
Preserving the math class of variables

Hans Hagen

If there is one thing that OpenType math has made clear, it's that we have lots of alphabets. It is customary in a \TeX document to key in regular (ASCII) letters and expect them to become for instance math italic, bold upright, script or whatever.

One way to do this is to relate a character (directly or by name) to a specific slot in a font assigned to a so-called math family, which groups text, script and scriptscript sizes. Here are a couple ways to do this, using the Unicode `\Umathcode` primitive:

```
\Umathcode'a = "0 "9 'a
\Umathcode'a = "0 "5 "1D44E
```

In the first line we map the input character `a` (the first `'a`) to the glyph slot of `'a` (the second one; that is, 97) in family 9. In the second line, the input `a` is mapped to the Unicode math italic alphabet's `a`, using family 5. The `"0` in both lines is the math class, in this case specifying an “ordinary” character.

Switching families can be done directly, although more usually it is wrapped in a command:

```
$ a + {\fam"9 a} + {\fam"5 a} $
```

For our next example, we take a colon from family zero (`"0`) and assign it class 6 (`"6`) which means that it will get punctuation spacing (like `\Colon`):

```
\Umathchardef\foo "6 "0 ': % punct
```

In the following line we do the same but with class 7, which is “variable”, meaning \TeX uses the current family, as stored in the `\fam` primitive parameter.

```
\Umathchardef\foo "7 "0 ': % ord
```

Doing this, we lose the prior class value (3), so we end up with ordinary (which normally means no) spacing. In Lua \TeX (>1.15.1) we can now preserve the class by declaring and using a special “variable” family instead:

```
\variablefam"24
\Umathchardef\foo "6 "24 123 % punct
```

When a character has family `\variablefam` assigned, it will get the current `\fam` value and the class can remain 3, as specified.

This is a relatively cheap extension which we prototyped in LuaMeta \TeX and backported to Lua \TeX . We don't use this in Con \TeX t (just to warn its users) but it might be handy in other macro packages.

◇ Hans Hagen
Pragma ADE