Generalized mediation operation in METAFONT

Hu Yajie

Abstract

The macros on page 299 of *The METAFONTbook*, 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

- `1/3[z1,z2]` for the point one-third of the way from z_1 to z_2,
- `1/2[z1,z2]` for the point midway between z_1 and z_2,
- `.8[z1,z2]` for the point eight-tenths of the way from z_1 to z_2,

and, in general, `t[z1,z2]` 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[z1,z2,z3]` and `.4[z1,z2,z3,z4]`, computed as in the construction of Bézier curves (see Figure 1).

2 The original macros

Page 299 of *The METAFONTbook* 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

```
path p[][]a
x[n]
1/3[z1,z2]
```

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

`[a,b,c]` expand to `Bernstein 3`,
`[a,b,c,d]` expand to `Bernstein 4`,
...

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 *METAFONTbook* macros have two bugs which can cause innocent commands like

```
draw 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` says `z_[incr n_]`, which usually increases $n_{\_}$ once. But the `[` macro evaluates the expressions up to the matching `]` twice, once to count their number and once in primitive brackets, so $n_{\_}$ gets increased twice and you get index mismatch errors. This bug can be fixed by changing `if bn_<3: [[[t]]]` on line 6 to

```
    if bn_=0: [[[]]]
elseif bn_=1: [[[u_[[[1]]] ]]]
elseif bn_=2: [[[u_[[[1]]], u_[[[2]]] ]]]
```

to reuse the result of the first-time evaluation.

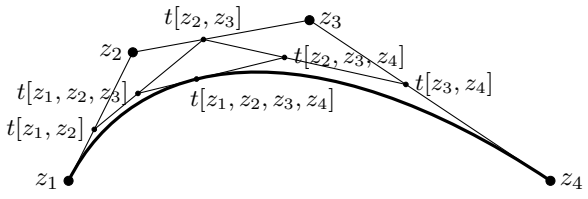

Figure 1: The generalized mediation operation

3 The improved macros

While fixing the bugs, I also discovered some other improvements to the *METAFONTbook* macros:

```
let [[[ = [; let ]]] = ];
def [ = for u = enddef;
def ] = , hide(bn_ := 0; let v_ = \; ):
 if incr bn_ = 1: hide(def v_ = u enddef)
 else: hide(expandafter def expandafter v_
  expandafter = v_, u enddef) fi endfor
 if bn_ < 3: [[[v_]]]
 else: Bernstein bn_ fi enddef;
primarydef t Bernstein nn = begingroup
 c_[[[1]]] := 1; for n = 1 upto nn - 1:
  c_[[[n + 1]]] := t * c_[[[n]]];
  for k = n downto 2: c_[[[k]]] :=
   t[[[c_[[[k]]], c_[[[k - 1]]] ]]]; endfor
  c_[[[1]]] := (1 - t) * c_[[[1]]]; endfor
 bn_ := 0; for u = v_: + c_[[[incr bn_]]] * u
 endfor 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:

```
*show 2[a,b,c];
>> 4c-4b+a  (formerly u_1 or %CAPSULE4691)
```

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

◇ Hu Yajie
https://github.com/dine2014

doi.org/10.47397/tb/42-1/tb130hu-bernstein