avm.sty

Christopher Manning
manning@csli.stanford.edu

Version 1.00 – December 12, 1993

avm.sty is a TEX macro package that makes it easy to draw attribute-value matrices (AVMs, also known as feature structures and signs), things like (1):

(1)
$$
{}_{\textit{indexed-obj}}\left[\begin{array}{ll}
\text{INDEX} & {}_{\textit{index}}\left[\begin{array}{ll}
\text{VARIABLE} & \boxed{1}\;{}_{\textit{variable}}\left[\begin{array}{ll}\text{PER} & \text{3rd}\\ \text{NUM} & \text{sing}\\ \text{GEND} & \text{neut}\end{array}\right]\\
\text{RESTRICTION} & {}_{\textit{psoa}}\left[\begin{array}{ll}\text{RELATION} & \text{book}\\ \text{INSTANCE} & \boxed{1}\end{array}\right]
\end{array}\right]
\end{array}\right]
$$

It is a special mode, like math mode, in which various things are redefined in order to make it easy to typeset a certain type of thing. In fact, avm.sty supports several sub-modes of AVM entry that are tailored to different needs. This document first describes the basic mode of avm.sty and then shows how the specialized modes deviate from it. Everything that can be done in one of the specialized sub-modes can also be done in the main mode; it's just that you might be able to do it with less keystrokes. avm.sty has been designed so that it will work with either TEX or LATEX; the few differences between using these two formats are mentioned at the appropriate places below.

1 Loading this option

If you are using LATEX, avm.sty should be included among the other document-style options that you load. For example, you might type at the beginning of your document:

(2) `\documentstyle[11pt,avm,lingmacros]{article}`

In TEX, you input the file by including the following near the beginning of your document:

(3) `\input avm.sty`

Either of these assumes that the file avm.sty is somewhere on the search path for input files (the TEXINPUTS environment variable). This search path usually includes the current directory.

2 The 'standard' environment

Most commands (except those that establish defaults) are only available within an AVM environment. To enter and exit this environment, in LATEX you type:

(4)
```
\begin{avm}
...
\end{avm}
```

while in TeX you type:

(5)
```
\avm
...
\endavm
```

2.1 Making AVMs in square brackets

The commands `\[` and `\]` begin and end a square-bracketed AVM. Within an AVM, a new line is begun by typing `\\`.[1] It is unnecessary to type `\\` following the last row of the AVM. Within a line columns are divided by the ampersand '`&`'. There can actually be any number of columns, although in most applications of AVMs there are only two. For example:[2]

(6) a.
$$\begin{bmatrix} \textsc{cat}\,|\,\textsc{subcat} & \left\langle \mathrm{NP}_{it},\ \mathrm{NP}_{\boxed{2}},\ \mathrm{S}[\textsc{comp}]{:}\boxed{3} \right\rangle \\ \textsc{content} & \begin{bmatrix} \textsc{relation} & \textbf{bother} \\ \textsc{bothered} & \boxed{2} \\ \textsc{soa-arg} & \boxed{3} \end{bmatrix} \end{bmatrix}$$

b.
$$\begin{bmatrix} \textsc{subj} & \begin{bmatrix} \textsc{pers} & 3 \\ \textsc{num} & \textsc{sg} \\ \textsc{gend} & \textsc{masc} \\ \textsc{pred} & \text{`pro'} \end{bmatrix} \\ \textsc{pred} & \text{`eat}\langle\text{SUBJ, OBJ}\rangle\text{'} \\ \textsc{obj} & \begin{bmatrix} \textsc{pers} & 3 \\ \textsc{num} & \textsc{pl} \\ \textsc{gend} & \textsc{fem} \\ \textsc{pred} & \text{`pro'} \end{bmatrix} \end{bmatrix}$$

(7) a.
```
\begin{avm}
\[ cat\|subcat & \<NP$_{it}$, NP$_{\@2}$, S[comp]:\@3\> \\
  content & \[ relation & \bf bother\\
             bothered & \@2 \\
             soa-arg & \@3  \] \]
\end{avm}
```

[1] TeX users can use `\cr`, if they prefer. LaTeX users (only!) can follow the `\\` with any of the things that can follow `\\` in `tabular` environments. So you can put extra space below one line of an AVM, by typing `\\[2ex]`, for example.

[2] Here is one in a footnote (note the smaller fonts):

(i)
$$\begin{bmatrix} \textsc{ind} & \begin{bmatrix} \textsc{pers} & 3 \\ \textsc{num} & \mathit{sg} \\ \textsc{gend} & \mathit{masc} \\ \textsc{refind} & \boxed{12} \end{bmatrix} \end{bmatrix}$$

b.
```
\begin{avm}
\[ subj & \[ pers & 3   \\
            num  & sg  \\
            gend & masc\\
            pred & \rm ‘pro’ \]\\
   pred & \rm ‘eat\q<SUBJ, OBJ\q>’\\
   obj  & \[ pers & 3   \\
            num  & pl  \\
            gend & fem \\
            pred & \rm ‘pro’ \]
\]
\end{avm}
```

The underlying environment is based on TEX's `\halign` (which also underlies LATEX's `tabular` environment), and so the normal conventions of these apply. Note that spaces are ignored on both sides of an `&` or `\\`, as in LATEX's `tabular` environment (but not in Plain TEX's `\halign`).

2.1.1 When you don't want to align columns

There are some occasions when you don't want things lined up in columns. For example, with a long HPSG path, we probably want the output to look as in (8a) not (8b).

(8) a. $$\begin{bmatrix} \text{SYNSEM}\,|\,\text{LOCAL}\,|\,\text{CONTENT} & \boxed{2} \\ \text{QSTORE} & \begin{bmatrix} \text{DET} & \text{forall} \\ \text{RESTPAR} & \boxed{2}\begin{bmatrix} \text{PARA} & \boxed{1} \\ \text{RESTR} & \begin{bmatrix} \text{RELN} & \text{book} \\ \text{INST} & \boxed{1} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

b. $$\begin{bmatrix} \text{SYNSEM}\,|\,\text{LOCAL}\,|\,\text{CONTENT} & \boxed{2} \\ \text{QSTORE} & \qquad\qquad\qquad \begin{bmatrix} \text{DET} & \text{forall} \\ \text{RESTPAR} & \boxed{2}\begin{bmatrix} \text{PARA} & \boxed{1} \\ \text{RESTR} & \begin{bmatrix} \text{RELN} & \text{book} \\ \text{INST} & \boxed{1} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

There are several ways to achieve this, which will be presented in order from the quick and dirty fix to the best solution. One way is just to not use &s. That way everything will be in one column:

(9) a. $$\begin{bmatrix} \text{SHORT value}_1 \\ \text{EXTREMELY-LOOOOOOOOOONG value}_2 \end{bmatrix}$$

b.
```
\begin{avm}
\[ short \rm value$_1$ \\
   extremely-looooooooooong \rm value$_2$ \]
\end{avm}
```

However, the discerning eye will notice that there is then only a normal space width between attributes and values, and that looks bad. But avm.sty defines a command `\;` analogous to the 'thick space' in math mode that will put in the normal spacing between attributes and values. So the above example can be improved by typing it as follows.

(10) a. $\left[\begin{array}{l} \text{SHORT}\;\; \text{value}_1 \\ \text{EXTREMELY-LOOOOOOOOOONG}\;\; \text{value}_3 \end{array}\right]$

b.
```
\begin{avm}
\[ short\; \rm value$_1$ \\
   extremely-loooooooooong\; \rm value$_3$ \]
\end{avm}
```

However, to make elegant AVMs it is often the case that you want most columns to be lined up, and in this case, you should use `&` in most columns as usual, but make the entire exceptional rows an argument of the command `\avmspan` as shown below:

(11) a. $\left[\begin{array}{ll} \text{CONTENT} & \left[\begin{array}{ll} \text{RELN} & \textbf{cause} \\ \text{CAUSER} & \boxed{1} \\ \text{CAUSEE} & \boxed{2} \\ \multicolumn{2}{l}{\text{CAUSED-EVENT} \mid \text{RELN}\;\; \boxed{3}} \end{array}\right] \end{array}\right]$

b.
```
\begin{avm}
\[ content & \[ reln & \bf cause  \\
                causer & \@1 \\
                causee & \@2 \\
                \avmspan{caused-event\|reln\;\@3} \] \]
\end{avm}
```

2.2 Other types of brackets and braces

avm.sty also provides other types of large extendable braces and brackets. One uses `\{` and `\}` for curly braces, `\(` and `\)` for round parentheses, `\<` and `\>` for angle brackets and `\avml` and `\avmr` for no brackets at all. Note that angle brackets have a maximum size, though, and are not infinitely stretchable like the other types of brackets. As is usual in TEX, an opening bracket must be matched by a closing bracket, but the two need not be of the same sort. All these commands provide exactly the same internal environment as was described above for square brackets (e.g., use of `&` and `\\`), although, in general, these fancier options are less used with these types of braces. Some examples follow (note the 'misuse' of a 3 column AVM in (12c)):

(12) a. $\left[\begin{array}{ll} \text{QSTORE} & \left\{\left[\begin{array}{ll} \text{DET} & \text{the} \\ \text{RESTPAR} & \left[\begin{array}{ll} \text{PARA} & \boxed{1} \\ \text{RESTR} & \left\{\left[\begin{array}{ll} \text{RELN} & \text{poss} \\ \text{POSSESSOR} & \boxed{3} \\ \text{POSSESSED} & \boxed{1} \end{array}\right]\right\} \textbf{ union } \{\boxed{2}\} \end{array}\right] \end{array}\right]\right\} \end{array}\right]$

b.
```
\begin{avm}
\[qstore & \{ \[det & \rm the \\
                restpar & \[para & \@1 \\
```

```
                               restr & \{ \[reln & \rm poss \\
                                   possessor & \@3 \\
                                   possessed & \@1\]\}\ \bf union \q\{\@2\q\}
\] \] \} \]
\end{avm}
```

c. kak $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ kak-ase

$$\left[\text{SUBCAT}\ \left\langle \begin{bmatrix}\text{POS} & \text{N}\\ \text{CASE} & \text{NM}\\ \text{IND} & \boxed{1}\end{bmatrix}, \begin{bmatrix}\text{POS} & \text{N}\\ \text{CASE} & \text{AC}\\ \text{IND} & \boxed{2}\end{bmatrix}\right\rangle\right] \Longrightarrow \left[\text{SUBCAT}\ \left\langle \begin{bmatrix}\text{POS} & \text{N}\\ \text{CASE} & \text{NM}\\ \text{IND} & \boxed{3}\end{bmatrix}, \begin{bmatrix}\text{POS} & \text{N}\\ \text{CASE} & \text{DT}\\ \text{IND} & \boxed{1}\end{bmatrix}, \begin{bmatrix}\text{POS} & \text{N}\\ \text{CASE} & \text{AC}\\ \text{IND} & \boxed{2}\end{bmatrix}\right\rangle\right]$$

d.
```
\begin{avm}
\avml
\avml \rm kak\\
\[subcat \< \[pos&n\\case&nm\\ind&\@1\], \[pos&n\\case&ac\\ind&\@2\] \>\]
\avmr
& $\Longrightarrow$ &
\avml \rm kak-ase\\
\[subcat \< \[pos&n\\case&nm\\ind&\@3\], \[pos&n\\case&dt\\ind&\@1\],
             \[pos&n\\case&ac\\ind&\@2\] \>\]
\avmr \avmr
\end{avm}
```

e.
$$\left\{\begin{matrix}\begin{bmatrix}\text{PRED} & \text{‘see}\langle\text{girl, Mary}\rangle\text{’}\\ \text{SUBJ} & \text{‘girl’}\\ \text{OBJ} & \text{‘Mary’}\end{bmatrix}\\ \begin{bmatrix}\text{PRED} & \text{‘heard}\langle\text{girl, Bill}\rangle\text{’}\\ \text{SUBJ} & \text{‘girl’}\\ \text{OBJ} & \text{‘Bill’}\end{bmatrix}\end{matrix}\right\}$$

f.
```
\begin{avm}
\{ \[ pred & \rm ‘see\q<girl, Mary\q>’ \\
      subj & \rm ‘girl’ \\
      obj & \rm ‘Mary’ \]                     \\
   \[ pred & \rm ‘heard\q<girl, Bill\q>’ \\
      subj & \rm ‘girl’ \\
      obj & \rm ‘Bill’ \]                     \}
\end{avm}
```

For normal text size brackets, `[`, `]`, `(` and `)` work in the usual way. Curly braces and angle brackets can be made using the `\q` (‘quote’ command), by writing `\q\{`, `\q\}`, `\q<` and `\q>`, as in (12b,f). Or they can be made by going into math mode in the usual way.

2.3 Other features

2.3.1 Sort labels

The commands `\sort` and `\osort` put sort labels at the corner of any structure. They differ in that the label of `\sort` takes up space in the column or text where it appears whereas `\osort` overlaps its label into the previous column or text. `\osort` does not check for collisions so you have to use your judgment. For example, (1) was typeset with these commands:

(13)
```
\begin{avm}
\sort{indexed-obj}{\[ index & \osort{index}{\[
        variable & \@1 \osort{variable}{\[ per & \rm 3rd \\
                                           num & \rm sing \\
                                           gend & \rm neut \]} \\
        restriction & \osort{psoa}{\[relation & \rm book \\
                                     instance & \@1 \]} \]} \]}
\end{avm}
```

If one wants (almost) all square-bracketed feature structures sorted, it is more convenient to use the sub-mode `sorted` (see Section 4.2). See also Section 4.1 for how to vary the sort label position.

2.3.2 Bars, index boxes, nodes and lines

The command `\|` will produce a vertical bar with a little space on either side. The command `\@{...}` will produce a labeled index box, as used in PATR and HPSG. If the index is a single digit, the braces may be omitted. The `\@` command only works within the AVM environment, but since you often want to refer to index boxes in the text of your paper, the additional command `\avmbox{...}` can be used anywhere to draw an index box.

For curved lines connecting AVMs (as widely used in LFG), Emma Pease's tree-dvips.sty can be used as normal, but avm.sty also adds the shortcut whereby `\!` can be used instead of `\node`. One often wants these curved lines to connect to a dash or an empty feature structure. So avm.sty provides the one-parameter commands `\avmd` and `\avmb` (for dash and box, respectively), that draw such an object and label it with their parameter. These are illustrated below:

(14) a.
$$\left[\begin{array}{ll} \text{SUBJ} & \left[\begin{array}{ll}\text{PRED} & \text{‘pro’}\\ \text{NUM} & \text{SG}\\ \text{PERS} & 1\end{array}\right]\\ \text{PRED} & \text{‘want}\langle\text{—, —}\rangle\text{’}\\ \text{XCOMP} & \left[\begin{array}{ll}\text{PRED} & \text{‘go}\langle\text{—}\rangle\text{’}\\ \text{SUBJ} & [\ \ ]\end{array}\right]\end{array}\right]$$

b.
```
\begin{avm}
  \[ subj & \!{a}{\[ pred & \rm ‘pro’\\
                     num & sg \\
                     pers & 1 \]} \\
     pred & \rm ‘want\q<\avmd{aa}, \avmd{bb}\ \q>’ \\
     xcomp & \!{b}{\[ pred & \rm ‘go\q<\avmd{aaaa}\q>’ \\
                      subj & \avmb{aaa} \]} \]
\end{avm}
\nodecurve[r]{a}[t]{aa}{.4in}
\nodecurve[b]{aa}[t]{aaaa}{.2in}
\nodecurve[r]{aaa}[b]{aaaa}{.15in}
\nodecurve[r]{bb}[r]{b}{.5in}[.5in]
```

If one wants (almost) all square-bracketed feature structures labeled, it is more convenient to use the sub-mode `labeled` (see Section 4.2).

3 Fonts

By default, avm.sty puts AVMs in the type size and style that is currently being used. This allows one to easily type things like the following (in LaTeX):

(15) a.

$$\begin{bmatrix} \textsf{content} & \begin{bmatrix} \textsf{reln} & \textbf{cause} \\ \textsf{causer} & \boxed{1} \\ \textsf{causee} & \boxed{2} \\ \multicolumn{2}{l}{\textsf{caused-event} \,|\, \textsf{reln} \;\; \boxed{3}} \end{bmatrix} \end{bmatrix}$$

b.
```
{\footnotesize\sf \begin{avm}
\[ content \[ reln & \bf cause  \\
              causer & \@1 \\
              causee & \@2 \\
              \avmspan{caused-event\|reln\;\@3} \] \]
\end{avm}}
```

However, one often wants all AVMs to be set in certain fonts and so avm.sty lets you specify default fonts for AVMs. This is done with the following commands:

(16) a. `\avmfont{...}`

b. `\avmvalfont{...}`

c. `\avmsortfont{...}`

This is how fonts work: (i) index box contents are set in the `\scriptstyle` (smaller subscript size) of the current math font; (ii) if `\avmsortfont` is defined, sort names for AVMs are set in the `\avmsortfont`; (iii) if `\avmvalfont` is defined, value names (items in the second and subsequent columns of the AVM environment) are set in the `\avmvalfont`; (iv) everything not assigned a font so far is set in the `\avmfont`, if this is defined, and in the current default font otherwise.

Most of the examples in this document are set in small capitals using the following (LaTeX) defaults:

(17)
```
\avmfont{\sc}
\avmsortfont{\scriptsize\it}
```

After setting up a default font, to reset things so that avm.sty uses the current font, simply call the same commands with an empty argument, for example, `\avmfont{}`.

Note for Plain TeX users: by default, neither the small capitals font nor the equivalent of LaTeX's `\scriptsize` italic font is loaded. To get similar output in Plain TeX, the minimum you must add somewhere near the beginning of your file is:

(18)
```
\font\tensc=cmcsc10
\font\seveni=cmti7
\avmfont{\tensc}
\avmsortfont{\seveni}
```

However, more setup would be required to fully integrate these fonts into Plain TeX or to get things to work with different font sizes or scalings.

The font of any individual item can be changed in the usual way with a font changing command, as shown in (13). Note that in any of the bracketed structures, font changes are always only local (having scope until the next `&` or `\\`), so it is generally not necessary to use braces to delimit font changes.

4 User-specified options

avm.sty has various (well, eight) options that you can specify. Three of these activate sub-modes which alter how you input AVMs, while the others alter the way AVMs appear in the output. You activate these options by using the `\avmoptions` command. This command takes a comma-separated list of options to be turned on and all the other options are turned off. So, three options are turned on with (19a) and all are turned off (returning things to the default mode) with the command (19b). Just `labeled` can be selected with command (19c). Note that the parameter to an `\avmoptions` command must not contain any spaces! The options are `active`; `sorted`, `labeled`; `center` and `bottom`; `topleft`, `topright`, and `bottomright`. Options within each semicolon-separated set are mutually exclusive, but otherwise the options can be freely combined.

(19) a. `\avmoptions{active,sorted,bottom}`

b. `\avmoptions{}`

c. `\avmoptions{labeled}`

Setting these options is local to a group and so the domain of effect can be delimited with braces. Also, since all LaTeX environments (things enclosed in `\begin` ... `\end` pairs) define their own group, including an options command within such a `\begin` ... `\end` pair will affect only the processing of the rest of that environment.

4.1 Sort labels in different corners: `topleft`, `topright`, and `bottomright`

While HPSG normally places sort labels in the bottom left corner of AVMs, with avm.sty you can actually have sort labels in any corner that you wish. You select different corners for the `\sort` and `\osort` commands and the `sorted` option with the options `topleft`, `topright`, and `bottomright`. This is illustrated in the next section.

4.2 The `sorted` and `labeled` sub-modes

These modes both affect just the behavior of the square brackets commands (in either the default mode or the `active` sub-mode) and are mutually exclusive. Left square-bracket commands take one argument, which is the sort or node label name for this AVM. The `labeled` mode gives the same results as using `\!` (i.e., `\node`), but lessens the typing required:

(20) a.

$$\begin{bmatrix} \text{SUBJ} & \begin{bmatrix} \text{NUM} & \text{PL} \end{bmatrix} \\ \text{TENSE} & \text{PRES} \\ \text{PRED} & \text{`bark}\langle - \rangle\text{'} \end{bmatrix}$$

b.
```
\avmoptions{labeled}
\begin{avm}
\[{a} subj  & \[{b} num & pl \] \\
      tense & pres \\
      pred  & \rm `bark\q<\avmd{c}\q>' \]
\end{avm}
\nodecurve[br]{b}[t]{c}{15pt}
```

The sorted sub-mode gives a combination of the effects of the \sort and \osort commands (trying to give you what you probably want). If the corner sub-mode is topright or bottomright then it always behaves as \sort. If the sub-mode is topleft, or the default which puts labels in the bottom left, then it behaves like \osort, except at the outermost level of the AVM, where it behaves like \sort.[3] Example (21) was produced by preceding (22) by each of the commands in (23) in turn.

(21)

$$ {}^{index}\left[\begin{array}{ll} \text{VARIABLE} & \boxed{1}\ {}^{variable}\left[\begin{array}{ll}\text{PER} & \text{3rd}\\ \text{NUM} & \text{sing}\\ \text{GEND} & \text{neut}\end{array}\right] \\ \text{RESTR} & {}^{psoa}\left[\begin{array}{ll}\text{RELATION} & \text{book}\\ \text{INSTANCE} & \boxed{1}\end{array}\right]\end{array}\right] \qquad \left[\begin{array}{ll} \text{VARIABLE} & \boxed{1}\ \left[\begin{array}{ll}\text{PER} & \text{3rd}\\ \text{NUM} & \text{sing}\\ \text{GEND} & \text{neut}\end{array}\right]^{variable} \\ \text{RESTR} & \left[\begin{array}{ll}\text{RELATION} & \text{book}\\ \text{INSTANCE} & \boxed{1}\end{array}\right]^{psoa}\end{array}\right]^{index} $$

$$ {}_{index}\left[\begin{array}{ll} \text{VARIABLE} & \boxed{1}\ {}_{variable}\left[\begin{array}{ll}\text{PER} & \text{3rd}\\ \text{NUM} & \text{sing}\\ \text{GEND} & \text{neut}\end{array}\right] \\ \text{RESTR} & {}_{psoa}\left[\begin{array}{ll}\text{RELATION} & \text{book}\\ \text{INSTANCE} & \boxed{1}\end{array}\right]\end{array}\right] \qquad \left[\begin{array}{ll} \text{VARIABLE} & \boxed{1}\ \left[\begin{array}{ll}\text{PER} & \text{3rd}\\ \text{NUM} & \text{sing}\\ \text{GEND} & \text{neut}\end{array}\right]_{variable} \\ \text{RESTR} & \left[\begin{array}{ll}\text{RELATION} & \text{book}\\ \text{INSTANCE} & \boxed{1}\end{array}\right]_{psoa}\end{array}\right]_{index} $$

(22)
```
\begin{avm}
\[{index} variable & \@1 \[{variable} per & 3rd \\
                                      num & sing \\
                                      gend & neut \] \\
          restr & \[{psoa} relation & book \\
                                 instance & \@1 \] \]
\end{avm}
```

(23) a.
```
\avmvalfont{\rm}
\avmoptions{topleft,sorted}
```

b. `\avmoptions{topright,sorted}`

c. `\avmoptions{sorted}`

d. `\avmoptions{bottomright,sorted}`

4.3 The active sub-mode

The AVMs above looked okay, but it can get tiring typing all the backslashes. So an alternative environment in which bracket characters are 'active' (i.e., they are commands) is also provided. Using it, one can type the AVM in (8a) as follows:

(24)
```
\avmoptions{active}
\begin{avm}
```

[3] This is so the outermost sort label will not overlap the text preceding it, whereas if you use the \osort command, no extra space is added, even at the outermost level. This means that if you are building AVMs with sorts in parts, you will want to use \osort rather than the sorted sub-mode.

```
[ synsem|local|content\; @2 \\
  qstore\; \{ [ det & \rm forall \\
                restpar & @2 [para & @1\\
                                restr & \{ [ reln & \rm book\\
                                               inst & @1 ] \} ] ] \} ]
\end{avm}
```

The functionality of this environment is exactly the same as that of the default mode described above; all that differs is how various commands are called. The following table defines the correspondence between commands in the two versions:

(25)

<table>
<tr><th>Default</th><th><code>active</code> mode</th><th>Default</th><th><code>active</code> mode</th></tr>
<tr><td><code>\[</code></td><td><code>[</code></td><td><code>\]</code></td><td><code>]</code></td></tr>
<tr><td><code>\(</code></td><td><code>(</code></td><td><code>\)</code></td><td><code>)</code></td></tr>
<tr><td><code>\<</code></td><td><code><</code></td><td><code>\></code></td><td><code>></code></td></tr>
<tr><td><code>\{</code></td><td><code>\{</code></td><td><code>\}</code></td><td><code>\}</code></td></tr>
<tr><td><code>[</code></td><td><code>\[</code></td><td><code>]</code></td><td><code>\]</code></td></tr>
<tr><td><code>(</code></td><td><code>\(</code></td><td><code>)</code></td><td><code>\)</code></td></tr>
<tr><td><code>\q<</code></td><td><code>\<</code></td><td><code>\q></code></td><td><code>\></code></td></tr>
<tr><td><code>\q\{</code></td><td><code>\q\{</code></td><td><code>\q\}</code></td><td><code>\q\{</code></td></tr>
<tr><td><code>\@{...}</code></td><td><code>@{...}</code></td><td><code>\|</code></td><td><code>|</code></td></tr>
</table>

All other commands in the `avm` environment are unchanged when the `active` sub-mode is in effect. Note that curly braces still require a preceding backslash. This is so curly braces can retain their usual TEX functions.

Unfortunately, there is a major restriction on being able to use this environment. Because of the way TEX works, such an environment cannot appear inside the argument of any command (such as `\footnote` or the `\enumsentence` macro of Emma Pease's lingmacros.sty).[4] However, the `active` sub-mode can be used within LATEX environments or between Plain TEX commands that serve as a preamble and postamble. For example, the author has an `example` environment that produces output similar to the `\enumsentence` macro, and using it and the `active` sub-mode, he could draw the structure in (26) by using the commands in (27).

(26)
$$\begin{bmatrix} \text{QCONTENT} & \left\langle \begin{bmatrix} \text{DET} & \text{some} \\ \text{RESTPAR} & \begin{bmatrix} \text{PARA} & \boxed{3} \\ \text{RESTR} & \left\{ \begin{bmatrix} \text{RELN} & \text{poem} \\ \text{INST} & \boxed{3} \end{bmatrix} \right\} \end{bmatrix} \end{bmatrix} \right\rangle \\ \text{DCONTENT} & \begin{bmatrix} \text{RELATION} & \text{know} \\ \text{KNOWER} & \boxed{2} \begin{bmatrix} \text{PERS} & 1 \\ \text{NUM} & \text{SG} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

(27)
```
\begin{example}
\avmoptions{active}
\begin{avm}
```

[4] This is because characters receive a permanent *catcode* when first read as a macro argument. This is a fundamental feature of TEX and nothing to do with this package. Hence it isn't likely to change.

```
[ qcontent & < [ det & \rm some \\
                 restpar & [ para & @3 \\
                             restr & \{ [ reln & \rm poem\\
                                          inst & @3      ] \} ] ] > \\
              dcontent & [ relation & \rm know \\
                           knower & @2 [ pers & 1 \\ num & sg ] ] ]
\end{avm}
\end{example}
```

Here is one more example of the `active` sub-mode (in combination with the `sorted` sub-mode), included mainly to exercise the other commands in the sub-mode.

(28) a. $${}_{\textit{test}}\left[\begin{array}{ll}\text{TESTING} & \left(\begin{array}{ll}\text{STUFF} & {}_{\textit{widget}}\left[\begin{array}{ll}\text{ATTR}_1 & \langle\ \text{VALUE}_{\boxed{1}}\ \rangle \\ \text{ATTR}_2 & (\ \text{VALUE}\ ) \\ \text{ATTR}_3 & [\ \{\ \text{VALUE}\ \}\ ]\end{array}\right]\end{array}\right)\end{array}\right]$$

b.
```
\avmoptions{active,sorted}
\begin{avm}
[{test} testing & ( stuff & [{widget} attr$_1$ & \< value$_{@1}$ \> \\
                    attr$_2$ & \( value \) \\
                    attr$_3$ & \[ \q\{ value \q\} \] ] ) ]
\end{avm}
```

4.4 The `center` and `bottom` options

These parameters determine the positioning of the whole AVM. Although an AVM usually appears in a display environment, it is actually like one huge character as far as TEX is concerned, and if you choose to put it near other characters, you might want any of the top, bottom or center of it aligned with those other characters. By default you get alignment along the top (usually what you want), but you can change this by specifying `bottom` or `center` to the `\avmoptions` command. Giving any `\avmoptions` command that does not specify `center` or `bottom` will return you to using alignment along the top. All three possibilities are shown in the below:

(29) a. Top $\left[\begin{array}{ll}\text{CASE} & \text{ACC}\\ \text{GEND} & \text{FEM}\\ \text{NUM} & \text{PL}\end{array}\right]$, bottom $\left[\begin{array}{ll}\text{CASE} & \text{ACC}\\ \text{GEND} & \text{FEM}\\ \text{NUM} & \text{PL}\end{array}\right]$ or center $\left[\begin{array}{ll}\text{CASE} & \text{ACC}\\ \text{GEND} & \text{FEM}\\ \text{NUM} & \text{PL}\end{array}\right]$.

b.
```
Top \avmoptions{active}\begin{avm}
[ case & acc \\ gend & fem \\ num  & pl ]
\end{avm},
bottom \avmoptions{active,bottom}\begin{avm}
[ case & acc \\ gend & fem \\ num  & pl ]
\end{avm}
or center \avmoptions{active,center}\begin{avm}
[ case & acc \\ gend & fem \\ num  & pl ]
\end{avm}.
```

Within the AVM environment, bracketed structures are always aligned along their centers. But you can get different alignments by nesting AVM environments (as opposed to nesting brackets within a single AVM environment).

5 Advanced topics

5.1 Layout parameters

Three parameters control the spacing in AVMs. `\avmvskip` determines the extra vertical space added at the beginning, end and between every row of an aligned structure. `\avmhskip` determines the spacing between two columns of an aligned structure. At the left and right borders of an aligned structure (between the bracket and the first or last column) the spacing is `\avmbskip`. The default values of these parameters (determined by eyeballing) are:

(30) a. `\avmvskip{0.385ex}`

b. `\avmhskip{1em}`

c. `\avmbskip{0em}`

These defaults are defined in terms of the font-dependent units `em` and `ex` so that they vary with the font size, but these parameters can be defined as any ⟨*dimen*⟩. For more compressed AVMs (such as those embedded in phrase-structure trees), the author tends to issue the following commands:

(31) `\avmvskip{.1ex}\avmhskip{.5em}`

5.2 Using boxes to build huge AVMs

Note that TeX has only a finite capacity for nesting, and if a structure is too complicated TeX may be unable to set it. The vertical length of an AVM is fairly unimportant; the critical parameter is the horizontal depth of nesting of stretchable brackets. On the TeX I am using, one can embed stretchable brackets to about the level shown in (26) (when already within a LaTeX list environment). To set a more complex AVM than this, you unfortunately have to set pieces separately, store them in box registers and then set the whole AVM (or else find a TeX with a larger *semantic nest size*). An example of how to do this is shown below. Not the use of the `center` sub-mode.[5]

[5]Note also that TeX box manipulation commands are shown. These work perfectly well under LaTeX as well, but if you want to find out about the official LaTeX box manipulation commands, see pp. 96–101 of the LaTeX book.

(32) a.

$$ {}_{sign}\left[\begin{array}{ll} \text{PHONOLOGY} & \langle \text{she} \rangle \\ \text{SYNSEM} & {}_{synsem}\left[\begin{array}{ll} \text{LOCAL} & {}_{local}\left[\begin{array}{ll} \text{CATEGORY} & {}_{cat}\left[\begin{array}{ll} \text{HEAD} & \textit{noun}[\text{CASE} \quad \textit{nom}] \\ \text{SUBCAT} & \langle\ \rangle \end{array}\right] \\ \text{CONTENT} & {}_{ref\text{-}obj}\left[\begin{array}{ll} \text{PARA} & \boxed{1}\ {}_{ppro}\left[\begin{array}{ll} \text{INDEX} & {}_{index}\left[\begin{array}{ll} \text{PER} & \textit{3rd} \\ \text{NUM} & \textit{sing} \\ \text{GEND} & \textit{fem} \end{array}\right] \end{array}\right] \\ \text{RESTR} & \{\ \} \end{array}\right] \\ \text{CONTEXT} & {}_{context}\left[\begin{array}{ll} \text{BACKGR} & \left\{ {}_{soa}\left[\begin{array}{ll} \text{RELN} & \textit{female} \\ \text{INST} & \boxed{1} \end{array}\right] \right\} \end{array}\right] \end{array}\right] \end{array}\right] \end{array}\right] $$

b.
```
\newbox\mybox \newbox\myboxtoo \avmoptions{center}

\setbox\mybox=\hbox{\begin{avm}
        \osort{ppro}{\[index & \osort{index}{\[ per & \it 3rd\\
        num & \it sing\\
        gend & \it fem \]} \]}
        \end{avm}}
\setbox\myboxtoo=\hbox{\begin{avm}
        \sort{soa}{\[reln & \it female \\
        inst & \@1 \]}
        \end{avm}}

\avmoptions{active,sorted}
\begin{avm}
[{sign} phonology\;\<\rm she\> \\
    synsem\;[{synsem} local & [{local} category & [{cat} head &
                                 \it noun{\sc \[case\;{\it nom}\]}\\
                                 subcat & \< \> ] \\
content & [{ref-obj} para & @1 \box\mybox \\
                     restr & \q\{ \q\}] \\
context & [{context} backgr & \{ \box\myboxtoo \} ]]]]
\end{avm}
```

It would be nice if avm.sty could be made more economical in its use of this TEX resource, but I don't know how to do that without compromising the functionality or the user interface of avm.sty.

5.3 Known problems

All known bugs were fixed for the Version 1.00 release. Please send bug reports and ideas for improvements to the email address given at the beginning of this document.