

The `taxonomy` package*

Tikitu de Jager
tikitu@gmail.com

November 26, 2006

Abstract

A dirty horrible hack to get cases-like environments but with braces limited to the size of the labels at each level, not the entire rightward closure of the tree.

1 Introduction

1.1 Usage

Add `\usepackage{taxonomy}` to the preamble of your document.

2 The macros

Everything goes inside the environment `taxon` (read this as “one taxonomic entry”). You’ll be constructed nested trees of taxons.

Within a `taxon`, you have four useful macros.

- Use `\taxLabel` *first* to introduce the label of the current subtree. It takes an optional arg for the width of the minipage, and a mandatory arg for the content. It typesets the minipage in `raggedright`, to override this include whatever setting you like in your content. A label is required, even if it’s empty.
- Use `\taxFirst` for the first sub-entry. If this is a subtree it should contain only a `taxon` environment, but if it’s just text you don’t need any additional markup. So: either `\taxFirst{\begin{taxon}(subtree)\end{taxon}}` or `\taxFirst{text}`.
- Use `\taxMid` for all entries after the first and before the last. Even when you have only two entries, it’s sometimes useful to put an empty `\taxMid{}` in for spacing. Same rules apply as for `\taxFirst`: subtrees contain *only* a `taxon` environment, text entries are just text.
- Use `taxLast` for the last entry, with the same rules as for the other two.

*This document corresponds to `taxonomy` v0.2, dated 2006/11/26.

All three subtree macros take an optional vertical alignment arg, which can be [t], [m] or [b]. By default they will be aligned so that the brace exactly covers them, with the subtree rooted at the outside of the brace (that is, top line for `\taxFirst` and bottom line for `\taxLast`). For long labels this can look a bit daft, but unfortunately altering the alignment of the subtree will necessarily make the labels poke out outside the surrounding braces — this is unavoidable given the approach taken here. Check out the example to see this in action.

3 Bugs and warnings

A `\taxLabel` is required, even if it's empty. Required. As in, it won't work without one.

The tree will spread to stop anything bumping into its immediate children, but *not* necessarily its grandchildren or nieces and nephews. (This is pretty much unavoidable; the spacing works by forgetting the extra height contributed by the deeper reaches of the tree, and once it's forgotten it's gone for ever.) So you'll sometimes have to add space by hand — especially to stop the top edge of the tree bumping into material completely outside the taxonomy.

The spacing will go haywire if you don't nest and order your `taxon` environments and `label/first/mid/last` macros correctly. That's the first thing to check, if it looks odd.

Horizontal alignment between text-only and subtree entries isn't perfect. I can't see why at the moment, and to be honest I can't be bothered figuring it out.

This is a dirty dirty hack, and I can't see any way to make it much prettier. Don't expect much from it.

There's an example on the next page, shifted so that it will show up in landscape view.

4 An Example

This is rewritten from an example taxonomy by the person who originally asked how to do this. I've added empty `\taxMid` entries in a few places for clarity, and altered the alignment for a couple of entries to let you see how that looks. (“Generalissimis” and “Aliquid” are given [m] alignment; not much else would really make any sense.) You’ll probably need to view the page in landscape to see the whole taxonomy.

The horizontal line shows where the taxonomy thinks its top edge is; I’ve moved it down so it doesn’t collide with this text.

<hr style="border: 0.5px solid black; margin-bottom: 10px;"/> <p><i>Metaphysicæ</i> (quæ est GREEK, quatenus ab homine naturali rationis lumine sine ullo materiæ conceptu est intelligibile.) partes sunt duæ; Altera</p>	{	<p><i>Vniversalis</i>, quæ cōsistit in Intelligibilium & Entium tūm <i>Particularis</i>. Vide EE.</p>	{	<p><i>Generalissimis</i> distributionibus. <i>Intelligibilis</i> autem notetur.</p>	{	<p>λόγος: <i>Intelligibile</i> dicitur omni</p>	{	<p><i>Distributio</i>: Intelligibile est vel</p>	{	<p><i>Nihil</i>: 1 <i>Aliquid</i> quodsi non es Estq*</p>
	}		}		}		}		}	

5 Implementation

You probably don't want to read this. It's dirty hackery and most likely won't help you use the package better. It *certainly* won't teach you anything useful about L^AT_EX.

The plan is this: typesetting the *first* entry of a taxonomy we forget the height of the box, typesetting the *last* entry we forget the depth. We typeset the material in a cases-like environment regardless, then we add a zero-width vertical of the correct height, outside the cases braces.

We're going to keep track of heights and depths separately (it's profligate, I know):

```
1 \newdimen\tax@ht
2 \newdimen\tax@dp
3 \newdimen\tax@local@ht
4 \newdimen\tax@local@dp
5 \tax@ht=0pt
6 \tax@dp=0pt
```

We're also going to use a vertical alignment macro (user-specifiable if they really want to play silly-buggers):

```
7 \newcommand{\taxAlign}{m}
```

The `taxon` environment doesn't do much to start with, that's mostly handled by `\taxLabel` and `\taxFirst`. Closing down it resets the height and depth variables, which will have been messed with globalled by the subtree.

```
8 \newenvironment{taxon}{
9   \begin{math}
10 }{
11   \endarray
12   \right.
13   \vrule height\tax@ht depth\tax@dp width 0pt \relax
14   \global\tax@ht=\tax@local@ht
15   \global\tax@dp=\tax@local@dp
16   \end{math}
17 }
```

The setup for the cases-like braces is actually done by the `\taxLabel`, which is *not* a good idea but is close enough for government work. The braces definition is ripped out of the `amsmath` package, since a full `cases` environment does stuff we don't want with the spacing.

```
18 \newcommand{\taxLabel}[2][3cm]{%
19   \begin{minipage}[\taxAlign]{#1}
20     \raggedright #2
21   \end{minipage}%
22   \tax@local@ht=\tax@ht
23   \tax@local@dp=\tax@dp
24   \global\tax@ht=0pt
25   \global\tax@dp=0pt
26   \left\lbrace
27   \array{@{}l@{}}
28 }
```

The `\taxFirst` definition throws the material into a box, remembers the height and depth of the box, then smashes the height and typesets it. The height and

depth are stored globally, because each cell of an array is a separate group; we can't use group nesting to restore the correct value except at the outermost level of the `taxon` environment.

```
29 \newcommand{\taxFirst}[2][t]{%
30   \def\taxAlign{#1}%
31   \setbox0\hbox{#2}%
32   \global\advance\tax@ht by \ht0\relax
33   \global\advance\tax@dp by \dp0\relax
34   \ht0=0pt \box0\relax\
35 }
```

The `\taxMid` definition is very slightly simpler, since it doesn't need to mess with the size of the box coming out.

```
36 \newcommand{\taxMid}[2][m]{%
37   \def\taxAlign{#1}%
38   \setbox0\hbox{\tabular{@{}l@{}}#2\endtabular}%
39   \global\advance\tax@ht by \ht0\relax
40   \global\advance\tax@dp by \dp0\relax
41   \box0\relax\
42 }
```

And `\taxLast` looks just like you'd expect by now.

```
43 \newcommand{\taxLast}[2][b]{%
44   \def\taxAlign{#1}%
45   \setbox0\hbox{#2}%
46   \global\advance\tax@ht by \ht0\relax
47   \global\advance\tax@dp by \dp0\relax
48   \dp0=0pt \box0\relax\
49 }
```

That's all, folks. Ugly, ain't it?