Abstract
The macros on page 299 of The METAFONT book, which generalize METAFONT’s mediation operation, have some bugs which went unnoticed for years. This article discusses how to fix the bugs, and some other improvements to the macros.

1 The problem
METAFONT’s mediation operation allows us to write

- \( \frac{1}{3}[z_1,z_2] \) for the point one-third of the way from \( z_1 \) to \( z_2 \),
- \( \frac{1}{2}[z_1,z_2] \) for the point midway between \( z_1 \) and \( z_2 \),
- \( .8[z_1,z_2] \) for the point eight-tenths of the way from \( z_1 \) to \( z_2 \),

and, in general, \( t[z_1,z_2] \) stands for the point that lies a fraction \( t \) of the way from \( z_1 \) to \( z_2 \).

Our goal is to extend METAFONT’s syntax so that it will accept generalized mediation formulas like \( 1/2[z_1,z_2,z_3] \) and \( .4[z_1,z_2,z_3,z_4] \), computed as in the construction of Bézier curves (see Figure 1).

2 The original macros
Page 299 of The METAFONT book gives some macros that implement the generalized mediation operation. The basic idea is to make \( [ \) a macro that counts how many comma-separated expressions follow, up to the matching \( ] \). If there are fewer than three, as in any of

\[
\begin{align*}
\text{path } p[1] a \\
x[n] \\
1/3[z_1,z_2]
\end{align*}
\]

we don’t need to do anything special, so we restore the expressions in primitive brackets. Otherwise we store away the expressions and make

\[
\begin{align*}
[a,b,c] & \text{ expand to } \text{Bernstein 3}, \\
[a,b,c,d] & \text{ expand to } \text{Bernstein 4},
\end{align*}
\]

The binary-operator-like Bernstein macro then absorbs the fraction to the left and computes the result

\[
t[u_1, \ldots, u_n] = \sum_{k=1}^{n} \binom{n-1}{k-1} (1-t)^{n-k} t^{k-1} u_k.
\]

However, the METAFONT book macros have two bugs which can cause innocent commands like

\[
draw \text{flex}(0,0),(100,100),(300,0);
\]

to stop working. The first bug is easy to fix: rename the private variable \( n_\) to \( bn_\) to avoid a name conflict with the \( n_\) in plain METAFONT’s flex routine. The second bug is harder to find: the definition of flex

\[
\text{primarydef } t \text{ Bernstein } nn = \text{begingroup} \\
(a_1, a_2, \ldots, a_{n+1}) := 1; \text{ for } n = 1 \text{ upto } nn - 1: \\
c_1[1] := t \ast c_1[nn]; \\
\text{for } k = n \text{ downto } 2: c_1[k] := t[\text{c}_1[k]] \ast c_1[k - 1]; \text{ endfor} \\
c_1[1] := (1 - t) \ast c_1[1]; \text{ endfor} \\
bn_ := 0; \text{ for } u = u_1 + c_1[\text{incr } bn_] \ast u \text{ endfor} \text{ endgroup enddef};
\]

The first improvement is that \( [ \) and \( ] \) are changed to macros which expand separately; this allows the \( ] \) to be buried in another macro like \( ] \), a single token which plain METAFONT expands to \( ] \). The second improvement is that the expressions between \( [ \) and \( ] \) are now stored in a “list macro” instead of an array. This makes the code simpler, readily adaptable to new types like METAPOST colors, and diagnostics with show and showdependencies more readable:

\[
* \text{show } 2[a,b,c];
\]

\[
>> 4c-4b+a \quad (\text{formerly } u_1 \text{ or } \%\text{CAPSULE4691})
\]

I gratefully thank Donald Knuth for suggesting that I write this note.

\[
\diamond \text{ Hu Yajie} \\
\text{https://github.com/dine2014}
\]