Hints and Tricks

‘Hey — it works!’

Jeremy Gibbons

Welcome to *‘Hey — it works!’*, a column devoted to (LA)TEX tips, tricks and techniques. In this issue we have three articles: one from Jeroen Nijhof showing how to get multi-letter ‘initials’ in BIBTEX; one from me, showing how to use a text symbol (in this case, a hyphen) as an operator in maths; and one from Christina Thiele on generating ornamental rules. My backlog of articles is running low, so please send them in!

◇ Jeremy Gibbons
CMS, Oxford Brookes University
Oxford OX3 0BP, UK
jgibbons@brookes.ac.uk
http://www.brookes.ac.uk/~p0071749/

1 Controlling abbreviations in BibTEX

In the TEX newsgroup, `comp.text.tex`, someone asked how to get BIBTEX to abbreviate Yuri to ‘Yu.’, in styles in which the author’s first names are abbreviated — after all, ‘Yu’ is one letter in Russian. Take for example

```
@Article{govorukhin,
  author =  {Yuri Govorukhin},
  title  =  {What we sow,
             that we reap the fruits of},
  ... }
```

How to get the author abbreviated to ‘Yu. Govorukhin’? Simply putting braces around the ‘Yu’ does not work. Somewhat surprisingly, BIBTEX will not abbreviate `{{Yu}ri Govorukhin}` at all.

But if one defines a macro `\Yu` in the the preamble that expands to ‘Yu’,

```
@Preamble{"\newcommand\Yu{Yu}"}
```

then `{{\Yu}ri Govorukhin}` will be abbreviated to ‘Yu. Govorukhin’.

But as Oren Patashnik, the author of BIBTEX, remarked, this solution has one problem. Namely, the ‘Yu’ will be neglected for sorting purposes, so this way the author would be sorted as if he were ‘ri Govorukhin’. To get both a proper abbreviation and proper sorting, one needs to use a feature of BIBTEX (which is discussed in the BIBTEX documentation `btxdoc.dvi`): BIBTEX considers a special character, which is everything from a left brace at the top level directly followed by a backslash up to the matching right brace, as one character, and all characters in it will be taken into account for sorting purposes. So one can define a one-parameter macro `\oneletter`:

```
@Preamble{"\newcommand\oneletter[1]{#1}"}
```

and `{{\oneletter{Yu}}ri Govorukhin}` will give the desired effect. There are more tricks in the preamble of `xampl.bib` in the BIBTEX distribution.

In our case, though, we can get by without a preamble. The macro in the special character does not have to do anything, it only has to provide a backslash after the opening brace. So we can simply write

```
author = {{\relax Yu}ri Govorukhin}
```

and here is some text.

◇ Jeroen H. B. Nijhof
Aston University
Photonics Research Group
Birmingham, UK
J.H.B.Nijhof@aston.ac.uk

2 A small minus sign

In a recent paper (*Deriving Tidy Drawings of Trees*, Journal of Functional Programming **6**(3) p. 535–562, May 1996) I had to make a visible distinction between unary minus (for example, the number ‘minus three’) and binary minus (for example, the subtraction ‘four minus three’). I decided to keep the ordinary minus symbol for subtraction, and to use a hyphen for negation: ‘$2 - \text{-}3$’.

The quick-and-dirty way to achieve this is to use an `\hbox`:

```
\def\minus{\hbox{-}}
```

This looks fine most of the time (‘$2 - \text{-}3$’). Unfortunately, the hyphen does not change size in superscripts and subscripts; compare ‘$x_{y_{2-\text{-}3}}$’ with how it should look (‘$x_{y_{2-\text{-}3}}$’).

A better way is to use TEX’s `\mathchardef`:

```
\mathchardef\minus="002D
```

This defines `\minus` to be an ordinary symbol (the first `0`), taken from maths family zero (the second `0`), using the character in position `45` (the `2D`) of that font, which in the case of Computer Modern Roman happens to be the hyphen character. (There are no guarantees in other fonts, of course!) The LATEX 2_ε way of writing this is

```
\DeclareMathSymbol{\minus}
  {\mathord}{operators}{"2D}
```

but this has to go in the preamble of the document.

◇ Jeremy Gibbons
Oxford Brookes University
jgibbons@brookes.ac.uk

3 Ornamental rules

Something you run across in books on occasion are 'type ornaments': those looping trellises and lattices and swirls often used to indicate the top or bottom of a titlepage, for example, or sometimes the top of a chapter titleblock (called 'headbands', I believe), or to separate sections within a chapter. What I noticed was that sometimes the ornaments were nothing more than a fairly simple character repeated a few times:

$\div \quad \div \quad \div \quad \div \quad \div \quad \div \quad \div$

or all the way across the page, like this:

◇◇

So how can I get TeX to do this? Poking around in the *The TeXbook* (pp. 353 and 357), I found the following macros:[1]

```
\def\m@th{\mathsurround=0pt }
\def\dotfill{\cleaders\hbox{$\m@th
  \mkern1.5mu . \mkern1.5mu$}\hfill}
```

(the `\m@th` is to switch off spacing around mathematical material, which complicates the measurements) You can use this in a table of contents, for example:

3. Typographic use of leaders 45

So that's how to get a repeating pattern going. Here is the macro that does all the work:

```
\mathsurround=0pt
\def\motif#1{\cleaders\hbox{#1}\hfil}
```

It's now a simple matter to replace the period and `\mkern` amounts with other values. Here's a definition using the `\diamond`, which I wanted to have very close together:

```
\def\diamondpattern{$\mkern-.6mu
  \diamond \mkern-.6mu$}
\motif{\diamondpattern}
```

But this only defines the repeating motif. To make use of it, I've got a macro which takes the motif, and allows you to specify how long the sequence should span: e.g., 5cm or even full page width:

```
\def\border#1#2{\leavevmode
  \hbox to #2{\motif{#1}}}
```

The first argument to `\border` is a motif to repeat, and the second argument is a dimension; for example,

```
\def\divpattern{$\div \mkern12mu$}
\border{\divpattern}{5cm}
```

or, to span the full width of the column,

```
\border{\divpattern}{\columnwidth}
```

[1] I'm sure there are other ways to do this; so far, this one works fine for me.

The macro is designed to be used in horizontal mode, so it can also be used inside LaTeX's `center` environment, as in the first example, above.

Some tips:

1. First place to raid: `cmmi` and `cmsy` have lots of promising items; I've already tried `\diamond`, `\vee`, `\wedge`, `\dagger` and `\ddagger`, `\bowtie`, `\div`, `\widetilde`, ...
2. Other fonts which may contain some surprises: `msam` and `msbm`, and of course, any dingbat font.
3. The value of `\mkern` must be given in '`mu`' math units; there are 18mu in 1em (see p. 68 of *The TeXbook*). While putting `\mkern` units both before and after the symbol works for some characters, others work better if only used after. Changing the spacing can really alter the flavour of the ornamental line, so experiments are useful.
4. Note that `\widetilde` needs an argument in order to know how wide it must be: I used `\widetilde{\qquad}`:

   $\widetilde{\qquad} \quad \widetilde{\qquad} \quad \widetilde{\qquad}$

5. It's also fun to combine symbols: `\times\div` yields ' $\times\div\times\div\times\div\times\div\times\div$ '. Making the pattern start and finish with the same symbol is a little trickier. Because the pattern produced by `\cleaders` is centered within the area to be filled, it may be padded with a little extra space at either end, which makes positioning the extra symbol difficult:

   $\times\div\times\div\times\div\times\div\times\div\ \times$

   A better solution is to use `\leaders` instead, as these are aligned with the smallest enclosing box (see p. 224 of *The TeXbook*); then the extra symbol can be placed at the start, with no intervening space. The width of the repeating section is the stated width less the width of the first symbol.

   ```
   \def\bordertwo#1#2#3{{%
     \setbox1=\hbox{#1}%
     \dimen0=#3\advance\dimen0 by -\wd1
     \leavevmode\hbox{#1\hbox to \dimen0
       {\leaders\hbox{#2#1}\hfil}}%
   }}
   ```

   Then '`\bordertwo{$\times$}{$\div$}{3cm}`' yields

   '$\times\div\times\div\times\div\times$ '

   (which is no longer nicely centered, but that's life).

◇ Christina Thiele
Nepean, Ontario, Canada
cthiele@ccs.carleton.ca