auch das ist vielleicht ganz angenehm: alle `myChemistry`-Befehle in den Listings sind nun anklickbare Hyperlinks, die auf ihre Beschreibung in der Befehlsreferenz verweisen.

1.2 Lizenz

`myChemistry` v1.3 steht unter der LaTeX Project Public License Version 1.3 oder später. (<http://www.latex-project.org/lppl.txt>)

1.3 Voraussetzungen

Damit `myChemistry` funktionieren kann, müssen ein paar Pakete installiert sein:

`ChemFig` ohne das ergibt die ganze Sache gar keinen Sinn;

`ifthen` für interne Abfragen;

`calc` für interne Berechnungen;

`xkeyval` Paketoptionen und Befehl-Keys werden damit erstellt;

`float` damit wird die `rxnscheme`-Umgebung definiert;

pgf/TikZ pgf ist nicht nur ein Paket sondern eine ganze Reihe von Paketen. Sie stellen die gesamte Basis für TikZ da. Damit myChemistry funktionieren kann, muss mindestens die Version vom 08.09.2010¹ installiert sein.

Genauer: der Befehl `\pgfpositionnodelater` muss verfügbar sein. Noch genauer benötigt der Key `both` des Befehls `\arrow` diese Version. Wenn Sie den Key nicht verwenden, sollte myChemistry auch mit pgf v2.00 problemlos funktionieren. Ältere Versionen wurden nicht getestet.

1.4 Die Idee

Seit August 2010 steht mit **ChemFig** eine wirklich flexible Lösung für organische Strukturformeln zur Verfügung. So kann man nun durch das Einbinden von **ChemFig** und 'mhchem' mehr oder weniger alle Struktur- und Summenformeln, die man als Chemiker so benötigt, mit L^AT_EX setzen. Was **ChemFig** gegenüber 'ochem' noch benachteiligt, ist das Erstellen richtiger Reaktionsmechanismen. Hier soll myChemistry Abhilfe schaffen.

myChemistry bindet die Pakete

- **ChemFig**²,
- wenn vorhanden 'mhchem'³ in der Version 3,
- wenn vorhanden 'chemexec'⁴ und
- wenn vorhanden 'chemcompounds'⁵ ein.

Zur Funktion der Befehle der oben genannten Pakete siehe deren Dokumentation. Wenn Sie die Pakete separat laden wollen, weil Sie ihnen Optionen mitgeben wollen, dann sollten Sie das machen, *bevor* Sie myChemistry laden, um Konflikte zu vermeiden. myChemistry prüft intern einerseits darauf, ob die Pakete installiert sind und falls ja, ob sie bereits geladen sind. Wenn nicht, werden sie von myChemistry aufgerufen.

Befehle, die durch die eingebundenen Pakete zur Verfügung stehen, sind unter anderem

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

In den Beispielen in diesem Manual wurden Befehle dieser Pakete verwendet *ohne sie speziell als solche zu kennzeichnen*.

Vor allem stellt myChemistry Befehle zum Erstellen von Reaktionsschemata zur Verfügung.

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

²von Christian Tellechea, <http://www.ctan.org/tex-archive/macros/generic/chemfig/>

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

⁴von mir, <http://www.ctan.org/tex-archive/macros/latex/contrib/chemexec/>

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

2 Verwendung

2.1 Hintergrund

myChemistry stellt zwei Umgebungen zur Verfügung, innerhalb derer die Reaktionsmechanismen erstellt werden. Beide Umgebungen sind letztlich eine ‘tikzpicture’-Umgebung. Die Frage, die sich aufdrängt, ist natürlich: wozu? **ChemFig** bringt doch schon einiges an Möglichkeiten mit. Und mit **TikZ** hat man wirklich alle Möglichkeiten offen. Zugegeben. Allerdings bin ich faul, also habe einige häufig verwendete **TikZ**-Befehle zu Makros zusammengefasst. Die sind immer mehr geworden und haben immer mehr Feinheiten erhalten, so dass dieses Paket dabei herausgekommen ist. Selbstverständlich bleibt man mit **TikZ** flexibler, aber die Möglichkeit bleibt einem ja immer offen.

2.2 Das Grundprinzip

In dem ‘tikzpicture’, das in den **myChemistry**-Umgebungen erstellt wird, werden Reaktanden und Reaktionspfeile mit einzelnen ‘nodes’¹ auf einer ‘chain’² angeordnet.

Beispiel 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

Dadurch ergeben sich einige Möglichkeiten, die ‘nodes’ relativ zueinander zu platzieren.

Beispiel 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

myChemistry macht vor allem von der Möglichkeit Gebrauch, ‘branches’ zu erstellen.

¹In einem tikzpicture kann man nahezu beliebig sogenannte ‘nodes’ setzen, mit allen möglichen Formen und Inhalten. Das sind „Knotenpunkte“ an bestimmten Koordinaten in einer ‘tikzpicture’-Umgebung.

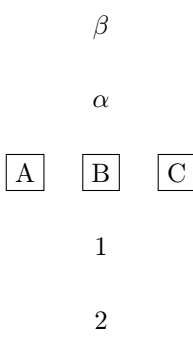
²Dafür ist die tikzlibrary ‘chains’ nötig.

Beispiel 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}

```



Sie müssen das nicht in allen Konsequenzen nachvollziehen, sollten aber die Richtungsangaben des letzten Beispiels in Erinnerung behalten, denn sie werden von **myChemistry** ebenfalls verwendet.

In manchen Beispielen der Dokumentation werden die Nodes farbig eingerahmt (siehe [Abschnitt 4.7](#)), damit man sehen kann, welchen Platz sie einnehmen und welchen Effekt eine Änderung der Ausrichtung hat.

2.3 Wie funktioniert's?**2.3.1 Basisbefehle**

Sehen wir uns zunächst ein Beispiel an:

Beispiel 4

```

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

```



Sie sehen hier die wichtigsten Befehle von **myChemistry** im Einsatz:

`\begin{rxn}[<keys>]` Die erste von zwei Umgebungen. Sie stellt die Reaktionschemata zwischen den Text und zentriert sie (siehe [Abschnitt 4.13](#)).

`\reactand[<ausrichtung>,<anker>,<tikz>]{<formeln>}` setzt eine node auf die chain, in die die chemischen Formeln geschrieben werden.

Die Standard-Ausrichtung ist `right` (siehe [Abschnitt 4.12](#)).

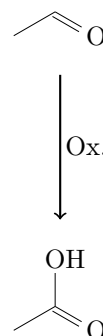
`\arrow[<keys>]{<oben>}{<unten>}` schreibt in der Standardeinstellung einen 5 em langen einfachen Pfeil nach rechts (siehe [Abschnitt 4.1](#)).

Beispiel 5

```

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

```



Wie Sie sehen, lässt sich das Reaktionschema durch optionale Argumente anders ausrichten. Durch die Angabe `below` wird die Carbonsäure unter den Pfeil gesetzt und nicht rechts daneben. Durch die Key-Angabe `direction=below` zeigt der Pfeil nach unten anstatt nach rechts.

2.3.2 Verzweigungen

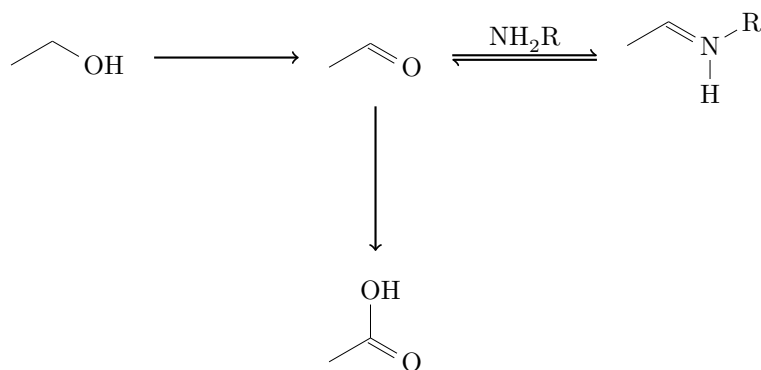
Bislang ist noch nicht recht einsichtig, wieso man `myChemistry` einsetzen sollte. Die waagerechten Reaktionen sind mit `'mhchem'` und `ChemFig` selbst zu verwirklichen. Und weshalb sollte man eine senkrechte Reaktion benötigen? Was den Einsatz von `myChemistry` aber interessant machen könnte, ist die Möglichkeit, verzweigte Reaktionsschemata zu erstellen.

Beispiel 6

```

1 \begin{rxn}
2 \reactand{ \chemfig{-[::30]-[::-60]OH} }
3 \arrow{}{}
4 \reactand[, carbonyl]{ \chemfig{-[::30]=_[::-60]O} }
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}

```

Im letzten Beispiel haben Sie einen weiteren wichtigen Befehl kennengelernt:

`\branch[<ausrichtung>,<anker>,<tikz>]{<zweig>}` (siehe [Abschnitt 4.2](#))

Der Zweig wurde mit `right=of carbonyl` rechts neben den ersten Reaktanden mit dem Anker `carbonyl` angesetzt. Innerhalb des Zweigs wurde beim Pfeil der Key `type={<=>}` verwendet, wodurch der Gleichgewichtspfeil dargestellt wurde. Andere `type`-Möglichkeiten wären `->` (Voreinstellung), `<-` oder `<->`.

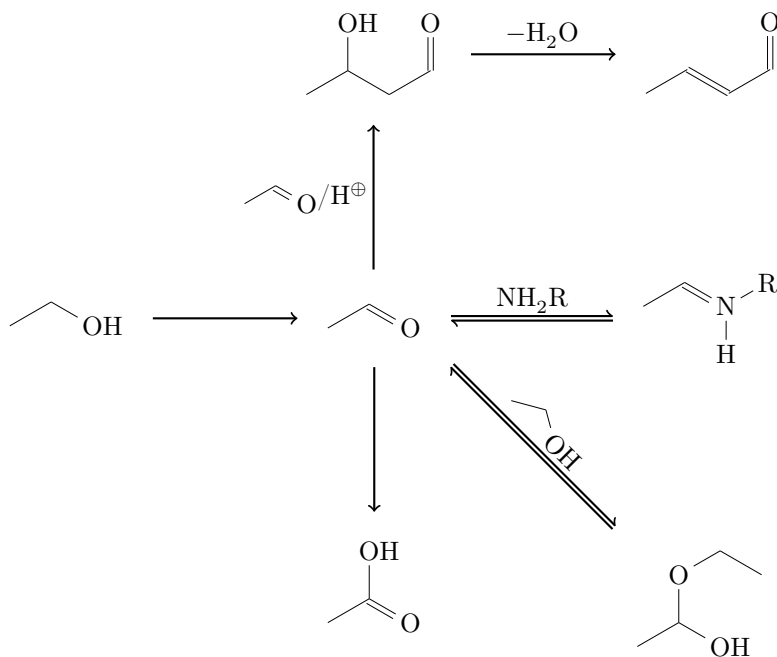
Durch mehrfaches Verwenden von `\branch` können so umfangreichere Reaktionsschemata entstehen:

Beispiel 7

```

1 \begin{rxn}
2 \reactand{ \chemfig{-[::30]-[::-60]OH} }
3 \arrow{}{}
4 \reactand[, carbonyl]{ \chemfig{-[::30]=[::-60]O} }
5 \arrow[direction=below]{}{}
6 \reactand[below]{ \chemfig{-[::30](-[::60]OH)=[::-60]O} }
7 \branch[right=of carbonyl,imin]{
8 \arrow[type={<=>},length=1.12]{\ce{NH2R}}{}
9 \reactand{ \chemfig{-[::30]=[::-60]N(-[:6]H)-[::60]R} }
10 }
11 \branch[below right=of carbonyl,halbacetal,yshift=-2pt,xshift=3pt]{
12 \arrow[type={<=>},direction=below right,length=1.12,aboveshift=3pt]{
13 \chemfig{[,.75]-[::30]-[::-60]OH} }{}
14 }
15 \branch[above=of carbonyl,aldol,xshift=5.2em]{
16 \arrow[direction=above]{ \chemfig{[,.75]-[::30]=[::-60]O}\Hpl }{}
17 \reactand[above]{ \chemfig{-[::30](-[::60]OH)-[::-60]-[::60]=[::60]O} }
18 } }
19 \arrow[{$-\ce{H2O}$}]{}
20 \reactand{ \chemfig{-[::30]=[::-60]-[::60]=[::60]O} }
21 \end{rxn}

```



2.3.3 Nummerierte Schemata

Die zweite Umgebung von **myChemistry** funktioniert genau wie die erste, setzt das Reaktionsschema allerdings in eine nummerierte Gleitumgebung mit Überschrift.

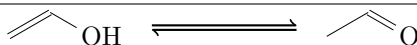
Beispiel 8

```

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

```

Reaktionsschema 1 Keto-Enol-Tautomerie



Hier kommt die Umgebung

```

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

```

zum Einsatz. Wie Sie die Ihren Vorstellungen gemäß anpassen können, lesen Sie in der Befehlsreferenz ([Abschnitt 4.14](#)).

2.4 Voreinstellungen

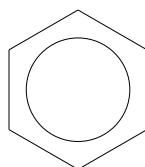
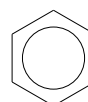
Es gibt einige Voreinstellungen, die zum Teil meinem persönlichen Geschmack geschuldet sind, die Sie aber nach Bedarf ändern können. So gelten für die **ChemFig**-Formeln innerhalb der *myChemistry*-Umgebungen folgende Voreinstellungen:

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

Außerhalb der Umgebungen gelten weiterhin die Voreinstellungen von **ChemFig**.

Beispiel 9

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



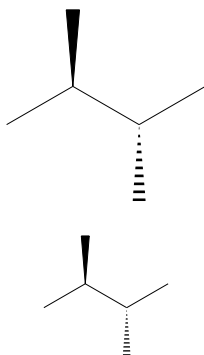
Sie können die Voreinstellungen von **myChemistry** über folgende Befehle ändern:

```
1 \setbondlength{<länge>}
2 \setbondshape{<basislänge>}{<strichdicke>}{<strichabstand>}
3 \setatomsiz{<schriftgröße>}
```

Damit werden die Einstellungen nachfolgend *für alle weiteren myChemistry*-Umgebungen geändert. Lassen Sie die Argumente leer, werden die Voreinstellungen wiederhergestellt. `\setatomsiz` hat die Voreinstellung `\small`.

Beispiel 10

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



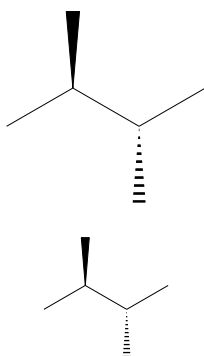
Wollen Sie nur die Parameter einer Umgebung ändern, verwenden Sie *innerhalb der Umgebung* die Befehle von **ChemFig** und die L^AT_EX-Befehle für die Schriftgröße.

Beispiel 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}

```



Reaktionspfeile haben als Standardwert die Länge 5 em oder $5\sqrt{2}$ em im Fall der schrägen Pfeile. Die Voreinstellung lässt sich mit

```
1 \setarrowlength{<länge>}
```

auf <länge> bzw. $\langle \text{länge} \rangle \cdot \sqrt{2}$ ändern.

2.5 Paket-Optionen

myChemistry verfügt über einige Paket-Optionen.

seit v1.2 `chemstyle` Mit dieser Option kann 'chemstyle' automatisch geladen werden, ohne dass Konflikte mit myChemistry entstehen.

`color=<farbe>` Mit dieser Option wird die entsprechende Farbe an ‘chemexec’ weitergereicht und dessen Option `shade=true` aufgerufen.

`english` Wird diese Option aufgerufen, dann lädt **myChemistry** ‘chemexec’ in der englischen Version, falls das Paket nicht vorher aufgerufen wurde. Außerdem wird der Name der `rxnscheme`-Umgebung (siehe [Abschnitt 4.14](#)) in „Reaction scheme“ geändert.

`nochemexec` Durch diese Option können Sie verhindern, dass **myChemistry** ‘chemexec’ lädt.

`nocolor` Mit dieser Option wird ‘chemexec’ ohne Farbe und mit der Option `shade=false` geladen (Default-Verhalten von **myChemistry**).

seit v1.2 `nocompounds` Durch diese Option können Sie verhindern, dass **myChemistry** ‘chemcompounds’ lädt.

seit v1.2 `nomhchem` Durch diese Option können Sie verhindern, dass **myChemistry** ‘mhchem’ lädt, vorausgesetzt, dass ‘chemexec’ auch nicht geladen wird.

seit v1.2 `nopackages` Durch diese Option werden (außer **ChemFig**) gar keine Pakete geladen¹.

`placement=<position>` Durch den Aufruf dieser Option kann das Standard-Platzierungsverhalten der `rxnscheme`-Umgebung (siehe [Abschnitt 4.14](#)) in `<position>` geändert werden.

`shade` Mit dieser Option wird ‘chemexec’ mit der Option `shade=true` geladen.

3 Fortgeschrittene Anwendung, Verwendung von TikZ

Die meisten der Befehle ermöglichen nach der Ausrichtungsangabe die Angabe weiteren TikZ-Codes. Dadurch lassen sich viele Feinjustierungen vornehmen. Wenn Sie sich mit TikZ einigermaßen auskennen, können Sie sowieso noch weitaus mehr realisieren, als durch **ChemFig** und **myChemistry** vorgegeben (siehe [Abschnitt 5.3](#)).

3.1 Die Ausrichtungsfrage

Da Reaktanden, Pfeile und Zweige mittig zu dem Objekt, auf das sie sich beziehen, ausgerichtet werden, erzeugt die Default-Ausrichtung nicht immer schöne Ergebnisse.

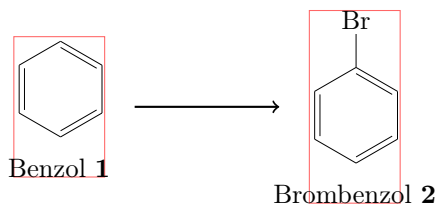
¹Außer denen, die **myChemistry** benötigt, um zu funktionieren (TikZ etc.).

Beispiel 12

```

1 \makevisible
2 \begin{rxn}
3 \reactand{ \chemname{\chemfig{*6(-----)}}{Benzol \compound{benzol}} }
4 \arrow{}{}
5 \reactand{ \chemname{\chemfig{*6(-----(-Br)-)}}{Brombenzol \compound{
  brombenzol}} }
6 \end{rxn}

```



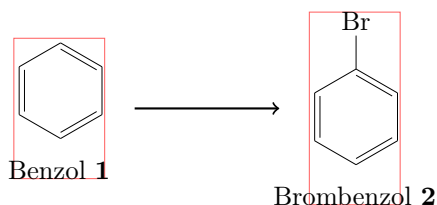
Wie Sie sehen können, sind die beiden Reaktanden aus Sicht der Benzolringe nicht gleich zu dem Pfeil ausgerichtet. Der erste Reaktand scheint nach oben geschoben zu sein. Der Versuch, das mit TikZ-Code zu korrigieren, versagt.

Beispiel 13

```

1 \makevisible
2 \begin{rxn}
3 \reactand[, , yshift=-1em]{ \chemname{\chemfig{*6(-----)}}{Benzol \
  compound{benzol}} }
4 \arrow{}{}
5 \reactand{ \chemname{\chemfig{*6(-----(-Br)-)}}{Brombenzol \compound{
  brombenzol}} }
6 \end{rxn}

```



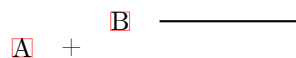
Das kommt daher, da der erste Reaktand relativ zu dem Objekt verschoben wird, auf das er sich bezieht. Da er das erste Objekt auf der Chain ist, wird er gar nicht verschoben. Der nachfolgende Pfeil richtet sich in Bezug auf den ersten Reaktanden aus.

Beispiel 14

```

1 \makevisible
2 \begin{rxn}
3 \reactand{A}
4 \chemand
5 \reactand[, , yshift=1em]{B}
6 \arrow{}{}
7 \end{rxn}

```



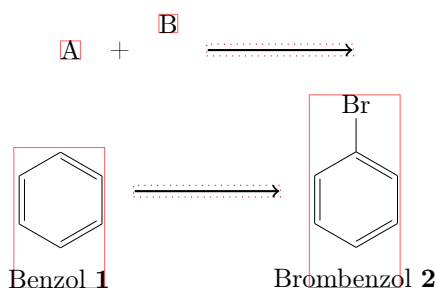
Da es (noch) keine Möglichkeit gibt, die Ausrichtung des Pfeils selbst zu ändern, könnte man ihn stattdessen in einen Zweig stecken.

Beispiel 15

```

1 \makevisible
2 \begin{rxn}
3 \reactand{A}
4 \chemand
5 \reactand[, , yshift=1em]{B}
6 \branch[, , yshift=-1em]{\arrow{}{}}
7 \end{rxn}
8 \begin{rxn}
9 \reactand{ \chemname{\chemfig{*6(-----)}}{Benzol \compound{benzol}} }
10 \branch[, , yshift=1em]{\arrow{}{}}
11 \reactand{ \chemname{\chemfig{*6(-----(-Br)-)}}{Brombenzol \compound{
    brombenzol}} }
12 \end{rxn}

```



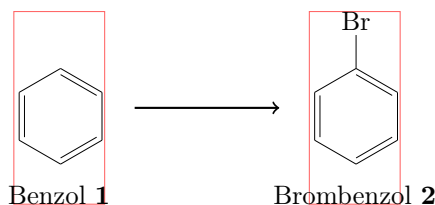
Das ist für das letzte Beispiel aber nicht die beste Lösung, da die exakte Ausrichtung auf diese Weise immer eine ganze Reihe von Versuchen benötigt, bis man das gewünschte Ergebnis erzielt. Es gibt eine andere Lösung: ein unsichtbares Brom am ersten Benzol.

Beispiel 16

```

1 \makevisible
2 \begin{rxn}
3 \reactand{ \chemname{\chemfig{*6(---(-[,,,,draw=none]\phantom{Br})--)} }{Benzol \compound{benzol}} }
4 \arrow{}{}
5 \reactand{ \chemname{\chemfig{*6(---(-Br)--)} }{Brombenzol \compound{brombenzol}} }
6 \end{rxn}

```



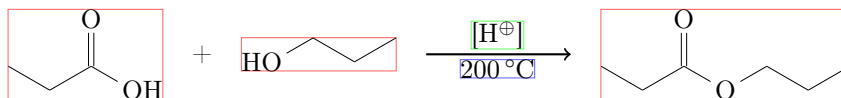
In anderen Fällen ist der TikZ-Code aber die beste Lösung:

Beispiel 17

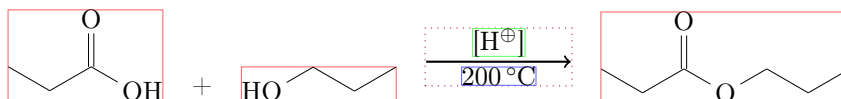
```

1 \makevisible
2 default:
3 \begin{rxn}
4 \reactand{\chemfig{-[: -30] -[:30] (= [2] O) -[: -30] OH}}
5 \chemand
6 \reactand{\chemfig{HO -[:30] -[: -30] -[:30]}}
7 \arrow{[\Hpl]}{\SI{200}{\celsius}}
8 \reactand{\chemfig{-[: -30] -[:30] (= [2] O) -[: -30] O -[:30] -[: -30] -[:30]}}
9 \end{rxn}
10 Hydroxy-Gruppen auf gleicher H"ohe:
11 \begin{rxn}
12 \reactand{\chemfig{-[: -30] -[:30] (= [2] O) -[: -30] OH}}
13 \chemand[, , yshift=-1.2em]
14 \reactand[, , yshift=.12em]{\chemfig{HO -[:30] -[: -30] -[:30]}}
15 \branch[, , yshift=1em]{\arrow{[\Hpl]}{\SI{200}{\celsius}}}
16 \reactand{\chemfig{-[: -30] -[:30] (= [2] O) -[: -30] O -[:30] -[: -30] -[:30]}}
17 \end{rxn}
default:

```



Hydroxy-Gruppen auf gleicher Höhe:



Ich fürchte, in vielen Fällen müssen Sie mit `xshift` und `yshift` spielen, bis das Schema

aussieht, wie Sie Sich das vorstellen. Vielleicht wird eine zukünftige Version von **myChemistry** eine benutzerfreundlichere Ausrichtungsmöglichkeit bieten.

3.2 Mit TikZ andere Ziele erreichen

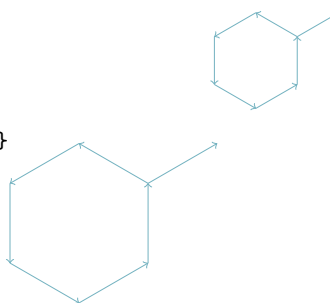
Man könnte natürlich, nur zum Spaß?, das Aussehen von Molekülen mit **TikZ** ändern.

Beispiel 18

```

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

```



Das Beispiel ist natürlich kein gutes, da mit **ChemFig** dasselbe Ergebnis erzielt werden kann. Vielfache andere Anwendungen sind aber denkbar:

Beispiel 19

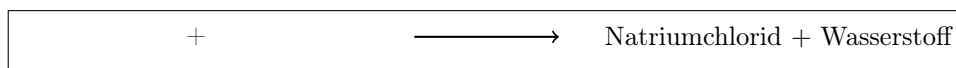
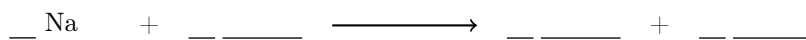
```

1 \newcommand{\leer}{\reactand[, 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 Salzbildung (I)}
5 \reactand[below, a, yshift=1em]{F\"ulle die L\"ucken}
6 \branch[below=of a, b]{ \reactand[, minimum width=5em]{ \stoich\ Na }\
   chemand \leer \arrow{}{} \leer \chemand \leer }
7 \branch[below=of b, , draw, inner sep=3pt]{\reactand[, minimum width=5em
   ]{\chemand\reactand[, minimum width=5em]{\arrow{}{} \reactand[,
   minimum width=5em]{Natriumchlorid $+$ Wasserstoff}}
8 \end{rxn}

```

Salzbildung (I)

Fülle die Lücken



4 Alphabetische Befehlsreferenz

Im folgenden Abschnitt werden alle Befehle von **myChemistry** in alphabetischer Reihenfolge vorgestellt.

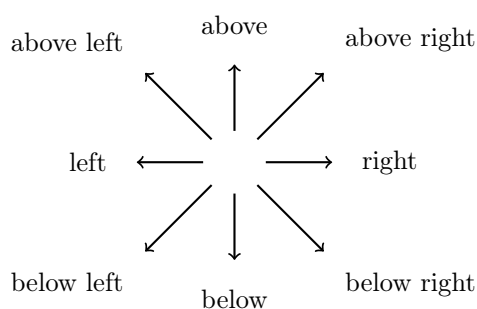
4.1 arrow

Reaktionspfeile werden mit `\arrow` erstellt.

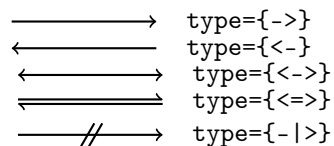
```
1 \arrow [<keys>]{<oben>}{<unten>}
```

Mit mehreren Keys können die Reaktionspfeile angepasst werden. Sie werden nach dem Muster `key=wert` angegeben.

`direction=<richtung>` – mögliche Einstellungen sind:



`type=<typ>` – mögliche Einstellungen sind:



`length=<faktor>` – mit dem Faktoren, den Sie hier angeben, wird die Pfeillänge (5.0 em bei Faktor = 1.0, Standard) multipliziert.

`name=<anker>` – hier können Sie dem Pfeil einen Anker geben, auf den z. B. mit einem Branch referenziert werden kann.

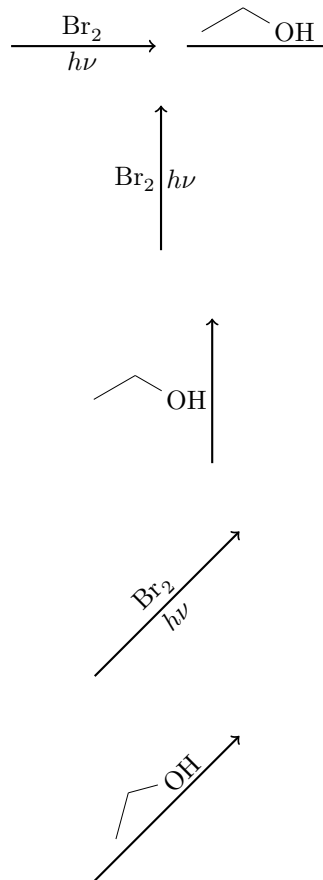
seit VI.2 `both` – durch diesen Key haben die beiden Nodes, in die die Beschriftungen geschrieben werden, die gleichen Maße.

Beispiel 20

```

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

```



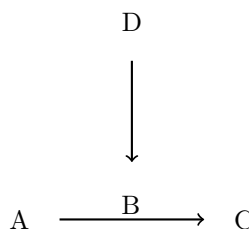
Einmal die meisten Keys im Einsatz:

Beispiel 21

```

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

```



Liegt der Pfeil in einem Branch (siehe [Abschnitt 4.2](#)), dann wird die Ausrichtung des Branch bestimmt durch die Größe der Nodes, mit denen die Pfeilbeschriftung platziert

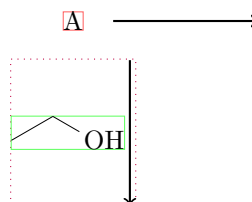
wird. Hat der Pfeil nun nur eine oder zwei unterschiedlich große Beschriftungen, dann ist die Ausrichtung falsch.

Beispiel 22

```

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

```



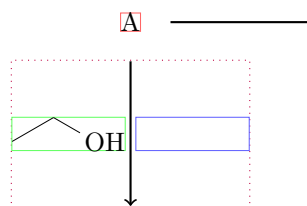
Durch die Verwendung des Keys `both` bekommen die Nodes beider Pfeilbeschriftungen die gleichen Maße, wodurch die Ausrichtung korrigiert werden kann.

Beispiel 23

```

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

```



Mehr zu dem Problem der Ausrichtung lesen Sie in [Abschnitt 4.2.1](#).

4.2 branch

NEU in v1.3

Der Befehl `\branch` wird verwendet, um eine Verzweigung der Reaktion zu realisieren. Wenn Sie ältere Versionen von **myChemistry** eingesetzt haben, beachten Sie, dass sich die Befehl-Syntax verändert hat.

```

1 \branch[<ausrichtung>,<anker>,<tikz>]{<formel(n)>}

```

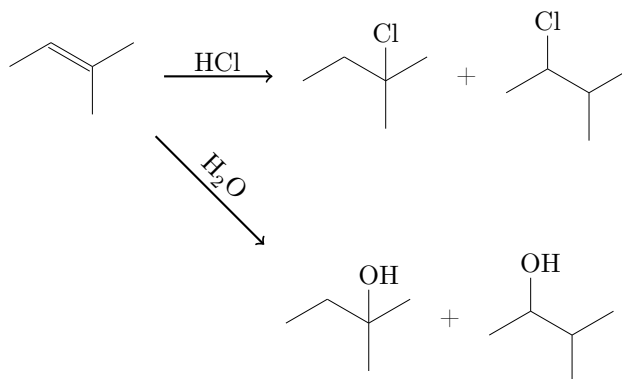
Für den `\branch` wird die Ausrichtung und der Anker wichtig. Sehen wir uns ein Beispiel an.

Beispiel 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)
10 \chemand
11 \reactand{\chemfig{-[:30](-[:60]OH) -[: -60](-[: -60]) -[:60]}}
12 }
13 \end{rxn}

```



In diesem Beispiel hat der erste Reaktand den Anker `start` bekommen (Zeile 2, siehe auch [Abschnitt 4.12](#)).

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

`\branch` bezieht sich nun in seiner Ausrichtung darauf (Zeile 6):

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

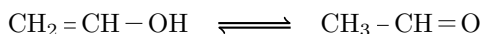
Gibt man die Ausrichtung nicht in Bezug auf einen Anker an, bezieht sie sich immer auf den letzten `reactand` oder `\arrow`. Lässt man das optionale Argument leer, dann platziert sich der Branch automatisch rechts.

Beispiel 25

```

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

```



Zur Ausrichtung haben Sie mehrere Möglichkeiten: Sie können den Branch entweder selbst auf die `chain` setzen oder ihn relativ zu einem Objekt platzieren.

chain In diesem Fall geben Sie als Ausrichtung folgendes an: `on chain=going <wert>`.

relativ In diesem Fall geben Sie folgendes an: `<wert>=of <anker>`.

Als `<wert>` können Sie die gleichen Werte einsetzen wie bei `\reactand`, siehe [Abschnitt 4.12](#). Die Voreinstellung ist `on chain=going right`.

4.2.1 Ausrichtungsprobleme

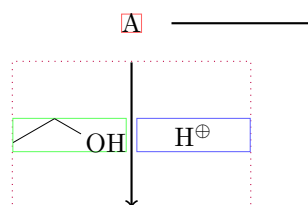
Wenn ein Pfeil zwei verschieden große Beschriftungen hat und in einem Branch liegt, wird der Branch nicht mehr richtig ausgerichtet. Der `\arrow`-Key `both` ist nicht wirklich eine Lösung, weil die kleinere Beschriftung dann nicht mehr am Pfeil liegt, sondern wegrutscht.

Beispiel 26

```

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}}{\Hpl}
7 }
8 \end{rxn}
9 \makeinvisible

```



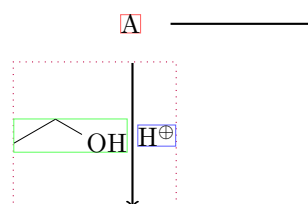
In diesem Fall können Sie den Branch mit den `TikZ`-Keys `xshift` und `yshift` verschieben.

Beispiel 27

```

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

```



4.3 chemand

neu in v1.3

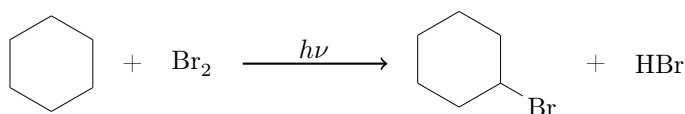
Der Befehl

```
1 \chemand [<ausrichtung>,<anker>,<tikz>]
```

erzeugt und platziert ein + auf die gleiche Weise, wie `\reactand` beliebigen Text platziert.

Beispiel 28

```
1 \begin{rxn}
2 \reactand{\chemfig{*6(-----)}}
3 \chemand
4 \reactand{\ce{Br2}}
5 \arrow{$h\nu$}{>}
6 \reactand{\chemfig{*6(--(-Br)----)}}
7 \chemand
8 \reactand{\ce{HBr}}
9 \end{rxn}
```



Die optionalen Argumente von `\chemand` und `\reactand` sind die gleichen, siehe [Abschnitt 4.12](#).

4.4 dummy

Mit `\dummy` zeichnet man eine leere Node. Die Pfeile, die mit `\arrow` erzeugt werden, müssen einer Node nachfolgen. `\arrow` ruft intern `\tikzchainprevious` auf. Ist *vor* einem Pfeil noch *keine* Node auf die Chain geschrieben worden, erzeugt das eine Fehlermeldung. Durch setzen des `\dummy` kann ein Schema dennoch mit einem Pfeil beginnen.

Beispiel 29

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



4.5 elmove

`\elmove` ist lediglich ein Abkürzungsmakro für den `ChemFig`-Befehl `\chemmove`.

```
1 \elmove [<tikz>]{<start>}{<startrichtung>}{<ende>}{<endrichtung>}
```

Das schreibt den Befehl

```
1 \chemmove{\draw [<tikz>](<start>).. controls +(<startrichtung>) and +(<endrichtung>)..(<ende>);}
```

mit `[->,red,shorten <=3pt,shorten >=1pt]` als Voreinstellung für `<tikz>`.

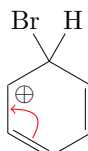
Wie `\chemmove` funktioniert, können Sie im Manual zu `ChemFig` nachlesen.

Beispiel 30

```

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}

```



4.6 makeinvisible

seit **VI.2** Dieser Befehl hebt die Änderungen von `\makevisible` (siehe [Abschnitt 4.7](#)) auf und stellt das normale Verhalten von `myChemistry` wieder her. `\makeinvisible` wirkt sich nur auf nachfolgende Reaktanden aus.

4.7 makevisible

seit **VI.2** Mit `\makevisible` können Sie die Nodes, innerhalb derer sich die Reaktanden befinden, farblich hervorheben. Das kann z. B. bei der Positionierung und Feinjustierung von Branches ganz nützlich sein. Ein Beispiel dafür sehen Sie in [Abschnitt 4.1](#). Je nach Art der Node ist die Markierung eine andere:

`\reactand{}{}`, `\arrow{above}{}{}`, `\arrow{}{below}` und `\branch{}{}`. Siehe auch [Abschnitt 4.6](#).

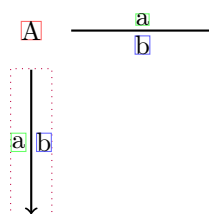
`\makevisible` wirkt sich nur auf nachfolgende Reaktanden aus.

Beispiel 31

```

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 \makeinvisible

```



4.8 marrow

Der Befehl `\marrow` zeichnet einen Mesomeriepfeil.

```
1 \marrow [<ausrichtung>]
```

Die Ausrichtung funktioniert analog zu `\reactand` (Abschnitt 4.12), siehe auch Abschnitt 4.11.

4.9 mCsetup

Neu in v1.3

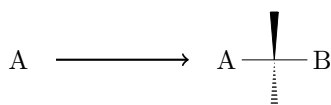
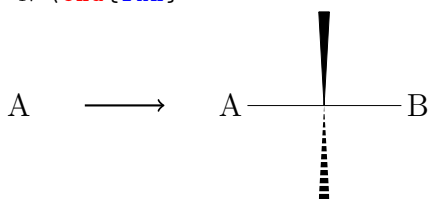
Der Befehl

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

kann verwendet werden, um alle Voreinstellungen zu verändern. Für jeden der `\set<command>`-Befehle von `myChemistry` mit Ausnahme von `\setbondshape` gibt es einen Key `<command>=<value>`. Zusätzlich gibt es den Key `align=<value>`, mit dem das Ausrichtungsverhalten von `rxn` und `rxnscheme` gleichzeitig geändert werden kann und den Key `reset`, durch den alle Voreinstellungen wiederhergestellt werden.

Beispiel 32

```
1 \mCsetup{
2   arrowlength=3em,
3   rcndist=2em,
4   atomsize=\large,
5   bondlength=3em,
6   %rxnalign=right,
7   %schemealign=left,
8   align=left
9 }
10 \setbondshape{4pt}{2pt}{1pt}
11 \begin{rxn}
12 \reactand{A}\arrow{}{\}\reactand{\chemfig{A-(<[2])(<:[6])>-B}}
13 \end{rxn}
14 \mCsetup{reset}
15 \begin{rxn}
16 \reactand{A}\arrow{}{\}\reactand{\chemfig{A-(<[2])(<:[6])>-B}}
17 \end{rxn}
```



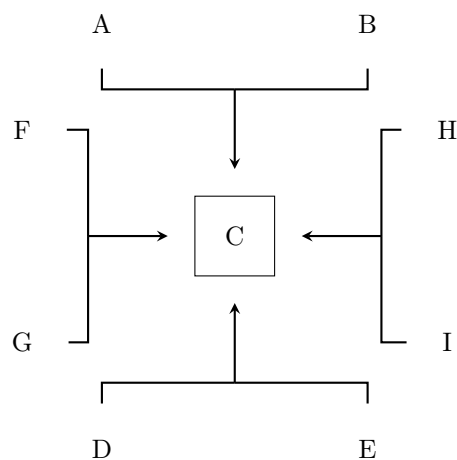
4.10 merge

Der `merge`-Befehl sind nicht nur für den direkten Einsatz in den `myChemistry`-Umgebungen gedacht, sondern können flexibler in einem `tikzpicture` eingesetzt werden. Mit diesem Befehl können verschiedene Reaktionsstränge zu einem vereint werden. Dafür müssen die einzelnen zu vereinenden Reaktanden als `nodes` mit Namen gekennzeichnet sein.

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

Beispiel 33

```
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 \node(f) at (-3em,-4em) {F};
11 \node(g) at (-3em,-12em) {G};
12 \merge[direction=right]{c}{f}{g}
13 \node(h) at (13em,-4em) {H};
14 \node(i) at (13em,-12em) {I};
15 \merge[direction=left]{c}{h}{i}
16 \end{tikzpicture}
17 \end{center}
```



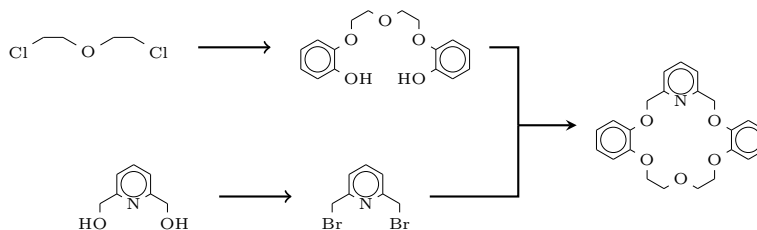
Gibt man den Start- und dem Zielreaktanden Anker, funktioniert `\merge` natürlich auch in den `myChemistry`-Umgebungen.

Beispiel 34

```

1 \begin{rxn}
2 \setatomsep{1em}\tiny
3 % Strang 1
4 \reactand[,oben]{ \chemfig{Cl-[:30,1.5]--[: -30,1.5]O
  -[:30,1.5]--[: -30,1.5]Cl}{ } }
5 \arrow[length=.5]{ }
6 \reactand[,start_oben]{ \chemfig{O(-[: -150]**6(-----(-OH)-))
  -[:90]-[:30]-[: -30]O-[:30]-[: -30]-[: -90]O-[: -30]**6(-(-HO)-----)} }
7 % Strang 2
8 \branch[below=of oben,start_unten,xshift=8em,yshift=-4em]{
9 \reactand{ \chemfig{**6((--[6,,2]HO)-N(--[6]OH)----)} }
10 \arrow[length=.5]{ }
11 \reactand{ \chemfig{**6((--[6]Br)-N(--[6]Br)----)} }
12 }
13 % Ziel
14 \branch[right=of start_oben,ziel,xshift=5em,yshift=-4em]{
15 \reactand[,c]{ \chemfig{O(-[: -150]**6(-----(-O?) -)-[:90]-[:30]**6(-N
  -(-[: -90]O-[: -30]**6(-(-O-[6]-[: -150]-[:150]O-[: -150]-[:150]?)-----))
  ----)} }
16 }
17 % Zusammenfuehren:
18 \merge[direction=right]{ziel}{start_oben}{start_unten}
19 \end{rxn}

```



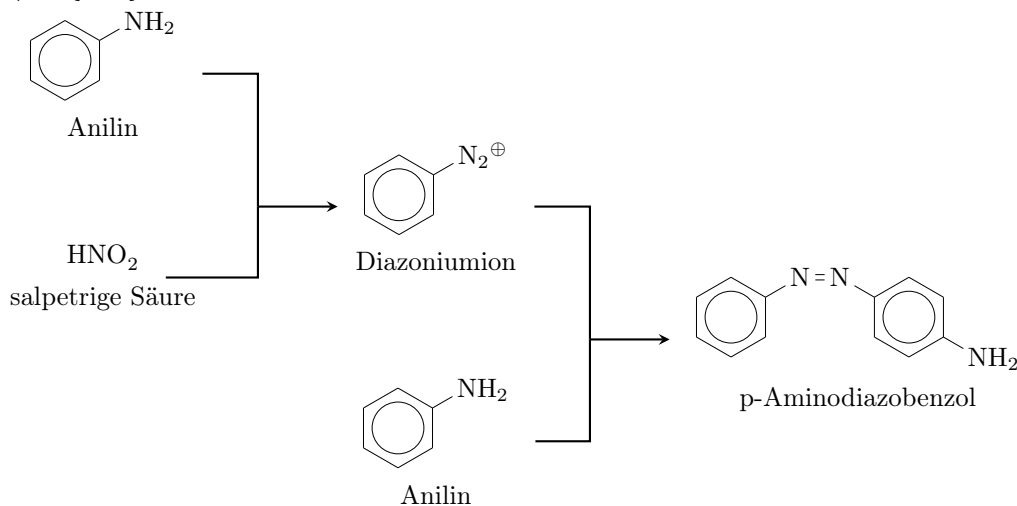
Beachten Sie, dass für die ‘nodes’ in der Regel Branches verwendet werden sollten, wenn Sie `\merge` in den `myChemistry`-Umgebungen verwenden. Die Verwendung von `\merge` erfordert unter Umständen einige Spielerei mit Branches, `xshift` und `yshift`, bis man das gewünschte Ergebnis erhält.

Beispiel 35

```

1 \begin{rxn}
2 \setatomsep{1.5em}
3 \reactand[,start_aa]{ \chemname{\chemfig{**6(---(-NH_2)---)}}{Anilin}
4 \reactand[below,start_ab,yshift=-3em]{ \chemname{\ce{HNO2}}{salpetrige
  S"auere} }
5 \branch[right=of start_aa,ziel_a,xshift=6em,yshift=-5em]{
6 \reactand{ \chemname{\chemfig{**6(---(-N|_2\op)---)}}{Diazoniumion}
7 }% = start_ba
8 \branch[below=of ziel_a,start_bb,yshift=-3em]{
9 \reactand{ \chemname{\chemfig{**6(---(-NH_2)---)}}{Anilin} }
10 }
11 \branch[right=of ziel_a,ziel_b,xshift=6em,yshift=-5em]{
12 \reactand{ \chemname{\chemfig{N(-[: -150]**6(-----))=N
  -[: -30]**6(---(-NH_2)---)}}{p-Aminodiazobenzol} }
13 }
14 \merge[direction=right]{ziel_a}{start_aa}{start_ab}
15 \merge[direction=right]{ziel_b}{ziel_a}{start_bb}
16 \end{rxn}

```

**4.11 mesomeric**

Neu in V1.3

Der `\mesomeric`-Befehl funktioniert wie ähnlich wie `\branch` (Abschnitt 4.2). Sein Zweck ist es, eckige Klammern zu setzen. **Wenn Sie ältere Versionen von myChemistry eingesetzt haben, beachten Sie, dass sich die Befehl-Syntax verändert hat.**

```
1 \mesomeric [<ausrichtung>,<anker>,<tikz>]{<formel(n)>}
```

In `<formel(n)>` werden die mesomeren Grenzstrukturen geschrieben. Mit `\marrow` (Abschnitt 4.8) werden die Mesomeriepfeile gesetzt. Man kann `\mesomeric` falls nötig

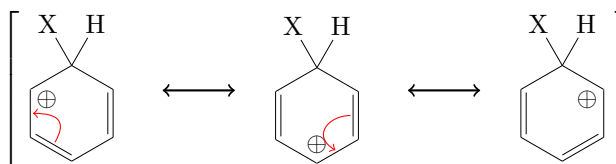
mit einem Anker (<anker>) versehen (Abschnitt 4.2). Die Ausrichtung funktioniert analog `\reactand`.

Beispiel 36

```

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}

```



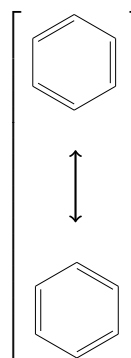
Oder auch von oben nach unten:

Beispiel 37

```

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

```



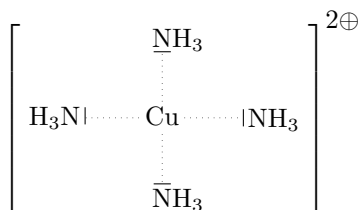
Vielleicht auch einen Komplex?

Beispiel 38

```

1 \begin{rxn}
2 \setatomsep{3em}
3 \mesomeric[,a]{
4 \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} }
5 }
6 \node[above right=of a,yshift=-1em] {$2\oplus$};
7 \end{rxn}

```

**4.12 reactand****Neu in v1.3**

Der Befehl `\reactand` ist so etwas wie der Basisbefehl. **Wenn Sie ältere Versionen von myChemistry eingesetzt haben, beachten Sie, dass sich die Befehl-Syntax verändert hat.**

```

1 \reactand[<ausrichtung>,<anker>,<tikz>]{<formel(n)>}

```

In diesen Befehl werden die Formeln (`<formel>`) geschrieben und können, falls nötig, mit einem Anker (`<anker>`) versehen werden. Die Ausrichtung kann die 8 Werte (a) `right`, (b) `above right`, (c) `above`, (d) `above left`, (e) `left`, (f) `below left`, (g) `below`, (h) `below right` annehmen, Voreinstellung ist (`right`). Dieses Argument wird verwendet, wenn die Reaktionsgleichung nicht von links nach rechts, sondern z. B. von oben nach unten verlaufen soll.

Beispiel 39

<pre> 1 untereinander: 2 \begin{rxn} 3 \reactand{\ce{Br2}} 4 \reactand[below]{\ce{Cl2}} 5 \end{rxn} 6 7 Beispiel mit mehreren Reaktanden 8 : 9 \begin{rxn} 10 \reactand{\ce{Br2}} 11 \reactand[below]{\ce{I2}} 12 \reactand{\ce{Cl2}} 13 \end{rxn} 14 Reaktion von oben nach unten: 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} </pre>	<pre> untereinander: Br2 Cl2 Beispiel mit mehreren Reaktanden: Br2 I2 Cl2 Reaktion von oben nach unten: Br-Br hv 2 Br• </pre>
--	---

4.13 rxn (Umgebung)

Die Umgebung `rxn` ist eine unnummerierte nicht gleitende Umgebung für Reaktionsschemata. Die Reaktionsschemata werden per Default zentriert. Die Voreinstellungen `\setbondlength`, `\setbondshape`, `\setarrowlength` und `\setatomsiz` gelten hier ebenso wie `beirxnscheme`.

```

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

```

4.13.1 Optionen

seit VI.2 `rxn` hat zwei Keys:

`align=<ausrichtung>` das Ausrichtungsverhalten der `rxn`-Umgebung, Default: center

`scale=<factor>` Skalierung der `rxn`-Umgebung, Default: 1.0

Beispiel 40

```

1 \begin{rxn}[align=center]
2 \reactand{center}\arrow{}{}\reactand{zentriert}
3 \end{rxn}
4 \begin{rxn}[align=right]
5 \reactand{right}\arrow{}{}\reactand{rechts}
6 \end{rxn}
7 \begin{rxn}[align=left]
8 \reactand{left}\arrow{}{}\reactand{links}
9 \end{rxn}

```

center → zentriert

right → rechts

left → links

4.14 rxnscheme (Umgebung)

Die Umgebung `\rxnscheme` ist eine Gleitumgebung für Reaktionsschemata.

```

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

```

4.14.1 Optionen

`label=<label>` Wie jede Gleitumgebung kann auch `rxnscheme` mit einem Label versehen werden. Setzen Sie z. B.

```

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

```

ein, können Sie mit `\ref{rs:schema}` wie gewohnt referenzieren.

`scale=<scalefactor>` Mit diesem Key kann das Reaktionsschema skaliert werden. Beachten Sie, dass er sich nicht auf die Schriftgröße und die Größe der `ChemFig`-Formeln auswirkt.

```

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

```

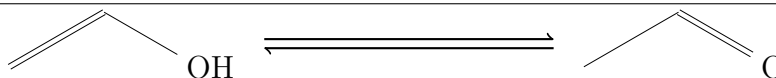
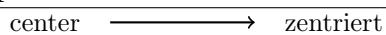
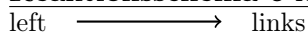
seit v1.2 `align=<ausrichtung>` Mit diesem Key kann man auswählen, ob das Schema links, rechts oder mittig ausgerichtet wird.

Beispiel 41

```

1 \begin{rxnscheme}[scale=2]{Gro\ss es Schema}
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]{Kleines Schema}
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{zentriert}
15 \end{rxnscheme}
16 \begin{rxnscheme}[align=right]{right}
17 \reactand{right}\arrow{}{}\reactand{rechts}
18 \end{rxnscheme}
19 \begin{rxnscheme}[align=left]{left}
20 \reactand{left}\arrow{}{}\reactand{links}
21 \end{rxnscheme}

```

Reaktionsschema 2 Großes Schema**Reaktionsschema 3** Kleines Schema**Reaktionsschema 4** center**Reaktionsschema 5** right**Reaktionsschema 6** left**4.14.2 rxnscheme anpassen**

Stil Wenn Ihnen der Stil nicht gefällt, können Sie diesen mit

```

1 \floatstyle{<neuer Stil>}
2 \restylefloat{rxnfloat}

```

ändern. Als Stile stehen durch das ‘float’-Paket

plain Ohne spezielle Formatierungen, Legende erscheint unter dem Objekt

plaintop Wie **plain**, aber Legende oberhalb des Objekts

boxed Objekt ist gerahmt, Legende unterhalb

ruled Legende erscheint von Linien umgeben oberhalb des Objekts, Objekt wird unterhalb von einer weiteren Linie begrenzt; Voreinstellung für **rxnscheme**

zur Verfügung.

Beispiel 42

```

1 \begin{rxnscheme}{ruled}
2 \reactand{Standard-Stil}
3 \end{rxnscheme}
4 \floatstyle{boxed}
5 \restylefloat{rxnfloat}
6 \begin{rxnscheme}{boxed}
7 \reactand{mit Rahmen}
8 \end{rxnscheme}
9 \floatstyle{plain}
10 \restylefloat{rxnfloat}
11 \begin{rxnscheme}{plain}
12 \reactand{ohne Schnickschnack}
13 \end{rxnscheme}

```

Reaktionsschema 7 ruled

Standard-Stil

mit Rahmen

Reaktionsschema 8: boxed

ohne Schnickschnack

Reaktionsschema 9: plain

Platzierung Auch das Platzierungsverhalten, das in der Voreinstellung H ist, können Sie entsprechend ändern.

```

1 \floatplacement{rxnfloat}{<position>}

```

Einfacher ist allerdings der Aufruf von **myChemistry** mit entsprechender Option.

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

Sie können auch das Verhalten einer einzigen Umgebung durch Angabe eines Keys ändern.

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

Benennung Wollen Sie den Namen der Beschriftung ändern, können Sie das mit

```
1 \setschemename{<neuer name>}
```

machen. Voreinstellung ist „Reaktionschema“ bzw „Reaction scheme“ bei der Paketoption ‘english’.

Zähler Um den Zähler zu ändern, gehen Sie wie üblich vor. Durch

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

wird der Zähler der Schemata z. B. mit jeder neuen `section` zurückgesetzt und die Nummer nach den Muster `section.rxnfloat` ausgegeben. Beachten Sie, dass Sie wegen des @ den Aufruf mit `\makeatletter` und `\makeatother` begrenzen müssen.

Verzeichnis Mit

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

können Sie eine Liste aller Reaktionsschemata erzeugen:

Beispiel 43

Reaktionsschemata

1	Keto-Enol-Tautomerie	10
2	Großes Schema	33
3	Kleines Schema	33
4	center	33
5	right	33
6	left	33
7	ruled	34
8	boxed	34
9	plain	34
10	Additionsreaktion	38
11	Elektrophile Substitution . . .	41
12	Synthese von Chrysanthemum- säure	53

```
1 \listof{rxnfloat}{
  Reaktionsschemata}
```

4.15 setarrowlength

NEU in V1.3

Reaktionspfeile haben als Standardwert die Länge 5.0em oder $5.0 \cdot \sqrt{2}$ em im Fall der schrägen Pfeile. Die Voreinstellung lässt sich mit

```
1 \setarrowlength{<länge>}
```

auf <länge> bzw. $\langle \text{länge} \rangle \cdot \sqrt{2}$ ändern. Beachten Sie, dass Sie eine Längeneinheit verwenden müssen. Lassen Sie das Argument leer, wird die Voreinstellung wiederhergestellt.

Dieser Befehl ersetzt \arrowlength, der bis Version 1.2 noch verfügbar war.

4.16 setatomsiz

Mit

```
1 \setatomsiz{<größe>}
```

lässt sich die Schriftgröße der Atomgruppen verändern. Standard ist `\small`. Lassen Sie das Argument leer, wird die Voreinstellung wiederhergestellt.

Dieser Befehl ersetzt \atomsiz, der bis Version 1.2 noch verfügbar war.

4.17 setbondlength

Mit

```
1 \setbondlength{<länge>}
```

lässt sich `\setatomsep{<länge>}` für die `ChemFig`-Formeln *innerhalb* der `myChemistry`-Umgebungen einstellen. Standard ist 1.8em. Lassen Sie das Argument leer, wird die Voreinstellung wiederhergestellt.

Dieser Befehl ersetzt \bondlength, der bis Version 1.2 noch verfügbar war.

4.18 setbondshape

Mit

```
1 \setbondshape{<basislänge>}{<strichdicke>}{<strichabstand>}
  >}
```

lässt sich `\setcrambond{<basislänge>}{<strichdicke>}{<strichabstand>}` für die `ChemFig`-Formeln *innerhalb* der `myChemistry`-Umgebungen einstellen. Standard sind in dieser Reihenfolge 3 pt, 0.5 pt und 1 pt. Lassen Sie die Argumente leer, wird die jeweilige Voreinstellung wiederhergestellt.

Dieser Befehl ersetzt \bondshape, der bis Version 1.2 noch verfügbar war.

4.19 setrcndist

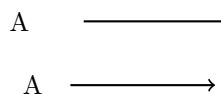
seit v1.2 Die einzelnen Nodes, in denen die Reaktanden und Pfeile geschrieben werden, haben in den myChemistry-Umgebungen einen bestimmten Abstand voneinander. Per Default ist das 1 em. Wenn Sie das ändern wollen, können Sie das mit

```
1 \setrcndist{<länge>}
```

machen. Lassen Sie das Argument leer, wird der Abstand wieder auf 1 em zurückgesetzt.

Beispiel 44

```
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}
```



4.20 setrxnalign/setschemealign

seit v1.2 Mit den Befehlen

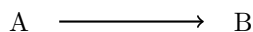
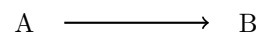
```
1 \setrxnalign{<alignment>}
2 \setschemealign{<alignment>}
```

lässt sich das Default-Ausrichtungsverhalten (siehe [Abschnitt 4.13.1](#) & [Abschnitt 4.14.1](#)) der Umgebungen festlegen. Es gibt die Einstellungsmöglichkeiten `left`, `center` oder `right`.

Lassen Sie das Argument leer, wird die Defaulteinstellung von myChemistry (`center`) wiederhergestellt.

Beispiel 45

```
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}
```



4.21 setschemename

Siehe [Abschnitt 4.14.2](#).

4.22 transition

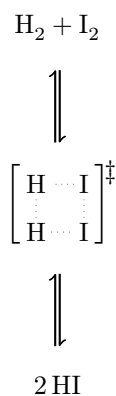
NEU in V1.3

`\transition` funktioniert genau wie `\reactand` (siehe [Abschnitt 4.12](#)). Wenn Sie ältere Versionen von **myChemistry** eingesetzt haben, beachten Sie, dass sich die Befehl-Syntax verändert hat.

```
1 \transition [<ausrichtung>,<anker>,<tikz>]{<formel>}
```

Beispiel 46

```
1 \begin{rxn}
2 \reactand{ \ce{H2 + I2} }
3 \arrow[type=<=>,length=.5,
4 \transition[below]{ \chemfig[
5 \arrow[type=<=>,length=.5,
6 \reactand[below]{ \ce{2 HI} }
7 \end{rxn}
```

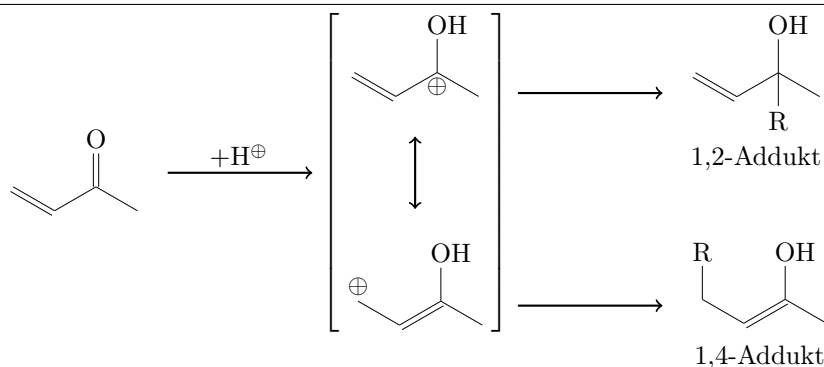


5 Beispiele

5.1 Addition

Ein einfaches Reaktionsschema mit zwei unterschiedlichen Produkten.

Reaktionsschema 10 Additionsreaktion

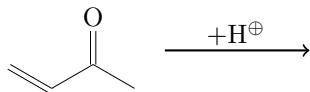


Schritt für Schritt. Zunächst das Edukt und der erste Reaktionspfeil.

```

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

```

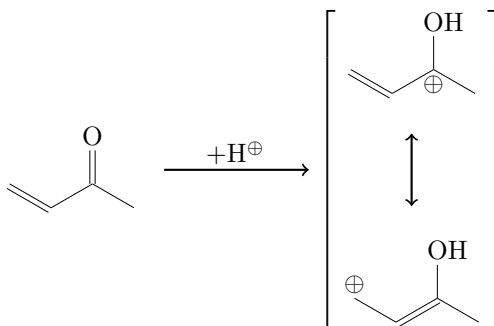


Anschließend die mesomeren Grenzformeln:

```

3 \mesomeric[,gf]{
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 }

```

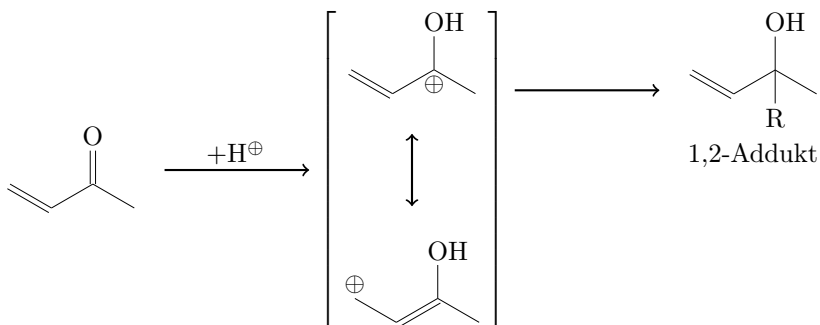


Nun den Branch zum 1,2-Addukt, mit yshift nach oben verschoben:

```

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

```

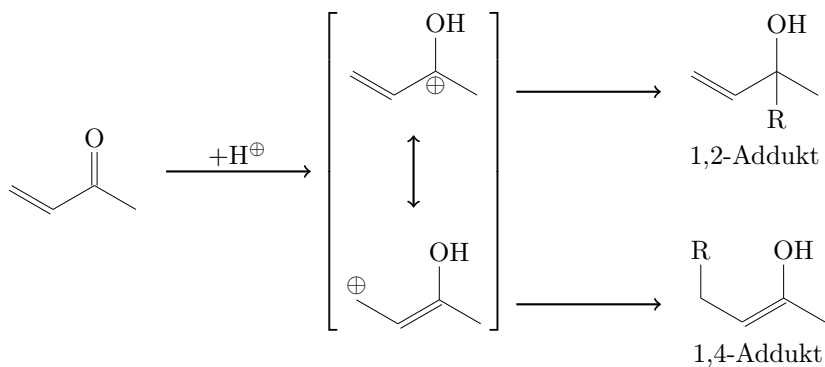


Zuletzt den Branch zum 1,4-Addukt, mit yshift nach unten verschoben:

```

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

```



Der komplette Code ist also der folgende:

```

1 \begin{rxnscheme}{Additionsreaktion}
2   \reactand{ \chemfig{=_[: -30] -[: :60] (=[: :60] O) -[: : -60]}}
3   \arrow{ $+ \Hpl$ }{}
4   \mesomeric[,gf]{
5     \reactand{ \chemfig{=_[: -30] -[: :60] (-[: :60] OH)
6       (-[: : -120, .3, , , white)\oplus} -[: : -60]}} }
7     \marrow[below]
8     \reactand[below]{ \chemfig{\oplus -[6, .3, , , white
9       ] -[: -30]=_[: :60] (-[: :60] OH) -[: : -60]}} }
10  }
11 \branch[right=of gf,,yshift=3em]{
12   \arrow{}{}
13   \reactand{ \chemname{\chemfig{=_[: -30] -[: :60] (-[: :60]
14     OH) (-[: : -120] R) -[: : -60]}}{1,2-Addukt} }
15 }
16 \branch[right=of gf,,yshift=-5em]{
17   \arrow{}{}
18   \reactand{ \chemname{\chemfig{R
19     -[6] -[: -30]=_[: :60] (-[: :60] OH) -[: : -60]}}{1,4-Addukt} }
20 }
21 \end{rxnscheme}

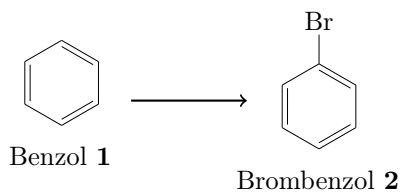
```



```

3  \reactand{
4  \chemname{\chemfig{*6(-----)}}{Benzol \compound{benzol}}
5  }
6  \arrow{>}
7  \reactand{
8  \chemname{\chemfig{*6(-----(-Br)---)}}{Brombenzol \
\compound{brombenzol}}
9  }
10 \end{rxn}

```



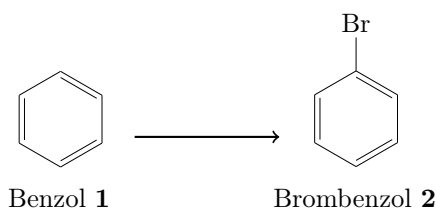
Damit die beiden Benzol-Ringe auf gleicher Höhe erscheinen, haben wir zwei Möglichkeiten. Entweder, wir verschieben den zweiten mit *TikZ*-Code nach oben:

```

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

```

Diese Lösung ist nicht optimal, da dann der Reaktionspfeil nicht mittig sondern etwas zu tief erscheint.

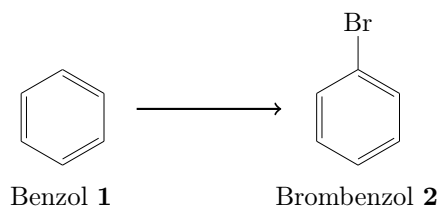


Die zweite Variante wäre, den ersten Ring nach unten zu verschieben. Das können wir nicht mit *TikZ*-Code verwirklichen, da der Pfeil und der zweite Ring jeweils relativ zum vorhergehenden gesetzt werden. Durch ein unsichtbares Brom erreichen wir aber den gewünschten Effekt:

```

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

```



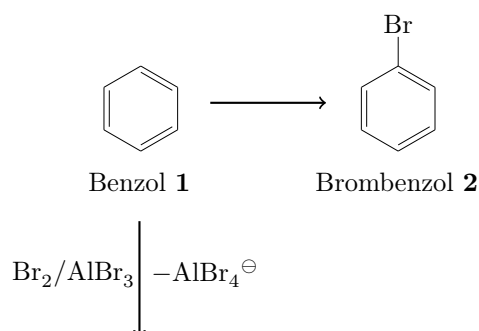
Damit wir nun unterhalb dieser Reaktion einen Reaktionszweig erstellen können, benötigen wir den Befehl `\branch` und wir müssen der ersten Formel einen Namen als Anker geben.

```

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

```

Wir nennen also die erste Substanz `start` und sagen `\branch` mit `below=of start`, dass die Verzweigung unterhalb beginnen soll. Damit der anschließende Reaktionspfeil nach unten zeigt, bekommt `\arrow` den Key `direction=below`. Damit erhalten wir folgendes Bild:

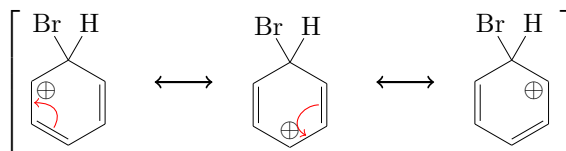


Als nächstes erstellen wir die mesomeren Grenzformeln des Wheland-Komplexes. Hier setzen wir drei weitere Befehle ein: `\mesomeric`, `\marrow` und `\elmove`.

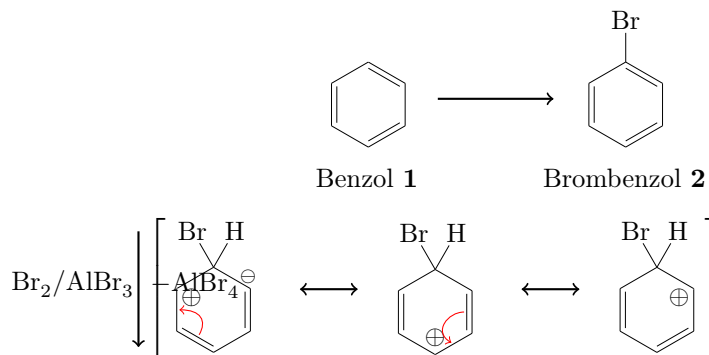
```

1  \mesomeric{
2  \reactand{
3    \chemfig{*6(=[@{e1}] -- (-[:120] Br) (-[:60] H)
4    -(-[: -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}] =[@{e
10   3]) -(-[:120] Br) (-[:60] H) -)}
11   \elmove{e3}{180:4mm}{e4}{150:4mm}
12 }
13 \marrow
14 \reactand{
15   \chemfig{*6(-(-[: -150, .4, , , white]\oplus) -(-[:120] Br
16   ) (-[:60] H) -)}
17 }

```



Setzen wir den Code *in den* `\branch` nach dem Pfeil, ergibt sich folgendes Gesamtbild:



Die Ausrichtung der mesomeren Formeln stimmt offensichtlich nicht und der Pfeil ist nicht mehr, wo er sein soll. Wir könnten folgendes versuchen:

```

1  \begin{rxn}[scale=.8]
2  \setatomsep{1.6em}
3  \reactand[, start]{
4    \chemname{\chemfig{*6(-(-[: , , , white]\phantom{Br})
5    -)}{Benzol} \compound{benzol}}

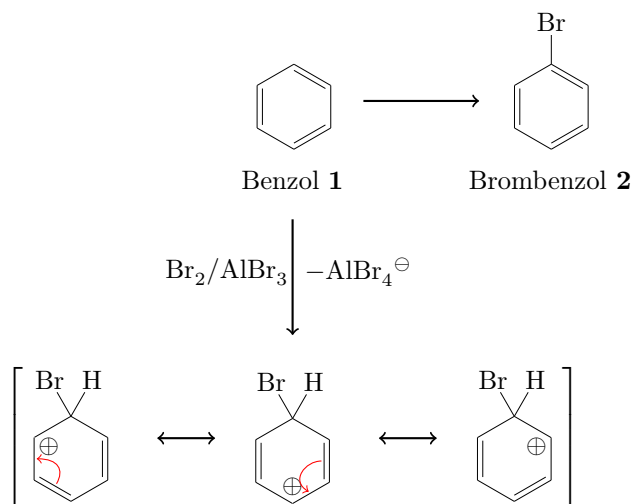
```

```

5   }
6
7   \branch[below=of start]{
8     \arrow[direction=below,both]{\ce{Br2 / AlBr3}}{\$-\ce{
9       AlBr4\om}}{\$}
10    \mesomeric[below]{
11      \reactand{
12        \chemfig{*6(=[@{e1}]--(-[:120]Br)(-[:60]H)
13          -(-[: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}]=[@{
19          e3}]-(-[:120]Br)(-[:60]H)-=)}
20        \elmove{e3}{180:4mm}{e4}{150:4mm}
21      }
22      \marrow
23      \reactand{
24        \chemfig{*6(-=-(-[:150,.4,,,white]\oplus)
25          -(-[:120]Br)(-[:60]H)-=)}
26      }
27    }
28  }
29  \arrow{}{}
30  \reactand{
31    \chemname{\chemfig{*6(-=-(-Br)-=)}}{Brombenzol \
32      compound{brombenzol}}
33  }
34  \end{rxn}

```

Das Ergebnis ist schon besser:

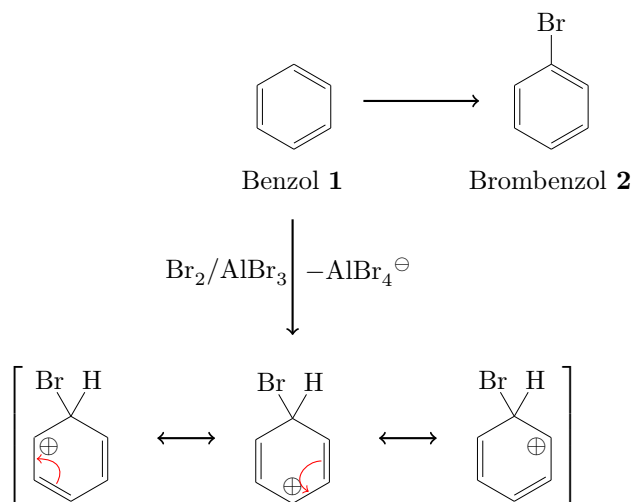


Es ist allerdings für unser Beispiel unbefriedigend, dass die Platzierung zwar unterhalb, aber zentriert erscheint. Um das zu umgehen, werden wir dem Pfeil darüber einen Namen als Anker geben und die mesomeren Formeln als eigenen `\branch` setzen.

```

6 ...
7 \branch[below=of start]{
8   \arrow[direction=below,name=pfeil_a,both]{\ce{Br2 /
9     AlBr3}}{\$-\ce{AlBr4\om}\$}
10 }
11 \branch[below=of pfeil_a]{
12   \mesomeric{
13     \reactand{
14       \chemfig{*6(=[@{e1}]--(-[:120]Br)(-[:60]H)
15         -(-[: -30,.4,,,white]\oplus)-[@{e2}])}
16       \elmove{e1}{60:4mm}{e2}{0:4mm}
17     }
18     \marrow
19     \reactand{
20       \chemfig{*6(-(-[:90,.4,,,white]\oplus)-[@{e4}]=[@{e
21 3}]-(-[:120]Br)(-[:60]H)-=)}
22       \elmove{e3}{180:4mm}{e4}{150:4mm}
23     }
24     \marrow
25     \reactand{
26       \chemfig{*6(-(-[: -150,.4,,,white]\oplus)-(-[:120]
27 Br)(-[:60]H)-=)}
28     }
29   }
30 }
31 ...

```

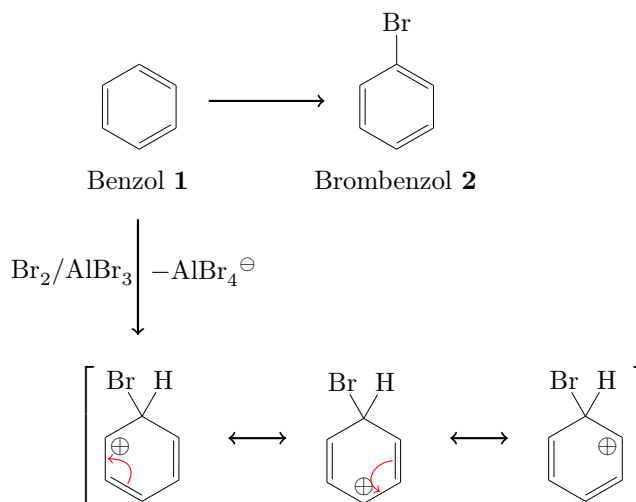


Das scheint auf den ersten Blick keine Verbesserung zu sein. Allerdings können wir den `\branch` mit den TikZ-Keys `xshift` und `yshift` noch beliebig verschieben.

```

6 ...
7 \branch[below=of start]{
8   \arrow[direction=below,name=pfeil_a,both]{\ce{Br2 /
9     AlBr3}}{\$-\ce{AlBr4\om}}\$}
10 }
11 \branch[below=of pfeil_a,mesomerie,xshift=8.5em]{
12   \mesomeric{
13     \reactand{
14       \chemfig{*6(=[@{e1}]--(-[:120]Br)(-[:60]H)
15         -(-[: -30,.4,,,white]\oplus)-[@{e2}])}
16       \elmove{e1}{60:4mm}{e2}{0:4mm}
17     }
18     \marrow
19     \reactand{
20       \chemfig{*6(-(-[:90,.4,,,white]\oplus)-[@{e4}]=[@{e
21         3]}-(-[:120]Br)(-[:60]H)-=)}
22       \elmove{e3}{180:4mm}{e4}{150:4mm}
23     }
24     \marrow
25     \reactand{
26       \chemfig{*6(--(-[: -150,.4,,,white]\oplus)-(-[:120]
27         Br)(-[:60]H)-=)}
28     }
29   }
30 }
31 ...

```

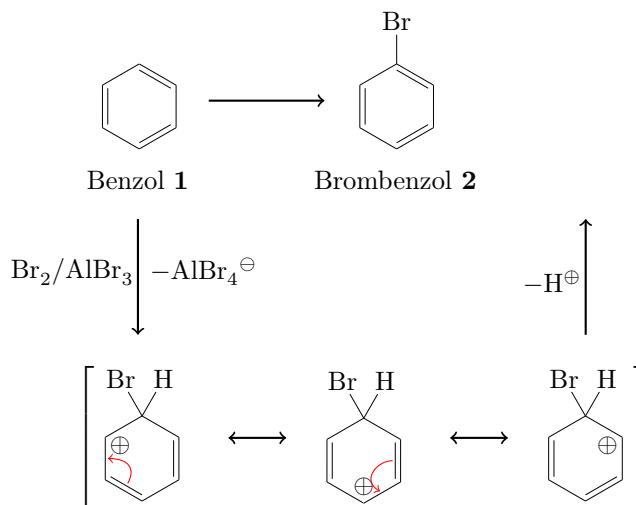


Damit wir mit dem Pfeil zurück nicht wieder das Problem mit der Zentrierung haben, bekommt er ebenfalls seinen eigenen `\branch`.

```

6 ...
7 \branch[below=of start]{
8   \arrow[direction=below,name=pfeil_a,both]{\ce{Br2 /
9     AlBr3}}{\$-\ce{AlBr4\om}}\$}
10 }
11 \branch[below=of pfeil_a,mesomerie,xshift=8.5em]{
12   \mesomeric{
13     \reactand{
14       \chemfig{*6(=[@{e1}]---(-[:120]Br)(-[:60]H)
15         -(-[: -30,.4,,white]\oplus)-[@{e2}])}
16       \elmove{e1}{60:4mm}{e2}{0:4mm}
17     }
18     \marrow
19     \reactand{
20       \chemfig{*6(-(-[:90,.4,,white]\oplus)-[@{e4}]=[@{e
21         3]}-(-[:120]Br)(-[:60]H)-=)}
22       \elmove{e3}{180:4mm}{e4}{150:4mm}
23     }
24     \marrow
25     \reactand{
26       \chemfig{*6(---(-[: -150,.4,,white]\oplus)-(-[:120]
27         Br)(-[:60]H)-=)}
28     }
29   }
30 }
31 \branch[above=of mesomerie,,xshift=7.25em]{
32   \arrow[direction=above]{\$-\Hpl\$}{}
```


29 }
30 ...



Nun sind wir fast am Ziel. Der Hauptreaktion ist noch zu kurz.

```

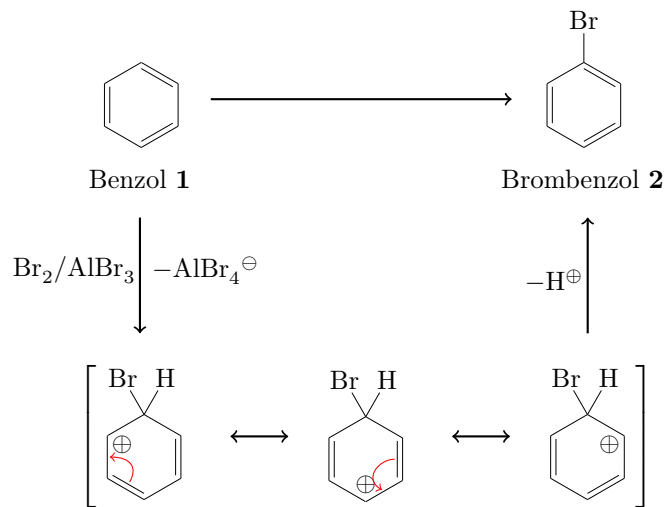
1 \begin{rxn}[scale=.8]
2 \setatomsep{1.6em}
3 \reactand[,start]{\chemname{\chemfig{*6(---(-[,,,,white
4 \phantom{Br})--)}}}{Benzol \compound{benzol}} }
5 \branch[below=of start]{
6 \arrow[direction=below,name=pfeil_a,both]{\ce{Br2 /
7 AlBr3}}{\$-\ce{AlBr4\om}\$}
8 }
9 \branch[below=of pfeil_a,mesomerie,xshift=8.5em]{
10 \mesomeric{
11 \reactand{
12 \chemfig{*6(=[@{e1}]---(-[:120]Br)(-[:60]H)
13 -(-[: -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}]=[@{e
19 3]--(-[:120]Br)(-[:60]H)--)}
20 \elmove{e3}{180:4mm}{e4}{150:4mm}
21 }
22 \marrow
23 \reactand{
24 \chemfig{*6(---(-[: -150,.4,,,white]\oplus)-(-[:120]
25 Br)(-[:60]H)--)}

```

```

22     }
23   }
24 }
25 \branch[above=of mesomerie,,xshift=7.25em]{
26   \arrow[direction=above]{$-\text{Hpl}$}{}
27 }
28
29 \arrow[length=2.6]{}{}
30 \reactand{ \chemname{\chemfig{*6(-==(-Br)-=)}}{
31   Brombenzol \compound{brombenzol}} }
31 \end{rxn}

```



5.3 TikZ, myChemistry und ChemFig für eine umfangreichere Synthese

Da die ChemFig-Befehle innerhalb einer tikzpicture-Umgebung problemlos funktionieren, lassen sich mit dem \merge-Befehl von myChemistry auch größere Synthesen realisieren. Die anderen myChemistry-Befehle funktionieren nicht ohne weiteres, da sie alle auf einer chain angeordnet werden. Solange Sie myChemistry eingebunden haben, müssen Sie allerdings kaum eine tikzlibrary zusätzlich einbinden. Im Beispiel wurde direkt auf die Gleitumgebung rxnfloat von myChemistry zugegriffen.

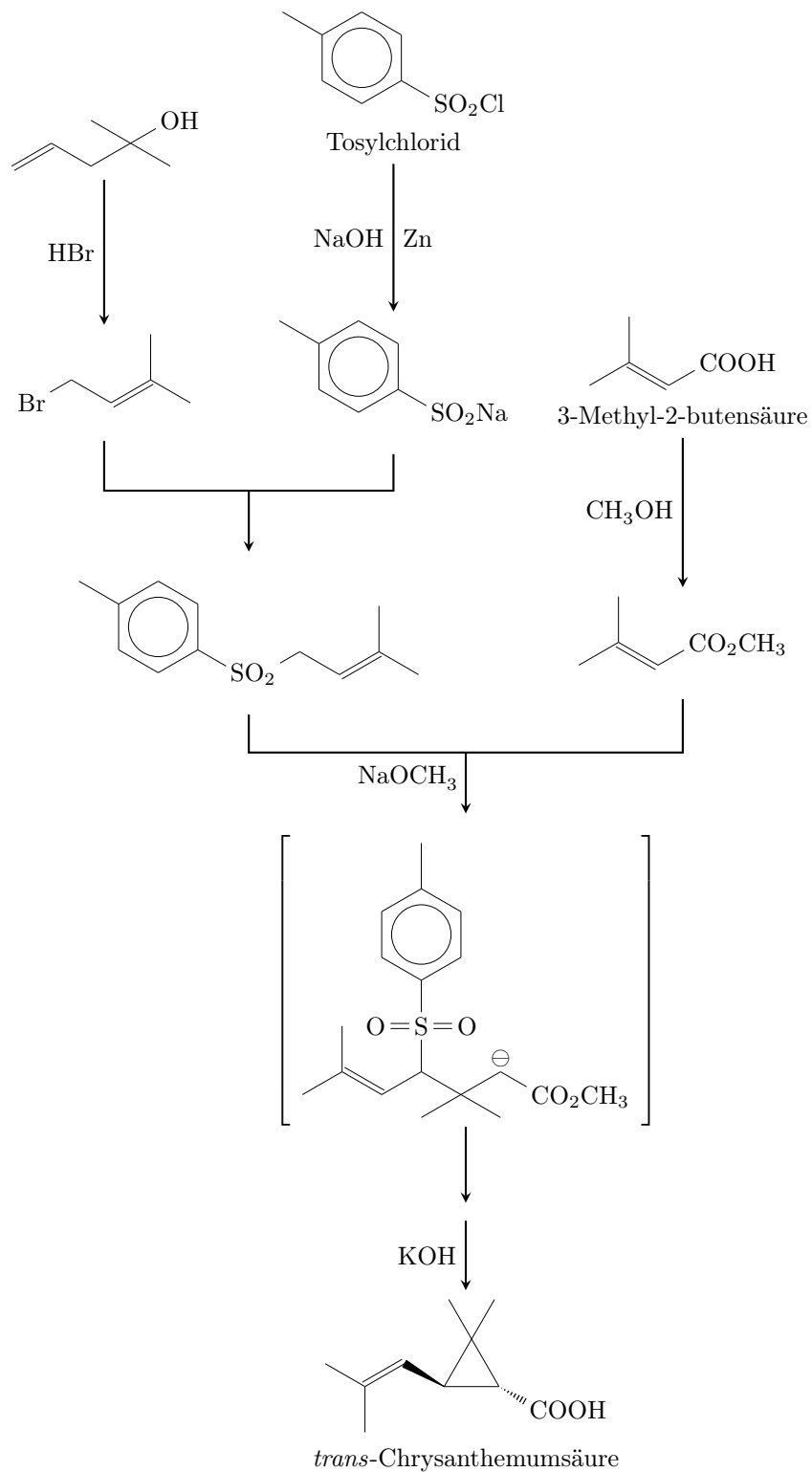
```

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
8   {=_[:30]-[:60]-[:60](-[:60])(-[:120])-[:0]OH}};
9 \node(b) at (0,-4) {\chemfig{Br
10  -[:30]-[:60]=[:60](-[:60])-[:60]}};
11 \draw[-stealth,thick] (a.south) -- node[left]{HBr} (b.
12  north);
13 \node(c) at (5,1) {\chemname{\chemfig{**6(--(-SO_2Cl)
14  ---(-)-)}}{Tosylchlorid}};
15 \node(d) at (5,-4) {\chemfig{**6(--(-SO_2Na)---(-)-)}};
16 \draw[-stealth,thick] (c.south) -- node[left]{NaOH} node
17  [right]{Zn} (d.north);
18 \node(e) at (2.5,-8.5) {\chemfig{**6(--(-SO
19  _2-[:30]-[:60]=[:60](-[:60])-[:60])---(-)-)}};
20 \node(f) at (10,-4) {\chemname{\chemfig{-[:30](-[:60])
21  =[:60]-[:60]COOH}}{3-Methyl-2-butensäure}};
22 \node(g) at (10,-8.5) {\chemfig{-[:30](-[:60])
23  =[:60]-[:60]CO_2CH_3}};
24 \draw[-stealth,thick] (f.south) -- node[left]{\ce{CH3OH
25  }} (g.north);
26 \merge{e}{b}{d}
27 \node[left delimiter={},right delimiter={}] (h) at
28  (6.25,-14.5) {\chemfig{-[:30](-[:60])
29  =^[:60]-[:60](-[:60]S(=[90]O)(=[-90]O)
30  -[:0]**6(---(-)---))-[:60](-[:0])(-[:120])
31  -[:60](-[:60],.5,,white)\ominus)-[:60]CO_2CH_3}};
32 \node at (5.25,-11) {\ce{NaOCH3}};
33 \merge{h}{e}{g}
34 \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}-Chrysanthemumsäure}}};
23 \draw[-stealth,thick] (h.south) -- (i.north);
24 \draw[-stealth,thick] (i.south) -- node[left]{KOH} (j.
    north);
25 \end{tikzpicture}
26 \caption{Synthese von Chrysanthemumsäure}
27 \end{rxnfloat}
28
```

Reaktionsschema 12 Synthese von Chrysanthemumsäure



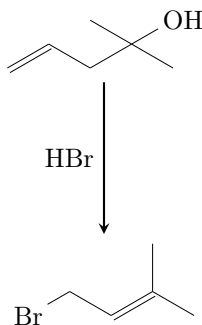
Gehen wir den Code Stück für Stück durch.

```

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 den Zeilen 1 – 6 wird die Umgebung begonnen und die Voreinstellungen vorgenommen, damit die Formeln nicht zu groß werden. In den Zeilen 7 – 9 werden die beiden ersten Formeln erstellt (Zeilen 7 und 8) und mit Reaktionspfeil (Zeile 9) verbunden.

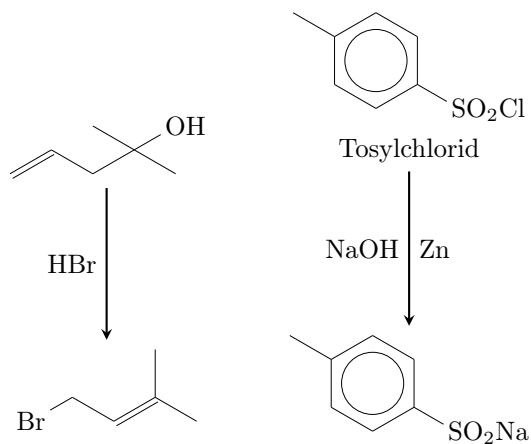


```

10 \node(c) at (5,1) {\chemname{\chemfig{**6(--(-SO_2Cl)
   ---(-)-)}}{Tosylchlorid}};
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 den drei folgenden Zeilen 10 – 12 wird der zweite Syntheseast erstellt und mit Reaktionspfeil verbunden.

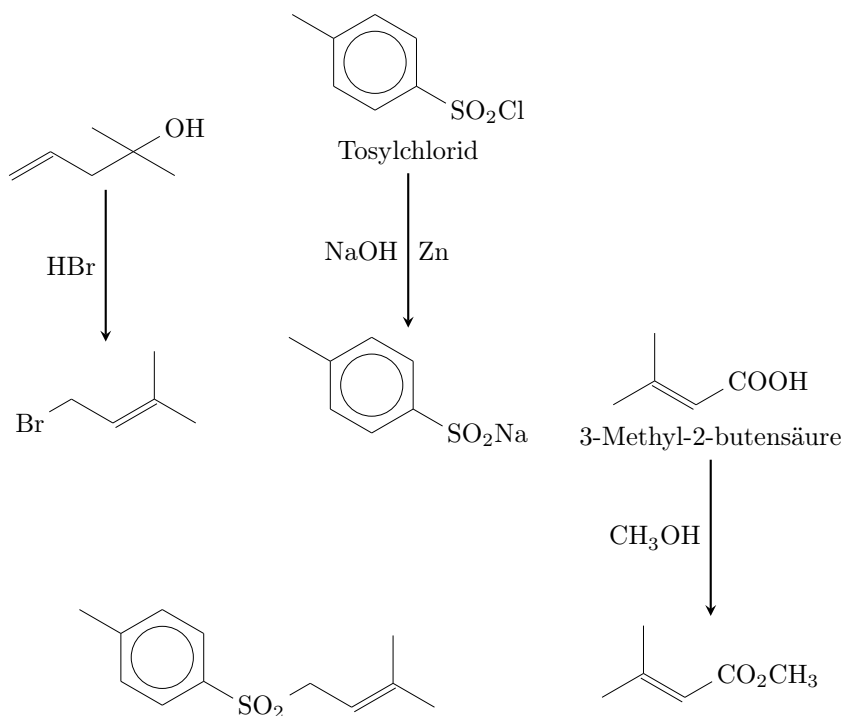


```

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

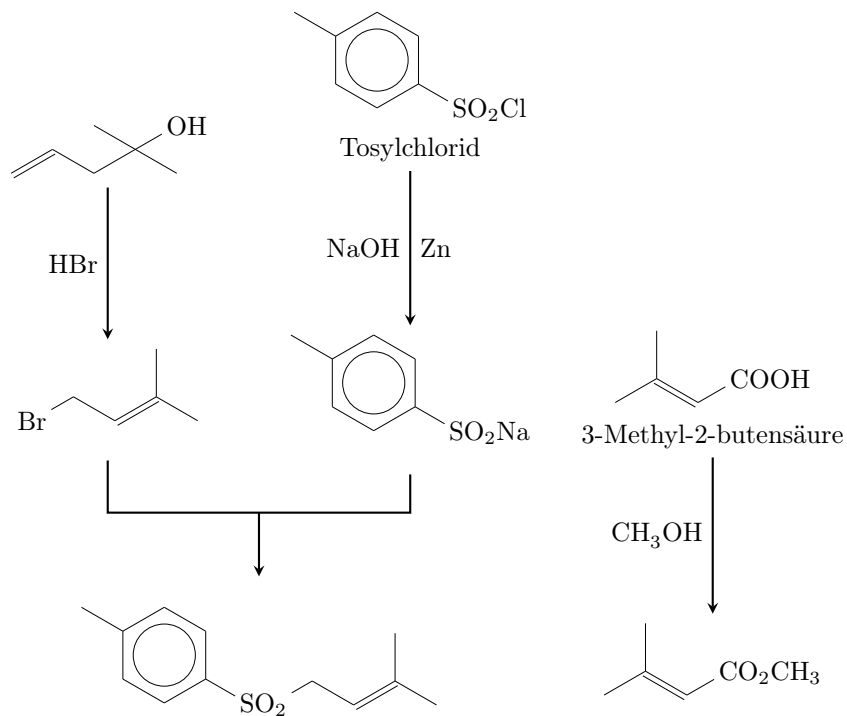
```

In den Zeilen 13 – 16 wird der dritte Ast sowie das Ergebnis der ersten beiden Äste erstellt.



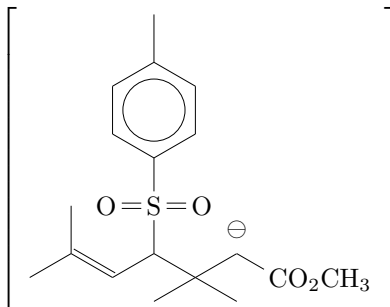
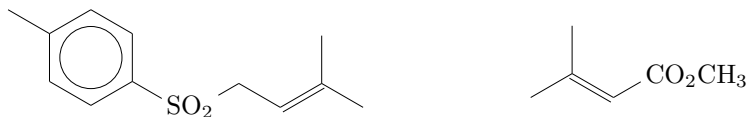
17 `\merge{e}{b}{d}`

In Zeile 17 werden nun die beiden ersten Äste zusammengeführt.



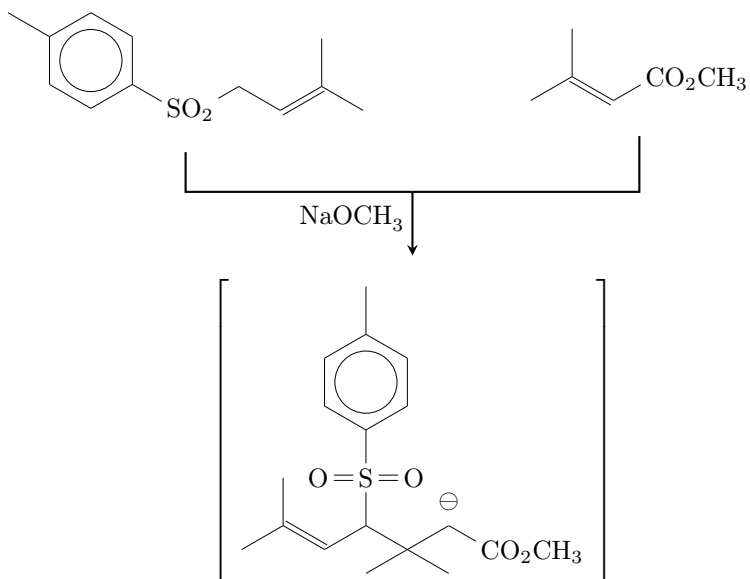
18 `\node[left delimiter={},right delimiter={}] (h) at (6.25,-14.5) {\chemfig{-[:30](-[:60])=^[:-60]-[:60](-[:60]S(=[:90]O)(=[::-90]O)-[:0]**6(---(-)---))-[:-60](-[:0])(-[:-120])-[:60](-[:60],.5,,white)\ominus)-[:-60]CO_2CH_3}};`

In Zeile 18 erstellen wir den Übergangszustand.



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

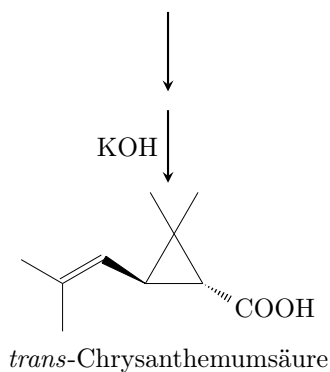
In den Zeilen 19 und 20 werden die Äste zusammengeführt und der Pfeil beschriftet.



```
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}-Chrysanthemumsäure}};
```

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

In den abschließenden Zeilen 21 – 28 wird zunächst eine leere node erstellt (Zeile 21), dann das Produkt (Zeile 22). In den Zeilen 23 und 24 werden die beiden letzten Reaktionspfeile erstellt, in den letzten vier Zeilen die Umgebung dann beendet.



6 Nachwort

myChemistry steckt noch in den Kinderschuhen. Das bedeutet, dass vermutlich noch eine ganze Reihe von Bugs enthalten sind. Bestimmt fehlt auch noch das eine oder andere Feature, das nützlich wäre. Da ich das Paket nur in meiner Freizeit testen und verbessern kann, bin ich über *jede* Art von Feedback sehr froh. Wenn Ihnen myChemistry gefällt, dann helfen Sie doch, es zu verbessern, indem Sie mir Ihre Erfahrungen mitteilen.

Auch wenn ich mich bemüht habe, sinnvolle chemische Reaktionen einzusetzen, habe ich nicht extra überprüft, ob jedes Beispiel chemisch sinnvoll ist. Vertrauen Sie den Beispielen diesbezüglich nicht, sondern sehen Sie in einem Lehrbuch der Chemie nach.

Viel Spaß mit myChemistry!

Clemens Niederberger, Berlin, 04. April 2011