

# Andrew Bredenkamp

---

- [Loading \*avm.sty\*](#)
  - [Using \*avm.sty\*](#)
  - [Large avms](#)
  - [Advanced topics](#)
  - [Other packages](#)
  - [The package](#)
- 

Most people use Chris Manning's [avm.sty](#) for producing sorted and unsorted attribute-value matrices.

(If you just want to make a single avm, you could just use a LaTeX array environment (see below), but *avm.sty* is very simple, and much easier).

## Loading *avm.sty*

As with other packages, you can load *avm.sty* with the following line in the preamble (before the `\begin{document}`):

```
\usepackage{avm}
```

There are a number of options for *avm.sty* which can be set in the preamble, these define the type of avm you want and the fonts for values, sorts, etc. All examples given below assume that the following options are set:

```
\avmfont{\sc}
\avmoptions{sorted,active}
\avmvalfont{\rm}
\avmsortfont{\scriptsize\it}
```

## Using *avm.sty*

Using *avm.sty* in 'active/sorted' mode allows avms for 'typed feature structures' to be constructed quickly and clearly.

Here is an example:

```


$$\text{headed-phrase} \left[ \begin{array}{l} \text{SYNSEM} | \text{LOCAL} | \text{CAT} | \text{HEAD } \boxed{1} \\ \text{DTRS} \left[ \begin{array}{l} \text{HEAD-DTR} | \dots | \text{HEAD } \boxed{1} \end{array} \right] \\ \text{hd-str} \end{array} \right]$$

\begin{avm}
[ {headed-phrase}
  synsem|local|cat|head @{1} \\
  dtrs [ {hd-str}
    head-dtr|\ldots|head @{1}] ]
\end{avm}
```

The position of the sort label can be changed (see [the full documentation](#)).

It is very easy to get other sorts of bracket (again, see [the full documentation](#)).

Untyped avms of the kind used in LFG functional descriptions can be created in the same way (just give an

empty sort):

$$\begin{bmatrix} \text{PRED} & \text{'likes'} \langle -, - \rangle \\ \text{TENSE} & \text{pres} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{'Fido'} \end{bmatrix} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{'bones'} \end{bmatrix} \end{bmatrix}$$

```
\begin{avm}
[{} pred & `likes< \_ , \_ >'      \\
  tense & pres                      \\
    subj & [{} pred & `Fido']      \\
    obj  & [{} pred & `bones']]
\end{avm}
```

If you only ever produce avms without the sort, you can turn off the `sorted' option, and so not write the empty `{}' that was used above (again, see [the full documentation](#)).

The advantage of using the `active' option is that the commands for doing things like drawing the square brackets do not need a `backslash' in front of them. The disadvantage is that this means you cannot define a command to draw an avm (e.g. to draw the same avm twice). If you want to do this (and it is a very useful thing to do, especially when making large avms (a) because they can be difficult for LaTeX, and (b) because they can be difficult for you -- it is easier to avoid mistakes if you make your avms in small pieces separately, and then just combine the pieces together), you need the `inactive' option. We look at this in the next section.

## Large avms

Due to a limitation on LaTeX, large avms can cause problems (you might get errors referring to semantic nest size. To get round this, you have to build complex avms by nesting smaller ones (it is also easier to avoid mistakes if you do things like this). The best way to do this is by defining and using `boxes':

```
\newbox\matrixsynsem \newbox\headdtr
\newbox\compdtrone   \newbox\compdtrtwo

{\scriptsize
\avmoptions{center}
\setbox\matrixsynsem=\hbox{\begin{avm}
\osort{synsem}{\local & \cat & \[ \] \\
               content & \[ \]\]\]}
\end{avm}}

\setbox\headdtr=\hbox{\begin{avm}
\osort{word}{\[ phon\;\<\rm likes\>\\
             synsem\;\local\cat\;\head \; \@{5}\\
             arg-s\;\< \@{1},\@{2}\>\] \\
             content\;\@{6} \]\]\]} \end{avm}}

\setbox\compdtrone=\hbox{\begin{avm}
\sort{phrase}{\[ phon\;\<\rm fido\>\\
              synsem\;\@{1}\;\local\cat\;\head\;noun\\
              arg-s\;\< \>\] \\
              content\;\@{3} \]\]\]} \end{avm}}

\setbox\compdtrtwo=\hbox{\begin{avm}
\sort{phrase}{\[ phon\;\<\rm bones\>\\
              synsem\;\@{2}\;\local\cat\;\head\;noun\\
              arg-s\;\< \>\] \\
              content\;\@{4} \]\]\]} \end{avm}}

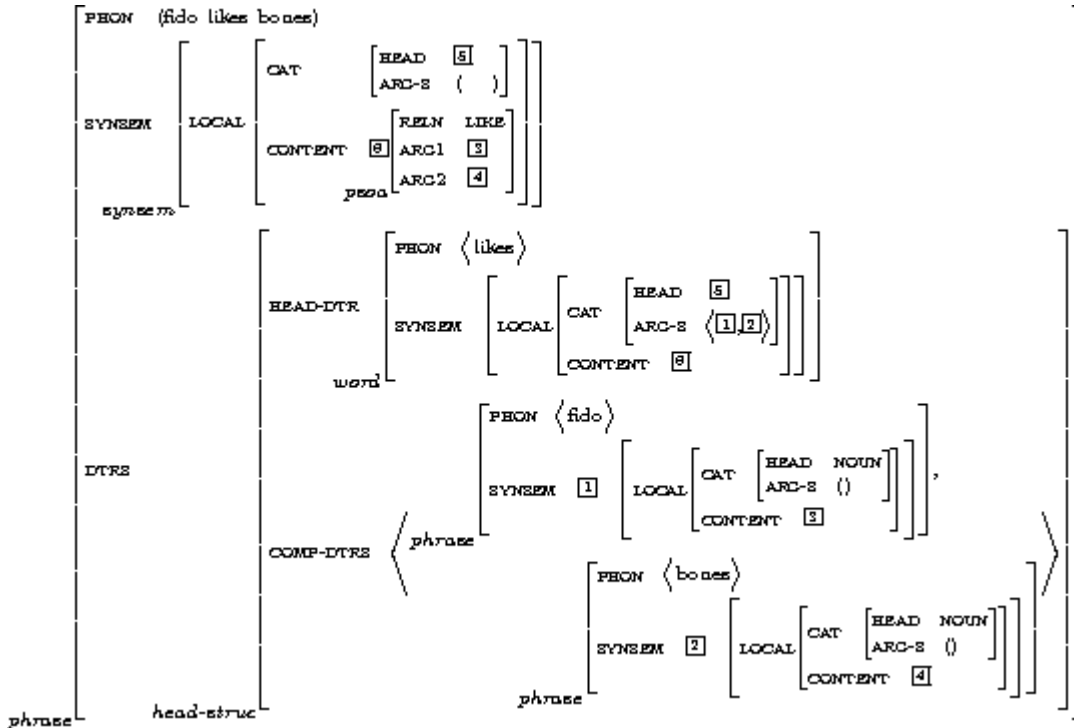
\avmoptions{active,sorted}
\begin{avm} \hspace{-1in}
[{}phrase] phon\;\<\rm fido likes bones\> \\
  synsem\;[{}synsem] local [{} cat & [{} head & @{5}\\
```

```

                                arg-s & \< \; \>] \\\
                                content & @{{6}}[{{psoa}} reln\;like \\\
                                                arg1\;@{{3}} \\\
                                                arg2\;@{{4}} ]]]\\
dtrs\;\sort{head-struct}{{{{}} head-dtr\;\box\headdtr \\\
                                comp-dtrs\;< \box\compdtrone , \\\ \hspace{.5in}
                                \box\compdtrtwo>]]]
\end{avm}}

```

produces:



## Advanced topics

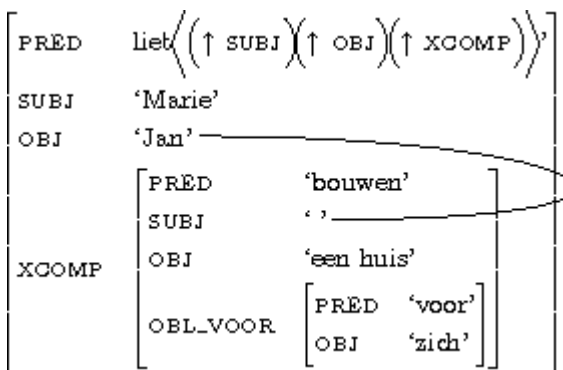
It is possible to draw curves between points inside an avm by using [tree-dvips.sty](#):

```

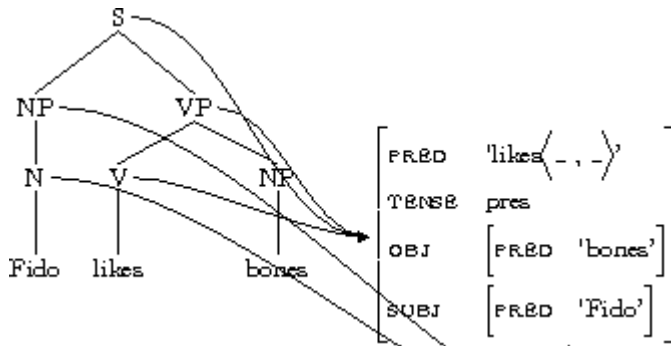
\begin{avm}
[{{}} pred & liet<($\uparrow$ subj)($\uparrow$ obj)($\uparrow$ xcomp)>' \\\
    subj & `Marie' \\\
    obj & \node{uobj}{{`Jan'}} \\\
    xcomp & [{{}} pred & `bouwen' \\\
        subj & \node{dsubj}{{` '}} \\\
        obj & `een huis' \\\
        obl_{voor} & [{{}} pred & `voor' \\\
            obj & `zich' ]]]
\end{avm}
\nodecurve[r]{uobj}[r]{dsubj}{2in}

```

will produce:



You can combine [trees](#) with avms using features from [tree-dvips.sty](#):



This was generated by the following code:

```
\begin{small}
\ vbox{
\ begin{tabular}{cccc}
& \ node{s}{S} \ \ [.3in]

\ node{np1}{NP} && \ node{vp}{VP} \ \ [.2in]

\ node{n}{N} & \ node{v}{V} && \ node{np2}{NP} \ \ [.3in]

\ node{fido}{Fido} & \ node{likes}{likes} && \ node{bones}{bones}
\ end{tabular}\ hspace*{.2in}
\ begin{avm}
\ node{f1}{[{} pred & `likes< \_ , \_ >' \ \
tense & pres \ \
obj & [{} pred & `bones'] \ \
subj & \ node{f2}{[{} pred & `Fido']] } ]}
\ end{avm}}
\ nodeconnect[b]{s}[t]{np1}
\ nodeconnect[b]{s}[t]{vp}
\ nodeconnect[b]{np1}[t]{n}
\ nodeconnect[b]{vp}[t]{v}
\ nodeconnect[b]{vp}[t]{np2}
\ nodeconnect[b]{n}[t]{fido}
\ nodeconnect[b]{v}[t]{likes}
\ nodeconnect[b]{np2}[t]{bones}
\ anodecurve[r]{s}[l]{f1}{.5in}
\ anodecurve[r]{vp}[l]{f1}{.3in}
\ anodecurve[r]{v}[l]{f1}{.3in}
\ anodecurve[r]{np1}[b]{f2}{1in}
\ anodecurve[r]{n}[b]{f2}{1in}
\ end{small}
```

## Other packages

There is an up-dated version, avm3e.sty, available. It seems to be backwards compatible with avm.sty. It is available from "[the usual places](#)". If you have only one avm to make, you could always just use a LaTeX array:

$$\left[ \begin{array}{l} TENSE \ pres \\ PRED \ 'like< \_ , \_ >' \\ SUBJ \ \left[ \begin{array}{l} PRED \ 'Fido' \\ PRED \ 'bones' \end{array} \right] \\ OBJ \ \left[ \begin{array}{l} PRED \ 'bones' \end{array} \right] \end{array} \right]$$

```
\[ % we go into display maths mode
\left [ % draw the opening left bracket
\begin{array}{ll} % begin a two column array
TENSE & pres\ \
PRED & `like<\_ , \_ >' \ \
SUBJ & \left [ \begin{array}{l} PRED `Fido' \\ PRED `bones' \end{array} \end{array}
```

```

                                PRED & 'Fido\\
                                \end{array}
                                \right ] \\
OBJ   & \left [ \begin{array}{ll}
                                PRED & 'bones'
                                \end{array}
                                \right ]
\end{array} % end the outer array
\right ]   % closing right bracket
\]         % out of math mode

```

# The package

The package is available here in the following bundles:

- [The package and documentation](#) - (*avm.sty*, *avm-doc.ps* and this guide *avm-intro.ps*)
- [The package](#) - (*avm.sty.gz*)
- [The documentation](#) - (*avm-doc.ps.gz*)
- [This mini documentation](#) - (*avm-intro.ps.gz*)

---

Last real modification: July 10, 1997  
 Last Modified (trivially) September 25, 2007.

---

[LaTeX for Linguists](#),  
[Doug Arnold](#),  
[doug@essex.ac.uk](mailto:doug@essex.ac.uk),  
 September 25, 2007.