

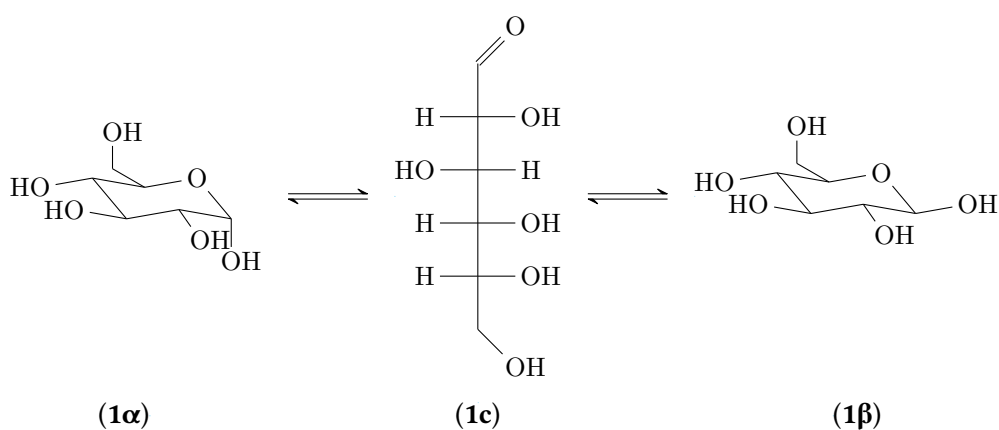
CHEMNUM

v0.3e 2011/12/01

Clemens NIEDERBERGER

<http://www.mychemistry.eu/>
contact@mychemistry.eu

Ein umfassender Ansatz für das Nummerieren chemischer Verbindungen
mit $\text{\LaTeX 2}_{\epsilon}$.



Schema 1: Das α - und das β -Anomer von D-Glucose (**1 α** und **1 β**) stehen in Lösung über die offenkettige Form (**1c**) im Gleichgewicht.

Inhaltsverzeichnis

1	Lizenz, Voraussetzungen	3
2	Paket-Optionen	3
3	Motivation	3
4	Verbindungen Nummerieren	5
4.1	Basisbefehl	5
4.2	Sublabel	6
4.3	Listen	6
5	Label zentral initialisieren	7
5.1	Verwendung	7
5.1.1	Haupt-Label	7
5.1.2	Sublabel	8
5.2	Fehler statt Warnung	8
5.3	Warum initialisieren?	9
6	Zurücksetzen des Zählers	9
7	Besondere Verwendung	10
7.1	Abschnitts-Titel	10
7.1.1	Test-Überschrift mit Verbindungslabel (7a)	11
7.1.2	Test-Überschrift mit Verbindungslabel (7b)	11
7.2	Gleitumgebungen	11
7.3	Schemata externer Programme	11
8	Listen und Bereiche von Sublabeln	14
9	Personalisieren	16
9.1	Das Prinzip	16
9.2	Schriftstil	17
9.3	Eigene Labels	18
9.4	Eigene Marker in Bildern	18
9.5	Zähler	18
9.6	Begrenzer	19
9.7	Suffix und Präfix	20
9.8	Trenner	21
9.8.1	Bei Labeln	21
9.8.2	Bei Listen – Eingabe	22
9.8.3	Bei Listen – allgemein	22
9.8.4	Bei Listen – letzter Trenner	23
9.9	Sublabel-Marker	24

10 Sprachunterstützung	24
10.1 Zusammenarbeit mit babel	24
10.2 Harvard-Komma	25
11 Übersicht	27
11.1 Keys	27
11.2 Befehle	28
12 Dank	28

1 Lizenz, Voraussetzungen

CHEMNUM v0.3e steht unter der L^AT_EX Project Public License Version 1.3 oder später.
(<http://www.latex-project.org/lppl.txt>)

CHEMNUM ruft intern die Pakete expl3 und xparse¹ auf. xparse ist Teil des l3packages²-Bundles, expl3 ist Teil des l3kernel³-Bundles. Das heißt, **CHEMNUM** benötigt L^AT_EX3-Unterstützung.

CHEMNUM benötigt außerdem die Pakete etoolbox⁴ und psfrag⁵ und entweder das Paket textgreek⁶ oder das Paket bm⁷, abhängig von den Paketoptionen (siehe Abschnitt 2).

2 Paket-Optionen

CHEMNUM hat eine Option.

```
\usepackage[textgreek = <value>]{chemnum}
```

Damit können Sie sowohl einen der drei Stile des textgreek-Pakets wählen⁸ oder **CHEMNUM** ohne textgreek laden. Folgende Werte stehen zur Auswahl: artemisia, cbgreek, euler und false. Wenn Sie <value> weglassen, wird textgreek mit dem cbgreek-Stil geladen. Die Verwendung von `\usepackage[textgreek]{chemnum}` entspricht also `\usepackage[textgreek = cbgreek]{chemnum}`. Wenn Sie **CHEMNUM** ohne Option verwenden, wird textgreek nicht verwendet. Das entspricht `\usepackage[textgreek = false]{chemnum}`, der Voreinstellung.

Dieses Dokument wurde mit `\usepackage[textgreek]{chemnum}` gesetzt.

3 Motivation

Soweit mir bekannt ist, gab es bislang drei Pakete, die das Nummerieren von Verbindungen erleichtern sollen. Sie alle haben ihre Schwächen.

¹<http://www.ctan.org/pkg/xparse>

²<http://www.ctan.org/pkg/l3packages>

³<http://www.ctan.org/pkg/l3kernel>

⁴<http://www.ctan.org/pkg/etoolbox>

⁵<http://www.ctan.org/pkg/psfrag>

⁶<http://www.ctan.org/pkg/textgreek>

⁷<http://www.ctan.org/pkg/bm>

⁸Zu Details siehe Dokumentation des textgreek-Pakets.

Das erste, chemcono⁹, hat für diesen Zweck Bibliographie-Befehle umdefiniert. Dort müssen Verbindungen in einer Art Literaturliste definiert werden. Dadurch kann dann im Text auf sie referenziert werden. Da es die „Verbindungs-Bibliothek“ immer mit ausgibt, schlägt der Paket-Autor folgendes vor:

After compilation and printout, discard the last page.

Stefan Schulz

Ganz offensichtlich ist das keine perfekte Lösung.

Das zweite, chemcompounds¹⁰, ist entstanden, weil der Paket-Autor ebenfalls mit den Schwächen von chemcono nicht mehr arbeiten wollte. Bei der Erstellung des Pakets hielt er sich stark an den gleichen prinzipiellen Mechanismus wie chemcono.

When taking a closer look at the chemcono package, I realised that the only thing one has to do is to get rid of everything which produces text. Thus, as a basis I used the mechanism of `\bibitem` and `\cite` in pretty much the same way as chemcono does by extracting the corresponding code from `article.cls` and `latex.ltx` but deleting any unnecessary commands producing output. I also introduced several lines of code to make the printing of the compound names more customisable.

Stephan Schenk

Manche Schwächen hat auch dieses Paket:

1. Verbindungen müssen in der Regel mit `\declarecompound` deklariert werden. Sie müssen es auf jeden Fall, wenn man Label der Art **1a** haben möchte. Dann muss man die Label sogar von Hand vergeben, was den Sinn einer automatischen Nummerierung untergräbt.
2. Das Layout lässt sich nicht einzeln ändern, sondern nur für alle Label.
3. Die Nummerierung kann nicht zurückgesetzt werden. Obwohl das in der allermeisten Fällen *weder nötig noch zu empfehlen ist*, kann es Einzelfälle geben, in denen solches Verhalten gewünscht ist.
4. Eine Liste mehrerer Verbindungen `\compound{a,b,c}` kann nicht oder nur umständlich eigenen Vorstellungen angepasst werden.

Dann gibt es noch bpchem¹¹, das Befehle ähnlich den Befehlen `\label` und `\ref` zur Verfügung stellt: `\CNlabel{}`, `\CNlabelnoref{}` und `\CNref{}`. Außerdem hat es Befehle für Sublabel: `\CNlabelsub{ }{ }`, `\CNlabelsubnoref{ }{ }` und `\CNrefsub{ }{ }`. Damit ist es flexibler als die anderen, stellt allerdings kaum Möglichkeiten zur Personalisierung bereit, Listen sind nicht möglich und die Tatsache, dass es verschiedene Befehle für Label und Label mit Sublabeln gibt, ist ebenfalls nicht die beste Lösung.

CHEMNUM soll diese Schwächen beheben. Dafür wurden alle Befehle vollständig neu geschrieben. Allerdings wurden Ideen von chemcompounds aufgegriffen was Darstellungsmöglichkeiten wie Begrenzer usw. betrifft.

Wenn Ihnen irgendwelche fehlenden Features auffallen, geben Sie mir bitte Bescheid, indem Sie mir eine E-Mail senden.

⁹<http://www.ctan.org/pkg/chemcono>

¹⁰<http://www.ctan.org/pkg/chemcompounds>

¹¹<http://www.ctan.org/pkg/bpchem>

4 Verbindungen Nummerieren

4.1 Basisbefehl

Der eigentliche Befehl des Pakets ist

`\cmpd{<label name>}`

Beim ersten Aufruf mit <label name> wird das Label erzeugt und gespeichert (= deklariert) und ausgegeben. Bei jedem weiteren Aufruf wird nun dieses Label verwendet.

```
1 Verbindungen \cmpd{a} und \cmpd{b} werden deklariert und k\"onnen nun
   jederzeit wieder aufgerufen werden: \cmpd{a}. Es ist keine vorherige
   Deklaration n\"otig. Die Substanzen wie \cmpd{c} werden in der
   Reihenfolge ihres ersten Erscheinens im Text nummeriert.\par
2 Noch einmal: \cmpd{b}, \cmpd{a}, \cmpd{c}.
```

Verbindungen 1 und 2 werden deklariert und können nun jederzeit wieder aufgerufen werden: 1.
Es ist keine vorherige Deklaration nötig. Die Substanzen wie 3 werden in der Reihenfolge ihres
ersten Erscheinens im Text nummeriert.
Noch einmal: 2, 1, 3.

Falls es nötig sein sollte, eine Verbindung zu deklarieren, ohne ein Label auszugeben, kann man an entsprechender Stelle

`\cmpd*{<label name>}`

verwenden. Dadurch wird ein entsprechendes Label deklariert aber nicht ausgegeben. **Dieser Befehl wird nicht benötigt, wenn man `\cmpdinit{}` verwendet, siehe Abschnitt 5 (S. 7ff).**

```
1 Die versteckte Version \cmpd*{d} erzeugt zwar das Label, schreibt es aber
   nicht aus. Der n\"achste \cmpd{e} f\"uhrt die Z\"ahlung einfach weiter.
   Mit \cmpd{d} kann nat\"urlich auf das erzeugte Label zur\"uckgegriffen
   werden.
```

Die versteckte Version erzeugt zwar das Label, schreibt es aber nicht aus. Der nächste 5 führt die
Zählung einfach weiter. Mit 4 kann natürlich auf das erzeugte Label zurückgegriffen werden.

Als Label-Namen können Sie im Prinzip einsetzen, was Sie wollen¹². Sie sollten allerdings darauf achten, dass der Name keine Leerzeichen enthält. Das führt zwar nicht unbedingt zu einem Fehler, könnte aber verschiedenen Labeln die gleiche Nummer zuweisen.

¹²Es gibt ein Einschränkungen, siehe Abschnitt 4.3 (S. 6), Abschnitt 9.8.2 (S. 22), Abschnitt 4.2 (S. 6) und Abschnitt 9.9 (S. 24).

```

1  \cmpd{aa}, \cmpd{a_a}, \cmpd{a_aa}, \cmpd{_a_a}, \cmpd{_aa} und \cmpd{
    aa} erzeugen alle das gleiche Label. So, wie auch \cmpd{aa}, \cmpd{a_a
    }, \cmpd{_aa} und \cmpd{aa}.

```

6, 6, 6, 6, 6 und 6 erzeugen alle das gleiche Label. So, wie auch 6, 6, 6 und 6.

4.2 Sublabel

Möchte man Label der Form **1a** erzeugen, geschieht das nach folgender Syntax:

```
\cmpd{<label name>.<subname>}
```

Dabei muss <label name> der gleiche bleiben, <subname> variiert. Diese Syntax bedeutet, dass der Punkt . weder Teil von <label name> noch von <subname> sein darf.

```

1  \cmpd{f.one} und \cmpd{f.two} sind eng verwandt, genauso \cmpd{g.one} und \
    cmpd{g.two}. Nat\"urlich bleibt auch hier bei sp\"aterem Aufruf das
    Label gleich: \cmpd{g.two} und \cmpd{f.one}.

```

7a und **7b** sind eng verwandt, genauso **8a** und **8b**. Natürlich bleibt auch hier bei späterem Aufruf das Label gleich: **8b** und **7a**.

Diese Syntax funktioniert auch, wenn der Hauptname bereits vergeben wurde.

```

1  \cmpd{a} und seine Varianten          1 und seine Varianten 1a und 1b
2  \cmpd{a.one} und \cmpd{a.two}

```

Genauso kann der Hauptname von kombiniert vergebenen Labeln einzeln aufgerufen werden.

```

1  \cmpd{f} und \cmpd{g}                7 und 8

```

Wie Sie ein Label der Art **7a,b** erzeugen können, wird in Abschnitt 8 erklärt.

4.3 Listen

Tatsächlich wurde bislang nur die halbe Wahrheit über den Befehl erzählt. Er kann auch für mehrere Label auf einmal verwendet werden.

Die richtige Beschreibung des Befehls lautet also eher:

```
\cmpd{<(possibly comma separated list of)label name(s)>}
```

Das bedeutet, dass in der Voreinstellung das Komma nicht Teil des Label-Namens sein kann, es sei den versteckt in geschweiften Klammern. Sie können das ändern, siehe Abschnitt 9.8.2 (S. 22).

```
1 Mehrere Label k\"onnen einfach durch Kommata getrennt eingegeben werden.  
   Dann wird eine Liste wie \cmpd{a, b, c, e, g.two} ausgegeben.
```

Mehrere Label können einfach durch Kommata getrennt eingegeben werden. Dann wird eine Liste wie **1, 2, 3, 5** und **8b** ausgegeben.

5 Label zentral initialisieren

Die in diesem Abschnitt beschriebenen Befehle sind für den Einsatz von CHEMNUM nicht nötig, stellen aber Funktionen bereit, die der ein oder andere Anwender nützlich finden könnte.

5.1 Verwendung

5.1.1 Haupt-Label

Vielleicht vermissen Sie ja den `\declarecompound`-Befehl von chemcompounds. Schließlich hat man dadurch am Beginn einen Überblick, welche Namen man bereits vergeben hat. Und man wird durch eine Fehlermeldung unter Umständen gewarnt, wenn man ein falsches Label setzt¹³. Dieses Verhalten lässt sich mit den folgenden Befehlen realisieren.

```
\cmpdinit{<comma separated list of label names>}  
\cmpdinit*{<comma separated list of label names>}
```

Alle Verbindungen, die innerhalb dieses Befehls gesetzt werden, werden *in der verwendeten Reihenfolge* initialisiert. Verwendet man ein Label, das nicht in der Liste enthalten ist, wird eine Warnung ausgegeben.

```
1 \cmpdinit{A, B, C} \cmpd{B}           2 1a 3 4  
2 \cmpd{A.a} \cmpd{C} \cmpd{D}
```

```
*****  
* chemnum warning: "cmpd-init"  
*  
* You used \cmpdinit but didn't initiate compound "D" on line 1.  
*****
```

Beachten Sie, dass Sie bei Labeln mit Sublabeln *nur den Hauptnamen* initialisieren können und sollten. Dadurch sind dann *alle* Verbindungen mit gleichem Hauptnamen initialisiert.

`\cmpdinit*{}` gibt die gleiche Warnung aus, allerdings werden die Nummern noch nicht vergeben, sondern erst bei Verwendung im Text und in der im Text verwendeten Reihenfolge.

¹³z. B. aufgrund eines Schreibfehlers

```

1  \cmpdinit*{X, Y, Z} \cmpd{Y}          1 2a 3 4
2  \cmpd{X.x} \cmpd{Z} \cmpd{W}

```

```

*****
* chemnum warning: "cmpd-init"
*
* You used \cmpdinit but didn't initiate compound "W" on line 1.
*****

```

Wenn Sie also eigene Label vergeben wollen oder müssen (siehe Seite 18) und die Label geschlossen initialisieren wollen, sollten Sie `\cmpdinit*{}` verwenden, da dort bei der Initialisierung *keine* Label vergeben werden.

Der Befehl kann mehrfach verwendet werden, mit oder ohne *. Sinnvoll ist aber wohl eine *einmalige* Verwendung, vielleicht in der Dokumentenpräambel nach dem `CHEMNUM`-Setup (siehe Abschnitt 9).

5.1.2 Sublabel

Wenn Sie zusätzlich auch für jede Label-Sublabel-Kombination gewarnt werden wollen oder die Reihenfolge im Voraus festlegen wollen, dann können Sie

```
\cmpdinit[sub-init = true]{<comma separated list of label names>}
```

verwenden. So werden Sie auch in diesen Fällen gewarnt.

```

1  \cmpdinit[sub-init = true]{E.e}      1a und 1b
2  \cmpd{E.e, E.f}

```

```

*****
* chemnum warning: "cmpd-sub-init"
*
* You used \cmpdinit and "sub-init = true" but didn't initiate sub-compound
* "E.f" on line 1.
*****

```

5.2 Fehler statt Warnung

Wenn Sie lieber eine Fehler-Meldung anstelle einer Warnung mögen, können Sie das realisieren, indem Sie `\cmpdinit` folgendermaßen aufrufen:

```
\cmpdinit[strict = true]{<comma separated list of label names>}
```

Die Verwendung `strict=false` entspricht der Verwendung ohne optionalem Argument. Die Option kann natürlich auch bei der Stern-Variante eingesetzt werden.

Durch `strict=true` wird der \LaTeX -Lauf mit einer entsprechenden Fehlermeldung abgebrochen, wenn man einen Label-Namen verwendet, der nicht initialisiert wurde.

Natürlich können Sie `\cmpdinit` auch mit beiden Keys aufrufen:

```
\cmpdinit[strict = true, sub-init = true]{<comma separated list of label names>}
```

Wenn Sie einen anderen Sub-Label-Marker ausgewählt haben (siehe Seite 24), müssen Sie mit `\cmpdinit[sub-init = true]` ebenfalls verwenden.

5.3 Warum initialisieren?

Wie zu Beginn dieses Abschnitts erwähnt, ist die Initialisierung von Labeln nicht nötig. Es gibt jedoch Gründe, warum man `\cmpdinit` vielleicht trotzdem verwenden möchte.

1. Der Einsatz von `\cmpdinit*` ist eine Möglichkeit, den Überblick darüber zu behalten, welche Label man verwendet hat. Man bekommt eine Warnung/Fehlermeldung, wenn man ein bestehendes falsch geschrieben hat. Abhängig davon, wieviele Labels man verwendet (oder verwenden muss) kann das aber viel zusätzliche Arbeit bedeuten.
2. Der Einsatz von `\cmpdinit` bewirkt erst einmal dasselbe wie `\cmpdinit*`, deklariert aber *zusätzlich* die Label in der vorgegebenen Reihenfolge. Damit wird die Verwendung von `\cmpd*` (Abschnitt 4.1) und `\cmpd+` (Abschnitt 7.1) überflüssig. Man hat damit direkte Kontrolle, welche Verbindung welche Nummer erhält.

6 Zurücksetzen des Zählers

Es ist möglich, mit folgendem Befehl

```
\cmpdreset[<number>]
```

den Zähler wieder zurückzusetzen. Ohne Argument wird er auf 1 gesetzt und entspricht damit `\cmpdreset[1]`. Dieser Befehl wirkt sich global aus, nicht nur lokal innerhalb einer Gruppe – im Gegensatz zu den meisten anderen vorgestellten Befehlen.

```
1 \cmpdreset
2 Die Verbindungen werden jetzt wieder von 1 gezählt: \cmpd{h, i, j}
```

Die Verbindungen werden jetzt wieder von 1 gezählt: 1, 2 und 3

Sie sollten mit diesem Befehl sehr vorsichtig umgehen! In der Regel ist es nicht zu empfehlen, den Zähler zurückzusetzen. Dadurch können ein und dieselbe Nummer an verschiedene Verbindungen vergeben werden!

```
1   Gleiche Nummern: \cmpd{a}, \cmpd{h}   Gleiche Nummern: 1, 1
```

7 Besondere Verwendung

7.1 Abschnitts-Titel

Die Verwendung von `\cmpd` in einer Überschrift scheint auf den ersten Blick kein Problem zu sein. Es gibt keine Fehler und der Output sieht wie erwartet aus:

```
1   \section{Compound \cmpd[cmpd-delim]{b}}
2   \ldots
```

Wenn man allerdings `\tableofcontents` verwendet – was ziemlich wahrscheinlich ist, wie ich vermute –, geschieht etwas unerwünschtes: da das Inhaltsverzeichnis am Anfang des Dokuments geschrieben wird, wird das Label beim Erscheinen im Inhaltsverzeichnis deklariert, anstatt an der Stelle im Text, an der es zuerst erscheint. Es gibt zwei Möglichkeiten, das zu umgehen:

1. Sie deklarieren die Label in der Präambel mit `\cmpdinit{}` (siehe Seite 7).
2. Sie verwenden `\cmpd+`, um das Label zu setzen. Damit wird das Label von der Datei `<jobname>.cmpd` ausgelesen (vorausgesetzt ihre Hauptdatei heißt `<jobname>.tex`). Das bedeutet, dass *mindestens* zwei, unter Umständen mehr \LaTeX Durchläufe nötig sind, bis alle Labels richtig gesetzt sind.

Das gleiche gilt für jede andere Liste von beweglichen Argumenten, die *vor* dem Erscheinen des Labels im Text eingesetzt wird.

Der Befehl

`\cmpd+{<label name>}`

liest den Labelnamen, der ausgegeben werden soll, aus der Datei `<jobname>.cmpd`. Damit ist er sozusagen das Gegenteil von `\cmpd*`, da das Label nicht deklariert sondern nur referenziert und ausgegeben wird. Damit man also `\cmpd+{<label>}` einsetzen kann, muss `<label>` entweder mit `\cmpd{<label>}` oder `\cmpd*{<label>}` (davor oder danach) deklariert worden sein. `\cmpd+` ist nützlich, wenn ein Label in einem beweglichen Argument wie in `\section{}` eingesetzt werden soll. **Er wird nicht benötigt, wenn `\cmpdinit{}` verwendet wird, siehe Abschnitt 5 (S. 7ff).**

```

1 \subsubsection{Test-\textit{Überschrift mit Verbindungslabel \textit{cmpd}+[cmpd-delim]{f
  .one}}}
2 % mit hyperref:
3 \subsubsection{Test-\textit{Überschrift mit Verbindungslabel \textit{texorpdfstring}{\
  cmpd+[cmpd-delim]{f.two}}{(7b)}}}

```

7.1.1 Test-Überschrift mit Verbindungslabel (7a)

7.1.2 Test-Überschrift mit Verbindungslabel (7b)

7.2 Gleitumgebungen

Die Verwendung von `\cmpd` in Gleitumgebungen sollte kein Problem darstellen: die Labels werden in der Reihenfolge deklariert, wie Sie im Code auftauchen, eingeschlossen die Gleitumgebungen. Um auf der sicheren Seite zu sein, können Sie natürlich dennoch `\cmpd+` verwenden.

```

1 % preamble:
2 % \usepackage{chemscheme}
3 % \renewcommand*\schemename{Schema}
4 % document:
5 \cmpdreset\cmpd{float1, float2}. Und jetzt ein Schema:
6 \begin{scheme}[ht]
7   \centering
8   \caption{Dieses Label sollte eine 2 sein: \cmpd{float2}}
9   Wenn 2 = \cmpd{float2} wahr ist, ist alles ok.
10  \end{scheme}

```

1 und 2. Und jetzt ein Schema:

Wenn 2 = 2 wahr ist, ist alles ok.

Schema 2: Dieses Label sollte eine 2 sein: 2

Es funktioniert auch alles wie gewünscht, wenn die Gleitumgebung vor den aktuellen Code rutscht, etwa mit der Option [t].

7.3 Schemata externer Programme

Wenn Sie Ihre Schemata mit externen Programmen wie z. B. CHEMDRAW oder MARVINSKETCH erstellen und dort ebenfalls `CHEMNUM` einsetzen wollen, dann können Sie das mit

`\cmpdref [<keyval> [<tag>]{<label name>}`

tun. Dieser Befehl ist vom Befehl `\schemeref` des Pakets `chemscheme`¹⁴ inspiriert und funktioniert letztlich gleich. Sie erstellen das Schema und speichern es mit temporären Labels als eps-Datei.

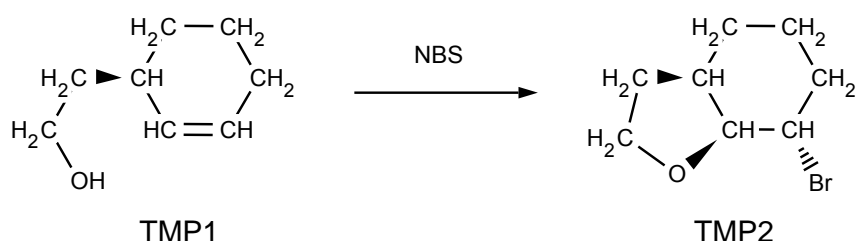
¹⁴<http://www.ctan.org/pkg/chemscheme>

```

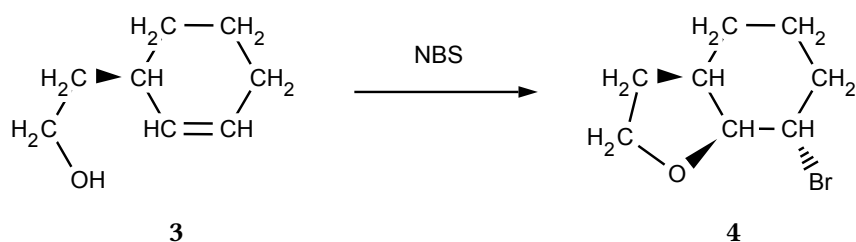
1 \begin{scheme}[ht]
2 \centering
3 \includegraphics{scheme-tmp}
4 \caption{Schema mit temporären Markern.}
5 \end{scheme}
6 \begin{scheme}[ht]
7 \centering
8 \cmpdref{Alc} % ersetzt TMP1
9 \cmpdref{EtherBr} % ersetzt TMP2
10 % \cmpdref{drittes} wuerde TMP3 ersetzen
11 \includegraphics{scheme-tmp}
12 \caption{Schema mit automatisierten Labeln.}
13 \end{scheme}

```

Sie nummerieren also im Schema die Substanzen mit TMP1, TMP2 usw. durch. Diese Marker



Schema 3: Schema mit temporären Markern.



Schema 4: Schema mit automatisierten Labeln.

werden dann durch die entsprechenden Label ersetzt. Dafür verwendet `CHEMNUM` genau wie `chemscheme` den Befehl `\psfrag`. Damit er funktioniert, müssen Sie also entweder über L^AT_EX, D_VIPS, P_S2PDF kompilieren, oder z. B. das Paket `auto-pst-pdf`¹⁵ zusammen mit P_DF_LA_TE_X einsetzen. X_EL_AT_EX wird leider nicht funktionieren, L_UA_LA_TE_X voraussichtlich ebensowenig, wurde aber vom Autor nicht getestet.

In jedem Fall ist es wichtig, dass der Marker als *Text* im eps gespeichert ist.

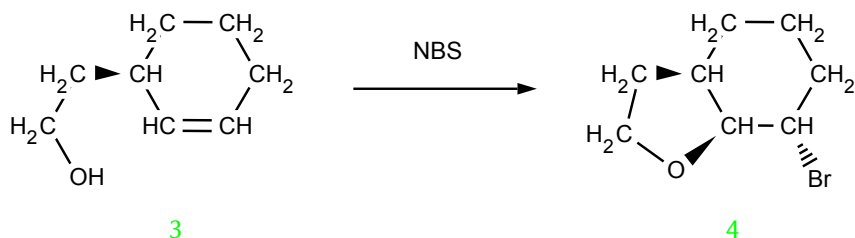
Mit den Keys, die in Abschnitt 9 vorgestellt werden, können Sie die Label hier auch personalisieren.

¹⁵<http://www.ctan.org/pkg/auto-pst-pdf>

```

1 \begin{scheme}[ht]
2 \centering
3 \cmpdref[cmpd-style=\bf\sf\textcolor{green}]{Alc} % ersetzt TMP1
4 \cmpdref[cmpd-style=\bf\sf\textcolor{green}]{EtherBr} % ersetzt TMP2
5 \includegraphics{scheme-tmp}
6 \caption{Schema mit automatisierten Labeln.}
7 \end{scheme}

```



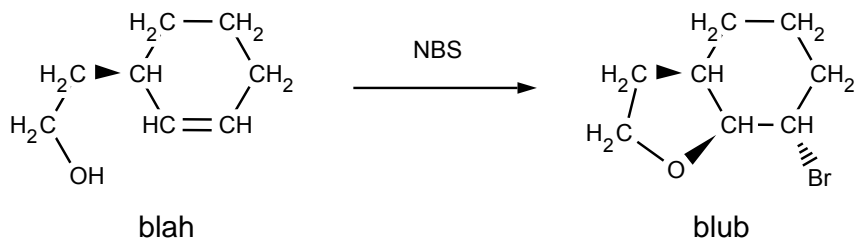
Schema 5: Schema mit automatisierten Labeln.

Sie können auch beliebigen Text ersetzen. Dafür gibt es das zweite optionale Argument.

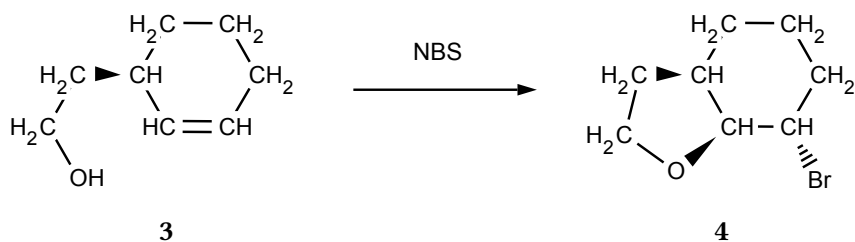
```

1 \begin{scheme}[ht]
2 \centering
3 \includegraphics{scheme-bla}
4 \caption{Schema mit beliebigen Markern.}
5 \end{scheme}
6 \begin{scheme}[ht]
7 \centering
8 \cmpdref[][blah]{Alc}
9 \cmpdref[][blub]{EtherBr}
10 % \cmpdref{drittes} wuerde TMP1 ersetzen
11 \includegraphics{scheme-bla}
12 \caption{Schema mit explizit gesetzten Labeln.}
13 \end{scheme}

```



Schema 6: Schema mit beliebigen Markern.



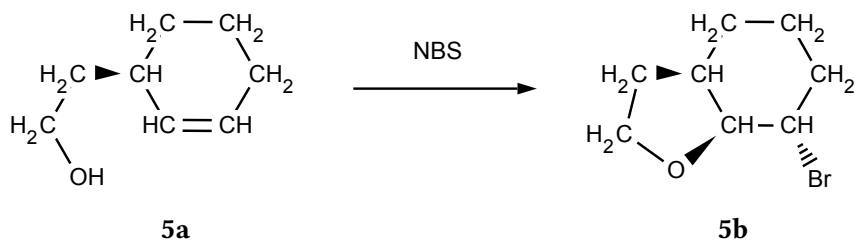
Schema 7: Schema mit explizit gesetzten Labeln.

Natürlich funktioniert das mit Sublabeln nach dem gewohnten Muster:

```

1 \begin{scheme}[ht]
2 \centering
3 \cmpdref{cpd.red}
4 \cmpdref{cpd.ox}
5 \includegraphics{scheme-tmp}
6 \caption{Schema mit automatisierten Labeln und Sublabeln.}
7 \end{scheme}

```



Schema 8: Schema mit automatisierten Labeln und Sublabeln.

8 Listen und Bereiche von Sublabeln

Manchmal kann es nützlich oder erwünscht sein, ein Label mit einem Bereich von Sublabeln anzuzeigen. Nehmen wir an, Sie haben die Verbindungen **9a**, **9b**, **9c**, **9d** und **9e**, die sich z. B. in ihren Substituenten unterscheiden. Unter Umständen möchte man auf alle gleichzeitig verweisen: **9a–e**.

CHEMNUM stellt zwei Keys (siehe Abschnitt 9) zur Verfügung, mit denen solche Label erzeugt werden können, und zwei, mit denen sie angepasst werden können.

<code>cmpd-all = true/false</code>	Typ: boolean
<code>sub-list = {<list of sublabel names>}</code>	Typ: other
<code>sub-range-sep = <separator></code>	Typ: literal
<code>sub-range-marker = <marker></code>	Typ: literal

Der Typ des Keys sub-list unterscheidet sich von dem anderer Keys. Am besten schauen Sie Sich das Beispiel unten an.

```
1 Liste von Labeln: \cmpd{q.one, q.two, q.three, q.four, q.five} \\
2 Alle auf einmal: \cmpd[cmpd-all]{q}
```

Liste von Labeln: **9a, 9b, 9c, 9d und 9e**
 Alle auf einmal: **9a-e**

Wenn Sie nicht auf alle Varianten auf einmal verweisen wollen, können Sie den Key sub-list verwenden. Als Eingabe benötigt er eine Komma-separierte Liste der Sublabel-Namen, die zu dem Hauptlabel gehören, das Sie verwenden.

```
1 \cmpd[sub-list={one,three,four}]{q} 9a,c,d
```

Sie können mit diesem Key auch Bereiche von Sublabeln darstellen. Als Marker zwischen den Namen der beiden Sublabel werden zwei Punkte .. verwendet.

```
1 \cmpd[sub-list={two..four}]{q} \\
2 \cmpd[sub-list={one,three..five}]{q} \\
3 \cmpd[sub-list={one..three,five}]{q}
```

9b-d
9a,c-e
9a-c,e

Mit den Keys sub-range-sep und sub-range-marker kann man sowohl das Symbol, mit dem der Bereich angezeigt wird als auch den Marker für die Eingabe ändern.

```
1 \cmpdsetup{sub-range-sep = {---}, sub-range-marker = : }
2 \cmpd[sub-list={two:four}]{q}
```

9b---d

Damit die Keys die richtigen Sublabel verwenden können, müssen diese deklariert worden sein. Daher werden sie aus der Datei <jobname>.cmpd (angenommen Ihre Quelldatei heißt <jobname>.tex) gelesen. Daher müssen Sie zweimal (oder öfter) kompilieren, bis alle Labels richtig dargestellt werden.

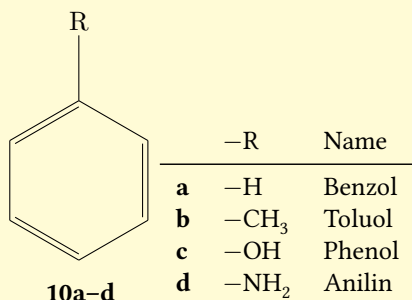
Verwendet man Label mit Sublabel-Bereichen, dann möchte man vielleicht an anderer Stelle *nur* das Sublabel ausgeben. Das ist möglich mit folgendem Key:

sub-only = true/false	Typ: boolean
-----------------------	--------------

```

1 % use packages 'chemfig', 'mhchem' and 'booktabs'
2 \cmpd*{benzol.H,benzol.Me,benzol.OH,benzol.NH2}
3 \chemname{\chemfig{*6(=--(-R)=)}}{\cmpd[cmpd-all]{benzol}}
4 \begin{tabular}{lll}
5                                     & & & \ce{{}-R} & & Name \\
6 \cmpd[sub-only]{benzol.H} & & \ce{{}-H} & & Benzol \\
7 \cmpd[sub-only]{benzol.Me} & & \ce{{}-CH3} & & Toluol \\
8 \cmpd[sub-only]{benzol.OH} & & \ce{{}-OH} & & Phenol \\
9 \cmpd[sub-only]{benzol.NH2} & & \ce{{}-NH2} & & Anilin
10 \end{tabular}

```



In der Zwischenzeit benötigt der Key sub-list etwas Aufmerksamkeit. Leider ist automatisches Sortieren nicht so trivial, wie man sich wünschte. Daher müssen Sie die Sublabel-Namen in der richtigen Reihenfolge einsetzen, um Label wie diese zu vermeiden:

```

1 \cmpd[sub-list={five..three}]{q} or \cmpd[sub-list={three,one,four}]{q}.

```

9e–c or 9c,a,d.

9 Personalisieren

9.1 Das Prinzip

Das Personalisieren geschieht mit einem Schlüssel-Wert-System. Einstellungen können dabei entweder zentral über

```
\cmpdsetup{key1 = value1, key2 = value2}
```

oder lokal direkt beim Befehl

```
\cmpd[key1 = value1, key2 = value2]{<label name>}
```

vorgenommen werden.

Dabei gibt es verschiedene Arten von Schlüsseln. Je nach Art können sie andere Werte bekommen.

macro Diese Schlüssel erwarten als Eingabe ein Makro, das auch ein obligatorisches Argument besitzen kann, z. B. `\bfseries` oder `\textbf`. **CHEMNUM** hat nur einen einzigen Schlüssel dieser Art, siehe Abschnitt 9.2.

literal Diese Sorte Schlüssel verwendet das eingegebene direkt, also buchstäblich. Die meisten Schlüssel von **CHEMNUM** entsprechen dieser Art.

choice Auswahl-Schlüssel, für den es vorgegebene Optionen gibt, aus denen gewählt werden kann.

Wenn Sie Schlüssel ohne Wertzuweisung verwenden `\cmpdsetup{key1,key2}`, werden Defaultwerte verwendet. Diese sind *nicht* unbedingt leer und stimmen auch *nicht* unbedingt mit der Voreinstellung des Pakets überein. *Alle* Schlüssel von **CHEMNUM** haben Defaultwerte. Sie werden in Abschnitt 11.1 auf Seite 27 aufgelistet.

Die Schlüssel sind in verschiedene Klassen unterteilt, die auf ihren Verwendungsbereich hindeuten. Sie folgen alle der Form `<class>-key = <value>`:

cmpd Schlüssel, die damit beginnen, wirken sich auch das Label insgesamt oder speziell auf das Hauptlabel aus.

sub Diese Schlüssel haben mit den Sublabeln direkt zu tun.

list Schlüssel, die die Ausgabe der Listen ändern.

ref Schlüssel, die speziell mit `\cmpdref` zu tun haben.

9.2 Schriftstil

Um den Stil zu ändern, mit dem die Nummern geschrieben werden, verwenden Sie

<code>cmpd-style = <style></code> <code>cmpd-weight = bold/normal</code>	Typ: macro Typ: choice
---	---------------------------

```
1 % preamble:
2 % \usepackage[normalem]{ulem}
3 % document:
4 \cmpd{a, b, f.two, k} \\ % default
5 \cmpd[cmpd-style = \uline]{a, b, f.two, k} \\ % unterstrichen
6 \cmpd{a, b, f.two, k} \\ % wieder default
7 \cmpdsetup{cmpd-style = \itshape\uline}
8 \cmpd{a, b, f.two, k} \\ % kursiv und unterstrichen
9 \cmpd[cmpd-weight=normal]{a, b, f.two, k}
```

1, 2, 7b und 11

1, 2, 7b und 11

1, 2, 7b und 11

1, 2, 7b und 11

1, 2, 7b und 11

Dass die Schriftstärke separat eingestellt wird, hat seine Ursache bei den griechischen Labels (siehe Abschnitt 9.5). Abhängig von der Paketooption muss die Schriftstärke dort unterschiedlich realisiert werden.

9.3 Eigene Labels

Mit dem Schlüssel

```
cmpd-label = <name>
```

Typ: literal

kann man eigene Labels vergeben.

```
1 Eigenes Label: \cmpd[cmpd-label = XYZ]{1}, wird bei einer Liste an \emph{
  alle neuen} vergeben! \cmpd[cmpd-label = XYZ]{1, a, m}
```

Eigenes Label: XYZ, wird bei einer Liste an *alle neuen* vergeben! XYZ, 1 und XYZ

9.4 Eigene Marker in Bildern

Wenn Sie für die temporären Label, die durch `\cmpdref` (siehe Seite 11) nicht TMP verwenden wollen, können Sie das ändern, z. B. in tmp:

```
1 \cmpdsetup{ref-marker = tmp}
```

9.5 Zähler

Wenn Sie mögen, können Sie auch das Zählerformat **1a** ändern.

```
cmpd-counter = <counter>
```

Typ: choice

```
sub-counter = <counter>
```

Typ: choice

Zur Auswahl stehen arabic, alph, Alph, greek, Greek, roman, Roman und Symbol.

Bitte beachten Sie: ein Ändern des Zählerformats wirkt sich nur auf *noch nicht erzeugte* Label aus!

```
1 \cmpdsetup{cmpd-counter = Alph, sub-counter = arabic}
2 \cmpd{a, b, f.two, k}\\ % keine Wirkung bei bereits erzeugten Labels!
3 \cmpd{n.one, n.two, o.one}\\
4 \cmpd{f.three, f.four}\\ % Vorsicht: Hauptlabel wurde bereits erzeugt und
  wird nicht geändert!
5 \cmpd[cmpd-counter, sub-counter = greek]{p.one, p.two}
```

1, 2, **7b** und **11**
L1, L2 und M1
73 und 74
14α und **14β**

9.6 Begrenzer

Sie können sowohl den einzelnen Nummern wie auch eine ganze Liste mit Begrenzern versehen. Dafür gibt es die Schlüssel:

<code>compd-delim = <odelim><cdelim></code>	Typ: literal
<code>list-delim = <odelim><cdelim></code>	Typ: literal
Benötigen <i>zwei</i> Token als Wert!	

```
1 \compd{a, b, f.two, k} \\ % default
2 \compd[compd-delim = ()]{a, b, f.two, k} \\ % eingeklammert
3 \compd{a, b, f.two, k} \\ % wieder default
4 \compdsetup{compd-delim = ()}
5 \compd{a, b, f.two, k} % eingeklammert
```

1, 2, **7b** und **11**
(1), (2), (7b) und (11)
1, 2, **7b** und **11**
(1), (2), (7b) und (11)

Wenn Sie `\compd[compd-delim]{<label>}` schreiben, also keinen Wert spezifizieren, werden `()` verwendet. Dasselbe gilt für `list-delim`. Beachten Sie, dass die Listenbegrenzer erst ab *zwei* Labeln verwendet werden.

```
1 \compd{a, b, f.two, k} \\ % default
2 \compd[list-delim = {[}{]}]{a, b, f.two, k} \\ % eingeklammert
3 \compd{a, b, f.two, k} \\ % wieder default
4 \compdsetup{list-delim = []}
5 \compd{a, b, f.two, k} \\ % eingeklammert
6 \compd{a} % KEINE Liste!
```

1, 2, **7b** und **11**
[1, 2, **7b** und **11**]
1, 2, **7b** und **11**
[1, 2, **7b** und **11**]
1

Wenn Sie generell die Label eingeklammert verwenden, aber ein einzelnes einmal ohne Klammern verwenden wollen, haben Sie mehrere Möglichkeiten:

```

1 \cmpdsetup{cmpd-delim}
2 Normal \cmpd{b}, \cmpd{c}, \cmpd{d.one}, aber manchmal so \cmpd[cmpd-delim
   = ]{e} oder so \cmpd-{e}.

```

Normal (2), (3), (4a), aber manchmal so 5 oder so 5.

Mit

```

\cmpd-{\<label name>}
\cmpdref-{\<label name>}

```

kann man die Klammern eines Labels schnell entfernen (nicht jedoch die der Liste).

```

1 \cmpd{a, b, f.two, k}\\ % default
2 \cmpdsetup{cmpd-delim = (), list-delim = []}
3 \cmpd{a, b, f.two, k}\\ % doppelt eingeklammert
4 \cmpd-{a, b, f.two, k} % einfach eingeklammert

```

1, 2, 7b und 11

[(1), (2), (7b) und (11)]

[1, 2, 7b und 11]

Durch leere Argumente, `\cmpdsetup{cmpd-delim = }` und `\cmpdsetup{list-delim = }`, können Sie die Voreinstellung wiederherstellen.

9.7 Suffix und Präfix

Falls Sie wollen, können sie sowohl einzelnen Labeln als auch den Listen Prä- und Suffixe geben.

<code>cmpd-prefix = <prefix></code>	Typ: literal
<code>cmpd-suffix = <prefix></code>	Typ: literal
<code>list-prefix = <prefix></code>	Typ: literal
<code>list-suffix = <prefix></code>	Typ: literal

Wie bei den Begrenzern gilt: Listen-Attribute werden erst bei einer Liste – also zwei oder mehr Labeln – verwendet.

Beispiel für die Label-Attribute:

```

1 \cmpd{a, b, f.two, k}\\ % default
2 \cmpd[cmpd-prefix = Nr.]{a, b, f.two, k}\\
3 \cmpdsetup{cmpd-prefix = \(\rightarrow\)}
4 \cmpd{a, b, f.two, k}\\
5 \cmpd{a} % KEINE Liste!

```

1, 2, 7b und 11
 Nr. 1, Nr. 2, Nr. 7b und Nr. 11
 → 1, → 2, → 7b und → 11
 → 1

Beispiel für die Listen-Attribute:

```
1 \compd{a, b, f.two, k}\\ % default
2 \compd[list-prefix = Liste:]{a, b, f.two, k}\\
3 \compdsetup{list-prefix = Sammlung:}
4 \compd{a, b, f.two, k}\\
5 \compd{a} % KEINE Liste!
```

1, 2, 7b und 11
 Liste: 1, 2, 7b und 11
 Sammlung: 1, 2, 7b und 11
 1

Zwischen Präfix, Label und Suffix ist per Default ein Leerraum:

```
1 \compd[compd-suffix=Sx, compd-prefix=Px]{k}
```

Px 11 Sx

Dieser kann durch

compd-space = <space>

Typ: literal

geändert werden.

```
1 \compd[compd-suffix=Sx, compd-prefix=Px, compd-space={}]{k}
```

Px11Sx

9.8 Trenner

9.8.1 Bei Labeln

Haben Sie Label mit Sublabeln, können Sie dort auch ein Trenn-Symbol verwenden.

compd-sep = <separator>

Typ: literal

```

1 \cmpd{a, b, f.two, f.three, k}\\ % default
2 \cmpd[cmpd-sep = -]{a, b, f.two, f.three, k}\\
3 \cmpdsetup{cmpd-sep = $\cdot$}
4 \cmpd{a, b, f.two, f.three, k}\\
5 \cmpd[cmpd-sep = :]{a, b, f.two, f.three, k}

```

1, 2, 7**b**, 73 und 11
 1, 2, 7-**b**, 7-3 und 11
 1, 2, 7·**b**, 7·3 und 11
 1, 2, 7:**b**, 7:3 und 11

9.8.2 Bei Listen – Eingabe

Wenn Sie mögen, können Sie das Trennsymbol der Listen-Eingabe ändern. Per Default wird hier das Komma verwendet.

`list-input-sep = <separator>`

Typ: literal

```

1 \cmpd{a, b, f.two, k}\\ % default
2 \cmpd[list-input-sep = +]{a + b + f.two + k}\\
3 \cmpd{a, b, f.two, k}\\
4 \cmpdsetup{list-input-sep = ;}
5 \cmpd{a; b; f.two; k}

```

1, 2, 7**b** und 11
 1, 2, 7**b** und 11
 1, 2, 7**b** und 11
 1, 2, 7**b** und 11

Das ermöglicht Ihnen z. B., IUPAC-Namen als Label-Namen zu verwenden¹⁶.

9.8.3 Bei Listen – allgemein

Auch bei Listen lässt sich ein Trennsymbol festlegen, das bestimmt, wie einzelne Label voneinander getrennt werden. Per Default ist das das Komma.

`list-output-sep = <separator>`

Typ: literal

```

1 \cmpd{a, b, f.two, k}\\ % default
2 \cmpd[list-output-sep = ]{a, b, f.two, k}\\
3 \cmpd{a, b, f.two, k}\\
4 \cmpdsetup{list-output-sep = ;}
5 \cmpd{a, b, f.two, k}

```

¹⁶Vielen Dank an Christina Lüdingk, die dieses Feature in gewisser Weise vorgeschlagen hat.

1, 2, **7b** und **11**
1 2 7b und **11**
1, 2, **7b** und **11**
1; 2; **7b** und **11**

9.8.4 Bei Listen – letzter Trenner

Der letzte Trenner bei Listen, derjenige zwischen dem vorletzten und letzten aufgelisteten Label, ist ein besonderer Fall. Hierfür gibt es zwei Schlüssel, mit dem er angepasst werden kann.

<code>list-last-sep = <separator></code>	Typ: literal
<code>list-lang = <lang></code>	Typ: choice

Zum einen gibt es den choice-Schlüssel `list-lang`, mit dem Sprachspezifische Einstellungen geändert werden können. Zur Auswahl stehen US (default), GB, DE, FR, ES und IT.

```
1 \cmpd[list-lang = US]{a, b, f.two, k} \cmpd[list-lang = US]{a, b}\\
2 \cmpd[list-lang = GB]{a, b, f.two, k} \cmpd[list-lang = GB]{a, b}\\
3 \cmpd[list-lang = DE]{a, b, f.two, k} \cmpd[list-lang = DE]{a, b}\\
4 \cmpd[list-lang = FR]{a, b, f.two, k} \cmpd[list-lang = FR]{a, b}\\
5 \cmpd[list-lang = ES]{a, b, f.two, k} \cmpd[list-lang = ES]{a, b}\\
6 \cmpd[list-lang = IT]{a, b, f.two, k} \cmpd[list-lang = IT]{a, b}
```

1, 2, **7b**, and **11** 1 and 2
1, 2, 7b and **11** 1 and 2
1, 2, **7b** und **11** 1 und 2
1, 2, **7b** et **11** 1 et 2
1, 2, **7b** y **11** 1 y 2
1, 2, **7b** e **11** 1 e 2

Wie Sie sehen, verwendet die Spracheinstellung US per Default das Harvard-Komma, alle anderen Sprachen verwenden es nicht. Mehr zur Sprach-Unterstützung und zum Harvard-Komma finden Sie in Abschnitt 10.

Sie können den letzten Trenner mit `list-last-sep` auch individuell festlegen.

```
1 \cmpdsetup{list-lang = DE}%
2 \cmpd[list-last-sep = {und auch}]{a, b, f.two, k}\\
3 \cmpd[list-last-sep = ]{a, b, f.two, k}\\
4 \cmpd[list-last-sep = sowie]{a, b, f.two, k}\\
5 \cmpd[list-last-sep = empty]{a, b, f.two, k}
```

1, 2, **7b** und auch **11**
1, 2, 7b **11**
1, 2, **7b** sowie **11**
1, 2, 7b, 11

Ein spezieller Wert ist `empty`. Wird er verwendet, wird überall der Trenner eingesetzt, der mit `list-output-sep` festgelegt wurde.

```
1 \cmpdsetup{list-last-sep = empty}%
2 \cmpd{a, b, f.two, k}\\
3 \cmpd[list-output-sep = ;]{a, b, f.two, k}\\
4 \cmpd[list-output-sep = {\ und}]{a, b, f.two, k}
```

1, 2, 7b, 11
1; 2; 7b; 11
1 und 2 und 7b und 11

9.9 Sublabel-Marker

Als Voreinstellung verwendet `CHEMNUM` den Punkt `.` als Zeichen, um Hauptlabelnamen von Sublabelnamen zu unterscheiden. Sie können das nach Belieben ändern. Die Zeichen `,` `%` `#` können Sie nicht verwenden und `@` sollten Sie nicht verwenden.

sub-marker = <separator>

Typ: literal

```
1 \cmpdsetup{sub-marker = !}%
2 \cmpd{f!one, g!two}\\
3 \cmpd[sub-marker= +]{f+one, g+two}\\
4 \cmpd[sub-marker= ~]{f~one, g~two}\\
5 \cmpd[sub-marker= &]{f&one, g&two}\\
6 \cmpd[sub-marker= *]{f*one, g*two}\\
7 \cmpd[sub-marker=\marker]{f\marker one,g\marker two}
```

7a und 8b
7a und 8b
7a und 8b
7a und 8b
7a und 8b
7a und 8b

Sie sollten – nicht nur aus Gründen der Konsistenz – nur *einmal* am Anfang ihres Dokuments entscheiden, welches Zeichen Sie verwenden. Und zwar *bevor* Sie `\tableofcontents` einsetzen. Am besten in der Präambel. Ansonsten kann `\tableofcontents` nicht wissen, welchen Marker Sie verwenden und Label im Inhaltsverzeichnis können falsch dargestellt werden.

10 Sprachunterstützung

10.1 Zusammenarbeit mit babel

`CHEMNUM` unterstützt die Sprachauswahl durch das `babel`¹⁷-Paket für eine Reihe von Sprachen. Dies geschieht mit Hilfe des `translator` Pakets, falls es vorhanden. Kann `translator` nicht geladen

¹⁷<http://www.ctan.org/pkg/babel>

werden, muss die Sprachunterstützung von Hand vorgenommen werden, siehe Abschnitt 9.8.4. Die Einstellung von Hand funktioniert natürlich auch, wenn translator geladen wurde.

```
1 \selectlanguage{ngerman}DE: \cmpd{r,s,t} \selectlanguage{german}\cmpd{r,s,t}
  \\\
2 \selectlanguage{naustrian}AU: \cmpd{r,s,t} \selectlanguage{austrian}\cmpd{r
  ,s,t}\\\
3 \selectlanguage{english}EN: \cmpd{r,s,t}\\\
4 \selectlanguage{british}GB: \cmpd{r,s,t}\\\
5 \selectlanguage{american}US: \cmpd{r,s,t}\\\
6 \selectlanguage{canadian}CDN-EN: \cmpd{r,s,t}\\\
7 \selectlanguage{canadien}CDN-FR: \cmpd{r,s,t}\\\
8 \selectlanguage{french}FR: \cmpd{r,s,t}\\\
9 \selectlanguage{spanish}ES: \cmpd{r,s,t}\\\
10 \selectlanguage{italian}IT: \cmpd{r,s,t}
```

DE: 15, 16 und 17 15, 16 und 17

AU: 15, 16 und 17 15, 16 und 17

EN: 15, 16, and 17

GB: 15, 16 and 17

US: 15, 16, and 17

CDN-EN: 15, 16, and 17

CDN-FR : 15, 16 et 17

FR : 15, 16 et 17

ES: 15, 16 y 17

IT: 15, 16 e 17

Bei weitem nicht alle Sprachen werden unterstützt, schon allein aus Machbarkeitsgründen. Falls Ihnen eine Sprache fehlen sollte, senden Sie mir bitte eine E-Mail¹⁸ mit der dazugehörigen babel Option und der richtigen Übersetzung, und ich werde sie gerne hinzufügen.

10.2 Harvard-Komma

Im US-Amerikanischen ist die Verwendung des sogenannten Harvard- oder Oxford-Kommas weit verbreitet, im Britischen und in anderen europäischen Sprachen ist sie eher unüblich.

The serial comma (also known as the Oxford comma or Harvard comma, and sometimes referred to as the series comma) is the comma used immediately before a coordinating conjunction (usually and or or, and sometimes nor) preceding the final item in a list of three or more items. For example, a list of three countries can be punctuated as either "Portugal, Spain, and France"(with the serial comma) or as "Portugal, Spain and France"(without the serial comma).

Opinions vary among writers and editors on the usage or avoidance of the serial comma. In American English, the serial comma is standard usage in non-journalistic writing that follows the CHICAGO MANUAL OF STYLE. Journalists, however, usually follow the AP Stylebook, which advises against it. It is used less often in British English, where it is standard usage to leave it out, with some notable exceptions such as FOWLER'S MODERN ENGLISH USAGE. In many languages (e.g., French, German, Italian, Polish, Spanish), the serial comma is not the norm

¹⁸contact@mychemistry.eu

and may even go against punctuation rules. It may be recommended in many cases, however, to avoid ambiguity or to aid prosody.

Wikipedia, 2011/11/07

CHEMNUM verwendet das Harvard-Komma bei der Verwendung von American English, nicht aber bei anderen Sprachen. Sie haben jedoch die Möglichkeit, das Harvard-Komma mit dem Schlüssel

<code>list-serial-comma = true/false</code>

Typ: boolean

nach Belieben zu (De-)Aktivieren.

```
1 \selectlanguage{ngerman}DE: \cmpd[list-serial-comma]{r,s,t} vs. \cmpd{r,s,t}
  }\\
2 \selectlanguage{american}US: \cmpd[list-serial-comma=false]{r,s,t} vs. \
  cmpd{r,s,t}
```

DE: 15, 16, und 17 vs. 15, 16 und 17

US: 15, 16 and 17 vs. 15, 16, and 17

Wenn Sie die Spracheinstellung manuell vornehmen, beachten Sie, dass dabei auf die Voreinstellung der Sprache zurückgesetzt wird. Daher sollte `list-serial-comma` *nach* `list-lang` verwendet werden.

```
1 falsch: \cmpdsetup{list-serial-comma=false, list-lang=US}\cmpd{r,s,t}\\
2 richtig: \cmpdsetup{list-lang=US, list-serial-comma=false}\cmpd{r,s,t}
```

falsch: 15, 16, and 17

richtig: 15, 16 and 17

11 Übersicht

11.1 Keys

Im folgenden sind alle verfügbaren Schlüssel aufgelistet.

key	default	type
cmpd-style		macro
cmpd-weight	bold	choice
cmpd-label		literal
cmpd-delim ^a		literal
cmpd-odelim		literal
cmpd-cdelim		literal
cmpd-prefix		literal
cmpd-suffix		literal
cmpd-space	\penalty \@m\	literal
cmpd-sep		literal
cmpd-counter	arabic	choice
cmpd-all	true	boolean
sub-marker	.	literal
sub-counter	alph	choice
sub-list		other ^b
sub-range-sep	-	literal
sub-range-marker	..	literal
sub-only		boolean
list-delim ^a		literal
list-odelim		literal
list-cdelim		literal
list-prefix		literal
list-suffix		literal
list-input-sep	,	literal
list-output-sep	,	literal
list-lang	US	choice
list-last-sep	and ^c	literal
list-serial-comma		boolean
ref-tag	TMP	literal

^a Benötigt zwei Token als Wert.

^b Eingabe ist eine Komma-separierte Liste, siehe Seite 15.

^c Hängt vom Wert von `list-lang` und der Verwendung von `babel` ab.

11.2 Befehle

Im folgenden werden alle von **CHEMNUM** definierten Befehle und deren Varianten aufgelistet.

Befehl	Beschreibung
<code>\cmpd [] {}</code>	Hauptbefehl, erzeugt und schreibt Label oder Liste von Labeln, siehe Seite 5ff.
<code>\cmpd *{} </code>	unsichtbar, erzeugt Label, siehe Seite 5
<code>\cmpd - [] {}</code>	ohne Begrenzer, siehe Seite 20
<code>\cmpd +- [] {}</code>	Label wird aus Hilfsdatei gelesen, siehe Seite 10
<code>\cmpdinit {}</code>	Label initialisieren und erzeugen, siehe Seite 7f.
<code>\cmpdinit *{} </code>	Label initialisieren, siehe Seite 7f.
<code>\cmpdref [] [] {}</code>	temporäre Label in eps-Dateien durch Label ersetzen, siehe Seite 11
<code>\cmpdref - [] [] {}</code>	temporäre Label in eps-Dateien durch Label ohne Begrenzer ersetzen, siehe Seite 20
<code>\cmpdreset []</code>	Zähler zurücksetzen, siehe Seite 9
<code>\cmpdsetup {}</code>	CHEMNUM einstellen, siehe Seite 16ff.

12 Dank

Ich möchte Joseph WRIGHT und Russell HEWITT danken, die mir wertvolle Vorschläge und Feedback gaben, um **CHEMNUM** zu verbessern.