Preliminary documentation for `gb4e.sty`*and `cgloss4e.sty`

hans-peter kolb <kolb@sfs.nphil.uni-tuebingen.de>
Craig Thiersch <thiersch@kub.nl>

Contents

Important note:

The file `gb4e.sty`[1] allows _ (subscript) and ^ (superscript) to be used in ordinary text, which is handy, but because it changes their definitions it must be loaded **after** any file that uses them in their TeX meaning.

---

*Parts of "Government-Binding style" `gb4e.sty` are loosely based on ideas and code by M. Covington, M. deGroot, J. Frampton and L. Holt. In particular it automatically includes `cgloss4e.sty`, a (heavily adapted) version of M. Covington's LaTeX-adaptation of M. de Groot's plainTeX Midnight gloss macros.

[1]This version of `gbx.sty` redesigns the entire series of \ex commands. It also contains changes to the gloss-macros to make them work in footnotes etc. **NB: The e*xport* version does not provide a soft-landing system of command-compatibility with earlier versions**.

NB: Make sure to get `cgloss4e.sty` together with `gb4e.sty`.

As time permits, we'd like to revise the rest of `gb4e.sty`, so please experiment and let us know about bugs and/or suggestions.

1 Linguistic examples in LaTeX

`gb4e.sty` is a style option to be used with LaTeX's major document styles. To make its commands available, add it to your document's preamble like this:

(in LaTex2e:) `\usepackage{gb4e}`

(in LaTex 2.09:) `\documentstyle[...gb4e]{...}`

Its core is an environment for the formatting of linguistic examples.

The format is simple:
(Batches of) examples are introduced by `\begin{exe}` which sets up the top level example environment. Obviously, this environment has to be closed after the example(s) by `\end{exe}`. You can create sub- and subsubexamples by embedding several `xlist`-environments (up to three levels deep).

Here is an elaborate example:

```
\begin{exe}                       % sets up the top-level example environment
\ex\label{here} Here is one.  % example with running number
\ex[*]{Here another is.}      % judged ex. with running number
\ex Here are some with judgements.
 \begin{xlist}                    % first embedding (alphabetical numbering)
 \ex[]{A grammatical sentence am I.}
 \ex[*]{An ungrammatical sentence is you.}
 \ex[??]{A dubious sentence is she.}
 \ex                               % just the number
  \begin{xlist}                   % second embedding (roman numbering)
  \ex[**]{Need one a second embedding?}
  \ex[\%]{sometime.}
  \end{xlist}                    % end second embedding
 \ex Dare to judge me!
 \end{xlist}                     % end first embedding
\ex This concludes our demonstration.
\end{exe}                        % end example environment
```

This produces:

(1)   Here is one.

(2)   * Here another is.

(3)   Here are some with judgements.
      a.   A grammatical sentence am I.
      b.   * An ungrammatical sentence is you.

    c. ?? A dubious sentence is she.

    d.   i. ** Need one a second embedding?

        ii. % sometime.

    e. Dare to judge me!

(4)   This concludes our demonstration.

All example-list commands have an optional argument which allows you to set the labelwidth to, e.g., the widest label to come (in the style of `\begin{bibliography}`). So, `\begin{exe}[(234)]` will set the labelwidth to the width of "`(234)`"—which is also the default. The other defaults are "`m.`" for `xlist`, and "`iii.`" for `xlisti`. The default labelwidth for the `exe`-environment can also be changed globally by issuing a `\exewidth{<string>}` command in the preamble of your document (i.e. between the `documentstyle` and the `\begin{document}` lines).

1.1   Examples with running numbers

As just shown, every example with a running number is introduced by `\ex`. Obviously, `\ex` is *context-sensitive*: at the top level the number produced is *arabic* and accumulative throughout the document; in embedded `xlist`-environments it is *alphabetic* on the first level, *roman* on the second, and *arabic* on the third, starting with *a.*, *i.*, and 1., repectively. For total control over the shape of your subexample-numbering there are some special forms listed below:[2]

Environment	Default numbering
`\begin{exe}` … `\end{exe}`	(1) arabic (Toplevel)
`\begin{xlist}` … `\end{xlist}`	depends on level of embedding
`\begin{xlista}` … `\end{xlista}`	a. alphabetical
`\begin{xlisti}` … `\end{xlisti}`	ii. roman
`\begin{xlistn}` … `\end{xlistn}`	3. arabic
`\begin{xlistI}` … `\end{xlistI}`	IV. Roman
`\begin{xlistA}` … `\end{xlistA}`	E. Alphabetical

In all cases, `\label`/`\ref` commands will do The Right Thing.

The syntax of `\ex` is as follows:

---

[2]Note that for all `\begin{<env>}` … `\end{<env>}` pairs the forms `\<env>` … `\end<env>` are also permissible in LATEX. So if you like you can go on using `\xlist` … `\endxlist` for `\begin{xlist}` … `\end{xlist}` etc., but LATEX-syntaxcheckers and utilities like AUC-TEX won't recognize them as necessarily paired and won't report unmatched expressions. (Have you ever tried `<Ctrl>-C` `<Ctrl>-C` (= LaTeX-environment) or `<Ctrl>-C` `<Ctrl>-F` (= LaTeX-close-environment) in EMACS' AUC-TEX mode? You should—it saves a lot of typing.)

> `\ex` $\Longrightarrow$ a label with running number (e.g., if a sublist of examples follows).

> `\ex <example>` $\Longrightarrow$ a numbered example.

> `\ex[<judgement>]{<example>}` $\Longrightarrow$ a numbered example with judgement.

The space reserved for judgements is governed by the `\judgewidth{<string>}` command which sets it to the width of `<string>` (Default: "`??`"). It can be issued in the preamble to change the default globally, or within an `exe`-environment for a local change.

In short: Use `\ex` for all numbered examples; if they have no optional argument (judgement) they will be flush left; if they have `[ ]` 's then there will be a space reserved for judgements. (NB: In this case you have to indicate the scope of the example itself with {...}'s.)

1.2   Examples with other identifiers

Sometimes it is necessary not to use running numbers but to specify the identifier of an example manually as in:

> ...let us now contemplate Chomsky (1981:264, ex.29f):

> (29) $\alpha[_{\text{INFL}}$ AGR$][_{\text{VP}}$ V...$]$
> (30)   (i) $\beta[_{\text{VP}}$ V-AGR...$]$
>   (ii) $\beta[_{\text{VP}}$ $[_{\text{VP}}$ V-AGR...$]$ NP$]$

For cases like this there is the command `\exi{<identifier>}` which works exactly like `\ex`:

> `\exi{<identifier>}` $\Longrightarrow$ the identifier as label

> `\exi{<identifier>} <example>` $\Longrightarrow$ the example labelled with identifier

> `\exi{<identifier>}[<judgement>]{<example>}` $\Longrightarrow$ the judged example,
>   labelled with identifier.

So the example above is produced by:

```
\ldots let us now contemplate Chomsky (1981:264, ex.29f):
\begin{exe}
\exi{(29)} \al\lb{INFL} AGR]\lb{VP} V\ldots]
\exi{(30)}
```

```
 \begin{xlist}[(ii)]
   \exi{(i)} \be\lb{VP} V-AGR\ldots]
   \exi{(ii)} \be\lb{VP}\lb{VP} V-AGR\ldots] NP]
 \end{xlist}
\end{exe}
```

\exi can be used to deal with many labelling problems. The three most commons ones are predefined:

- Crossreferences (\exr{<label>}):

    . . . a sentiment already expressed in (1) on page 2, repeated here:

    (1)   Here is one.

    ```
    \ldots a sentiment already expressed in (\ref{here})
    on page \pageref{here}, repeated here:
    \begin{exe}
    \exr{here} Here is one.
    \end{exe}
    ```

- Crossreferences with primes (\exp{<label>}):

    . . . which contrasts sharply with

    (1′)   ?? Here is two

    ```
    \ldots which contrasts sharply with
    \begin{exe}
    \exp{here}[??]{Here is two.}
    \end{exe}
    ```

- Unnumbered examples (\sn):

    . . . here we have an example

    % Colourless green ideas sleep furiously.

    of a sentence too famous to be numbered. . .

    ```
    \ldots here we have an example
    \begin{exe}
    \sn[\%]{Colourless green ideas sleep furiously.}
    \end{exe}
    of a sentence too famous to be numbered\ldots
    ```

Again, these commands work exactly as \ex or \exi{<identifier>}.

2 Glosses

`gb4e.sty` offers a set of commands for typesetting glosses based on M. Covington's LaTeX-adaptation of M. deGroot's Midnight gloss TeX-macros.[3]

A gloss basically contains two elements:

`\gll` (obligatory): introduces the actual sentence/gloss pair;[4]

`\glt` or `\trans` (optional): introduces a free translation.

So the following

```
\begin{exe}
\ex
\gll Wenn jemand in die W\"uste zieht ... \\
If someone in the desert draws and lives ... \\
\trans 'if one retreats to the desert and ... '
\end{exe}
```

produces:

(5)   Wenn jemand in die Wüste zieht und lebt dort von
If   someone in the desert draws and lives there from
Heuschrecken oder sich im   Wald   verirrt hat und nährt
grasshoppers or   self in the woods errored has and nourisches
sich von   Wurzeln und Beeren, ...
self from roots   and berries ...

'if one retreats to the desert and lives there from grasshoppers or gets lost in the woods and lives off of roots and berries'

Since the macro counts words, to match one word with a multi-word gloss or vice versa, you either have to put the words in braces or use hyphens. For example, German *zum* could be glossed as either `{to the}` or `to-the`. And of course, if you have a word that is not glossed, like WH-trace e_i or a proper name, you have to tell the gloss line that there's something there, e.g. with { }.

(6)   Den Fritz$_1$ habe ich $_{--1}$ zum   Essen eingeladen.
the fred   have I   to the eating invited.

I invited Fred for dinner.

is produced by

---

[3] They are defined in the file `cgloss4e.sty` which is automatically read by `gb4e.sty`.

[4] There is also a three-line version of `\gll`, `\glll` for special morphological/phonological information (not for the free translation!).

```
\begin{exe}
\ex
\gll Den Fritz_1 habe ich \_\_{}_1 zum Essen eingeladen.\\
     the fred have I {} {to the} eating invited.\\
\glt I invited Fred for dinner.
\end{exe}
```

Although it is possible to treat an optional judgement just as the first word of the sentence with a null-translation as in (7), the standard way of using an optional argument with `\ex` as in (8) is simpler (and conceptually more consistent).

(7)　* Dit　is een voorboeldje.
　　　This is an　example.

　　‘Here's where the translation would be’

(8)　　* Dit　is een voorboeld.
　　　　This is an　example.

　　　‘and this would be the translation line’

This output is produced by

```
\ex
\gll * Dit is een voorboeldje.\\
     {} This is an example.\\
\trans 'Here's where the translation would be'
```

and

```
\ex[*]{
\gll Dit is een voorboeld.\\
     This is an example.\\
\trans 'and this would be the translation line'}
```

respectively.

By default the glosses appear in "Linguistic Inquiry style" with all lines in `\rm`. To change a line (e.g. to `\it` if required for a particular journal), just put the appropriate lines in the preamble of your document, e.g., `\let\eachwordone=\it`. The same for `\eachwordtwo` and `\eachwordthree` (used by the three line glosses introduced by `\glll`). Note that it will *not* work to put `\it` before the line in your text, as the words are parsed separately. For example, `\let\eachwordone=\it` yields

(9)   ? *Und hier können wir sehen was für Unfug   wird   gemacht*
       and hier can   we see   what for nonsense will-be made
       wenn er einen ganz langen Satz   binnen kriegt
       if   he a   very long   sentence inside gets.
       'and here's what happens to a looooong sentence'

and `\let\eachwordtwo=\sf` gives

(10)   * Cette phrase n'a   pas   une traduction.
       This   sentence not has nought a   translation.

As you can see by these examples, you can also re-define the `eachword` variables in your file for individual examples by using the `\let` command within the `\begin{exe}` environment, and then the style change is local.

By default, the glosses will be single-spaced even in double-spaced environments. If you—or rather the journal you're submitting to—don't like that, just give the command `\nosinglegloss` in the preamble of your document.

3   Varia

`gb4e.sty` contains a few additional goodies (mostly abbreviations for frequently used constructions) which will be properly documented in a subsequent life. For now just some sample of uses:

- X-bar styles: Per default `\obar{X} \ibar{X} \iibar{X} \mbar{X}` yields X^0 $\overline{X}$ $\overline{\overline{X}}$ X^{max}. With `\primebars` in the preamble, the outcome is X^0 X' XP X^{max}.

- Labelled bracketings: `\lb{NP}[1]\lb{D}[1] the]\lb{\ibar{N}} example]]` yields $[_{NP_1}[_{D_1}$ the]$[_{\overline{N}}$ example]]. An analogous right-bracket (`\rb`) is also defined.

- Greek letters: the frequently used letters $\alpha\beta\gamma\delta$ and θ have abbreviations `\al \be \ga \de \th` which also work outside mathmode.

- Sub- & superscripts (`_` and `^`) work outside mathmode, too.

At the very end of the style-file a couple of special accents for "exotic" languages are defined. Also, check out the `\attop`-, `\atcenter`-macros (by Lex Holt) for aligning naughty (`picture`-)examples to their numbers, as well as J. Frampton's movement-arrows:

(11)   Did John t_{Agr} appear t to be likely t to win.

```
\begin{exe}
\ex\atcenter{
 \arrowalign{\fillright\pd&\link3&\fillleft{\vrule\hfil}\cr
  Did&\ &John&\ &t_{\mathrm{Agr}}&\ appear &t&\ to be likely &t&\ to win.\cr
  &&\fillright\pu&\link3&\centr{\vrule\spacer\pu}&\lf&\fillleft\vrule\cr}
           }
\end{exe}
```