# Pst-jTree, Version 2.4: Extensions and one bug fix

(September 2008)

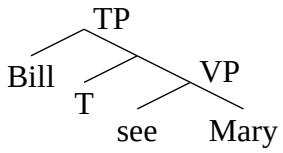
1. `\brokenbranch` and `\etcbranch` now work as advertised.

2. Previous versions of jTree assumed that the characters ^, >, <, and ", which all have special meaning to the jTree parser, had their normal category codes when the macro `\jtree` was invoked. This made jTree incompatible with Babel, gb4e, and any other macro packages which alter the category code of these characters. This incompatibility has now been removed. Invoking `\jtree` changes the category codes of these characters to the normal values. The change does not apply within labels, so the Babel shortcuts can be used in labels.

3. There is a new macro `\elc` (empty label comment) which can be used as follows.

---

```
\jtree
\! = {\elc{TP}}
:{Bill}
:{T} {\elc{VP}}
:{see} {Mary}.
\endjtree
```




---

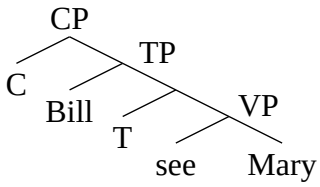
The term “comment” is used rather than “label” because the text is not treated as label by jTree. Like `{\pnode{...}}`, `{\elc{...}}` is considered to have an empty label, but there is a side effect in typesetting it. The argument of `\elc` is typeset in a certain position relative to the empty label. The position is determined by the parameters `elcxoffset`, `elcyoffset`, and `elcref`. The default settings are established by

```
\psset{elcxoffset=.8ex,elcyoffset=0,elcref=bl}
```

With these settings, `\elc{TP}` above (for example) is typeset by putting `TP` in an hbox and positioning its bottom left (`bl`) corner with respect to the empty node using the hoffset and voffset settings.

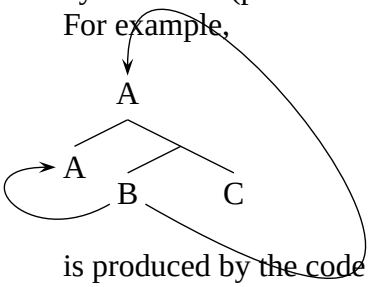
`\elc` takes parameters, as illustrated below. The default elc parameters are assumed to be in force. `elcxoffset` and `elcyoffset` are incremental parameters.

```
\jtree
\! =
{\elc[elcref=b,
   elcyoffset=.5ex,
   elcxoffset=0]{CP}}
:{C} {\elc[elcxoffset=!.5ex]{TP}}
:{Bill}
:{T} {\elc{VP}}
:{see} {Mary}.
\endjtree
```




\rput is used to typeset the node comments, so they do not contribute to size of the bounding box which jTree creates.

4. jTree adjusts the size of the hbox which \jtree. . . \endjtree creates so that it encloses all the labels. But paths drawn by PSTricks and text which is positioned by PSTricks (path annotations, for example) can fall outside this bounding box. For example,

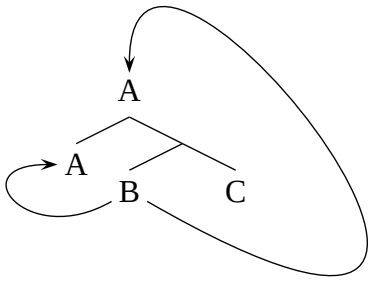



is produced by the code below.

```
For example,
\bigskip
\jtree[nodesep=.5ex]
\! = {A}@A1
:{A}@A3
:{B}@A2 {C}.
\nccurve[angleA=-30,ncurvA=9,
   angleB=90,ncurvB=4,arrows=->]{A2}{A1}
\nccurve[angleA=210,
   angleB=180,ncurv=3,arrows=->]{A2}{A3}
\endjtree
\bigskip
is produced by the code below.
```

It is usually sufficient to add suitable kerns to get such trees properly positioned. Many users, however, are not familiar with the Tex's kerning system. So jTree has been extended by providing a parameter bbadjust which can be used to adjust the size of the bounding box in tree constructions.

Contrast the tree spacing above with




which is produced by the code below.

```
Contrast the tree spacing above with
\bigskip
\jtree[nodesep=.5ex,bbadjust=height 4ex depth 3ex left 2em]
\! = {A}@A1
:{A}@A3
:{B}@A2 {C}.
\nccurve[angleA=-30,ncurvA=9,
   angleB=90,ncurvB=4,arrows=->]{A2}{A1}
\nccurve[angleA=210,
   angleB=180,ncurv=3,arrows=->]{A2}{A3}
\endjtree
\bigskip
which is produced by the code below.
```

The right edge of the bounding box above has not been adjusted even though the path projects out of the right side, because this is irrelevant here. But it can be adjusted in the obvious way if desired. Side by side trees might need such adjustment, for example.

The default settings are established in *pst-jtree* by

```
\psset{bbadjust=height 0pt depth 0pt left 0pt right 0pt}
```

Dimensions furnished to `bbadjust` must be Tex dimensions, not Postcript dimensions, and they must be absolute, not incremental.