Preparing for 16-bit math fonts with Ω

John Plaice
School of Computer Science and Engineering
The University of New South Wales
Sydney 2052, Australia
plaice@cse.unsw.edu.au
http://www.cse.unsw.edu.au/~plaice

Yannis Haralambous
Atelier Fluxus Virus
187, rue Nationale
F-59800 Lille, France
yannis@fluxus-virus.com
http://www.fluxus-virus.com

Introduction

One of the fundamental limitations of TEX3 is that most quantities are limited to 8-bit values. Fonts are limited to 256 characters each, only 256 fonts are allowed simultaneously, only 256 of any given kind of can be used simultaneously, etc. Ω loosens these restrictions, allowing 65 536 (0–65 535) of each of these entities. In this paper, we give a complete summary of the 16-bit capacities of Ω, with a focus on 16-bit mathematics.

A forthcoming version of LATEX package `yhmath` will use the math font features to implement (even) bigger delimiters of all kinds.

Characters

Each font can allow up to 65 536 characters, ranging between 0 and 65 535. Unless other means are provided, using Ω Translation Processes, the input and output mechanisms for characters between 256 (hex 100) and 65 535 (hex ffff) use four circumflexes. For example, `^^^^cab0` means hex value `cab0` and `^^^^0020` is the space character.

Fonts

Up to 65 536 fonts may be used. Since font numbers are never explicitly mentioned, all of this is handled automatically, and space is allocated as needed.

Registers

Up to 65 536 registers of each kind may be used. For each of `\count`, `\dimen`, `\skip`, `\muskip` and `\box`, one can place a number from 0 to 65535. Note that `\box255` remains the box used by the output routine.

Math fonts

There has been a lot of work recently on Math fonts (see `http://www.tug.org/twg/mfg/`). One of the severe restrictions that is constantly being faced is that TEX only allows the use of 16 (2^4) font families, where each font contains 256 (2^8) characters. To access the characters in the math fonts, and to define how they are to be used, there are several basic primitives:

- `\mathcode` ⟨*8-bit number*⟩ = ⟨*15-bit number*⟩:
  Defines 15-bit math code for character;
- `\mathcode` ⟨*8-bit number*⟩:
  Outputs 15-bit math code associated with character;
- `\mathchar` ⟨*15-bit number*⟩:
  Generates a math character with 15-bit math code;
- `\mathaccent` ⟨*15-bit number*⟩:
  Generates a math accent with 15-bit math code;
- `\mathchardef` ⟨*control-sequence*⟩ =
  ⟨*15-bit number*⟩ :
  Defines a control sequence with a 15-bit math code;
- `\delcode` ⟨*8-bit number*⟩ = ⟨*27-bit number*⟩:
  Defines 27-bit delimiter code for character;
- `\delcode` ⟨*8-bit number*⟩:
  Outputs 27-bit delimiter code associated with character;
- `\delimiter` ⟨*27-bit number*⟩:
  Generates a math delimiter with 27-bit delimiter code;
- `\radical` ⟨*27-bit number*⟩:
  Generates a math radical with 27-bit delimiter code;

where

- ⟨*8-bit number*⟩ means an 8-bit character;
- ⟨*15-bit number*⟩ means value `0x8000` or a triple
  - 3 bits for math category,
  - 4 bits for font family,
  - 8 bits for character in font,

  called a *math code*;
- ⟨*27-bit number*⟩ means a negative number or a quintuple
  - 3 bits for math category,
  - 4 bits for first font family,
  - 8 bits for first character in font,
  - 4 bits for second font family,
  - 8 bits for second character in font,

  called a *delimiter code*.

Ω, on the other hand, allows 256 (2^8) font families (yes, that means 768 math fonts!), where each font can contain 65 536 (2^{16}) characters. So, in addition to the TEX math font primitives, which continue to work, 16-bit versions are needed.

The way it works is that for each *primitive*, there is also an \\o*primitive*, with the following substitutions:

⟨*8-bit number*⟩ → ⟨*16-bit number*⟩
⟨*15-bit number*⟩ → ⟨*27-bit number*⟩
⟨*27-bit number*⟩ → ⟨*51-bit number*⟩

where

- ⟨*16-bit number*⟩ means a 16-bit character;
- ⟨*27-bit number*⟩ means value `0x8000000` or a triple
  - 3 bits for math category,
  - 8 bits for font family,
  - 16 bits for character in font,

  called an *Ω math code*;
- ⟨*51-bit number*⟩ means a pair of numbers; either both are negative or they must be arranged as ⟨*27-bit number*⟩ ⟨*24-bit number*⟩, with the first number being:
  - 3 bits for math category,
  - 8 bits for first font family,
  - 16 bits for first character in font,

  and the second number being:
  - 8 bits for second font family,
  - 16 bits for second character in font,

  called an *Ω delimiter code*.

Here are the new primitives:

- `\omathcode` ⟨*16-bit number*⟩ = ⟨*27-bit number*⟩ :
  Defines 27-bit math code for character;
- `\omathcode` ⟨*16-bit number*⟩:
  Outputs 27-bit math code associated with character;
- `\omathchar` ⟨*27-bit number*⟩:
  Generates a math character with 27-bit math code;
- `\omathaccent` ⟨*27-bit number*⟩:
  Generates a math accent with 27-bit math code;
- `\omathchardef` ⟨*control-sequence*⟩ = ⟨*27-bit number*⟩ :
  Defines a control sequence with a 27-bit math code;
- `\odelcode` ⟨*16-bit number*⟩ = ⟨*51-bit number*⟩:
  Defines 51-bit delimiter code for character;
- `\odelcode` ⟨*16-bit number*⟩:
  Outputs 51-bit delimiter code associated with character;
- `\odelimiter` ⟨*51-bit number*⟩:
  Generates a math delimiter with 51-bit delimiter code;
- `\oradical` ⟨*51-bit number*⟩:
  Generates a math radical with 51-bit delimiter code;

Since Ω is upwardly compatible with TEX, the older primitives continue to function as expected. Internally, math codes are 27-bit numbers and delimiter codes are 51-bit numbers. However, if a `\mathcode`⟨*15-bit number*⟩ occurs in text mode, a 15-bit number continues to be generated, to remain upwardly compatible with TEX. For example, file `plain.tex` contains these four lines

```
\mathchardef\@cclvi=256
\mathchardef\@m=1000
\mathchardef\@M=10000
\mathchardef\@MM=20000
```

When any of `\@cclvi`, `\@m`, `\@M` and `\@MM` act as numerical constants, the correct values are inserted.