

CHEMNUM

v0.4 2011/12/08

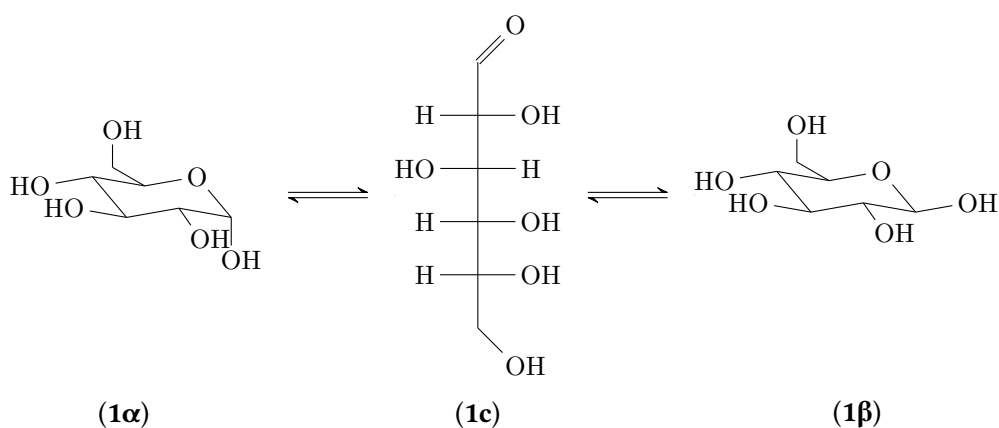
Clemens NIEDERBERGER

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

contact@mychemistry.eu

An approach for the numbering of chemical compounds with $\text{\LaTeX 2}_{\epsilon}$.

deutsche Dokumentation



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	Ein sehr kurzes Readme	3
2	Neues und ein Wort der Entschuldigung	3
3	Paket-Optionen	4
4	Motivation	5
5	Verbindungen Nummerieren	6
5.1	Basisbefehl	6
5.2	Sublabel	7
5.3	Listen	8
5.4	Listen und Bereiche von Sublabeln	8
6	Label zentral initialisieren	10
6.1	Verwendung	10
6.1.1	Haupt-Label	10
6.1.2	Sublabel	11
6.2	Fehler statt Warnung	11
6.3	Personalisierte Eingabe	12
6.4	Warum initialisieren?	12
7	Zurücksetzen des Zählers	12
8	Besondere Verwendung	13
8.1	Abschnitts-Titel	13
8.1.1	Test-Überschrift mit Verbindungslabel (1a)	14
8.1.2	Test-Überschrift mit Verbindungslabel (1b)	14
8.2	Gleitumgebungen	14
8.3	Schemata externer Programme	15
9	Verwendung mit hyperref	17
9.1	Grundprinzip	17
9.2	Gezielter Einsatz	18
10	Personalisieren	19
10.1	Das Prinzip	19
10.2	Schriftstil	20
10.3	Eigene Labels	20
10.4	Eigene Marker in Bildern	21
10.5	Zähler	21
10.6	Begrenzer	21
10.7	Suffix und Präfix	23
10.8	Trennzeichen	24
10.8.1	Bei Labeln – Eingabe	24

10.8.2	Bei Labeln – Ausgabe	25
10.8.3	Bei Sublabeln – Eingabe	25
10.8.4	Bei Sublabeln – Ausgabe	26
10.8.5	Bei Listen – Eingabe	26
10.8.6	Bei Listen – Ausgabe	27
10.8.7	Bei Listen – letzter Trenner	27
10.8.8	Bei <code>\cmpdinit</code> – Eingabe	28
10.8.9	Ein Beispiel	28
11	Sprachunterstützung	29
11.1	Zusammenarbeit mit babel	29
11.2	Harvard-Komma	29
12	Übersicht	30
12.1	Keys	30
12.2	Befehle	31
13	Dank	32
	Index	33

1 Ein sehr kurzes Readme

CHEMNUM v0.4 steht unter der \LaTeX Project Public License Version 1.3 oder später.

(<http://www.latex-project.org/lppl.txt>)

CHEMNUM ruft intern die Pakete `expl3` und `xparse`¹ auf. `xparse` ist Teil des `l3packages`²-Bundles, `expl3` ist Teil des `l3kernel`³-Bundles. Das heißt, **CHEMNUM** benötigt \LaTeX 3-Unterstützung.

CHEMNUM benötigt außerdem die Pakete `etoolbox`⁴ und `psfrag`⁵ und entweder das Paket `textgreek`⁶ oder das Paket `bm`⁷, abhängig von den Paketoptionen (siehe Abschnitt 3).

In diesem Dokument können Sie leicht zwischen einem `\befehl`, einem `key` und sonstigem code unterscheiden.

2 Neues und ein Wort der Entschuldigung

Mit dem Update auf v0.4 hat sich einiges bei den Keys getan. Es gibt ein paar neue, ein paar von den alten haben andere Namen. Sie haben einheitlichere Namen erhalten, die meiner Erachtens intuitiver ihre jeweilige Funktion widerspiegeln. Dadurch kann es sein, dass Ihre mit einer früheren Version von **CHEMNUM** erstellten Dokumente nicht mehr reibungslos kompilieren. Dafür möchte

¹CTAN: `xparse`

²CTAN: `l3packages`

³CTAN: `l3kernel`

⁴CTAN: `etoolbox`

⁵CTAN: `psfrag`

⁶CTAN: `textgreek`

⁷CTAN: `bm`

ich mich entschuldigen. **CHEMNUM** befindet sich noch in der Entwicklungsphase. Ist erst einmal v1.0 erreicht, wird es solche Änderungen nicht mehr⁸ geben.

Geändert haben sich folgende Keys:

alter Name	neuer Name
<code>cmpd-sep</code>	<code>sub-output-sep</code>
<code>sub-marker</code>	<code>sub-input-sep</code>
<code>sub-range-sep</code>	<code>subrange-output-sep</code>
<code>sub-range-marker</code>	<code>subrange-input-sep</code>
<code>strict</code>	<code>init-strict</code>
<code>sub-init</code>	<code>init-sub</code>

Neu hinzugekommen sind diese:

Key	Info
<code>sublist-output-sep</code>	Aufzählungszeichen in Labeln wie 1a,b , siehe Seite 27.
<code>sublist-input-sep</code>	Eingabe von Labeln wie 1a,b , siehe Seite 26.
<code>init-input-sep</code>	Eingabe bei <code>\cmpdinit</code> , siehe Seite 28.
<code>hyper-use</code>	Verwendung von Hyperlinks, siehe Seite 17.
<code>hyper-target</code>	
<code>hyper-link</code>	
<code>ref-style</code>	Darstellung von Labels mit <code>\cmpdref</code> , siehe Seiten 15 und 20.

Ebenfalls neu ist die Art und Weise, wie Sublabel-Listen und -Bereiche eingegeben werden. Der Key `sub-list` entfällt deshalb. Näheres siehe Abschnitt 5.4.

Dann gibt es nun die Möglichkeit, klickbare Links zu den Labeln herzustellen, vorausgesetzt, Sie haben `hyperref`⁹ eingebunden, siehe Abschnitt 9.

Diese Dokumentation hat mit v0.4 einen Index bekommen.

3 Paket-Optionen

CHEMNUM hat eine Option.

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

Damit können Sie sowohl einen der drei Stile des `textgreek`-Pakets wählen¹⁰ 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.

⁸oder nur in größten Ausnahmefällen

⁹CTAN: `hyperref`

¹⁰Zu Details siehe Dokumentation des `textgreek`-Pakets.

4 Motivation

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

Das erste, chemcono¹¹, 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, chemcompounds¹², 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 **2a** 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 bpchem¹³, 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.

¹¹CTAN: chemcono

¹²CTAN: chemcompounds

¹³CTAN: bpchem

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.

5 Verbindungen Nummerieren

5.1 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 2 und 3 werden deklariert und können nun jederzeit wieder aufgerufen werden: 2.
Es ist keine vorherige Deklaration nötig. Die Substanzen wie 4 werden in der Reihenfolge ihres ersten Erscheinens im Text nummeriert.
Noch einmal: 3, 2, 4.

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 6 (S. 10ff).**

```
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 6 führt die Zählung einfach weiter. Mit 5 kann natürlich auf das erzeugte Label zurückgegriffen werden.

Als Label-Namen können Sie im Prinzip einsetzen, was Sie wollen, außer den als Trennzeichen verwendeten Symbolen (siehe Abschnitt 5.2, Abschnitt 5.3, Abschnitt 5.4 und Abschnitt 10.8). 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.

```
1 \cmpd{aa}, \cmpd{a_a}, \cmpd{a_a}, \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}.
```

7, 7, 7, 7, 7 und 7 erzeugen alle das gleiche Label. So, wie auch 7, 7, 7 und 7.

5.2 Sublabel

Möchte man Label der Form **2a** 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. An Stelle des Punktes können Sie auch ein anderes Zeichen verwenden, siehe Abschnitt 10.8.1.

```
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}.
```

1a und **1b** sind eng verwandt, genauso **8a** und **8b**. Natürlich bleibt auch hier bei späterem Aufruf das Label gleich: **8b** und **1a**.

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

```
1 \cmpd{a} und seine Varianten          2 und seine Varianten 2a und 2b
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}                1 und 8
```

Wie Sie ein Label der Art **1a,b** erzeugen können, wird in Abschnitt 5.4 erklärt.

5.3 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 denn versteckt in geschweiften Klammern. Sie können das ändern, siehe Abschnitt 10.8.5 (S. 26).

```
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 **2, 3, 4, 6** und **8b** ausgegeben.

5.4 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**.

Das Prinzip ist recht intuitiv – Sie geben einfach eine Liste von Sublabeln ein:

```
1 Liste von Labeln: \cmpd{q.one, q.two, q.three, q.four, q.five}\\  
2 Label mit Sublabel-Liste: \cmpd{q.{one,two,three,four,five}}
```

Liste von Labeln: **9a, 9b, 9c, 9d** und **9e**

Label mit Sublabel-Liste: **9a,b,c,d,e**

Da die Subliste in der Voreinstellung ebenfalls mit Komma getrennt eingegeben wird, müssen Sie die Subliste in geschweiften Klammern eingeben.

Sie können auf diese Weise auch Bereiche von Sublabeln darstellen. Als Marker zwischen den Namen der beiden Sublabel werden zwei Punkte `..` verwendet.

1	<code>\cmpd{q.{two..four}}\\</code>	9b–d
2	<code>\cmpd{q.{one,three..five}}\\</code>	9a,c–e
3	<code>\cmpd{q.{one..three,five}}</code>	9a–c,e

Durch die Eingabe stehen Ihnen in der Voreinstellungen für Sublabel-Namen weder das Komma noch die zwei Punkte als Teils des Namens zur Verfügung. Wie Sie andere Symbole als Marker verwenden, damit Sie sie doch verwenden können, steht in den Abschnitten 10.8.3 und 10.8.5. In Abschnitt 10.8.9 wird ein Beispiel gezeigt, wie eine geänderte Eingabe aussehen könnte.

CHEMNUM stellt einen Key (siehe Abschnitt 10 zur näheren Erläuterung) zur Verfügung, mit denen alle Sublabel als Bereich ausgegeben werden können

`compd-all = true/false`

Typ: boolean

1 `\compd[compd-all]{q}` **9a-e**

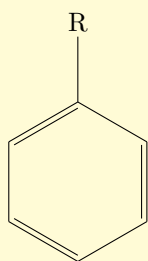
Damit der Key die richtigen Sublabel verwenden kann, müssen diese deklariert worden sein. Daher werden sie aus der Datei `<jobname>.compd` (angenommen Ihre Quelldatei heißt `<jobname>.tex`) gelesen. Das bedeutet, dass Sie zweimal (oder öfter) kompilieren müssen, bis alle Labels richtig dargestellt werden.

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 \compd*{benzol.H,benzol.Me,benzol.OH,benzol.NH2}
3 \chemname{\chemfig{*6(==(-R)=)}}{\compd[compd-all]{benzol}}
4 \begin{tabular}{lll}
5 & & & & \ce{{}-R} & & Name \\
6 \compd[sub-only]{benzol.H} & & \ce{{}-H} & & Benzol \\
7 \compd[sub-only]{benzol.Me} & & \ce{{}-CH3} & & Toluol \\
8 \compd[sub-only]{benzol.OH} & & \ce{{}-OH} & & Phenol \\
9 \compd[sub-only]{benzol.NH2} & & \ce{{}-NH2} & & Anilin
10 \end{tabular}
```



10a-d

	—R	Name
a	—H	Benzol
b	—CH ₃	Toluol
c	—OH	Phenol
d	—NH ₂	Anilin

Bereiche von Sublabeln benötigen eine gewisse Sorgfalt. Man muss erstens genau wissen, welches Label das kleinere und welches das größere ist, und zweitens, welche Label in einem Bereich dazwischen liegen. **CHEMNUM** kann Ihnen diese Sorgfalt nicht abnehmen. Daher kann es passieren, dass Sie Label wie diese erhalten, wenn Sie nicht aufpassen:

```
1 \cmpd{q.{five..three}} or \cmpd{q.{three,one,four}}.
9e-c or 9c,a,d.
```

6 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.

6.1 Verwendung

6.1.1 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 setzt¹⁴. 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.

¹⁴z. B. aufgrund eines Schreibfehlers

```

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

```

1 2a 3 4

```

*****
* 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 20) 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 10).

6.1.2 Sublabel

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

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

verwenden. So werden Sie auch in diesen Fällen gewarnt.

```

1 \cmpdinit[init-sub = true]{E.{e,f,g}}
2 \cmpd{E.e, E.h}

```

1a und 1d

```

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

```

6.2 Fehler statt Warnung

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

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

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

Durch `init-strict = true` wird der L^AT_EX-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[init-strict = true, init-sub = true]{<comma separated list of  
label names>}
```

oder `\cmpdsetup` verwenden, siehe Abschnitt 10.

6.3 Personalisierte Eingabe

Wenn Sie andere Eingabezeichen für `sub-input-sep` (siehe Seite 24) oder `sublist-input-sep` (siehe Seite 26) ausgewählt haben, müssen Sie sie mit `\cmpdinit[sub-init = true]` ebenfalls verwenden.

Die Änderung der Listen-Eingabe `list-input-sep` (siehe Seite 26) wirkt sich auf `\cmpdinit` nicht aus. Wenn Sie die Eingabe dort ebenfalls ändern wollen, verwenden Sie `init-input-sep`, siehe Seite 28.

6.4 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 5.1) und `\cmpd+` (Abschnitt 8.1) überflüssig. Man hat damit direkte Kontrolle, welche Verbindung welche Nummer erhält.

7 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: 2, 1
```

8 Besondere Verwendung

8.1 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 10).
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 6 (S. 10ff).**

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

8.1.1 Test-Überschrift mit Verbindungslabel (1a)

8.1.2 Test-Überschrift mit Verbindungslabel (1b)

8.2 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].

8.3 Schemata externer Programme

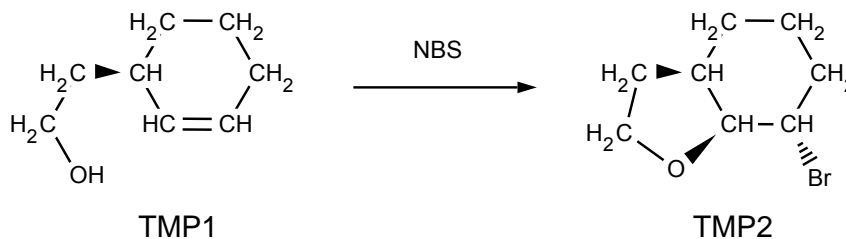
Wenn Sie Ihre Schemata mit externen Programmen wie z. B. CHEMDRAW oder MARVINSKETCH erstellen und dort ebenfalls CHEMNUM einsetzen wollen, dann können Sie das mit

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

tun. Dieser Befehl ist vom Befehl `\schemeref` des Pakets chemscheme¹⁵ inspiriert und funktioniert letztlich gleich. Sie erstellen das Schema und speichern es mit temporären Labeln als eps-Datei.

```
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.

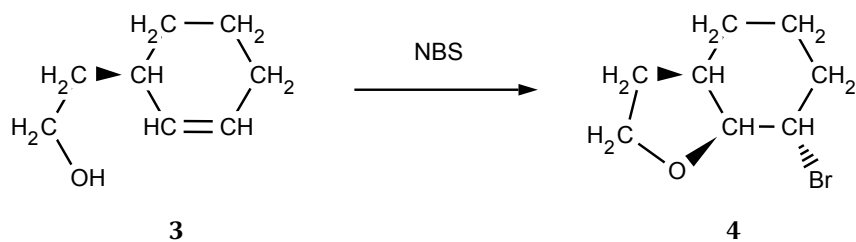
werden dann durch die entsprechenden Label ersetzt. Dafür verwendet CHEMNUM genau wie chemscheme den Befehl `\psfrag` des psfrag-Pakets. Damit er funktioniert, müssen Sie also entweder über LATEX, DVIPS, PS2PDF kompilieren, oder z. B. das Paket auto-pst-pdf¹⁶ zusammen mit PDFLATEX einsetzen. XELATEX wird leider nicht funktionieren, LUALATEX 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 10 vorgestellt werden, können Sie die Label hier auch personalisieren.

¹⁵CTAN: chemscheme

¹⁶CTAN: auto-pst-pdf

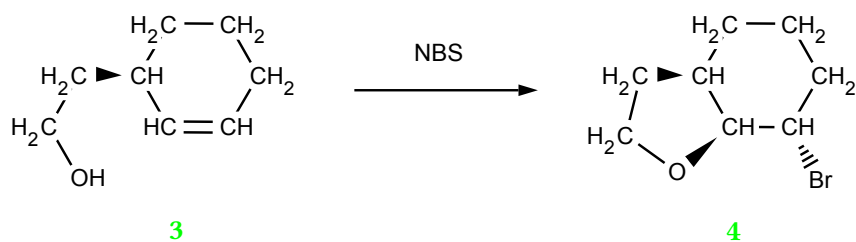


Schema 4: Schema mit automatisierten Labeln.

```

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

```



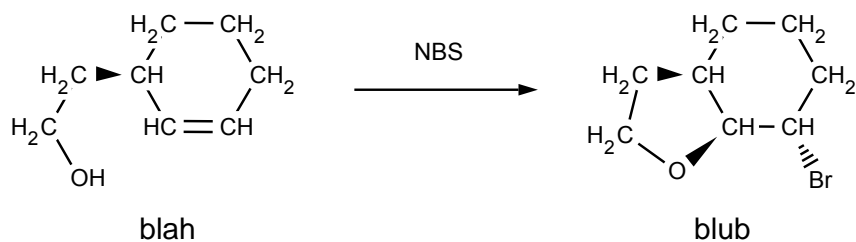
Schema 5: Schema mit automatisierten Labeln.

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

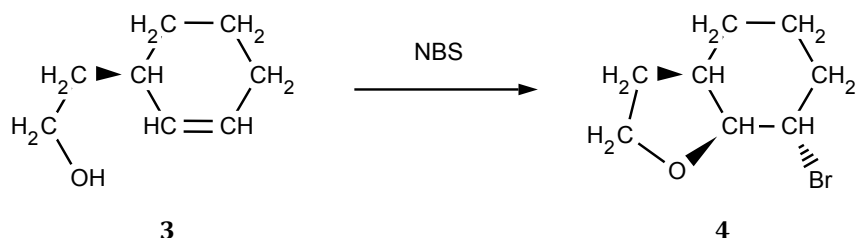
```

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.



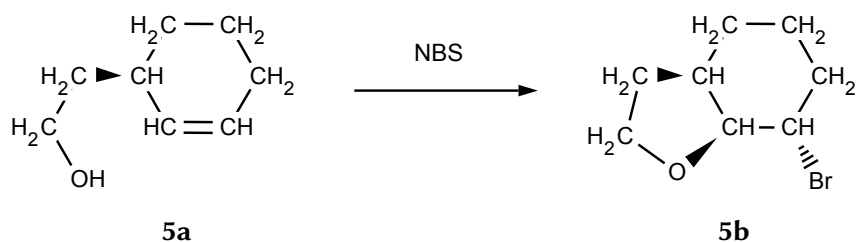
Schema 7: Schema mit explizit gesetzten Labeln.

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.

9 Verwendung mit hyperref

9.1 Grundprinzip

Wenn Sie das Paket hyperref eingebunden haben, können Sie die Label zu klickbaren Links im PDF machen.

`hyper-use` = true/false

Typ: boolean

Wenn Sie diesen Key (siehe Abschnitt 10 für nähere Informationen) auf true setzen, dann wird beim ersten Einsatz eines Labels danach ein Ziel festgelegt und bei jedem weiteren ein Link zu diesem Ziel erzeugt. Das funktioniert allerdings nur, solange das Label *ohne* Liste oder Bereich von Sublabeln gesetzt wird. Auch ein mit `\cmpd*` gesetztes Label kann nicht als Ziel markiert werden.

```
1 \cmpdsetup{hyper-use}
2 Bei der ersten Verwendung wird das Ziel festgelegt: \cmpd{a.two,b,c,d,e,f},
  Bei der n\"achsten dann ein Link gesetzt: \cmpd{a.two,d,e}.
3
4 Das funktioniert nur bei Labeln ohne Sublabel-Liste oder -Bereich: \cmpd{a,
  f.{one,two},c}. Da \cmpd{a.two} und \cmpd{a} zwei unterschiedliche
  Bedeutungen haben, haben sie auch unterschiedliche Linkziele.
```

Bei der ersten Verwendung wird das Ziel festgelegt: **2b**, **3**, **4**, **5**, **6** und **1**, Bei der nächsten dann ein Link gesetzt: **2b**, **5** und **6**.

Das funktioniert nur bei Labeln ohne Sublabel-Liste oder -Bereich: **2**, **1a,b** und **4**. Da **2b** und **2** zwei unterschiedliche Bedeutungen haben, haben sie auch unterschiedliche Linkziele.

9.2 Gezielter Einsatz

Der automatisierte Einsatz ist vielleicht nicht der, den man möchte. Vielleicht reicht es, an ein oder zwei Stellen einen Verweis zu erzeugen. Das kann mit folgenden Keys erfolgen.

`hyper-target` = <link name>

Typ: literal

`hyper-link` = <link name>

Typ: literal

Mit `hyper-target` kann man ein Ziel festsetzen. **Achtung:** wenn Sie diesen Key mit einer Liste einsetzen, dann erhält *jedes* Label den gleichen Linkzielnamen und hyperref wird eine Warnung

destination with the same identifier (name<link name>) has been already used, duplicate ignored

ausgeben.

Mit `hyper-link` kann man nun einen Link auf dieses Ziel erzeugen. **Achtung:** wenn Sie diesen Key mit einer Liste einsetzen, wird *jedes* Label auf das gleiche Ziel verweisen.

```
1 Einen Link setzen \cmpd[hyper-link=mylink,cmpd-all]{f} und das
  entsprechende Ziel dazu nicht vergessen \cmpd[hyper-target=mylink,cmpd-
  all]{f}.
```

Einen Link setzen **1a-4** und das entsprechende Ziel dazu nicht vergessen **1a-4**.

Wenn Sie `hyper-link` einsetzen, ohne das entsprechende `hyper-target` festgelegt zu haben, dann wird `hyperref` eine Warnung

```
name<link name> has been referenced but does not exist, replaced by a
fixed one
```

ausgeben.

10 Personalisieren

10.1 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 10.2.

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 12.1 auf Seite 30 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, sublist, subrange 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.

`hyper` Schlüssel, die mit der Verwendung von `hyperref` zu tun haben.

`init` Schlüssel, die mit der Initialisierung von Labels zu tun haben.

10.2 Schriftstil

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

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

Da es nicht unüblich ist, Schemata serifenlos zu setzen, während der Text mit Serifen gesetzt wird, ist es möglich, für `\cmpdref` einen eigenen Stil festzulegen. In der Voreinstellung werden mit `\cmpdref` erzeugt Label mit `\sffamily` gesetzt.

```
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}
```

2, 3, **1b** und **10**

2, 3, **1b** und **10**

2, 3, **1b** und **10**

2, 3, **1b** und **10**

2, 3, *1b* und *10*

Dass die Schriftstärke separat eingestellt wird, hat seine Ursache bei den griechischen Labels (siehe Abschnitt 10.5). Abhängig von der Paketooption muss die Schriftstärke dort unterschiedlich realisiert werden.

10.3 Eigene Labels

Mit dem Schlüssel

<code>cmpd-label = <name></code>	Typ: literal
--	--------------

kann man eigene Labels vergeben.

```

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

```

Eigenes Label: XYZ, wird bei einer Liste an *alle neuen* vergeben! XYZ, 2 und XYZ

10.4 Eigene Marker in Bildern

Wenn Sie für die temporären Label, die durch `\cmpdref` (siehe Seite 15) nicht TMP verwenden wollen, können Sie das ändern, z. B. in tmp:

```

1   \cmpdsetup{ref-marker = tmp}

```

10.5 Zähler

Wenn Sie mögen, können Sie auch das Zählerformat **2a** ändern.

```

cmpd-counter = <counter>
sub-counter = <counter>

```

Typ: choice

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 Labeln!
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}

```

2, 3, **1b** und **10**

K1, **K2** und **L1**

13 und **14**

13α und **13β**

10.6 Begrenzer

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

```

cmpd-delim = <odelim><cdelim>
list-delim = <odelim><cdelim>
Benötigen zwei Token als Wert!

```

Typ: literal
Typ: literal

```

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

```

2, 3, **1b** und 10
(2), (3), (**1b**) und (10)
2, 3, **1b** und 10
(2), (3), (**1b**) und (10)

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

```

2, 3, **1b** und 10
[2, 3, **1b** und 10]
2, 3, **1b** und 10
[2, 3, **1b** und 10]
2

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 (3), (4), (5a), aber manchmal so 6 oder so 6.

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
```

2, 3, 1b und 10

[(2), (3), (1b) und (10)]

[2, 3, 1b und 10]

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

10.7 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!
```

2, 3, 1b und 10

Nr. 2, Nr. 3, Nr. 1b und Nr. 10

→ 2, → 3, → 1b und → 10

→ 2

Beispiel für die Listen-Attribute:

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

2, 3, **1b** und **10**
Liste: 2, 3, **1b** und **10**
Sammlung: 2, 3, **1b** und **10**
2

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

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

Px 10 Sx
```

Dieser kann durch

<code>cmpd-space = <space></code>	Typ: literal
---	--------------

geändert werden.

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

Px10Sx
```

10.8 Trennzeichen

10.8.1 Bei Labeln – Eingabe

Als Voreinstellung verwendet **CHEMNUM** den Punkt . als Zeichen, um Hauptlabelnamen von Sublabelnamen zu unterscheiden. Sie können das Symbol nach Belieben ändern. Die Zeichen % # können Sie nicht verwenden und @ sollten Sie nicht verwenden. Außerdem sollten Sie darauf achten, für die anderen Trennsymbole (siehe folgende Abschnitte) nicht die gleichen Zeichen zu verwenden.

<code>sub-input-sep = <separator></code>	Typ: literal
--	--------------

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


1a und 8b
1a und 8b
1a und 8b
1a und 8b
1a und 8b

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

10.8.2 Bei Labeln – Ausgabe

Haben Sie Label mit Sublabeln, können Sie dort auch ein Trenn-Symbol verwenden. Das erscheint dann zwischen Hauptlabel und Sublabel.

`sub-output-sep = <separator>`

Typ: literal

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

2, 3, 1b, 13 und 10
2, 3, 1-b, 1-3 und 10
2, 3, 1·b, 1·3 und 10
2, 3, 1:b, 1:3 und 10

10.8.3 Bei Sublabeln – Eingabe

Für die Eingabe eines Bereiches von Sublabeln verwendet `CHEMNUM` in der Voreinstellung zwei Punkte ... Sie können aber eine beliebige andere Eingabe wählen. Die Zeichen % # können Sie nicht verwenden und @ sollten Sie nicht verwenden. Außerdem sollten Sie darauf achten, für die anderen Trennsymbole nicht die gleichen Zeichen zu verwenden.

`subrange-input-sep = <separator>`

Typ: literal

```
1 \cmpd[subrange-input-sep=:]{q.{one:four}}\\
2 \cmpdsetup{subrange-input-sep=-}
3 \cmpd{q.{one--four}}
```

9a-d
9a-d

10.8.4 Bei Sublabeln – Ausgabe

Vielleicht gefällt Ihnen die Ausgabe **9a–e** nicht und Sie wollen den Bereich anders darstellen? Das geht mit diesem Key:

```
subrange-output-sep = <separator>
```

Typ: literal

```
1 \cmpd[subrange-output-sep={ {\normalfont bis} }]{q.{one..four}}\
2 \cmpdsetup[subrange-output-sep=-]
3 \cmpd{q.{one..four}}
```

9a bis d

9a-d

10.8.5 Bei Listen – Eingabe

Wenn Sie mögen, können Sie das Trennsymbol der Listen-Eingabe ändern. Per Default wird hier das Komma , verwendet. Für die Eingabe eines Bereiches von Sublabeln verwendet **CHEMNUM** in der Voreinstellung zwei Punkte ... Sie können aber eine beliebige andere Eingabe wählen. Die Zeichen % # können Sie nicht verwenden und @ sollten Sie nicht verwenden. Außerdem sollten Sie darauf achten, für die anderen Trennsymbole nicht die gleichen Zeichen zu verwenden.

```
list-input-sep = <separator>
```

Typ: literal

```
sub-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}
```

2, 3, 1b und 10

2, 3, 1b und 10

2, 3, 1b und 10

2, 3, 1b und 10

Das ermöglicht Ihnen z. B., IUPAC-Namen als Label-Namen zu verwenden¹⁷.

```
1 \cmpdsetup[list-input-sep=;]
2 \cmpd{1,3-dichlorocyclohexanol}
```

14

¹⁷Vielen Dank an Christina Lüdigg, die dieses Feature in gewisser Weise vorgeschlagen hat.

10.8.6 Bei Listen – Ausgabe

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

<code>list-output-sep = <separator></code>	Typ: literal
<code>sub-output-sep = <separator></code>	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}
```

2, 3, **1b** und **10**
2 3 **1b** und **10**
2, 3, **1b** und **10**
2; 3; **1b** und **10**

10.8.7 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}
```

2, 3, **1b**, and **10** 2 and 3
2, 3, **1b** and **10** 2 and 3
2, 3, **1b** und **10** 2 und 3
2, 3, **1b** et **10** 2 et 3
2, 3, **1b** y **10** 2 y 3
2, 3, **1b** e **10** 2 e 3

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 11.

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}
```

2, 3, **1b** und auch **10**

2, 3, **1b** **10**

2, 3, **1b** sowie **10**

2, 3, **1b**, **10**

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}
```

2, 3, **1b**, **10**

2; 3; **1b**; **10**

2 und 3 und **1b** und **10**

10.8.8 Bei `\cmpdinit` – Eingabe

Wenn Sie mögen, können Sie das Trennsymbol der Listen-Eingabe ändern. Per Default wird hier das Komma , verwendet. Sie können aber eine beliebige andere Eingabe wählen. Die Zeichen % # können Sie nicht verwenden und @ sollten Sie nicht verwenden.

`init-input-sep = <separator>`

Typ: literal

10.8.9 Ein Beispiel

Im folgenden sehen Sie ein kleines Beispiel, wie eine personalisierte Variante der Trennzeichen aussehen könnte.

```
1 \cmpdsetup{
2   sub-input-sep      = ! ,
3   subrange-input-sep = : ,
4   sublist-input-sep  = / ,
5   list-input-sep     = +
6 }
7 \cmpd{a + f ! one + q ! one / three : five}
```

2, **1a** und **9a,c-e**

11 Sprachunterstützung

11.1 Zusammenarbeit mit babel

CHEMNUM unterstützt die Sprachauswahl durch das babel¹⁸-Paket für eine Reihe von Sprachen. Dies geschieht mit Hilfe des translator Pakets, falls es vorhanden ist. Kann translator nicht geladen werden, muss die Sprachunterstützung von Hand vorgenommen werden, siehe Abschnitt 10.8.7. 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-Mail¹⁹ mit der dazugehörigen babel Option und der richtigen Übersetzung, und ich werde sie gerne hinzufügen.

11.2 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).

¹⁸CTAN: babel

¹⁹contact@mychemistry.eu

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 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

```
list-serial-comma = true/false
```

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

12 Übersicht

12.1 Keys

Im folgenden sind alle verfügbaren Schlüssel aufgelistet.

Key	Leerwert	Voreinstellung	Typ
<code>cmpd-style</code>			macro
<code>cmpd-weight</code>	bold	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\	\penalty \@m\	literal
cmpd-counter	arabic	arabic	choice
cmpd-all	true	false	boolean
sub-only	true	false	boolean
sub-counter	alph	alph	choice
sub-output-sep		,	literal
sub-input-sep	.	.	literal
sublist-output-sep	,	,	literal
sublist-input-sep	,	,	literal
subrange-output-sep	-	-	literal
subrange-input-sep	literal
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	US ^b	choice
list-last-sep	and ^c		literal
list-serial-comma	true	true ^c	boolean
ref-tag	TMP	TMP	literal
ref-style		\sffamily	macro
init-strict	true	false	boolean
init-sub	true	false	boolean
init-input-sep	,	,	literal
hyper-use	true	false	boolean
hyper-target			literal
hyper-link			literal

^a Benötigt zwei Token als Wert.

^b Hängt von der Verwendung von babel ab.

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

12.2 Befehle

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

Befehl	Beschreibung
<code>\cmpd</code> [<keyval>] {<labels>}	Hauptbefehl, erzeugt und schreibt Label oder Liste von Labels, siehe Seite 6ff.
<code>\cmpd*</code> {<labels>}	unsichtbar, erzeugt Label, siehe Seite 6
<code>\cmpd-</code> [<keyval>] {<labels>}	ohne Begrenzer, siehe Seite 22
<code>\cmpd+-</code> [<keyval>] {<labels>}	Label wird aus Hilfsdatei gelesen, siehe Seite 13
<code>\cmpdinit</code> [<keyval>] {<labels>}	Label initialisieren und erzeugen, siehe Seite 10f.
<code>\cmpdinit*</code> [<keyval>] {<labels>}	Label initialisieren, siehe Seite 10f.
<code>\cmpdref</code> [<keyval>] [<marker>] {}	temporäre Label in eps-Dateien durch Label ersetzen, siehe Seite 15
<code>\cmpdref-</code> [<keyval>] [<marker>] {}	temporäre Label in eps-Dateien durch Label ohne Begrenzer ersetzen, siehe Seite 22
<code>\cmpdreset</code> [<num>]	Zähler zurücksetzen, siehe Seite 12
<code>\cmpdsetup</code> {<keyval>}	CHEMNUM einstellen, siehe Seite 19ff.

13 Dank

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

Index

Abschnittsüberschriften werden **fett**, Pakete serifenlos, Befehle **braun** und Keys **grün** dargestellt.

A

auto-pst-pdf 15

B

babel 29, 31

Basisbefehl 6 f.

Begrenzer 21 ff.

bm 3

bpchem 5

C

chemcompounds 5 f., 10

chemcono 5

chemscheme 15

\cmpd 6 ff., 13 f., 19, 22, 32

cmpd-all 9, 31

cmpd-cdelim 31

cmpd-counter 21, 31

cmpd-delim 22 f., 31

cmpd-label 20, 31

cmpd-odelim 31

cmpd-prefix 23, 31

cmpd-sep 4

cmpd-space 24, 31

cmpd-style 20, 30

cmpd-suffix 23, 31

cmpd-weight 20, 30

\cmpd* 6, 12, 14, 18, 32

\cmpdinit* 11

\cmpd+ 12 ff.

\cmpd- 22, 32

\cmpd+- 32

\cmpdinit 4, 6, 10–14, 32

\cmpdinit* 10, 12, 32

\cmpdref 4, 15, 20 f., 32

\cmpdref- 22, 32

\cmpdreset 12 f., 32

\cmpdsetup 12, 19, 23, 32

E

Eigene Labels 20 f.

Eigene Marker 21

etoolbox 3

expl3 3

H

Harvard-Komma 29 f.

hyper-link 4, 18 f., 31

hyper-target 4, 18 f., 31

hyper-use 4, 18, 31

hyperref 4, 17–20

I

init-input-sep 4, 12, 28, 31

init-strict 4, 12, 31

init-sub 4, 11 f., 31

L

l3kernel 3

l3packages 3

Label initialisieren 10 ff.

list-cdelim 31

list-delim 22 f., 31

list-input-sep 12, 26, 31

list-lang 27, 30 f.

list-last-sep 27 f., 31

list-odelim 31

list-output-sep 27 f., 31

list-prefix 23, 31

list-serial-comma 30 f.

list-suffix 23, 31

Listen 8

P

Paket-Optionen 4

psfrag 3, 15

R

ref-style 4, 20, 31

ref-tag 31

S

Schemata externer Programme 15 ff.

Schriftstil 20

Sprachunterstützung	29 f.
<code>strict</code>	4
<code>sub-counter</code>	21, 31
<code>sub-init</code>	4, 12
<code>sub-input-sep</code>	4, 12, 24, 26, 31
<code>sub-list</code>	4
<code>sub-marker</code>	4
<code>sub-only</code>	9, 31
<code>sub-output-sep</code>	4, 25, 27, 31
<code>sub-range-marker</code>	4
<code>sub-range-sep</code>	4
Sublabel	
Bereiche	8
Listen	8
Sublabel	7
<code>sublist-input-sep</code>	4, 12, 31
<code>sublist-output-sep</code>	4, 31
<code>subrange-input-sep</code>	4, 25, 31
<code>subrange-output-sep</code>	4, 26, 31
Suffix und Präfix	23 f.
T	
<code>textgreek</code>	3 f.
<code>translator</code>	29
Trennzeichen	
<code>\cmpdinit</code>	28
Label	
Input	24
Output	25
Listen	
Input	26
letzter Trenner	27
Output	27
Sublabel	
Input	25
Output	26
Trennzeichen	24–28
V	
Verwendung mit hyperref	17 ff.
X	
<code>xparse</code>	3
Z	
Zähler	21
Zurücksetzen des Zählers	12 f.