How many sizes of radicals do we need?

Matthias Clasen

1 The current approach

In his "Technical Report on Math Font Encoding", Justin Ziegler states that

> The present `cmsy` font contains one glyph that is set in a strange way — the radical sign, and thus makes that whole font unusable for the outer world. It would be a good idea to make sure that this does not happen again.
>
> … [`cmsy`] has always been loaded in three sizes, and must remain so. If it [the radical] is taken out of `cmsy`, and put in a `cmex` replacement, then this point must be taken into consideration.

And he comes to the conclusion that

> If the new `cmex` is loaded in one size, it must contain three different sizes of the radical in order to stay compatible with plain.

This makes script and scriptscript sizes of the radical available even if the MXP-encoded font is loaded in only one size. But it has the very ugly side effect of fixing the text/script/scriptscript size ratios, i. e. a MXP-encoded font designed for 10 pt / 7 pt / 5 pt will not be usable with e. g. 10 pt / 8 pt / 6 pt.

2 A new approach

If we assume that loading three sizes of the MXP-encoded font will be the normal situation and loading one size will only be necessary for compatibility with old documents, then the cleaner solution would be to put only a textsize radical into MXP.

I claim that we can do this and still enable loading one size. The idea is to use three sizes *locally* for radicals. The following macros are my first try at such a solution, inspired by some old macros to set math material in cramped style:

```
\def\fsqrt#1{{\mathpalette\@fsqrt{#1}}}
\def\@fsqrt#1#2{%
  % set #2 in cramped style #1
  \setbox0=\hbox{%
     \nulldelimiterspace=0pt
     $\m@th#1\radical0{#2}$}%
  % remove the extra vertical space
  % added by \radical (see appendix G)
  \ifx#1\displaystyle
     \dimen0=\fontdimen8\textfont3
     \advance\dimen0 .25\fontdimen5\textfont2
  \else \dimen0=1.25\fontdimen8
     \ifx#1\textstyle \textfont
     \else \ifx#1\scriptstyle \scriptfont
          \else \scriptscriptfont
          \fi
     \fi 3
  \fi
  \advance\dimen0-\ht0 \ht0=-\dimen0
  % add the radical, using exscale locally
  \setbox1=\hbox{%
     $\m@th\exsc@le#1\fsqrtsign{\box0}$}%
  \box1\relax}
```

The macro `\exsc@le` has to change the font allocations for the family from which we take the `\fsqrtsign` to use three sizes.

3 An example

My simple test version of `\exsc@le` (ignoring NFSS) used in this document is:

```
\font\fsyten=cmsy10
\font\fsyseven=cmsy7
\font\fsyfive=cmsy5
\def\exsc@le{%
 \textfont\symfsym=\fsyten
 \scriptfont\symfym=\fsyseven
 \scriptscriptfont\symfsym=\fsyfive}
```

For demonstration purposes within this document, I take the `\fsqrtsign` from a symbol font containing only one size of `cmsy`:

```
\DeclareFontFamily{U}{fsym}{}
\DeclareFontShape{U}{fsym}{m}{n}
   {<->sfixed*cmsy10}{}
\DeclareSymbolFont{fsym}{U}{fsym}{m}{n}
\DeclareMathRadical{\fsqrtsign}
   {fsym}{"70}{largesymbols}{"70}
```

Now compare

$$\sqrt{*}^{\sqrt{*}^{\sqrt{*}}} \sqrt{*^{*^{*}}}\ *^{*^{*}} \qquad \text{(using \texttt{\textbackslash sqrt})}$$

$$\sqrt{*}^{\sqrt{*}^{\sqrt{*}}} \sqrt{*^{*^{*}}}\ *^{*^{*}} \qquad \text{(using \texttt{\textbackslash fsqrt})}$$

$$\sqrt{*}^{\sqrt{*}^{\sqrt{*}}} \qquad \text{(using \texttt{\textbackslash fsqrt}, * from \texttt{fsym})}$$

I'd very much like to hear your opinions on the practicability of this approach.

◇ Matthias Clasen
Albert-Ludwigs-Universität Freiburg
Institut für Mathematische Logik
D-79104 Freiburg, Germany
clasen@mathematik.uni-freiburg.de