

Using this command `\N'65'\[radicalex]\N'66'\N'67'` with the ps.tmac .char which pdf.tmac .so's.

A \overline{BC}

Now do .rchar to get rid of the .char

A \overline{BC} <- a hole where the bar should be

Now with `\z\[radicalex]`

ABC <- hole gone bar static

Every font glyph has certain metrics, width/height/depth/italic correction which describes a bounding box for that glyph. The radicalex glyph in the S font has these numbers 500,917,0,590. That is huge italic correction, more than the total width of the glyph, but it gives a clue what is happening.

Inside the bounding box nothing is stroked, but from the NE corner of the bounding box a line is stroked to the right OUTSIDE the bounding box, so it goes above a following glyph, since its width is 500 but the measured width is probably 1090 with the extended line.

The .char you found is a fudge particular to the glyph in the S font, to make it overscore itself rather than the next character, which is what eqn expects this glyph to do.

The equivalent numbers for the XITSMR font are 408,973,0,20 and I suspect the line is drawn across the top of the bounding box, nothing outside, so it does not need any fudging.

I think I know why the font designer might done it this way, it allows you to do `\z\[\sqrt]\[radicalex]x` without involving eqn.

\sqrt{x}