

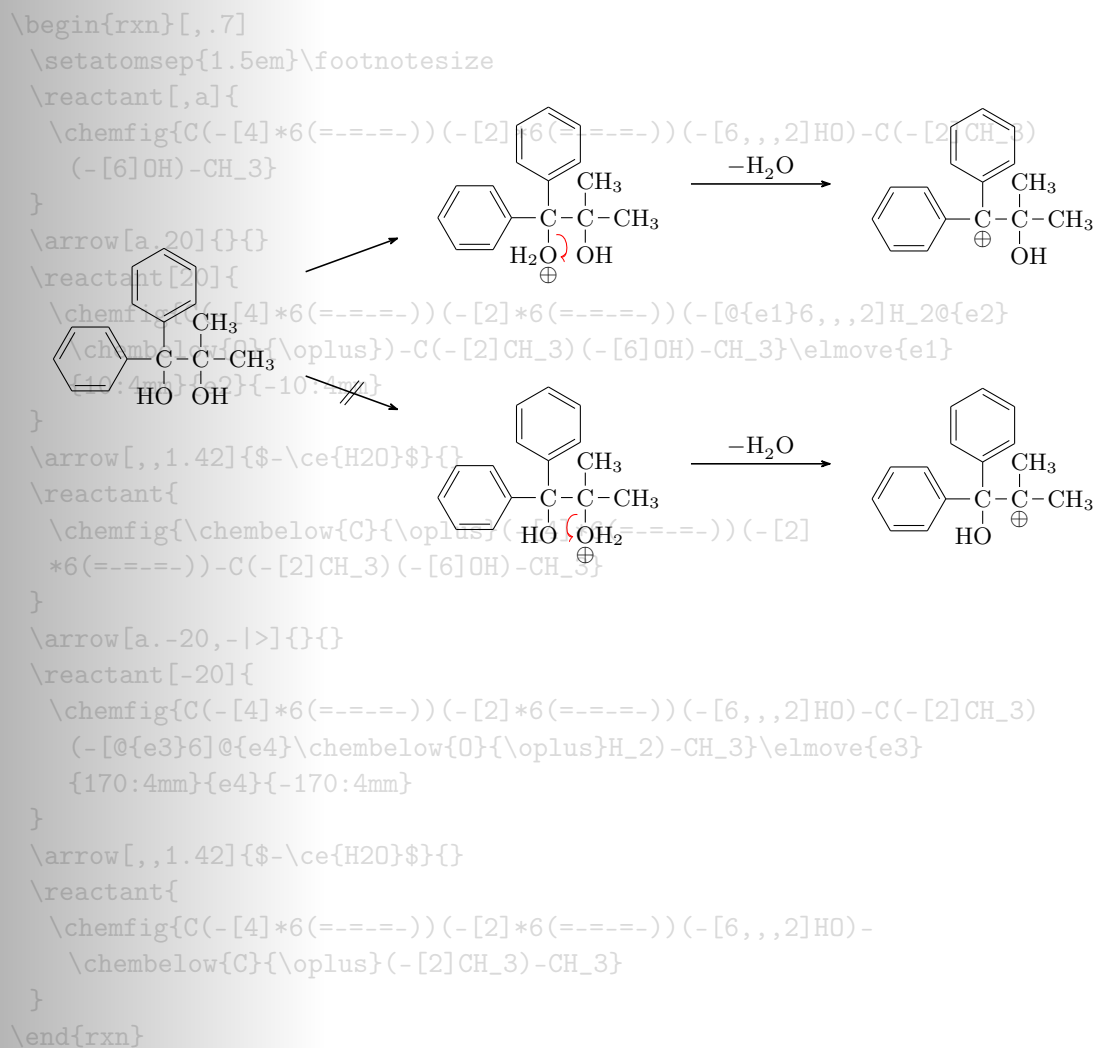
myChemistry

v1.5.1

27. April 2011

Clemens Niederberger

Reaktionsschemata mit L^AT_EX und ChemFig erstellen



Inhaltsverzeichnis

ABSCHNITT 1 Über	3
1.1 Änderungen	3
<i>Version 1.2, 3 • Version 1.3, 4 • Version 1.4, 4 • Version 1.5.1, 4</i>	
1.2 Lizenz	5
1.3 Voraussetzungen	5
1.4 Die Idee	5
ABSCHNITT 2 Verwendung	6
2.1 Hintergrund	6
2.2 Das Grundprinzip	7
2.3 Wie funktioniert's?	8
<i>Basisbefehle, 8 • Positionierung, 10 • Verzweigungen, 11 • Nummerierte Schemata, 14</i>	
2.4 Voreinstellungen	15
2.5 Paket-Optionen	17
ABSCHNITT 3 Fortgeschrittene Anwendung, Verwendung von TikZ	18
3.1 Die Ausrichtungsfrage	18
3.2 Mit TikZ andere Ziele erreichen	22
ABSCHNITT 4 Alphabetische Befehlsreferenz	24
4.1 anywhere	24
4.2 arrow	25
<i>Optionen, 25 • Ausrichtung, 28 • Aussehen, 29</i>	
4.3 branch	29
<i>Positionierung, 31 • Ausrichtungsprobleme, 32</i>	
4.4 chemand	33
4.5 dummy	33
4.6 elmove	34
4.7 makeinvisible	34
4.8 makevisible	35
4.9 marrow	35
4.10 mCsetup	35
4.11 merge	36
4.12 mesomeric	40
4.13 reactant	42
4.14 rxn (Umgebung)	43
<i>Optionen, 43</i>	

4.15 rxnscheme (Umgebung)	45
<i>Optionen, 45 • rxnscheme anpassen, 47</i>	
4.16 setarrowlabel	49
4.17 setarrowlength	50
4.18 setarrowline	50
4.19 setatomsizes	50
4.20 setbondlength	50
4.21 setbondshape	51
4.22 setmergelength	51
4.23 setrendist	51
4.24 setrxnalign/setschemealign	51
4.25 setschemename	52
4.26 transition	52
<hr/>	
ABSCHNITT 5 Nachwort	53
<hr/>	
ABSCHNITT 6 Dank	53
<hr/>	
STICHWORTVERZEICHNIS	54

1 Über

1.1 Änderungen

Die neuesten Änderungen sind mit **neu** gekennzeichnet.

1.1.1 Version 1.2

Neben einigen Bugfixes gibt es in Version v1.2 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 4.2](#)).

Auch die Umgebungen haben nun ein paar Möglichkeiten mehr, den persönlichen Vorstellungen angepasst zu werden (siehe [Abschnitt 4.14.1](#), [Abschnitt 4.15.1](#) und [Abschnitt 4.23](#)).

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

1.1.2 Version 1.3

Die Befehle `\branch`, `\mesomeric`, `\reactant` 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 [<pos>, <name>, <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 \reactant{\ce{Br2}}{\arrow{$h\nu$}}{\reactant[right,
   draw, inner sep=5pt]{\ce{2 \lewis{0., Br}}}}{}
3 % jetzt:
4 \reactant{\ce{Br2}}\arrow{$h\nu$}{\reactant[, , draw,
   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 4.17](#), [Abschnitt 4.19](#), [Abschnitt 4.20](#), [Abschnitt 4.21](#) und [Abschnitt 4.10](#).

Es gibt den neuen Befehl `\chemand`, der ein + erzeugt, siehe [Abschnitt 4.4](#).

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

1.1.3 Version 1.4

Der Befehl `\merge` wurde neu geschrieben, so dass der Pfeil auch beschriftet werden kann.

Der eigentliche Zweck des Befehls `\dummy` ist obsolet geworden. Der Befehl existiert aber weiterhin.

Bei den beiden Umgebungen `rxn` und `rxnscheme` hat sich die Verwendung der Optionen geändert, siehe [Abschnitt 4.14.1](#) und [Abschnitt 4.15.1](#).

Die Keys von `\arrow` sind in Optionen verändert worden, um die Syntax des Befehls mit den anderen anzugleichen. Zudem gibt es drei neue Pfeiltypen, siehe [Abschnitt 4.2](#). Neu ist außerdem das Aussehen der Pfeile und eine Möglichkeit, die Liniendicke der Pfeile anzupassen, siehe [Abschnitt 4.18](#).

Es gibt einen weiteren neuen Befehl: `\anywhere` ([Abschnitt 4.1](#)), mit dem Text oder Formeln außerhalb der Chain gesetzt werden können.

1.1.4 Version 1.5.1

NEU

Im Hintergrund ist viel passiert, was einige neue Features bei der Anwendung gebracht hat. Die wichtigste Neuerung ist eine deutlich flexiblere Richtungsangabe bei allen Befeh-

len (Reaktanden, Pfeile, Zweige ...). Anstelle der starren `right below` u.ä. kann jetzt der Winkel angegeben werden.

Es gibt die neuen Pfeiltypen `<=>` und `<<=>`, mit denen ein verschobenes Gleichgewicht angezeigt werden kann.

Die Default-Platzierung von `rxnscheme` wurde von H in htp geändert.

Die konkreten Einsatz-Beispiele wurden aus der Dokumentation ausgelagert und befinden sich in der Datei `examples.tex` bzw. `examples.pdf`.

Die Befehle `\arrow`, `\reactant`, `\mesomeric`, `\transition`, `\anywhere`, `\dummy`, `\branch` und `\chemand` sind nur innerhalb der Umgebungen `rxn` und `rxnscheme` definiert.

1.2 Lizenz

myChemistry v1.5.1 steht unter der L^AT_EX 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 die Option `both` des Befehls `\arrow` diese Version. Wenn Sie die Option 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

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

- ChemFig¹,
- wenn vorhanden ‘mhchem’² in der Version 3,
- wenn vorhanden ‘chemexec’³ und
- wenn vorhanden ‘chemcompounds’⁴ 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.

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, Reaktionsgleichungen zu erstellen. 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.

¹von Christian Tellechea, <http://www.ctan.org/tex-archive/macros/generic/chemfig/>

²von Martin Hensel, <http://www.ctan.org/tex-archive/macros/latex/contrib/mhchem/>

³von mir, <http://www.ctan.org/tex-archive/macros/latex/contrib/chemexec/>

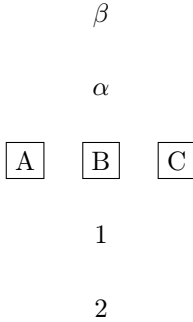
⁴von Stephan Schenk, <http://www.ctan.org/tex-archive/macros/latex/contrib/chemcompounds/>

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, das Grundprinzip sollte reichen.

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

Basis bilden folgende drei Befehle:

```

1 \begin{rxn}[<ausrichtung>,<skalierung>]
2 \reactant[<pos>,<name>,<tikz>]{<formel>}
3 \arrow[<pos>,<typ>,<längenfaktor>,<name>,<both>,<tikz>]{<
  oben>}{<unten>}
4 \end{rxn}

```

Die Schemata werden innerhalb der `rxn`-Umgebung erstellt. Dort werden Reaktanden und Pfeile gesetzt.

Beispiel 4

```

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

```



Ohne Optionen werden Reaktanden und Pfeile immer rechts voneinander gesetzt. Möchte man das nicht, hat man die Möglichkeit, über `<pos>` die Positionierung zu ändern.

Schlüsselwort	pos. Winkel	neg. Winkel
right	0	± 360
right above	45	-315
above	90	-270
above left	135	-225
left	180	-180
below left	225	-135
below	270	-90
below right	315	-45

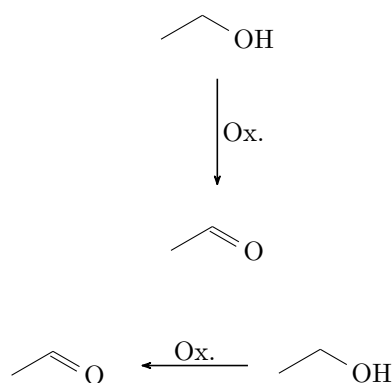
Tabelle 1: Schlüsselwörter zur Positionierung

Beispiel 5

```

1 % Anordnung mit
  Schlüsselwoertern:
2 \begin{rxn}
3 \reactant{ \chemfig
  {-[::30]-[::-60]OH} }
4 \arrow[below]{Ox.}{}
5 \reactant[below]{ \chemfig
  {-[::30]=[::-60]O} }
6 \end{rxn}
7 % Anordnung durch Angabe des
  Winkels zur Horizontalen:
8 \begin{rxn}
9 \reactant{ \chemfig
  {-[::30]-[::-60]OH} }
10 \arrow[180]{}{Ox.}
11 \reactant[180]{ \chemfig
  {-[::30]=[::-60]O} }
12 \end{rxn}

```



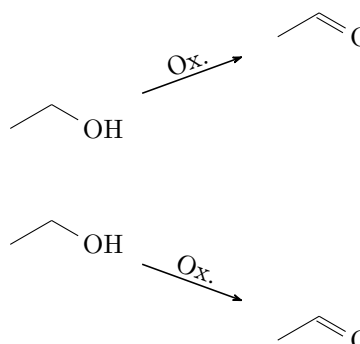
Sie sehen im letzten Beispiel, dass die Positionierung durch Schlüsselwörter (siehe [Tabelle 1](#)) wie `below` oder durch Angabe eines Winkels aus dem Intervall $[-360^\circ; 360^\circ]$ zur Horizontalen. 0° entspricht `right`, der Voreinstellung. *Positive Winkel* bedeuten eine Drehung im *Gegenuhreigersinn*, negative eine im *Uhrzeigersinn*, ganz in der mathematischen Bedeutung.

Beispiel 6

```

1 \begin{rxn}
2 \reactant{ \chemfig
3 {-[::30]-[::-60]OH} }
4 \arrow[20]{Ox.}{}
5 \end{rxn}
6 \begin{rxn}
7 \reactant{ \chemfig
8 {-[::30]-[::-60]OH} }
9 \arrow[-20]{Ox.}{}
10 \end{rxn}

```

**2.3.2 Positionierung**

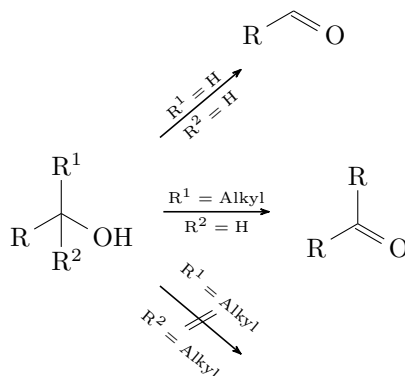
Die Reaktanden und Pfeile können nicht nur durch Schlüsselwörter und Winkel alleine positioniert werden, sondern auch in Bezug auf einen anderen Reaktanden oder Pfeil.

Beispiel 7

```

1 \begin{rxn}
2 \reactant[,start]{ \chemfig{R-[::30](-[::60]R|^1) (-[::-120]R|^2)
3 -[::-60]OH} }
4 \arrow[40]{\tiny$\text{R}^1=\text{H}$} {\tiny$\text{R}^2=\text{H}$}}
5 \reactant[40]{ \chemfig{R-[::30]=[::-60]O} }
6 \arrow[start.0]{\tiny$\text{R}^1=\text{Alkyl}$} {\tiny$\text{R}^2=\text{Alkyl}$}}
7 \reactant{ \chemfig{R-[::30](-[::60]R)=[::-60]O} }
8 \arrow[start.-40,-|>]{\tiny$\text{R}^1=\text{Alkyl}$} {\tiny$\text{R}^2=\text{Alkyl}$}}
9 \end{rxn}

```



Im letzten Beispiel wurden dem ersten Reaktanden der <name> start gegeben. Dar-

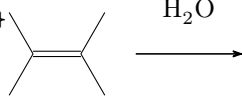
auf konnten sich die Pfeile in Zeilen 5 und 7 in der Positionierung beziehen. Der *zuvor* vergebene Name fungiert als Anker für den nächsten Reaktanden oder Pfeil, wenn die Positionierung nach dem Muster

```
1 <anker>.<winkel>
```

angegeben wird. Auch Pfeile können einen Anker bekommen. Der Ankerpunkt eines Pfeils sitzt immer in der Mitte eines Pfeils und hat *keine* Ausdehnung.

Beispiel 8

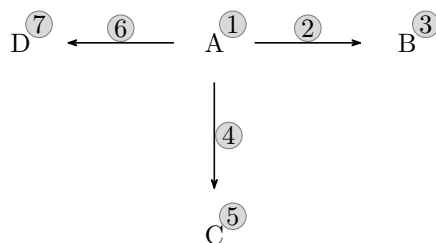
```
1 \begin{rxn}
2 \reactant{\chemfig
3 \arrow[,,,pfeil]{}{}
4 \reactant[pfeil.90]{\ce{H2O}}
5 \end{rxn}
```



Bei dieser Art der Positionierung wird die Kette, auf der die Reaktanden und Pfeile sitzen, *nicht* unterbrochen.

```
1 \begin{rxn}
2 \reactant[,a]{A}
3 \arrow{}{}
4 \reactant{B}
5 \arrow[a.-90]{}{}
6 \reactant[-90]{C}
7 \arrow[a.180]{}{}
8 \reactant[180]{D}
9 \end{rxn}
```

Alle sieben Objekte dieses Beispiels sitzen logisch gesehen auf einer Kette. Das nächste Objekt wird nun per Default rechts daneben geschrieben, wenn keine andere Positionierung angegeben wird.



2.3.3 Verzweigungen

Um die Kette zu unterbrechen, gibt es den folgenden Befehl:

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

Bei einem Branch funktioniert die Positionierung ein bisschen anders als bei den bisherigen Objekten, auch wenn die Syntax ähnlich ist. Zusätzlich zu den drei bisherigen Positionierungsmöglichkeiten kommen beim Branch noch zwei weitere dazu. Alle Positionierungen, die sich auf einen Anker beziehen, sorgen dafür, dass der Branch *nicht* auf der Kette sitzt, sondern ein echter Ast ist.

1	<winkel>	% auf der Kette
2	<schluessel>	% auf der Kette
3	<anker>.<winkel>	% nicht auf der Kette
4	on chain=going <schluessel>	% auf der Kette
5	<schluessel>=of <anker>	% nicht auf der Kette

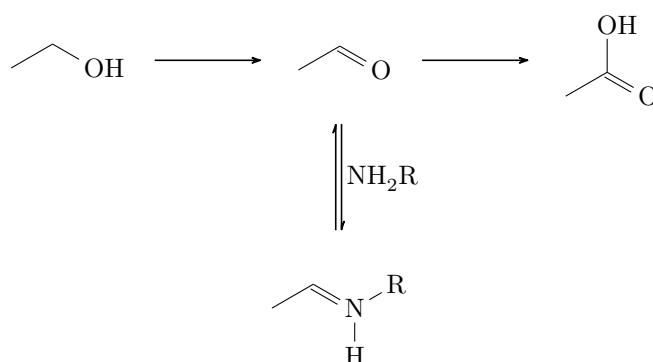
Als <schluessel> können die Werte aus [Tabelle 1](#) verwendet werden.

Beispiel 9

```

1 \begin{rxn}
2 \reactant{ \chemfig{-[::30]-[::-60]OH} }
3 \arrow{}{}
4 \reactant[, carbonyl]{ \chemfig{-[::30]=_[::-60]O} }
5 \branch[carbonyl.-90]{
6 \arrow[-90,<=>]{\ce{NH2R}}{}
7 \reactant[-90]{ \chemfig{-[::30]=_[::-60]N(-[6]H)-[::60]R} }
8 }
9 \arrow{}{}
10 \reactant{ \chemfig{-[::30](-[::60]OH)=_[::-60]O} }
11 \end{rxn}

```



Beachten Sie im letzten Beispiel, dass der Pfeil und der Reaktand nach dem Branch die ursprüngliche Kette fortsetzen.

```

1 \begin{rxn}
2 \reactant[, a]{A}
3 \arrow{}{}
4 \reactant{B}
5 \branch[a.-90]{
6 \arrow[-90]}{}

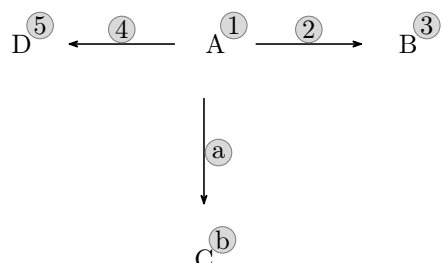
```

```

7     \reactant[-90]{C}
8   }
9   \arrow[a.180]{}{}
10  \reactant[180]{D}
11  \end{rxn}

```

Die Kette wird durch den Ast unterbrochen, der seinerseits eine weitere Kette startet.



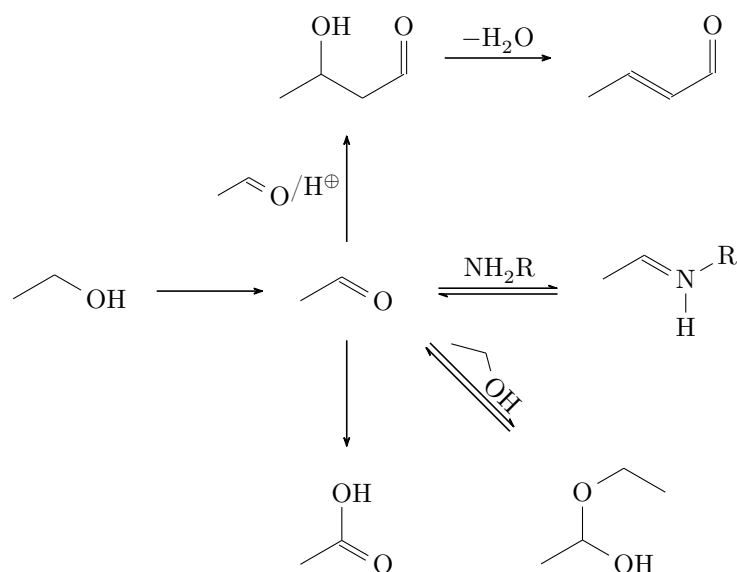
Durch `\branch` sind verzweigtere Schemata möglich:

Beispiel 10

```

1  \begin{rxn}
2  \reactant{ \chemfig{-[:30]-[:60]OH} }
3  \arrow{}{}
4  \reactant[,carbonyl]{ \chemfig{-[:30]=_[:60]O} }
5  \arrow[-90]{}{}
6  \reactant[-90]{ \chemfig{-[:30](-[:60]OH)=_[:60]O} }
7  \branch[right=of carbonyl]{
8    \arrow[,<=>,1.12]{\ce{NH2R}}{}
9    \reactant{ \chemfig{-[:30]=_[:60]N(-[6]H)-[:60]R} }
10 }
11 \branch[below right=of carbonyl]{
12   \arrow[-45,<=>,1.12]{ \chemfig{[,.75]-[:30]-[:60]OH} }{}
13   \reactant[-45]{ \chemfig{-[:30](-[:60]O-[:60]-[:60]) -[:60]OH} }
14 }
15 \arrow[carbonyl.90]{ \chemfig{[,.75]-[:30]=_[:60]O}/\Hpl }{}
16 \reactant[90]{ \chemfig{-[:30](-[:60]OH)-[:60]-[:60]=[:60]O} }
17 \arrow{ $\-\ce{H2O}$ }{}
18 \reactant{ \chemfig{-[:30]=[:60]-[:60]=[:60]O} }
19 \end{rxn}

```



2.3.4 Nummerierte Schemata

Es gibt noch eine weitere Umgebung, innerhalb derer Schemata erstellt werden können.

```

1  \begin{rxnscheme}[<label>,<platzierung>,<ausrichtung>,<
    skalierung>,<titel>]{<caption>}
2  ...
3  \end{rxnscheme}

```

Diese Umgebung stellt eine Gleitumgebung für Schemata bereit, die eine Beschriftung `<caption>`, ein Label `<label>` und die üblichen Platzierungen `<platzierung>` wie `hpt` (Voreinstellung) erhalten kann.

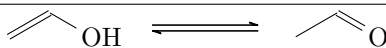
Beispiel 11

```

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

```

Reaktionsschema 1 Keto-Enol-Tautomerie



2.4 Voreinstellungen

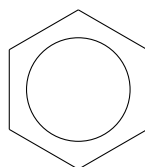
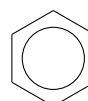
Es gibt einige Voreinstellungen, die zum Teil meinem persönlichen Geschmack geschuldet sind, die Sie aber nach Bedarf ändern können. So gelten für die `ChemFig`-Formeln innerhalb der `myChemistry`-Umgebungen folgende Voreinstellungen:

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

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

Beispiel 12

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



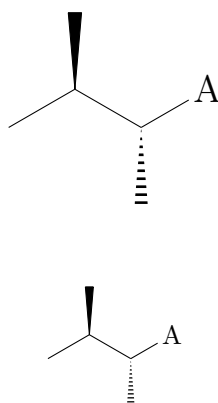
Sie können die Voreinstellungen von `myChemistry` über folgende Befehle ändern:

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

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

Beispiel 13

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



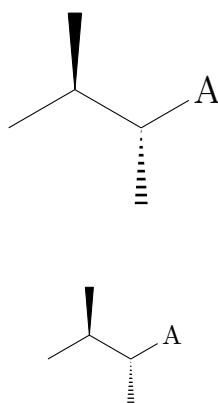
Wollen Sie nur die Parameter einer Umgebung ändern, verwenden Sie *innerhalb der Umgebung* die Befehle von **ChemFig** und die L^AT_EX-Befehle für die Schriftgröße.

Beispiel 14

```

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

```



Reaktionspfeile haben als Standardwert die Länge 4em, die Standarddicke **semithick** und einen Beschriftungsabstand von 0.2em. Die Voreinstellungen lassen sich mit den Befehlen

```

1 \setarrowlength{<länge>}
2 \setarrowline{<liniendicke>}
3 \setarrowlabel{<label abstand>}

```


oder

```
1 \mCsetup{arrowlength=<länge>,
2         arrowline=<liniendicke>,
3         arrowlabel=<label abstand>}
```

ändern.

2.5 Paket-Optionen

myChemistry verfügt über einige Paket-Optionen.

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 4.15](#)) 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**).

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

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

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 4.15](#)) in <position> geändert werden.

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

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

3 Fortgeschrittene Anwendung, Verwendung von TikZ

Das größte Problem bei der Verwendung von **myChemistry** ist in der Regel die korrekte Positionierung der einzelnen Reaktanden und Pfeile. Der [Abschnitt 3.1](#) geht etwas näher darauf ein.

Einige der Befehle ermöglichen als drittes optionales Argument die Angabe weiteren TikZ-Codes. Genauer gesagt können Sie dort im wesentlichen diejenigen TikZ-Keys einsetzen, die Sie in einem `tikzpicture` bei einer `\node` einsetzen können. Wenn die Syntax einer Node also `\node[<tikz>](<platzierung>){<irgendwas>};` ist, dann entspricht `<tikz>` dem entsprechenden Argument von z. B. `\reactant[, ,<tikz>]{}`. 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.

3.1 Die Ausrichtungsfrage

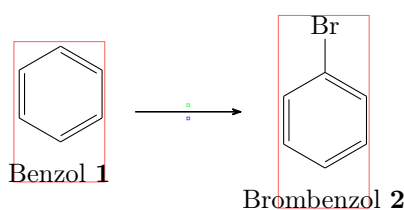
Da Reaktanden, Pfeile und Zweige mittig zu dem Objekt, auf das sie sich beziehen, ausgerichtet werden, erzeugt die Default-Ausrichtung nicht immer schöne Ergebnisse.

Beispiel 15

```

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

```



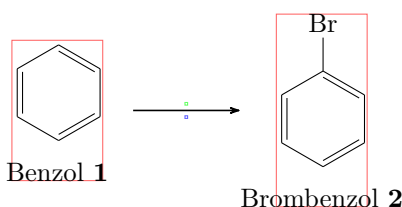
Wie Sie sehen können, sind die beiden Reaktanden aus Sicht der Benzolringe nicht gleich zu dem Pfeil ausgerichtet. Der erste Reaktand scheint nach oben geschoben zu sein. Der Versuch, das mit TikZ-Code wie `xshift` und `yshift` zu korrigieren, versagt.

Beispiel 16

```

1 \makevisible
2 \begin{rxn}
3 \reactant[, , yshift=-1em]{ \chemname{\chemfig{*6(---)}}{Benzol \
  compound{benzol}} }
4 \arrow{}{}
5 \reactant{ \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 17

```

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

```



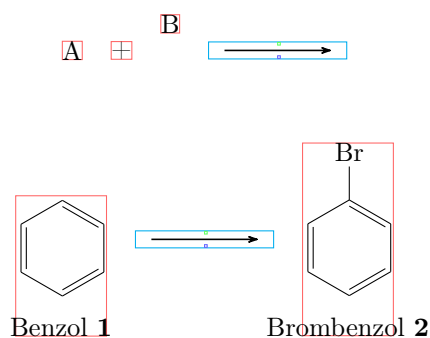
Da es (noch) keine Möglichkeit gibt, die Ausrichtung des Pfeils selbst zu ändern, könnte man ihn stattdessen in einen Zweig stecken.

Beispiel 18

```

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

```



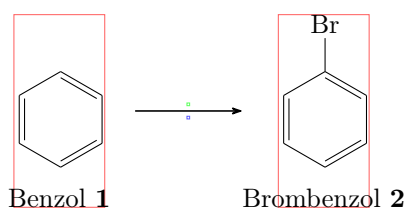
Das ist für das letzte Beispiel aber nicht die beste Lösung, da die exakte Ausrichtung auf diese Weise immer eine ganze Reihe von Versuchen benötigt, bis man das gewünschte Ergebnis erzielt. Es gibt eine andere Lösung: ein unsichtbares Brom am ersten Benzol.

Beispiel 19

```

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

```



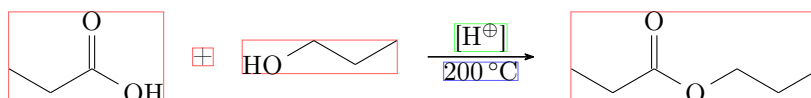
Auch in anderen Fällen ist der unsichtbare Substituent die bessere und einfachere Lösung gegenüber TikZ-Code:

Beispiel 20

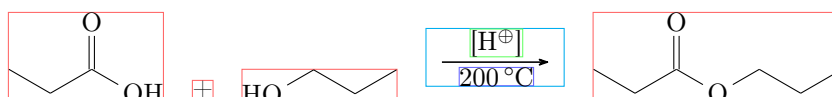
```

1 \makevisible
2 default:
3 \begin{rxn}
4 \reactant{\chemfig{-[: -30] -[:30] (= [2] O) -[: -30] OH}}
5 \chemand
6 \reactant{\chemfig{HO -[:30] -[: -30] -[:30]}}
7 \arrow{[\Hpl]}{\SI{200}{\celsius}}
8 \reactant{\chemfig{-[: -30] -[:30] (= [2] O) -[: -30] O -[:30] -[: -30] -[:30]}}
9 \end{rxn}
10 Hydroxy-Gruppen auf gleicher H"öhe mit Tikz:
11 \begin{rxn}
12 \reactant{\chemfig{-[: -30] -[:30] (= [2] O) -[: -30] OH}}
13 \chemand[, , yshift=-1.2em]
14 \reactant[, , yshift=.12em]{\chemfig{HO -[:30] -[: -30] -[:30]}}
15 \branch[, , yshift=1.08em]{\arrow{[\Hpl]}{\SI{200}{\celsius}}}
16 \reactant{\chemfig{-[: -30] -[:30] (= [2] O) -[: -30] O -[:30] -[: -30] -[:30]}}
17 \end{rxn}
18 Hydroxy-Gruppen auf gleicher H"öhe durch unsichtbaren Substituenten:
19 \begin{rxn}
20 \reactant{\chemfig{-[: -30] -[:30] (= [2] O) -[: -30] OH}}
21 \chemand
22 \reactant{\chemfig{HO -[:30] (= [2, , , draw=none]\phantom{O})
23 \arrow{[\Hpl]}{\SI{200}{\celsius}}
24 \reactant{\chemfig{-[: -30] -[:30] (= [2] O) -[: -30] O -[:30] -[: -30] -[:30]}}
25 \end{rxn}
default:

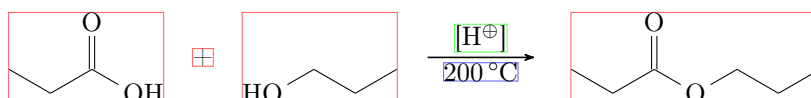
```



Hydroxy-Gruppen auf gleicher Höhe mit Tikz:



Hydroxy-Gruppen auf gleicher Höhe durch unsichtbaren Substituenten:



Ich fürchte aber, 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

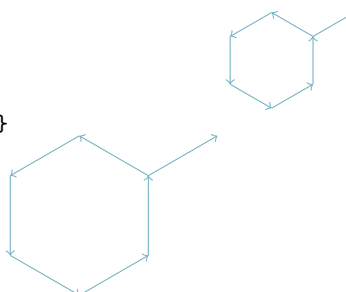
Man könnte natürlich, nur zum Spaß?, das Aussehen von Molekülen mit TikZ ändern.

Beispiel 21

```

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

```



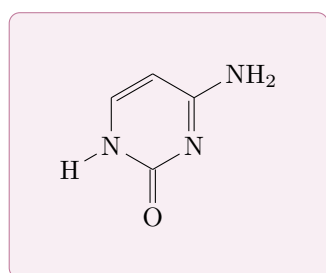
Das Beispiel ist natürlich kein gutes, da mit **ChemFig** dasselbe Ergebnis erzielt werden kann. Vielfache andere Anwendungen sind aber denkbar: beispielsweise könnte man sich einen Stil definieren, in dem Reaktanden angezeigt werden sollen:

Beispiel 22

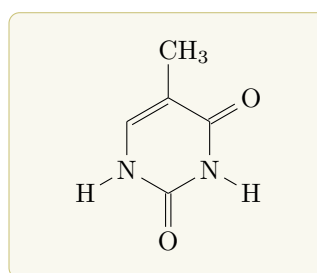
```

1 \colorlet{mCgreen}{green!50!gray}
2 \colorlet{mCblue}{cyan!50!gray}
3 \colorlet{mCred}{magenta!50!gray}
4 \colorlet{mCyellow}{yellow!50!gray}
5 \begin{rxn}
6 \tikzset{reactant/.style={draw=#1,fill=#1!10,inner sep=1em,minimum
   height=10em,minimum width=12em,rounded corners}}
7 \reactant[, cytosin, reactant=mCred]{\chemfig{H-[:30]N*6(- (=O) -N(-NH_2)
   --)}}
8 \anywhere{cytosin.-90,, yshift=-2mm}{Cytosin}
9 \reactant[, thymin, reactant=mCyellow]{\chemfig{H-[:30]N*6(- (=O) -N(-H)
   -(=O) -(-CH_3)=)}}
10 \anywhere{thymin.-90,, yshift=-2mm}{Thymin}
11 \reactant[cytosin.-90, adenin, yshift=-1.5em, reactant=mCblue]{\chemfig
   {[: -36]*5(-N(-H) -*6(-N=-N(-NH_2) --) --N=)}}
12 \anywhere{adenin.-90, Guanin, yshift=-2mm}{Adenin}
13 \reactant[, guanin, reactant=mCgreen]{\chemfig{[: -36]*5(-N(-H) -*6(-N=(-
   NH_2) -N(-H) -(=O) --) --N=)}}
14 \anywhere{guanin.-90,, yshift=-2mm}{Guanin}
15 \end{rxn}

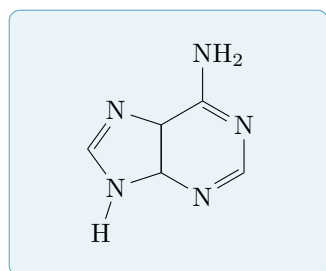
```



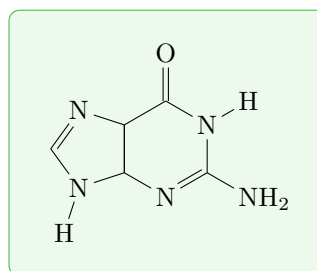
Cytosin



Thymin



Adenin



Guanin

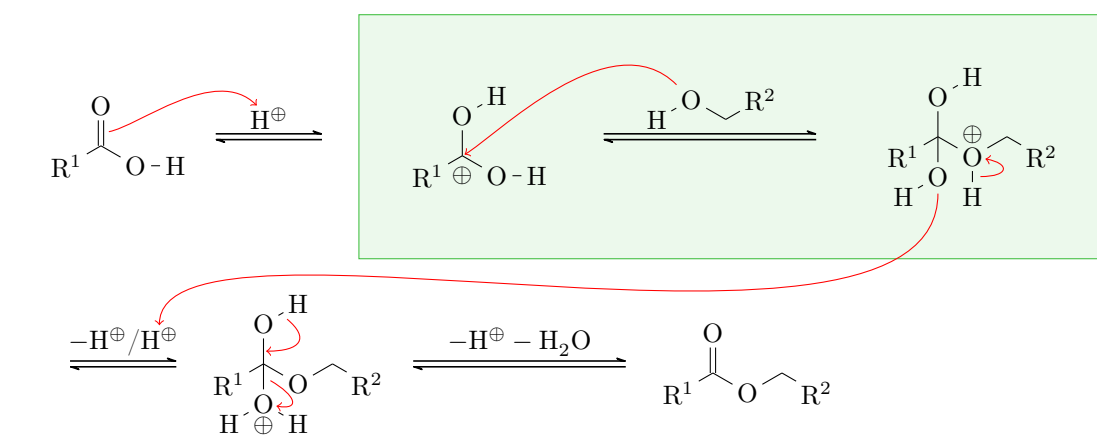
Auf diese Weise könnten Teile eines Schemas hervorgehoben werden:

Beispiel 23

```

1 \begin{rxn}
2 \setatomsep{1.5em}
3 \colorlet{mCgreen}{green!50!gray}
4 \tikzset{emph/.style={draw=mCgreen,fill=mCgreen!10,inner sep=1em}}
5 \reactant[,lineOne]{\chemfig{R^1-[:30](=[@{b1}2]O)-[: -30]O-H}}
6 \arrow[,<=>]{\chemfig{@{Hp11}\Hp1}}{f}
7 \branch[, ,emph]{
8 \reactant{\chemfig{R^1-[:30]@{C1}(-[2]O-[:30]H)(-[6,.5,,,draw=none]\oplus)-[: -30]O-H}}
9 \arrow[,<=>,2]{\chemfig{[:30]H-@{O1}O-[: -60]-R^2}}{f}
10 \reactant{\chemfig{R^1-[:30](-[2]O-[:30]H)(-[6]@{O2}O-[: -150]H)
-[: -30]@{O3}\chemabove{O}{\oplus}(-[@{b2}6]H)-[:30]-[: -30]R^2}}
11 }
12 \anywhere{lineOne.-90,lineTwo,xshift=-3em,yshift=-7em}{
13 \arrow[lineTwo.0,<=>]{\chemfig{@{Hp12}\Hp1}}{f}
14 \reactant{\chemfig{R^1-[:30](-[@{b3}2]O-[@{b4}:30]H)(-[@{b5}6]@{O4}\chembelow{O}{\oplus}(-[: -30]H)-[: -150]H)-[: -30]O-[:30]-[: -30]R^2}}
15 \arrow[,<=>,2]{\ce{- \Hp1 - H2O}}{f}
16 \reactant{\chemfig{R^1-[:30](=[2]O)-[: -30]O-[:30]-[: -30]R^2}}
17 \anywhere{lineTwo.-90}{
18 \elmove{b1}{10:5mm}{Hp11}{135:1cm}\elmove{O1}{135:1.5cm}{C1}{30:5mm}
19 \elmove{O2}{-90:3cm}{Hp12}{90:2cm}\elmove{b2}{0:5mm}{O3}{-10:5mm}
20 \elmove{b4}{-40:5mm}{b3}{0:5mm}\elmove{b5}{-30:5mm}{O4}{-10:5mm}
21 }
22 \end{rxn}

```



4 Alphabetische Befehlsreferenz

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

4.1 anywhere

NEU

Manchmal ist es nützlich, wenn man einen Reaktanden oder irgendetwas anderes außerhalb der Chain platzieren kann.

```
1 \anywhere{<pos>,<name>,<tikz>}{<irgendwas>}
```

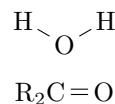
Für diesen Fall gibt es den Befehl `\anywhere`. Er wird über `<pos>` auf ähnliche Weise wie `\branch` platziert:

```
1 <anker>.<winkel> % nicht auf der Kette
2 on chain=going <schluessel> % auf der Kette
3 <schluessel>=of <anker> % nicht auf der Kette
```

Bitte beachten Sie, dass `<pos>` nicht leer bleiben kann.

Beispiel 24

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



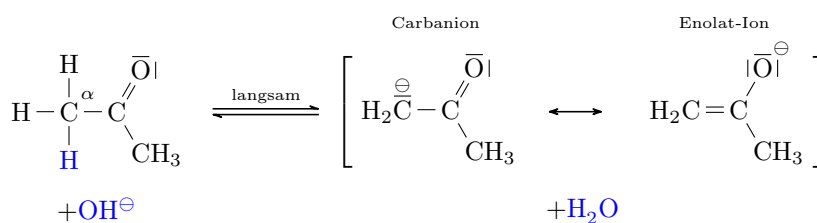
Der Befehl kann z. B. gut zum Beschriften von Reaktionspartnern eingesetzt werden.

Beispiel 25

```

1 \begin{rxn}
2 \reactant[,keton]{\chemfig{H-\chemabove{C}{\hspace*{5mm}\scriptstyle\alpha}(-[2]H)(-[6,,,2]{|\textcolor{blue}H})-C(=[:60]\lewis{02,0})-[: -60]C|H_3}}
3 \anywhere{below=of keton}{\color{blue}\Hyd}
4 \arrow[,<=>]{\tiny langsam}{}
5 \mesomeric[,mesomer]{
6 \reactant[,carbanion]{\chemfig{H_2|\chemabove[3pt]{\lewis{2,C}}{\scriptstyle\ominus}-C(=[:60]\lewis{02,0})-[: -60]C|H_3}}
7 \marrow
8 \reactant[,enolat]{\chemfig{H_2C=C(-[:60]\chemabove{\lewis{024,0}}{\hspace*{5mm}\scriptstyle\ominus})-[: -60]C|H_3}}
9 }
10 \anywhere{above=of enolat}{\tiny Enolat-Ion}
11 \anywhere{above=of carbanion}{\tiny Carbanion}
12 \anywhere{below=of mesomer}{\color{blue}\ce{H2O}}
13 \end{rxn}

```



Viele weitere Beispiele für Einsatzmöglichkeiten finden Sie in der Datei `examples.tex` bzw. `examples.pdf`.

4.2 arrow

Reaktionspfeile werden mit `\arrow` erstellt.

```

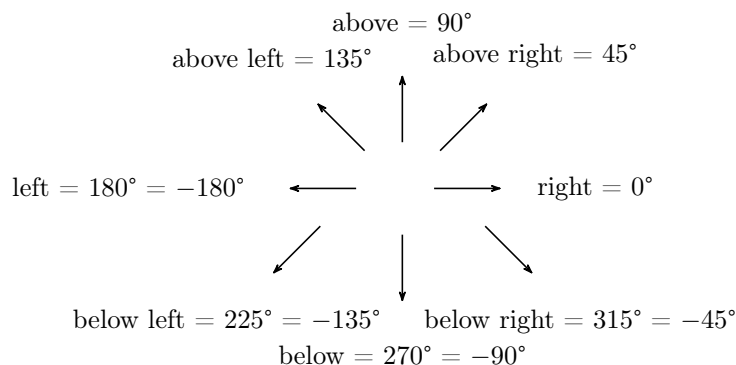
1 \arrow[<pos>,<typ>,<längenfaktor>,<name>,<both>,<tikz>]{<
  oben>}{<unten>}

```

4.2.1 Optionen

Mit mehreren Optionen können die Reaktionspfeile angepasst werden. Die Optionen müssen an entsprechender Stelle, durch Kommata separiert, angegeben werden.

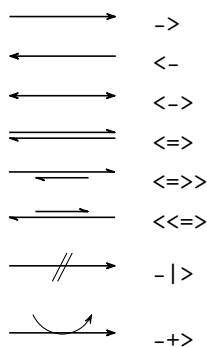
1. `<pos>` – mögliche Einstellungen sind:



NEU Außerdem kann jeder beliebige Winkel des Intervalls $[-360^\circ; 360^\circ]$ eingesetzt werden.

Zusätzlich kann der Winkel in Bezug auf ein durch `<name>` benanntes Objekt angegeben werden: `<name>.<winkel>` bedeutet im Winkel von `<winkel>` neben `<name>`. Im Dokument `examples.tex` bzw. `examples.pdf` können Sie einige Beispiele dafür finden.

2. `<typ>` – mögliche Einstellungen sind:



3. `<längenfaktor>` – mit dem Faktoren, den Sie hier angeben, wird die Pfeillänge (4em bei Faktor = 1.0, Standard) multipliziert.
4. `<name>` – hier können Sie dem Pfeil einen Anker geben, auf den z. B. mit einem Branch referenziert werden kann.
5. `both` – durch diese Option haben die beiden Nodes, in die die Label geschrieben werden, die gleichen Maße.
6. `<tikz>` – mit dieser Option können die Pfeile mit `TikZ`-Keys angepasst werden. Nicht alle `TikZ`-Keys zeigen Auswirkungen. So können Pfeile z. B. mit `shift=<coord>` *nicht* verschoben werden.

Beispiel 26

```

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

```



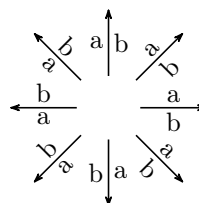
Beachten Sie, dass die Label <oben> und <unten> mit der Richtung gedreht werden. Bei einem Winkel von 180° liegt <oben> tatsächlich unterhalb des Pfeils.

Beispiel 27

```

1 \begin{rxn}
2 \setarrowlength{2.5em}
3 \dummy[a]
4 \arrow{a}{b}
5 \arrow[a.45]{a}{b}
6 \arrow[a.90]{a}{b}
7 \arrow[a.135]{a}{b}
8 \arrow[a.180]{a}{b}
9 \arrow[a.225]{a}{b}
10 \arrow[a.270]{a}{b}
11 \arrow[a.315]{a}{b}
12 \end{rxn}

```



NEU Beim Pfeiltypen \rightarrow gibt es ein, zwei Dinge zu beachten: lässt man die Beschriftung leer, dann wird lediglich der nach rechts zeigende Pfeil angezeigt. Das erste Argument beschriftet den hinzukommenden und das zweite Argument den wegzeigenden Pfeil. Verwendet man nur eines dieser Argumente, wird auch nur der jeweilige Teil gezeichnet.

Beispiel 28

```

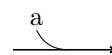
1 entspricht \verb"->":
2 \begin{rxn}
3 \arrow[,-->]{}{}
4 \end{rxn}
5 hinzu:
6 \begin{rxn}
7 \arrow[,-->]{a}{}
8 \end{rxn}
9 weg:
10 \begin{rxn}
11 \arrow[,-->]{}{b}
12 \end{rxn}
13 hinzu und weg:
14 \begin{rxn}
15 \arrow[,-->]{a}{b}
16 \end{rxn}
17 Leerzeichen sind \emph{kein}
   leeres Argument:
18 \begin{rxn}
19 \arrow[,-->]{}{}
20 \end{rxn}

```

entspricht ->:



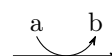
hinzu:



weg:



hinzu und weg:

Leerzeichen sind *kein* leeres Argument:

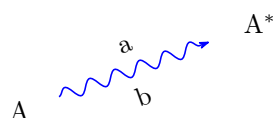
NEU Durch Verwendung von *TikZ*-Keys kann das Aussehen der Pfeile verändert werden:

Beispiel 29

```

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

```

**4.2.2 Ausrichtung**

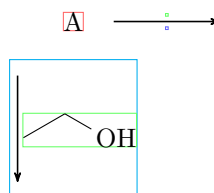
Liegt der Pfeil in einem Branch (siehe [Abschnitt 4.3](#)), dann wird die Ausrichtung des Branch bestimmt durch die Größe der Nodes, mit denen die Pfeilbeschriftung platziert wird. Hat der Pfeil nun nur eine oder zwei unterschiedlich große Beschriftungen, dann ist die Ausrichtung falsch.

Beispiel 30

```

1 \makevisible
2 \begin{rxn}
3 \reactant[,a]{A}
4 \arrow{}{}
5 \branch[below=of a]{
6 \arrow[-90]{\chemfig
7 {-[::30]-[::-60]OH}}{}
8 \end{rxn}
9 \makeinvisible

```



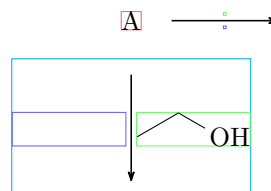
Durch die Verwendung der Option `both` bekommen die Nodes beider Pfeilbeschriftungen die gleichen Maße, wodurch die Ausrichtung korrigiert werden kann.

Beispiel 31

```

1 \makevisible
2 \begin{rxn}
3 \reactant[,a]{A}
4 \arrow{}{}
5 \branch[below=of a]{
6 \arrow[-90,,,both]{\chemfig
7 {-[::30]-[::-60]OH}}{}
8 \end{rxn}
9 \makeinvisible

```



Mehr zu dem Problem der Ausrichtung lesen Sie in [Abschnitt 4.3.2](#).

4.2.3 Aussehen

Mit den Befehlen `\setarrowlength` ([Abschnitt 4.17](#)) und `\setarrowline` ([Abschnitt 4.18](#)) lässt sich das prinzipielle Erscheinungsbild der Pfeile verändern.

4.3 branch

Der Befehl `\branch` wird verwendet, um eine Verzweigung der Reaktion zu realisieren.

```

1 \branch[<pos>,<name>,<tikz>]{<formel(n)>}

```

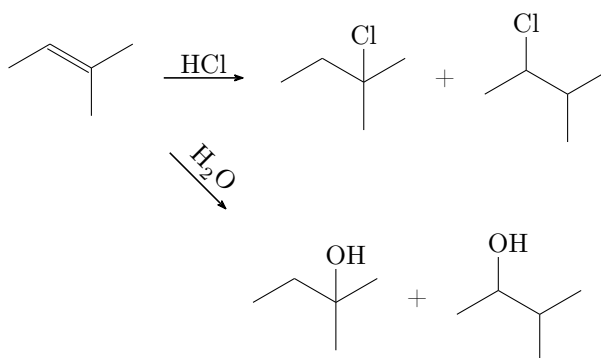
Für den `\branch` wird die Ausrichtung und der Anker wichtig. Sehen wir uns ein Beispiel an.

Beispiel 32

```

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

```



In diesem Beispiel hat der erste Reaktand den Anker `start` bekommen (Zeile 2, siehe auch [Abschnitt 4.13](#)).

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

`\branch` bezieht sich nun in seiner Ausrichtung darauf (Zeile 7):

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

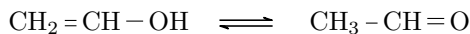
Gibt man die Ausrichtung nicht in Bezug auf einen Anker an, bezieht sie sich immer auf den letzten `reactant` oder `\arrow`. Lässt man das optionale Argument leer, dann platziert sich der Branch automatisch rechts.

Beispiel 33

```

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

```



4.3.1 Positionierung

Ein Ast kann auf verschiedene Arten positioniert werden. Je nachdem ist er dann Teil der Kette oder ein echter Ast.

1	<code><winkel></code>	% auf der Kette
2	<code><schluessel></code>	% auf der Kette
3	<code><anker>.<winkel></code>	% nicht auf der Kette
4	<code>on chain=going <schluessel></code>	% auf der Kette
5	<code><schluessel>=of <anker></code>	% nicht auf der Kette

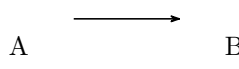
Die Voreinstellung entspricht `\branch[0]{}`. In unterschiedlichen Situationen kann eine unterschiedliche Verwendung angebracht sein. Wollen Sie zum Beispiel den Ast verwenden, um einen Pfeil zu verschieben, kann er durchaus Teil der Kette bleiben.

Beispiel 34

```

1 \begin{rxn}
2 \reactant{A}
3 \branch[, , yshift=1em]{\arrow
  {}{}}
4 \reactant[, , yshift=-1em]{B}
5 \end{rxn}

```



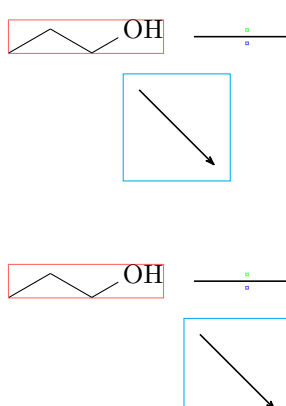
Beim Anfahren eines echten Astes ist es nicht immer egal, ob man Variante 3 (`<anker>.<winkel>`) oder 5 (`<schluessel>=of <anker>`) wählt:

Beispiel 35

```

1 \makevisible
2 \begin{rxn}
3 \reactant[, a]{\chemfig
  {[:30]--[::-60]-OH}}
4 \arrow{}{}
5 \branch[a. -45]{\arrow[-45]{}{}}
6 \end{rxn}
7 \begin{rxn}
8 \reactant[, a]{\chemfig
  {[:30]--[::-60]-OH}}
9 \arrow{}{}
10 \branch[below right=of a]{\
  arrow[-45]{}{}}
11 \end{rxn}

```



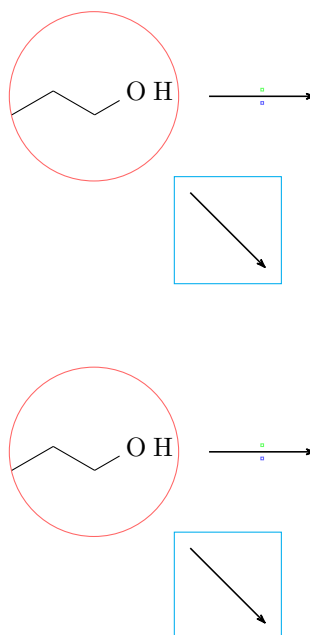
Im ersten Fall wird der Ast 45° unterhalb der Horizontalen ausgehend vom Zentrum des Reaktands positioniert. Da die Form des Reaktands rechteckig ist und nicht etwa quadratisch oder rund, stimmen die 45° nicht mit „unten rechts“ überein.

Beispiel 36

```

1 \makevisible
2 \begin{rxn}
3 \reactant[,a,circle]{\chemfig
4 {[:30]--[::-60]-OH}}
5 \arrow{}{}
6 \branch[a,-45]{\arrow[-45]{}{}}
7 \end{rxn}
8 \begin{rxn}
9 \reactant[,a,circle]{\chemfig
10 {[:30]--[::-60]-OH}}
11 \arrow{}{}
12 \branch[below right=of a]{\
13 \arrow[-45]{}{}}
14 \end{rxn}

```



4.3.2 Ausrichtungsprobleme

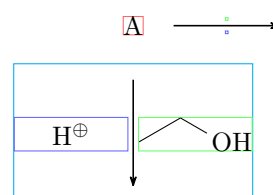
Wenn ein Pfeil zwei verschieden große Label hat und in einem Branch liegt, wird der Branch nicht mehr richtig ausgerichtet. Der `\arrow`-Key `both` ist nicht wirklich eine Lösung, weil die kleinere Beschriftung dann nicht mehr am Pfeil liegt, sondern wegrutscht.

Beispiel 37

```

1 \makevisible
2 \begin{rxn}
3 \reactant[,a]{A}
4 \arrow{}{}
5 \branch[below=of a]{
6 \arrow[-90,,,both]{\chemfig
7 {-[:30]-[::-60]OH}}{\Hopl}
8 }
9 \end{rxn}
10 \makeinvisible

```



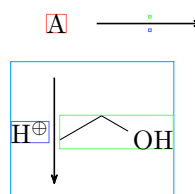
In diesem Fall können Sie den Branch mit den `TikZ`-Keys `xshift` und `yshift` verschieben.

Beispiel 38

```

1 \makevisible
2 \begin{rxn}
3 \reactant[,a]{A}
4 \arrow{}{}
5 \branch[below=of a,,xshift=1.35
  em]{
6 \arrow[-90]{\chemfig
  {-[::30]-[::-60]OH}}{\Hpl}
7 }
8 \end{rxn}
9 \makeinvisible

```

**4.4 chemand**

Der Befehl

```
1 \chemand [<pos>,<name>,<tikz>]
```

erzeugt und platziert ein + auf die gleiche Weise, wie `\reactant` beliebigen Text platziert. Letztlich ist er nur ein Shortcut für `\reactant`:

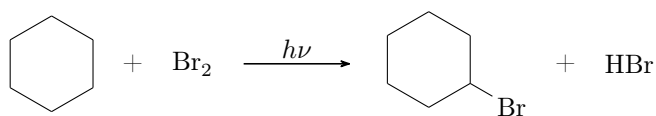
```
1 \newcommand*{\chemand}[1][\reactant[#1]{\chemsign[Opt]
  }{+}}}
```

Beispiel 39

```

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

```



Die optionalen Argumente von `\chemand` und `\reactant` sind also die gleichen, siehe [Abschnitt 4.13](#) für eine Beschreibung.

4.5 dummy**NEU**

Mit `\dummy` zeichnet man eine leere Node. Bis Version 1.3 mussten die Pfeile, die mit `\arrow` erzeugt werden, einer Node nachfolgen. `\arrow` ruft intern `\tikzchainprevious`

auf. Ist *vor* einem Pfeil noch *keine* Node auf die Chain geschrieben worden, erzeugt das eine Fehlermeldung. Ähnliches galt für `\branch`. Durch Setzen des `\dummy` konnte ein Schema dennoch mit einem Pfeil beginnen.

```

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

```

Das ist nun *nicht mehr nötig*. Dennoch kann unter Umständen eine leere Node am Anfang als Anker zur Ausrichtung anderer Objekte ganz nützlich sein, weshalb der Befehl weiter zur Verfügung steht.

4.6 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>`.

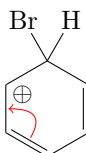
Wie `\chemmove` funktioniert, können Sie im Manual zu `ChemFig` nachlesen.

Beispiel 40

```

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}

```



Sie sollten `\elmove` *nur innerhalb* von `\anywhere`, `\reactant` oder `\transition` verwenden. Ansonsten könnten sich einige Ausrichtungsfehler ergeben.

4.7 makeinvisible

Dieser Befehl hebt die Änderungen von `\makevisible` (siehe [Abschnitt 4.8](#)) auf und stellt das normale Verhalten von `myChemistry` wieder her. `\makeinvisible` wirkt sich nur auf nachfolgende Reaktanden aus.

4.8 makevisible

Mit `\makevisible` können Sie die Nodes, innerhalb derer sich die Reaktanden befinden, farbig hervorheben. Das kann z. B. bei der Positionierung und Feinjustierung von Branches ganz nützlich sein. Ein Beispiel dafür sehen Sie in [Abschnitt 4.2](#). Je nach Art der Node ist die Markierung eine andere:

`\reactant{}`, `\arrow{above}{}`, `\arrow{below}{}` und `\branch{}`. Siehe auch [Abschnitt 4.7](#).

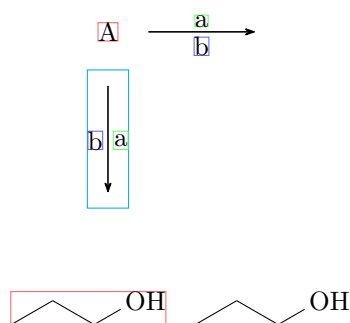
`\makevisible` wirkt sich nur auf nachfolgende Reaktanden aus und ist auch nur innerhalb der Gruppe wirksam, in der er gesetzt wurde.

Beispiel 41

```

1 \begin{rxn}
2 \makevisible
3 \reactant[,a]{A}
4 \arrow{a}{b}
5 \branch[below=of a]{
6   \arrow[-90,,,both]{a}{b}
7 }
8 \end{rxn}
9 \begin{rxn}
10 {\makevisible
11 \reactant{\chemfig
12   {[:30]--[::-60]-OH}}
13 }
14 \reactant{\chemfig
15   {[:30]--[::-60]-OH}}
16 \end{rxn}

```



4.9 marrow

Der Befehl `\marrow` zeichnet einen Mesomeriepfeil.

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

Er ist eine Abkürzung für `\arrow[<pos>,<->,.5]{}`.

4.10 mCsetup

Der Befehl

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

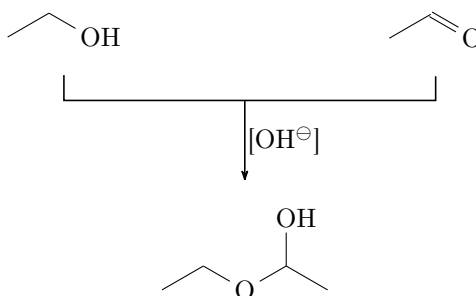
kann verwendet werden, um alle Voreinstellungen zu verändern. Für jeden der `\set<command>`-Befehle von *myChemistry* mit Ausnahme von `\setbondshape` gibt es einen Key `<command>=<value>`. Zusätzlich gibt es den Key `align=<value>`, mit dem das Ausrichtungsverhalten von `rxn` und `rxnscheme` gleichzeitig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

Beispiel 43

```

1 \begin{rxn}
2 \branch[,oben]{
3 \reactant[,start_a]{\chemfig{-[:30]-[: -30]OH}}
4 \reactant[,start_b,xshift=9em]{\chemfig{-[:30]=[: -30]O}}
5 }
6 \branch[below=of oben,ziel,yshift=-5em]{
7 \reactant{\chemfig{-[:30]-[: -30]O-[:30](-[2]OH)-[: -30]}}
8 }
9 \merge[\ce{\Hyd}]{ziel}{start_a}{start_b}
10 \end{rxn}

```



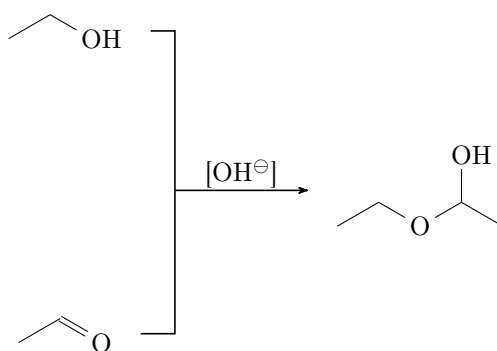
Die Standard-`<pos>` ist `below`, andere mögliche Werte sind `right`, `left` und `above`. Mit `<länge>` kann die Länge des Pfeils *ab der Zusammenführung* angegeben werden. Standard sind 3 em. Die Standardlänge kann über `\setmergelength` oder `\mCsetup` auch geändert werden.

Beispiel 44

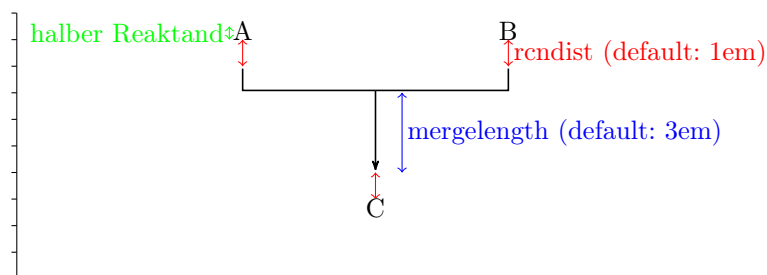
```

1 \begin{rxn}
2 \reactant[,start_a]{\chemfig{-[:30]-[: -30]OH}}
3 \branch[below=of start_a,start_b,yshift=-9em]{
4 \reactant{\chemfig{-[:30]=[: -30]O}}
5 }
6 \branch[right=of start_a,ziel,xshift=7em,yshift=-6em]{
7 \reactant{\chemfig{-[:30]-[: -30]O-[:30](-[2]OH)-[: -30]}}
8 }
9 \merge[\ce{\Hyd}],right,5em]{ziel}{start_a}{start_b}
10 \end{rxn}

```



Da man die Reaktanden vorher platzieren muss, ist es bestimmt hilfreich, die benötigten Abstände etwas im Auge zu behalten. Drei Werte bestimmen den benötigten Platz. Zum einen die **Ausdehnung** der beteiligten Reaktanden, der Abstand der „reaction-nodes“ **rcndist** (siehe [Abschnitt 4.23](#)) voneinander und die Länge des **\merge**-Pfeils **mergelength** (siehe auch [Abschnitt 4.22](#)).



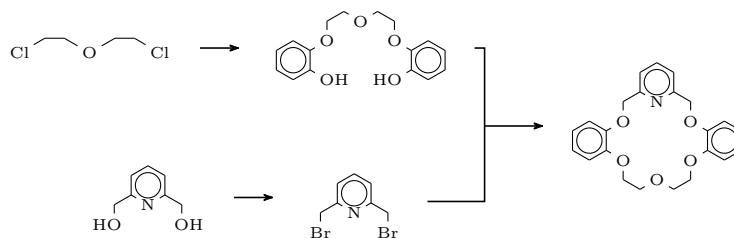
Noch ein letztes Beispiel, bei dem zwei Stränge vereint werden.

Beispiel 45

```

1 \begin{rxn}
2 \setatomsep{1em}\tiny
3 % Strang 1
4 \reactant[,oben]{ \chemfig{Cl-[:30,1.5]--[: -30,1.5]O
  -[:30,1.5]--[: -30,1.5]Cl}{ } }
5 \arrow[,,.5]{ }
6 \reactant[,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 \reactant{ \chemfig{**6((--[6,,2]HO)-N(--[6]OH)----)} }
10 \arrow[,,.5]{ }
11 \reactant{ \chemfig{**6((--[6]Br)-N(--[6]Br)----)} }
12 }
13 % Ziel
14 \branch[right=of start_oben,ziel,xshift=5em,yshift=-4em]{
15 \reactant[,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[,right]{ziel}{start_oben}{start_unten}
19 \end{rxn}

```



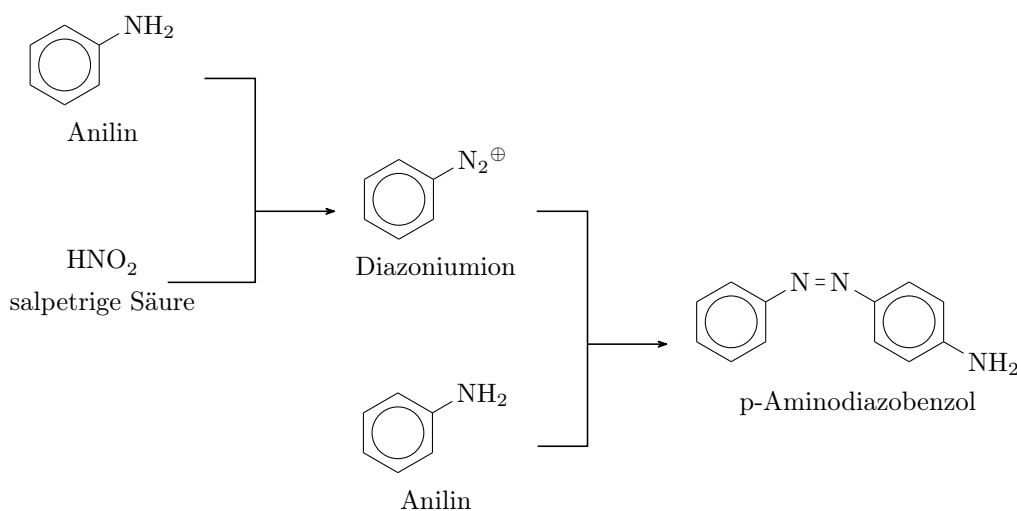
Die Verwendung von `\merge` erfordert unter Umständen einige Spielerei mit Branches, `xshift` und `yshift`, bis man das gewünschte Ergebnis erhält.

Beispiel 46

```

1 \begin{rxn}
2 \setatomsep{1.5em}
3 \reactant[,start_aa]{ \chemname{\chemfig{**6(---(-NH_2)---)}}{Anilin}
4 }
5 \reactant[below,start_ab,yshift=-3em]{ \chemname{\ce{HNO2}}{salpetrige
6 S"auere} }
7 }% = start_ba
8 \branch[right=of start_aa,ziel_a,xshift=6em,yshift=-5em]{
9 \reactant{ \chemname{\chemfig{**6(---(-N|_2\op)---)}}{Diazoniumion}
10 }
11 }
12 \branch[below=of ziel_a,start_bb,yshift=-3em]{
13 \reactant{ \chemname{\chemfig{**6(---(-NH_2)---)}}{Anilin} }
14 }
15 \branch[right=of ziel_a,ziel_b,xshift=6em,yshift=-5em]{
16 \reactant{ \chemname{\chemfig{N(-[: -150]**6(-----))=N
17 -[: -30]**6(---(-NH_2)---)}}{p-Aminodiazobenzol} }
18 }
19 \merge[,right]{ziel_a}{start_aa}{start_ab}
20 \merge[,right]{ziel_b}{ziel_a}{start_bb}
21 \end{rxn}

```

**4.12 mesomeric**

Der `\mesomeric`-Befehl funktioniert wie ähnlich wie `\branch` (Abschnitt 4.3). Sein Zweck ist es, eckige Klammern zu setzen.

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

In `<formel(n)>` werden die mesomeren Grenzstrukturen geschrieben. Mit `\marrow` (Abschnitt 4.9) werden die Mesomeriepfeile gesetzt. Man kann `\mesomeric` falls nötig

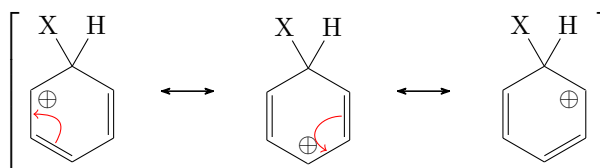
mit einem Anker (<name>) versehen (Abschnitt 4.3). Die <pos> funktioniert wie bei `\reactant[<pos>]{}`.

Beispiel 47

```

1 \begin{rxn}
2 \mesomeric{
3   \reactant{
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   \reactant{
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  \reactant{
14    \chemfig{*6(-=-(-[: -150,.4,, ,white]\oplus)-(-[:120]X)(-[:60]H)-=)}
15  }
16 }
17 \end{rxn}

```



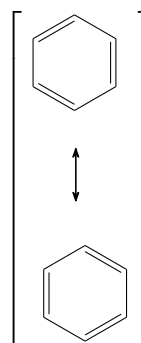
Oder auch von oben nach unten:

Beispiel 48

```

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

```



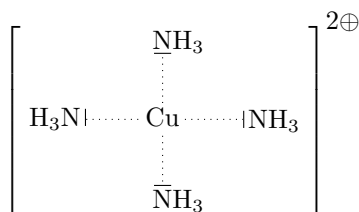
Vielleicht auch einen Komplex?

Beispiel 49

```

1 \begin{rxn}
2 \setatomsep{3em}
3 \mesomeric[,a]{
4 \reactant{ \chemfig{H_3\lewis{0,N}-[,1.35,,,dotted]{Cu}(-[2,,,,
dotted]\lewis{6,N}H_3)(-[6,,,,dotted]\lewis{2,N}H_3)-[,1.2,,,dotted]\
lewis{4,N}H_3} }
5 }
6 \anywhere{above right=of a,,yshift=-1em}{\$2\oplus\$}
7 \end{rxn}

```

**4.13 reactant**

Der Befehl `\reactant1` ist so etwas wie der Basisbefehl.

```
1 \reactant [<pos>,<name>,<tikz>]{<formel(n)>}
```

In diesen Befehl werden die Formeln (`<formel>`) geschrieben und können, falls nötig, mit einem Anker (`<name>`) versehen werden. `<pos>` kann die 8 Schlüssel-Werte

- a) right,
- b) above right,
- c) above,
- d) above left,
- e) left,
- f) below left,
- g) below,
- h) below right

NEU

annehmen, einen Winkel des Intervalls $[-360^\circ; 360^\circ]$ oder eine Anker-Winkel-Kombination `<anker>.<winkel>` sein. Die Voreinstellung entspricht `\reactant[0]{}`.

¹In älteren Versionen hieß der Befehl `\reactand`. Dieser Befehl ist noch immer verfügbar.

Beispiel 50

1 untereinander:	
2 <code>\begin{rxn}</code>	untereinander:
3 <code>\reactant{\ce{Br2}}</code>	Br_2
4 <code>\reactant[-90]{\ce{Cl2}}</code>	Cl_2
5 <code>\end{rxn}</code>	
6	
7 Beispiel mit mehreren Reaktanden	Beispiel mit mehreren Reaktanden:
8 <code>\begin{rxn}</code>	
9 <code>\reactant{\ce{Br2}}</code>	Br_2
10 <code>\reactant[-90]{\ce{I2}}</code>	I_2 Cl_2
11 <code>\reactant{\ce{Cl2}}</code>	
12 <code>\end{rxn}</code>	
13	Reaktion von oben nach unten:
14 Reaktion von oben nach unten:	$\text{Br}-\text{Br}$
15 <code>\begin{rxn}</code>	
16 <code>\reactant{\ce{Br-Br}}</code>	$\downarrow h\nu$
17 <code>\arrow[-90, , .5]{\\$h\nu\\$}</code>	
18 <code>\reactant[-90]{\ce{2 ~\lewis</code>	$2 \text{ Br}\cdot$
19 <code>\end{rxn}</code>	

Viele weitere Beispiele finden Sie in der Datei `examples.tex` bzw. `examples.pdf`.

4.14 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 bei `rxnscheme`.

```

1 \begin{rxn}[<ausrichtung>,<skalierung>]
2   ...
3 \end{rxn}

```

4.14.1 Optionen

`rxn` hat zwei Optionen, die in der angegebenen Reihenfolge, durch Komma separiert, einzusetzen sind:

1. `<ausrichtung>` das Ausrichtungsverhalten der `rxn`-Umgebung, Default: center
2. `<skalierung>` Skalierung der `rxn`-Umgebung, Default: 1.0

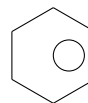
Wenn Sie die `<skalierung>`-Option verwenden, können Sie auf seltsame Effekte bei `ChemFig`-Formeln stoßen.

Beispiel 51

```

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

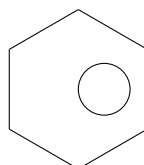
```



Die Skalierung hat keinen Einfluss auf die Größe der **ChemFig**-Formeln, skaliert aber den Aromatizitäts-Ring von Benzol und ähnlichen Molekülen. Die Ursache liegt an einer Unannehmlichkeit von **ChemFig**.

Beispiel 52

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



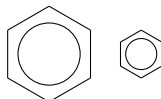
Sie können das lösen, indem Sie entweder mit dem *ersten* optionalen Argument von **\chemfig** den Ring wieder vergrößern, oder mit dem *zweiten* Argument den Rest auch verkleinern.

Beispiel 53

```

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

```



Beispiele zur Ausrichtung:

Beispiel 54

```

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

```

center \longrightarrow zentriert

right \longrightarrow rechts

left \longrightarrow links

4.15 rxnscheme (Umgebung)

Die Umgebung `rxnscheme` ist eine Gleitumgebung für Reaktionsschemata.

```

1 \begin{rxnscheme}[<label>,<platzierung>,<ausrichtung>,<
   skalierung>,<titel>]{<caption>}
2 ...
3 \end{rxnscheme}

```

4.15.1 Optionen

`rxnscheme` hat fünf Optionen, die in der angegebenen Reihenfolge, durch Komma separiert, einzusetzen sind:

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

```

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

```

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

2. `<platzierung>` Hier können sie Platzierungsangaben wie `htp` angeben. Default ist `H` (genau hier).

3. <ausrichtung> Mit dieser Option kann man auswählen, ob das Schema links, rechts oder mittig ausgerichtet wird.
4. <skalierung> Mit dieser Option kann das Reaktionsschema skaliert werden. Beachten Sie, dass sie sich nicht auf die Schriftgröße und die Größe der **ChemFig**-Formeln auswirkt. Sie können allerdings seltsame Auswirkungen auf **ChemFig**-Formeln beobachten, wenn Sie diese Option verwenden. Lesen Sie [Abschnitt 4.14.1](#) für weitere Informationen.

```

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

```

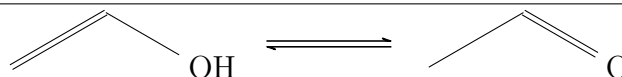
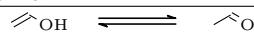
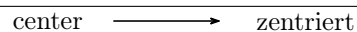
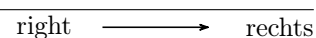
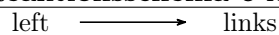
5. <titel> Mit dieser Option lässt sich der Name des konkreten Schemas von „Reaktionsschema“ (oder „Reaction scheme“) in <titel> ändern.

Beispiel 55

```

1  \begin{rxnscheme}[,H,,2]{Gro\ss es Schema}
2  \large\setatomsep{3.5em}
3  \reactant{ \chemfig{=[::30]-[::-60]OH} }
4  \arrow[,<=>]{}{}
5  \reactant{ \chemfig{-[::30]=[::-60]O} }
6  \end{rxnscheme}
7  \begin{rxnscheme}[,H,,.5]{Kleines Schema}
8  \tiny\setatomsep{1em}
9  \reactant{ \chemfig{=[::30]-[::-60]OH} }
10 \arrow[,<=>]{}{}
11 \reactant{ \chemfig{-[::30]=[::-60]O} }
12 \end{rxnscheme}
13 \begin{rxnscheme}[,H]{center}
14 \reactant{center}\arrow{}{}\reactant{zentriert}
15 \end{rxnscheme}
16 \begin{rxnscheme}[,H,right]{right}
17 \reactant{right}\arrow{}{}\reactant{rechts}
18 \end{rxnscheme}
19 \begin{rxnscheme}[,H,left]{left}
20 \reactant{left}\arrow{}{}\reactant{links}
21 \end{rxnscheme}

```

Reaktionsschema 2 Großes Schema**Reaktionsschema 3** Kleines Schema**Reaktionsschema 4** center**Reaktionsschema 5** right**Reaktionsschema 6** left**4.15.2 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 56

```

1 \begin{rxnscheme}[,H]{ruled}
2 \reactant{Standard-Stil}
3 \end{rxnscheme}
4 \floatstyle{boxed}
5 \restylefloat{rxnfloat}
6 \begin{rxnscheme}[,H]{boxed}
7 \reactant{mit Rahmen}
8 \end{rxnscheme}
9 \floatstyle{plain}
10 \restylefloat{rxnfloat}
11 \begin{rxnscheme}[,H]{plain}
12 \reactant{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 `hpt` 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 der entsprechenden Option ändern.

```

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

```

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

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

machen. Voreinstellung ist „Reaktionsschema“ 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 57

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

Reaktionsschemata

1	Keto-Enol-Tautomerie	14
2	Großes Schema	47
3	Kleines Schema	47
4	center	47
5	right	47
6	left	47
7	ruled	48
8	boxed	48
9	plain	48

4.16 setarrowlabel

NEU

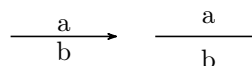
Die Beschriftungen eines Pfeils haben in der Voreinstellung einen Abstand von 0.2em vom Pfeil. Mit

```
1 \setarrowlabel{<abstand>}
```

kann der Abstand in `<abstand>` geändert werden.

Beispiel 58

```
1 \begin{rxn}
2 \arrow{a}{b}
3 \setarrowlabel{.5em}
4 \arrow{a}{b}
5 \end{rxn}
```



4.17 setarrowlength

Reaktionspfeile haben als Standardwert die Länge 4 em. Die Voreinstellung lässt sich mit

```
1 \setarrowlength{<länge>}
```

auf <länge> ändern. Beachten Sie, dass Sie eine Längeneinheit verwenden müssen. Lassen Sie das Argument leer, wird die Voreinstellung wiederhergestellt.

4.18 setarrowline

NEU Mit dem Befehl

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

lässt sich die Dicke der Pfeillinien einstellen. Mögliche Werte sind

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

Die Werte `very thick` und `ultra thick` sind nicht zu empfehlen.

Die Einstellung gilt auch für `\merge`.

4.19 setatomsiz

Mit

```
1 \setatomsiz{<größe>}
```

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

4.20 setbondlength

Mit

```
1 \setbondlength{<länge>}
```

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

4.21 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.

4.22 setmergelength

NEU

Mit

```
1 \setmergelength{<länge>}
```

kann die Länge des Pfeils beim `\merge`-Befehl geändert werden. Genauer ist damit die Länge ab Zusammenführung bis zur Pfeilspitze gemeint (siehe [Abschnitt 4.11](#)). Lässt man das Argument leer, wird die Voreinstellung (3 em) wiederhergestellt.

4.23 setrcndist

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 59

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

A \longrightarrow

A \longrightarrow

4.24 setrxnalign/setschemealign

Mit den Befehlen

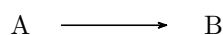
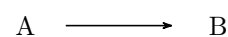
```
1 \setrxnalign{<alignment>}
2 \setschemealign{<alignment>}
```

lässt sich das Default-Ausrichtungsverhalten (siehe [Abschnitt 4.14.1](#) & [Abschnitt 4.15.1](#)) der Umgebungen festlegen. Es gibt die Einstellungsmöglichkeiten `left`, `center` oder `right`.

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

Beispiel 60

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



4.25 setschemename

Siehe [Abschnitt 4.15.2](#).

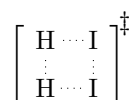
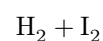
4.26 transition

`\transition` funktioniert genau wie `\reactant` (siehe [Abschnitt 4.13](#)).

```
1 \transition [<pos>, <name>, <tikz>]{<formel>}
```

Beispiel 61

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



5 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.

Viel Spaß mit myChemistry!

Clemens Niederberger, Berlin, 27. April 2011

6 Dank

Ich schulde Dank für Bugreports und Vorschläge:

F. Chervet, Ferghun, V. Garibal und C. Tellechea (der mir ein paar essentielle Dinge klar machte).

Stichwortverzeichnis

A

anywhere	24 f.
arrow	8, 25 - 29
Ausrichtung	28 f.
Aussehen	29
Optionen	25 - 28
both	26, 29
längenfaktor	26
name	26
pos	25
tikz	26, 28
typ	26 f.
Ausrichtung	18 - 21

B

Befehle

anywhere	24
arrow	25
branch	29
chemand	33
dummy	33
elmove	34
makeinvisible	34
makevisible	35
marrow	35
mCsetup	35
merge	36
mesomeric	40
reactant	42
rxn	<i>siehe rxn</i>
rxnscheme	<i>siehe rxnscheme</i>
setarrowlabel	49
setarrowlength	50
setarrowline	50
setatomsiz e	50
setbondlength	50
setbondshape	51
setmergelength	51
setrcndist	51
setrxnalign	51

setschemealign	51
setschemename	52
transition	52
branch	11, 29 - 33
Ausrichtung	32 f.
Positionierung	31 f.

C

chemand	33
chemcompounds	6
chemexec	6
ChemFig	6

D

dummy	33 f.
-------------	-------

E

elmove	34
--------------	----

M

makeinvisible	34
makevisible	35
marrow	35
mCsetup	17, 35 ff.
align	35
arrowlength	35
atomsiz e	35
bondlength	35
mergelength	35
rcndist	35
reset	35
rxnalign	35
schemealign	35
merge	36 - 40, 50
mesomeric	40 ff.
mhchem	6

O

Optionen	17
----------------	----

- chemstyle 17
 color 17
 english 17
 nochemexec 17
 nocolor 17
 nocompounds 17
 nomhchem 17
 nopackages 17
 placement 17
 shade 17
- R**
- reactant 8, 42 f.
 tikz 18
 rxn 8, 43 ff.
 Optionen 43 ff.
 ausrichtung 43
 skalierung 43
 rxnscheme 14, 45 - 49
 anpassen 47 ff.
 Benennung 48
 Platzierung 48
 Stil 47
 Verzeichnis 49
 Zähler 49
 Optionen 45 ff.
 ausrichtung 46
 label 45
 platzierung 45
 skalierung 46
 titel 46
- S**
- setarrowlabel 16, 49
 setarrowlength 16, 50
 setarrowline 16, 50
 setatomsep 15
 setatomsizes 15, 50
 setbondlength 15, 50
 setbondshape 15, 51
 setcrambond 15
 setmergelength 37, 51
 setrendist 51
- setrxnalign 51 f.
 setschemealign 51 f.
 setschemename 52
- T**
- transition 52 f.
- U**
- Umgebung
 rxn *siehe rxn*
 rxnscheme *siehe rxnscheme*
- V**
- Voraussetzungen 5
 calc 5
 ChemFig 5
 float 5
 ifthen 5
 pgf 5
 TikZ 5
 xkeyval 5
 Voreinstellungen 15 ff.
- X**
- xshift 18, 21, 32, 39
- Y**
- yshift 18, 21, 32, 39