The **synttree** package for typesetting syntactic trees.[*]

Matijs van Zuijlen[†]

November 7, 2004

Abstract

The **synttree** package provides a simple way to typeset syntactic trees as used in Chomsky's Generative Grammar.

Contents

2 Usage   2
   2.1 Package Options 2
   2.2 Defined Macros 2
   2.3 Examples 2

3 Wish List and Bugs   3

4 Known Bugs   3

5 Author and License   3

1 Introduction

The **synttree** package provides a macro for creating syntactic tree structures such as those used in Noam Chomsky's Generative Grammar. It is designed to create a tree that looks nice, with as little use of 'special effects', such as PostScript, as possible.

The current implementation can use either emTeX `\special`'s or LaTeX 2_ε's `\qbezier` macro to draw the lines in the tree, although the latter option is probably not very nice. The behavior defaults to the LaTeX 2_ε version.

Since the application is very specific, there is no need for a very complex drawing algorithm. A more complex package is the TreeTeX system [1]. This system, however, produces nodes consisting of a node symbol *and* a label, whereas in syntactic trees the label *is* the node symbol. Additionally, the method of specifying the tree structure itself makes the source code hard to read.

---

[*]Package version 1.3.1
[†]e-mail: `mvz@xs4all`; web: `http://www.matijs.net/`

2   Usage

2.1   Package Options

The following options are supported:

specials Use emTEX special commands.

nospecials Don't use emTEX special commands. This is the default.

2.2   Defined Macros

The main macro defined by this package is the `\synttree` macro. It takes as arguments first an optional parameter indicating the maximum level of the tree, and second a parameter describing the structure of the tree. Levels are surrounded by `[` and `]`.

Within a level, first, parameters may be added. These are specified by appending a `.` to the opening bracket, and appending letters that modify the appearance of the next level. Append a `t` to create a triangle instead of a line going from the label to the level above it, and a `b` to specify that the node has to appear at the bottom of the tree, on a line with the outermost branches. To use `b`, a maximum tree depth has to be specified. To combine the two features, use `x`.

Then, the label follows. If spaces are included in the label, surround them (or the entire label) by braces, or use `~`. After the label, the node's children follow. They are of course surrounded by brackets. No words may appear between the children.

The distance between levels can be adjusted by using `\branchheight`. It takes a length as its argument, and sets the distance to that length. Default is half an inch.
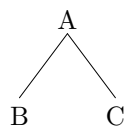
The balancing of the triangles can be adjusted by using `\trianglebalance`. It takes a number from 0 to 100, indicating the percentage of the triangle on the right side of its attachment point to the parent.

2.3   Examples
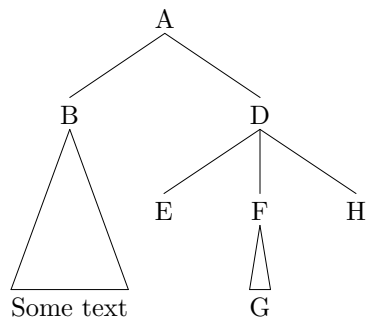
As a simple example, the line

    \synttree[A[B][C]]

creates the following tree:

A
B   C

Using the optional parameters, we may specify

    \synttree{4}[A[B[.x {Some text}]][D[E][F[.t G]][H]]]

to create:

A
B D
E F H
Some text G

3   Wish List and Bugs

I hope to add the following features sooner or later:

- Larger number of children. The current limit is three.

- Control over other parameters affecting the 'look' of the tree.

- The possibility of 'sinking' the tree on the current line, so that the top label aligns with the other text, instead of the bottom label.

- The possibility of creating trees within trees.

- Let labels ignore starting and trailing whitespace.

4   Known Bugs

Triangles on non-leaf nodes are not forbidden, but do not give the desired result.

5   Author and License

Copyright (c) 1998, 1999, 2001, 2004 Matijs van Zuijlen.

The `synttree` package is free software. It has been released under the terms of the Latex Project Public License, version 1.3a.

This work has the LPPL maintenance status 'maintained'. Its current maintainer is Matijs van Zuijlen. This work consists of the files `synttree.dtx` and `synttree.ins`.

References

[1] A. Brüggemann-Klein and D. Wood. *Drawing trees nicely with T$_E$X* File `tree_doc.tex`

[2] Jonathan Fine. *Some Basic Control Macros for T$_E$X* TUGboat, Volume 13 (1992), No. 1