

# Typing Linguistics with covington.sty

Michael A. Covington  
Artificial Intelligence Center  
The University of Georgia  
Athens, Georgia 30602 U.S.A.  
mcovingt@ai.uga.edu  
<http://www.ai.uga.edu/~mc>

2001 March 27

## Contents

1	Accents	2
2	Example numbers	3
3	The example environment	3
4	The examples environment	4
5	Glossing sentences word-by-word	5
6	Phrase structure rules	7
7	Feature structures	7
8	Discourse representation structures	7
9	Exercises	9
10	Reference Lists	9
11	Displayed sentences	10
12	Big curly brackets (disjunctions)	10

## New in This Version

- It is no longer necessary to type `\it` to get proper italic type in feature structures.
- Instructions have been rewritten with  $\text{\LaTeX} 2_{\epsilon}$  users in mind.

## New in Preceding Versions

- Multiple accents on a single letter (e.g.,  $\hat{\tilde{a}}$ ) are supported.
- This package is now called `covington` (with the o) and is compatible with  $\text{\LaTeX}$  2<sub>ε</sub> and NFSS as well as  $\text{\LaTeX}$  2.09.
- The vertical placement of labeled feature structures has been changed so that the category labels line up regardless of the size of the structures.

## Introduction

This file, `covington.tex`, is the documentation for the March 2001 version of `covington.sty`, which is a  $\text{\LaTeX}$  style option for typing many of the special notations common in linguistics.

In  $\text{em}\text{\TeX}$  under MS-DOS, `covington.sty` is called `covingto.sty`. The missing *n* has no effect.

To use `covington.sty`, you should have a copy of it in either your current directory or the directory where  $\text{\LaTeX}$  styles are kept on your system.

Then, under  $\text{\LaTeX}$  2<sub>ε</sub>, include the command `\usepackage{covington}` after your `\documentclass` command.

In  $\text{\LaTeX}$  2.09, include `covington` among the optional parameters of `\documentstyle`, like this:

```
\documentstyle[12pt,covington]{article}
```

Note the spelling `covington` (9 letters).

In what follows I presume that you know how to use  $\text{\LaTeX}$  and have access to the  $\text{\LaTeX}$  manual. Note that `covington.sty` does not provide any special fonts or character sets. However, it can be used in combination with other style sheets that do.

If you are using `covington.sty` and `uga.sty` (UGa thesis style) together, you should mention `uga` before `covington`.

## 1 Accents

$\text{\LaTeX}$  provides a generous range of accents that can be placed on any letter, such as:

$\grave{x}$   $\acute{x}$   $\hat{x}$   $\ddot{x}$   $\tilde{x}$   $\bar{x}$   $\check{x}$   $\text{\textcircled{x}}$   $\text{\textsubscript{x}}$   $\text{\textsuperscript{x}}$

which are typed, respectively, as:

`\grave{x}` `\acute{x}` `\hat{x}` `\ddot{x}` `\tilde{x}` `\bar{x}` `\check{x}` `\text{\textcircled{x}}` `\text{\textsubscript{x}}` `\text{\textsuperscript{x}}`

$\text{\LaTeX}$  also provides the foreign characters

$\text{\AA}$   $\text{\aa}$   $\text{\AE}$   $\text{\ae}$   $\text{\CE}$   $\text{\ce}$   $\text{\AA}$   $\text{\aa}$   $\text{\O}$   $\text{\o}$   $\text{\L}$   $\text{\l}$   $\text{\B}$   $\text{\b}$   $\text{\I}$   $\text{\i}$

which are typed as:

`\i \j \ae \AE \oe \OE \aa \AA \o \O \l \L \ss ? ‘ ! ‘`

But by itself, L<sup>A</sup>T<sub>E</sub>X doesn't give you a convenient way to put two accents on the same letter. To fill this gap, `covington.sty` provides the following macros:

`\twoacc[...|...]` to combine any 2 accents, e.g., `\twoacc[\~|\={a}]` =  $\tilde{a}$   
`\acm{...}` for acute over macron, e.g., `\acm{a}` =  $\acute{a}$   
`\grm{...}` for grave over macron, e.g., `\grm{a}` =  $\grave{a}$   
`\cim{...}` for circumflex over macron, e.g., `\cim{a}` =  $\hat{a}$

The first of these is the general case and the latter three are special cases that occur often in transcribing Greek. Now you can type *Koiné* with both accents in place.

Note the peculiar syntax of `\twoacc` — its arguments are in square brackets, not curly brackets, and are separated by `|`. The first argument is the upper accent (only) and the second argument is the letter with the lower accent indicated.

Note also that not all accents work in the `tabbing` environment. Use `tabular` or see the L<sup>A</sup>T<sub>E</sub>X manual for workarounds.

## 2 Example numbers

Linguistics papers often include numbered examples. The macro `\exempleno` generates a new example number and can be used anywhere you want the number to appear. For example, to display a sentence with a number at the extreme right, do this:

```
\begin{flushleft}
This is a sentence. \hfill (\exempleno)
\end{flushleft}
```

Here's what you get:

This is a sentence. (1)

The example counter is actually the same as L<sup>A</sup>T<sub>E</sub>X's equation counter, so that if you use equations and numbered examples in the same paper, you get a single continuous series of numbers. If you want to access the number without changing it, use `\theequation`.

Also, you can use `\label` and `\ref` with example numbers in exactly the same way as with equation numbers. See the L<sup>A</sup>T<sub>E</sub>X manual for details. This applies to the `example` and `examples` environments, described next, as well as to `\exempleno` itself.

## 3 The example environment

The `example` environment displays a single example with a generated example number to the left of it. If you type

```
\begin{example}
This is a sentence.
\end{example}
```

you get:

(2) This is a sentence.

The `example` environment is a lot like `flushleft`. The example can be of any length; it can consist of many lines (separated by `\\`), or even whole paragraphs.

One way to number sub-examples is to use `itemize` or `enumerate` within an example, like this:

```
\begin{example}
\begin{itemize}
\item[(a)] This is the first sentence.
\item[(b)] This is the second sentence.
\end{itemize}
\end{example}
```

This prints as:

(3) (a) This is the first sentence.  
(b) This is the second sentence.

However, the `examples` environment, described next, is usually more convenient.

## 4 The examples environment

To display a series of examples together, each with its own example number, use `examples` instead of `example`. The only difference is that there can be more than one example, and each of them has to be introduced by `\item`, like this:

```
\begin{examples}
\item This is the first sentence.
\item This is the second sentence.
\end{examples}
```

This prints as:

(4) This is the first sentence.  
(5) This is the second sentence.

## 5 Glossing sentences word–by–word

To gloss a sentence is to annotate it word–by–word. Most commonly, a sentence in a foreign language is followed by a word–for–word translation (with the words lined up vertically) and then a smooth translation (not lined up), like this:<sup>1</sup>

*Dit is een Nederlands voorbeeld.*  
This is a Dutch example.  
‘This is an example in Dutch.’

That particular example would be typed as:

```
\gll Dit is een Nederlands voorbeeld.  
      This is a Dutch example.  
\glt ‘This is an example in Dutch.’  
\glend
```

Notice that the words do not have to be typed lining up; instead,  $\TeX$  counts them. If the words in the two languages do not correspond one–to–one, you can use curly brackets to show the intended grouping. For example, to print

*Dit is een voorbeeldje in het Nederlands.*  
This is a little example in Dutch.  
‘This is a little example in Dutch.’

you would type:

```
\gll Dit is een voorbeeldje      in het Nederlands.  
      This is a {little example} in {} Dutch.  
\glt ‘This is a little example in Dutch.’  
\glend
```

All together, `covington.sty` gives you five macros for dealing with glosses:

- `\gll` introduces two lines of words vertically aligned, and activates an environment very similar to `flushleft`.
- `\glll` is like `\gll` except that it introduces *three* lines of lined–up words (useful for cited forms, morphology, and translation).
- `\glt` ends the set of lined–up lines and introduces a line (or more) of translation.
- `\gln` is like `\glt` but does not start a new line (useful when no translation follows but you want to put a number on the right).
- `\glend` ends the special `flushleft`–like environment.

---

<sup>1</sup>The macros for handling glosses are adapted with permission from `gloss.tex`, by Marcel R. van der Goot.

Here are several examples. First, a sentence with three lines aligned, instead of just two:

*Hoc est aliud exemplum.*  
n.sg.nom 3.sg n.sg.nom n.sg.nom  
This is another example.  
‘This is another example.’

This is typed as:

```
\glll Hoc est aliud exemplum.
      n.sg.nom 3.sg n.sg.nom n.sg.nom
      This is another example.
\glt  ‘This is another example.’
\glend
```

Next, an example with a gloss but no translation, with an example number at the right:

*Hoc habet numerum.* (6)  
This has number

That one was typed as:

```
\gll Hoc habet numerum.
      This has number
\gln \hfill (\exampleno)
\glend
```

Finally we’ll put a glossed sentence inside the `example` environment, which is a very common way of using it:

(7) *Hoc habet numerum praepositum.*  
This has number preposed  
‘This one has a number in front of it.’

This last example was, of course, typed as:

```
\begin{example}
\gll Hoc habet numerum praepositum.
      This has number preposed
\glt ‘This one has a number in front of it.’
\glend
\end{example}
```

Notice that every glossed sentence begins with either `\gll` or `\glll`, then contains either `\glt` or `\gln`, and ends with `\glend`. Layout is critical in the part preceding `\glt` or `\gln`, and fairly free afterward.

## 6 Phrase structure rules

To print the phrase structure rule  $S \rightarrow NP VP$  you can type `\psr{S}{NP~VP}`, and likewise for other phrase structure rules.

## 7 Feature structures

To print a feature structure such as:

$$\left[ \begin{array}{l} case : nom \end{array} \right]$$

you can type:

```
\fs{case:nom \\\ person:P}
```

The feature structure can appear anywhere — in continuous text, in a displayed environment such as `flushleft`, or inside a phrase-structure rule, or even inside another feature structure.

To put a category label at the top of the feature structure, like this,

$$\begin{array}{c} N \\ \left[ \begin{array}{l} case : nom \\ person : P \end{array} \right] \end{array}$$

here's what you type:

```
\lfs{N}{case:nom \\\ person:P}
```

And here is an example of a PS-rule made of labeled feature structures:

$$\begin{array}{c} S \\ \left[ \begin{array}{l} tense : T \end{array} \right] \end{array} \rightarrow \begin{array}{c} NP \\ \left[ \begin{array}{l} case : nom \\ number : N \end{array} \right] \end{array} \quad \begin{array}{c} VP \\ \left[ \begin{array}{l} tense : T \\ number : N \end{array} \right] \end{array}$$

which was of course typed as:

```
\psr{\lfs{S}{tense:T}}
{\lfs{NP}{case:nom \\\ number:N}
 \lfs{VP}{tense:T \\\ number:N} }
```

## 8 Discourse representation structures

Several macros in `covington.sty` facilitate display of discourse representation structures (DRSes) in the box notation originally used by Hans Kamp. The simplest is `\drs`, which takes two arguments: a list of discourse variables joined by `~`, and a list of DRS conditions separated by `\\`. Nesting is permitted. Note that the `\drs` macro itself does not give you a displayed environment; you must use `flushleft` or the like to display the DRS. Here are some examples:

`\drs{X}{donkey(X)\green(X)}`

$X$
$donkey(X)$ $green(X)$

`\drs{X}{  
{named(X, 'Pedro') \\  
\drs{Y}{donkey(Y)\owns(X,Y)}~~  
{\large $\rightarrow$}~  
\drs{~}{feeds(X,Y)}  
}`

$X$				
$named(X, 'Pedro')$ <table> <tr> <td><math>Y</math></td></tr> <tr> <td><math>donkey(Y)</math> <math>owns(X, Y)</math></td></tr> </table> $\Rightarrow$ <table> <tr> <td></td></tr> <tr> <td><math>feeds(X, Y)</math></td></tr> </table>	$Y$	$donkey(Y)$ $owns(X, Y)$		$feeds(X, Y)$
$Y$				
$donkey(Y)$ $owns(X, Y)$				
$feeds(X, Y)$				

To display a sentence above the DRS, use `\sdrs`, like this:

`\sdrs{A donkey is green.}{X}{donkey(X)\green(X)}`

*A donkey is green.*

$X$
$donkey(X)$ $green(X)$

Some DRS connectives are also provided (normally for forming DRSSes that are to be nested within other DRSSes). The macro `\negdrs` forms a DRS preceded by a negation symbol:

`\negdrs{X}{donkey(X)\green(X)}`

$\neg$
$X$
$donkey(X)$ $green(X)$



Finally, `\ifdrs` forms a pair of DRSes joined by a big arrow, like this:

$$\frac{\begin{array}{|l|} \hline X \\ \hline donkey(X) \\ hungry(X) \\ \hline \end{array}}{\begin{array}{|l|} \hline feeds(Pedro, X) \\ \hline \end{array}} \Rightarrow$$

If you have an “if”-structure appearing among ordinary predicates inside a DRS, you may prefer to use `\alifdrs`, which is just like `\ifdrs` but shifted slightly to the left for better alignment.

## 9 Exercises

The `exercise` environment generates an exercise numbered according to chapter, section, and subsection (suitable for use in a large book; in this example, the subsection number is going to come out as 0).

**Exercise 9.0.1 (Project)** *Prove that the above assertion is true.*

This was typed as

```
\begin{exercise}[Project]
Prove that the above assertion is true.
\end{exercise}
```

and the argument `[Project]` is optional (actually, any word could go there).

## 10 Reference Lists

To type an LSA-style hanging-indented reference list, use the `reflist` environment. (*Note:* `reflist` is not presently integrated with BibTeX in any way.) For example,

```
\begin{reflist}
Barton, G. Edward; Berwick, Robert C.; and Ristad, Eric Sven. 1987.
Computational complexity and natural language. Cambridge,
Massachusetts: MIT Press.

Chomsky, Noam. 1965. Aspects of the theory of syntax. Cambridge,
Massachusetts: MIT Press.

Covington, Michael. 1993. Natural language processing for Prolog
programmers. Englewood Cliffs, New Jersey: Prentice--Hall.
\end{reflist}
```

prints as:

Barton, G. Edward; Berwick, Robert C.; and Ristad, Eric Sven. 1987. Computational complexity and natural language. Cambridge, Massachusetts: MIT Press.

Chomsky, Noam. 1965. Aspects of the theory of syntax. Cambridge, Massachusetts: MIT Press.

Covington, Michael A. 1993. Natural-language processing for Prolog programmers. Englewood Cliffs, New Jersey: Prentice-Hall.

Notice that within the reference list, “French spacing” is in effect — that is, spaces after periods are no wider than normal spaces. Thus you do not have to do anything special to avoid excessive space after people’s initials.

## 11 Displayed sentences

The macro `\sentence` displays an italicized sentence (it is a combination of `flushleft` and `\em`). If you type

```
\sentence{This is a sentence.}
```

you get:

*This is a sentence.*

## 12 Big curly brackets (disjunctions)

Last of all, the 2-argument macro `\either` expresses alternatives within a sentence or PS-rule:

the `\either{big}{large}` dog = the  $\left\{ \begin{array}{c} \text{big} \\ \text{large} \end{array} \right\}$  dog

`\psr{A}{B~\either{C}{D}~E}` =  $A \rightarrow B \left\{ \begin{array}{c} C \\ D \end{array} \right\} E$

That’s all there is. Suggestions for improving `covington.sty` are welcome, and bug reports are actively solicited. Please note, however, that this is free software, and the author makes no commitment to do any further work on it.