The bpchem package*

Bjørn Pedersen
Bjoern.Pedersen@frm2.tum.de

2004/11/25

1 Introduction

This package has been written to alleviate the task of writing publications containing lots of chemistry. It provides methods for typesetting chemical names, sum formulae and isotopes. It provides the possibility to break very long names even over several lines.

This package also provides a way to automatically enumerate your chemical compounds, allowing for one-level subgrouping.

What this package does not provide: Methods to draw chemical compounds. Although there exist some packages,which where designed for this purpose (e.g. xymtex, PPChTex) they are quite limited once you get to complex organic, or metal organic compounds. I recommend using an external drawing program, possibly in conjunction with psfrag, in these cases.

2 Package options

Currently this package supports only one option:

`cbgreek`

this option causes the definitions of some macros to be changed to use the cbgreek fonts. As they are not available on all systems, and only in mf format, the default is to use the math fonts for greek symbols.

3 User commands in this package

3.1 Setting chemical sum formulae: BPChem<chemical formula>

`\BPChem` Within this macro you can use `\_` and `\^` for correct chemical sub- and superscripts. Example:

```
\BPChem{C\_2H\_5OH} or \BPChem{SO\_4\^{2-}}
```

C_2H_5OH or SO_4^{2-}

*This file has version number v1.06, last revised 2004/11/25.

3.2 Setting long chemical names: IUPAC<formula or name>

\IUPAC

in addition to sub/superscripts as above, `\-` is a hyphen which allows further breakpoints,`\|` is an (invisible) Multibreakpoint.

This environment is especially useful for your long IUPAC-compound names.

Example:

```
 %\IUPAC{Tetra\|cyclo[2.2.2.1\^{1,4}]\-
un\|decane-2\-dodecyl\-5\-(hepta\|decyl\|iso\|dodecyl\|thio\|ester)}
```

Tetracyclo[2.2.2.1^{1,4}]-undecane-2-dodecyl-5-(heptadecylisododecyl-thioester)

3.3 Enumerating and referencing chemical compounds: CNlabel{<label>}, CNlabelnoref{<label>}, CNref{<label>}

\CNlabel
\CNlabelnoref
\CNref

`CNlabel` defines and use `#1` (via `ref`) as label for numbering of chemical compounds. If the label has not yet been defined, it is created, otherwise it is just referenced. if you just want to define the label, use `\CNlabelnoref` instead.

If you want to get just the reference, use `\CNref`. This comes handy for figure captions or section titles, as you would get dissorder in the numbering due to the moving argument otherwise.

The default style is: `\textbf{\arabic{\counter}}`

To change, use something like

```
\renewcommand{\theBPCno}{\textbf{\arabic{BPCno}}}
```

Example:

```
Alkohol \CNlabel{al} is converted to aldehyd \CNlabel{ad}. \CNref{al}
can also be used otherwise, while \CNref{ad} cannot.
```

Alkohol **1** is converted to aldehyd **2**. **1** can also be used otherwise, while **2** cannot.

3.4 Using sub-labels for classes of compounds: CNlabelsub{<label>}{<sublabel>}, CNlabelsubnoref{<label>}{<sublabel>}, CNrefsub{<label>}{<sublabel>}

\CNlabelsub
\CNlabelsubnoref
\CNrefsub

These commands are the same as above, with additional sub identifier `#2` added. If the primary identifier is not yet used, it will be created and can also be referenced via the normal commands.

The default style is:`\textbf{\arabic{BPCno}\alph{BPCnoa}}}`

To change, use something like

```
\renewcommand{\theBPCnoa}{\textbf{\arabic{BPCno}\alph{BPCnoa}}}
```

```
To demonstrate the use of sublabels, methanol \CNlabelsub{alk}{a} and
ethanol \CNlabelsub{alk}{b} are both natural products. The acohols \CNref{alk}
can synthezied bio-chemically. \CNrebsub{alk}{a} is toxic, while
\CNrefsub{alk}{b} is only mildly toxic.
```

To demonstrate the use of sublabels, methanol
3a and ethanol **3b** are both natural products.
The alcohols **3** can synthezied bio-chemically.
3a is toxic, while **3b** is only mildly toxic.

3.5 Shortcuts for common idioms in chemical literature

^{1}H-NMR: δ `\HNMR`

^{13}C-NMR: δ `\CNMR`

cis `\cis`

trans `\trans`

α `\bpalpha`

β `\bpbeta`

Δ `\bpdelta`

$\eta^{<nummer>}$ `\hapto{<number>}`

Note: Some of these macros are influenced by the `cbgreek` option! Use is only recommended with the `\BPChem` and `\IUPAC` commands. Some will not even work outside those commands.

4 Example

```
\begin{minipage}[b]{15em}
some normal text and math: $A*2=B$

Test \BPChem{ C\_{2}H\_{4}\^{+}}
or using math in superscript  \BPChem{ C\_{2}H\_{4}\^{$+$}}

\BPChem{Example\_{longer subscript}\^{superscript}}

Isotope: \BPChem{\_{A}\^{B}X\^{C}\_{D}}

\IUPAC{Tetra\|cyclo[2.2.2.1\^{1,4}]\-^^A
un\|decane-2\-dodecyl-5-(heptadecyl\|iso\|dodecyl\|thio\|ester)}
\end{minipage}
```

and the resulting output:

some normal text and math: $A*2=B$, just to show it.
Test $C_2H_4^+$ or using math in superscript $C_2H_4^+$
Example$^{\text{superscript}}_{\text{longer subscript}}$ And normal Text again
Isotope: ${}^{\mathrm{B}}_{\mathrm{A}}\mathrm{X}^{\mathrm{C}}_{\mathrm{D}}$
Tetracyclo[2.2.2.1^{1,4}]-undecane-2-dodecyl-5-(heptadecylisododecyl-thioester)

5 The code

<*bpchem>˙ first comes some option setup

```
\newif\ifusecbgreek%
\usecbgreekfalse%
\DeclareOption{cbgreek}{\PackageInfo{bpchem}{cbgreek selected}\usecbgreektrue}
\ProcessOptions\relax
```

`\textsubscript` Define a textsubscript corresponing to textsuperscript. This is now also available as the package textsubscript by D.Arsenau or as part of KOMA-Script2 by M. Kohm.

```
\providecommand*\textsubscript[1]{%
  \@textsubscript{\selectfont#1}}
\def\@textsubscript#1{%
  {\m@th\ensuremath{_{\mbox{\fontsize\sf@size\z@#1}}}}}
```

a register to save the length to backspace two registers needed to get back to correct working position if one is longer than the other.

```
  \newlength{\BPClensub}
  \newlength{\BPClensuper}
  \newlength{\BPCdelta}
%
```

we are in subscript and maybe the superscript was longer

```
  \DeclareRobustCommand{\BPCadjustsub}{%
    \setlength\BPCdelta{\BPClensuper}\addtolength\BPCdelta{-\BPClensub}%
    \ifdim\BPCdelta>0pt{\kern\BPCdelta}\else\relax\fi%
    \setlength{\BPClensub}{0pt}% reset
    \setlength{\BPClensuper}{0pt}% reset
    }%
%
```

we are in superscript and maybe the subscript was longer

```
  \DeclareRobustCommand{\BPCadjustsuper}{%
    \setlength\BPCdelta{\BPClensub}\addtolength\BPCdelta{-\BPClensuper}%
    \ifdim\BPCdelta>0pt{\kern\BPCdelta}\else\relax\fi%
    \setlength{\BPClensub}{0pt}% reset
    \setlength{\BPClensuper}{0pt}% reset
    }%
%
```

make a subscript and remember length in BPClen

```
  \DeclareRobustCommand{\BPCsub}[1]{%
    \ifmmode_{#1}\settowidth\BPClensub{_{#1}}%
    \else\textsubscript{#1}\settowidth\BPClensub{\textsubscript{#1}}\fi%
    \futurelet\next\lookforsuper%
    }%
```

make a superscript and remember length in BPClen raise by 0.15 em, else e.g. + collides with subscript

```
  \DeclareRobustCommand{\BPCsuper}[1]{%
    \ifmmode^{#1}\settowidth\BPClensuper{^{#1}}%
    \else\raisebox{0.15em}{\textsuperscript{#1}}%
    \settowidth\BPClensuper{\textsuperscript{#1}}\fi%
    \futurelet\next\lookforsub%
    }%
```

see if next token is BPCsuper,

```
  \DeclareRobustCommand\lookforsuper{%
    \ifx\next\BPCsuper\let\next=\BPCsuperbs%
    \else\let\next=\BPCadjustsub\fi\next%
    }%
```

see if next token is BPCsub

```
  \DeclareRobustCommand\lookforsub{%
    \ifx\next\BPCsub\let\next=\BPCsubbs%
    \else\let\next=\BPCadjustsuper\fi\next%
    }%
%
```

backspace BPClen and make superscript eats the old \^

```
  \DeclareRobustCommand{\BPCsuperbs}[1]{\kern-\BPClensub\BPCsuper}%
%
```

backspace and make subscript eats the old _

```
  \DeclareRobustCommand{\BPCsubbs}[1]{\kern-\BPClensuper\BPCsub}%
%
```

needed to get catcodes right

```
  \DeclareRobustCommand{\DoBPChem}{}%
  \def\DoBPChem#1{%
    #1\endgroup%
    }%
\DeclareRobustCommand{\BPCSetupCat}{}
\def\BPCSetupCat{%
  %\catcode`^=\active%
  %\catcode`\_=\active%
  \BPCSetup%
}%
%
\DeclareRobustCommand{\BPCSetup}{}
\def\BPCSetup{%
%
 \let\_=\BPCsub%
 \let\^=\BPCsuper%
}%end BPCSetup
```

setup for chemical formula

```
\DeclareRobustCommand\BPChem{%
  \begingroup% endgroup in DoBPChem
  \BPCSetupCat%
  \DoBPChem%
  }
```

these are taken from german.sty and allow more than one break or breaks and hyphens in a word. Very useful for chemical names, as they tend to grow rather long. Two short versions are also defined

```
\DeclareRobustCommand{\allowhyphens}{\penalty\@M \hskip\z@skip}
\DeclareRobustCommand{\BreakHyph}{\penalty\@M -\allowhyphens}
\DeclareRobustCommand{\MultiBreak}%
                {\penalty\@M\discretionary{-}{}{\kern.03em}%
                \allowhyphens}
\let\MB=\MultiBreak \let\BH=\BreakHyph
```

```
\DeclareRobustCommand{\DoIUPAC}[1]{%
#1\endgroup}
\def\Prep{%
\let\-=\BreakHyph%
\let\|=\MultiBreak%
\DoIUPAC%
}
\DeclareRobustCommand*{\IUPAC}{%
  \begingroup\BPCSetup\ignorespaces%
  \Prep}%
```

Trick by David Kastrup <David.Kastrup@t-online.de> to make non-fragile. Otherwise `\|` would become `\delimiter"026B30D` in e.g. the toc

```
\expandafter\DeclareRobustCommand\expandafter\|\expandafter{\|}
\expandafter\DeclareRobustCommand\expandafter\-\expandafter{\-}
```

counters for numbering of chemical substances

```
\newcounter{BPCno}
\renewcommand{\theBPCno}{\textbf{\arabic{BPCno}}}

\newcounter{BPCnoa}[BPCno]
\renewcommand{\theBPCnoa}{\textbf{\arabic{BPCno}\alph{BPCnoa}}}
```

helper functions to mark first definition

```
\newcommand{\newchems@b}[2]{
\expandafter\gdef\csname cna@#1#2\endcsname{#2}%
}
```

reference a CNlabel (useful for section titles, captions etc.)

```
\DeclareRobustCommand*{\CNref}[1]{%
\ref{cn:#1}%
}
```

reference a CNlabel/sublabel

```
\DeclareRobustCommand*{\CNrefsub}[2]{%
\ref{cn:#1#2}
%%\textbf{\csname cna@#1#2\endcsname}%
}
```

label a substance and insert the number

```
\DeclareRobustCommand*{\CNlabel}[1]{%
\CNlabelnoref{#1}%
\CNref{#1}%
}

\DeclareRobustCommand*{\CNlabelnoref}[1]{%
\expandafter\ifx\csname cnd@#1\endcsname\relax%
{\refstepcounter{BPCno}\label{cn:#1}}%
\expandafter\gdef\csname cnd@#1\endcsname{x}%
\fi%
}

\DeclareRobustCommand*{\CNlabelsub}[2]{%
\CNlabelsubnoref{#1}{#2}%
\CNrefsub{#1}{#2}%
}
\DeclareRobustCommand*{\CNlabelsubnoref}[2]{%
```

```
\CNlabelnoref{#1}%
\expandafter\ifx\csname cna@#1#2\endcsname\relax%
{\refstepcounter{BPCnoa}\label{cn:#1#2}}%
\expandafter\gdef\csname cna@#1#2\endcsname{x}%
%% \newchems@b{#1}{#2}%
%%   \write\@auxout{\string\newchems@b{#1}{#2}}%
  \fi%
}
```

more helper mcors special symbols and macros for math-symbols without math-mode

```
\DeclareRobustCommand{\HNMR}{\IUPAC{\^{1}H-NMR}: $\delta$\xspace}
\DeclareRobustCommand{\CNMR}{\IUPAC{\^{13}C-NMR}: $\delta$\xspace}
\DeclareRobustCommand{\cis}{\textit{cis}\xspace}
\DeclareRobustCommand{\trans}{\textit{trans}\xspace}
%\DeclareRobustCommand{\R}{\textit{R}}
%\DeclareRobustCommand{\S}{\textit{S}}
%%%%%%%%%%%%%%%%%%%%
%
\ifusecbgreek% code with roman greek
\PackageInfo{bpchem}{using upright greek fonts from cbgreek}
\input{lgrenc.def}
\DeclareRobustCommand{\rm@greekletter}[1]{{\fontencoding{LGR}\selectfont%
     \def\encodingdefault{LGR}#1}}%
% some examples
\DeclareRobustCommand{\bpalpha}{\rm@greekletter{a}}
\DeclareRobustCommand{\bpbeta}{\rm@greekletter{b}}
\DeclareRobustCommand{\bpDelta}{\rm@greekletter{D}}
\DeclareRobustCommand{\hapto}[1]{\rm@greekletter{h}\^{#1}}
\else
% code with standard math greek
\PackageInfo{bpchem}{using default math greek fonts}
\DeclareRobustCommand{\bpalpha}{\ensuremath{\alpha}\xspace}
\DeclareRobustCommand{\bpbeta}{\ensuremath{\beta}\xspace}
\DeclareRobustCommand{\bpDelta}{\ensuremath{\Delta}\xspace}
\DeclareRobustCommand{\hapto}[1]{\ensuremath{\eta^{#1}}}
\fi
\let\talpha\bpalpha
\let\tbeta\bpbeta
%%%%%%
\DeclareRobustCommand*{\dreh}[1]%
{$\lbrack \alpha \rbrack _{\mathrm D}^{#1}$}
```

</bpchem>