Radical delimiters

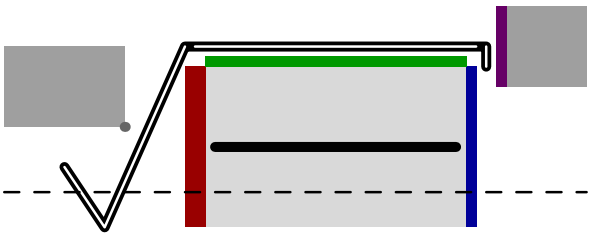
Hans Hagen, Mikael P. Sundqvist

Every TEX user who typesets math knows that left and right fences (parentheses) can grow with what they span. The same is true for the rule in a fraction, wide accents and braces (for instance) on top of a subformula. These are called horizontal and vertical extensibles. There are also special extensibles like integrals, sums and products. Integrals sometimes can grow indefinitely but the latter two come in a limited set of variants. Even (for instance) parentheses start with stepwise larger variants before we end up with an extensible.

A math radical is also an extensible: the left part of this symbol can grow but in traditional TEX the bar at the top is a rule. There is no real concept of a two-dimensional extensible and for reasons unknown to us OpenType didn't bother to add them. That would also introduce a right part being supported. In the next abstraction we show a bunch of properties that we have to deal with.

Because we have no hope that this will become available we've rolled our own. The middle piece can be a glyph like with any extensible and we support a right piece. For this the engine was adapted. But it's not enough. When a variant grows, the angle might change slowly till we go upwards. In many fonts the number of variants is not enough to accommodate proper rendering; think of plain symbols, symbols with a script, fractions, fractions that themselves contain symbols with scripts. It can be hard to come up with a configuration that works well for each of them when we lack variants.

The next graphic shows what we're dealing with. Here the radical shape is made from a single left, repeated middle and a single right piece but the left and right ones can also be made from pieces, when they are upright.



When a variant is sufficiently sloped, there is a danger that it will clash with the content, so we need some kerns that depend on the shape. In the picture above, they're shown as the vertical bars at left and right under the radical in red and blue, respectively (grayscaled for *TUGboat*), and hopefully also visible in this example:

This is a character-specific property. We already have the distance between content and top as a parameter (horizontal bar at top, in green) and of course this is different for text and display math. Then we have the degree of the radical, which has its own vertical positional parameter but again we need something per size. Because we have only a shared parameter the leftmost part of the symbol is always the same, although we could abuse some depth trickery here. So we need proper anchors so that the degree can stick out to the left of that anchor (gray dot) instead of using some heuristic (if at all: we can also overlap). This shape-dependent margin is not to be confused with margins that we add in ConTEXt, at the left and/or right, as well as enforced by struts.

We can even think of kerning between the radical left symbol and the first one seen in the content. This is kind of complicated because we get a chicken-egg situation as the symbol depends on the content and if that becomes wider we need to recalculate the assembly. So for now there is no extra kern (after the red) even if we do support kerning in some cases.

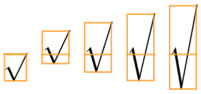
The radical as whole also has properties, for instance the right symbol can demand some top kern (magenta). Actually a prescript will kern before the left symbols but is not needed in case of a degree and given the white areas there, such a kern is unlikely. And the user might also expect the radicals in a formula made from more than one radical to have the same size. There is also the math axis (heavy black line) to deal with because the symbol gets vertically centered over the content.

So we have quite a few extra shape-dependent properties to deal with: margins, offsets and (corner) kerns. We also need multiple passes in order to meet demands like comparable sizes and calculating content dimensions that are needed for the sizing. Keep in mind that traditional TEX is eight bit and assumes a single extensible font (number three of the hard coded four family setup) and the 256 slots have to be distributed across variants and sizes and so TEX can provide only a limited solution space here.

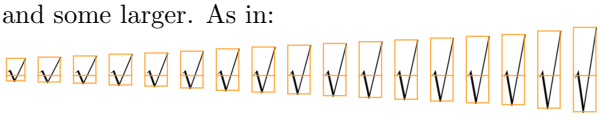
All this (plus some more) is supported in LuaMetaTEX but it only works out well if the macro package provides the information that is lacking in the fonts, which is yet another reason why we have companion fonts (adding sizes and fixing inconsistencies which are hard to tweak) as well as math font goodie files that add the information needed. It goes without saying that the authors spent a considerable

Hans Hagen, Mikael P. Sundqvist
doi.org/10.47397/tb/45-1/tb139hagen-radicals

amount of time on getting all this right. For example, we found out that the four variants of the radical in Latin Modern:



could benefit from extra sizes, some intermediate, and some larger. As in:



Compare the output of a few typical radicals:

$$\sqrt{2}+\sqrt{2^2}+\sqrt{\frac{1}{2}}$$

original

$$\sqrt{2}+\sqrt{2^2}+\sqrt{\frac{1}{2}}$$

companion

It is fair to mention that in ConTEXt we do use struts inside radicals and fractions to enforce consistency, and therefore the outcome of these examples might look different in other macro packages.

In principle all math symbols have these extra properties attached but their usage differs, so for instance in accents the margins are around the (wide) accent. With radicals the top and bottom margins are ignored but as we progress they might get some meaning. In some cases the implementation is less straightforward, for instance in a 'binop' we have a fraction with built-in fences so there we need to carry over the kerns that come with the chosen fences.

This all means that the math engine is more complex so it starts making sense to consider removing the traditional code paths: no new old school math fonts are likely to show up, and the ones that we had have acquired OpenType implementations by now anyway.

Sometimes, when we see what users try to compensate for, or ask for fixes about (on Stack Exchange, for instance), we wonder if this is a recent observation. After all, nothing like the above made it into OpenType math and fonts. Maybe observations get lost after some quick fix instead of being accumulated in some proposal, or maybe nothing got fixed anyway. We never get (or see) reports from editors (of journals) and none of them seem to follow developments like this (or we'd have noticed).[1] Of course much goes unnoticed when seen in print (at desktop or office printer resolution), but can be seen when proofing or reading on screen.

◇ Hans Hagen
◇ Mikael P. Sundqvist

[1] An exception is *TUGboat*'s Karl Berry who gives us valuable and inspiring feedback.