Appendix G illuminated

BOGUSŁAW JACKOWSKI
BOP s.c., Gdańsk, Poland
B_Jackowski (at) gust dot org dot pl

Introduction

This paper aims to provide a collection of illustrations to *Appendix G* of *The TEXbook* [1].

To begin with, I will summarize briefly the main issues of *The TEXbook* which will be dealt with here; next, I confine myself to the explanation of the figures. Naturally, I will use the same notation as is used in *Appendix G*.

I recommend reading this paper simultaneously with *Appendix G*, although they partly overlap.

Motivation

TEX's algorithm for typesetting mathematical formulas is precisely described by Donald E. Knuth in *Appendix G* of *The TEXbook*. The description suffices to implement the algorithm in other languages. For example, it was implemented in JavaScript by Davide P. Cervone [2].

The only drawback of *Appendix G* is that no illustrations are provided. Of course, it is only a relative drawback. Professor Knuth apparently can live without illustrations. My comprehension critically depends on pictures. When they are missing in the original text, I end up making sketches while reading.

A few years ago, during my umpteenth reading of *Appendix G*, I prepared a bunch of sketches for myself. I didn't think about publishing them, as I was convinced that it is just my predilection or idiosyncrasy; but, judging from the paucity of available math fonts, I concluded that perhaps others might have similar problems.

Therefore, I decided to publish my illustrations to *Appendix G* in hope that they may prove useful, for example, for those working on math extensions for the available non-math fonts. Moreover, they may turn out to be helpful in future works on the improvement of TEX; after all, the algorithm is older than a quarter of century, and the world is not sleeping. For example, Murray Sargent III from Microsoft published recently (April, 2006) an interesting note on using Unicode for coding math [3]. He apparently was inspired by TEX: the notation is certainly TEX-based and well-known names appear in the acknowledgements and bibliography (Barbara Beeton, Donald E. Knuth, Leslie Lamport).

Math styles

In math formulas, the following eight styles are used:

- D, D' — in display formulas, generated out of text placed between double dollars $$...$$ (*display style*);

- T, T' — in formulas occurring in a paragraph, i.e., placed between single dollars $...$ (*text style*);

- S, S' — in formulas occurring in lower or upper indices, i.e., after the symbols ^ and _ (*script style*);

- SS, SS' — in formulas occurring in indices of indices or deeper (*scriptscript style*).

Typically, in the plain format, 10-point fonts are used for the styles D and T, 7-point fonts for the style S, and 5-point fonts for the style SS. The "primed" styles, called cramped, use the same point sizes; they differ from the uncramped ones in the placement of subformulas. Table 1 defines the relations between styles of formulas and their subformulas.

The symbol C will denote the current style, the symbol $C{\uparrow}$ the corresponding style of a superscript, and the symbol $C{\downarrow}$ the corresponding style of a subscript. From the rules given in Table 1 it follows that $C{\downarrow} = (C{\uparrow})'$.

Math lists

When TEX reads math material, it makes a *math list* out of it. The math list contains the following math-specific objects:

Basic style	Superscript style	Subscript style
D, T	S	S'
D', T'	S'	S'
S, SS	SS	SS'
S', SS'	SS'	SS'

Basic style of the formula α \over β	Numerator (α) style	Denominator (β) style
D	T	T'
D'	T'	T'
T	S	S'
T'	S'	S'
S, SS	SS	SS'
S', SS'	SS'	SS'

Table 1: Rules of the style change—indices (top) and fractions (bottom)

Name	Description of the atom
Ord	ordinary, e.g., x
Op	holding a big operator, e.g., $\sum$
Bin	holding a binary operator, e.g., $+$
Rel	holding a relational symbol, e.g., $=$
Open	holding an opening symbol, e.g., (
Close	holding an closing symbol, e.g.,)
Punct	holding a punctuation symbol, e.g., ,
Inner	"inner", e.g., $\frac{1}{2}$
Over	holding an overlined symbol, e.g., $\overline{x}$
Under	holding an underlined symbol, e.g., $\underline{x}$
Acc	holding an accented symbol, e.g., $\hat{x}$
Rad	holding a radical symbol, e.g., $\sqrt{2}$
Vcent	holding a *vbox* produced by \vcenter

Table 2: The types of atoms which may appear in a math list

- *atom*, the basic element;
- *generalized fraction*, the result of \above, \over, etc.;
- *style change*, the result of \displaystyle, \textstyle, etc.;
- *boundary element*, result of \left or \right;
- *4-way choice*, the result of \mathchoice.

There are several types of atoms, depending on their contents. Table 2 (a duplicate of the table given on page 158 of *The TEXbook*) lists all possible cases.

Moreover, the math list may contain elements specific to a vertical list:

- *horizontal material* (*rules*, *penalties*, *discretionaries* or *whatsits*);
- *vertical material*, inserted by \mark, \insert or \vadjust;
- *horizontal space*, inserted by \hskip, \kern, or (acceptable only in math mode) \mskip, \nonscript or \mkern.

The math list is processed twice; after the second pass the "normal" vertical list is created. The scheme of the algorithm (consisting of 22 steps) is shown in Figure 1.

As we will see, font dimension parameters play an essential role. Following *The TEXbook*, I will denote the ith parameter of the second family (that is, \textfont2, \scriptfont2, \scriptscriptfont2) by σ_i, and the ith parameter of the third family (\textfont3, \scriptfont3, \scriptscriptfont3)

by ξ_i. Note that the σ_i parameters have different values for different styles, while the values of the ξ_i parameters do not depend on the current style—when Computer Modern fonts and the plain format are used.

Selected steps of the algorithm

In most cases, the steps of the algorithm are straightforward and a short verbal explanation suffices. More detailed elaboration is needed, in my estimation, only for the typesetting of: radicals (step 11), mathematical accents (step 12), operators with limits (step 13), generalized fractions (step 15), and formulas with indices (step 18). The ensuing subsections deal with these steps.

Typesetting radicals

The process of assembling a formula containing a radical is presented in Figure 2. The top part of the picture shows the components (the radicand is typeset using the C' style), the bottom part the result of the assembling. Let w_x, h_x, d_x denote respectively the width, the height and the depth of the radicand, and w_y, h_y, d_y those of the radical symbol. The height of the radical symbol is expected to be equal to the thickness of the math rule, i.e., $h_y = \theta = \xi_8$. (A font designer may decide that $h_y \neq \xi_8$ but I cannot see the rationale for such a decision.) The quantity ψ is defined as follows:

$$\psi = \begin{cases} \xi_8 + \frac{1}{4}\sigma_5, & \text{styles } D, D', \\ \frac{5}{4}\xi_8, & \text{other styles.} \end{cases}$$
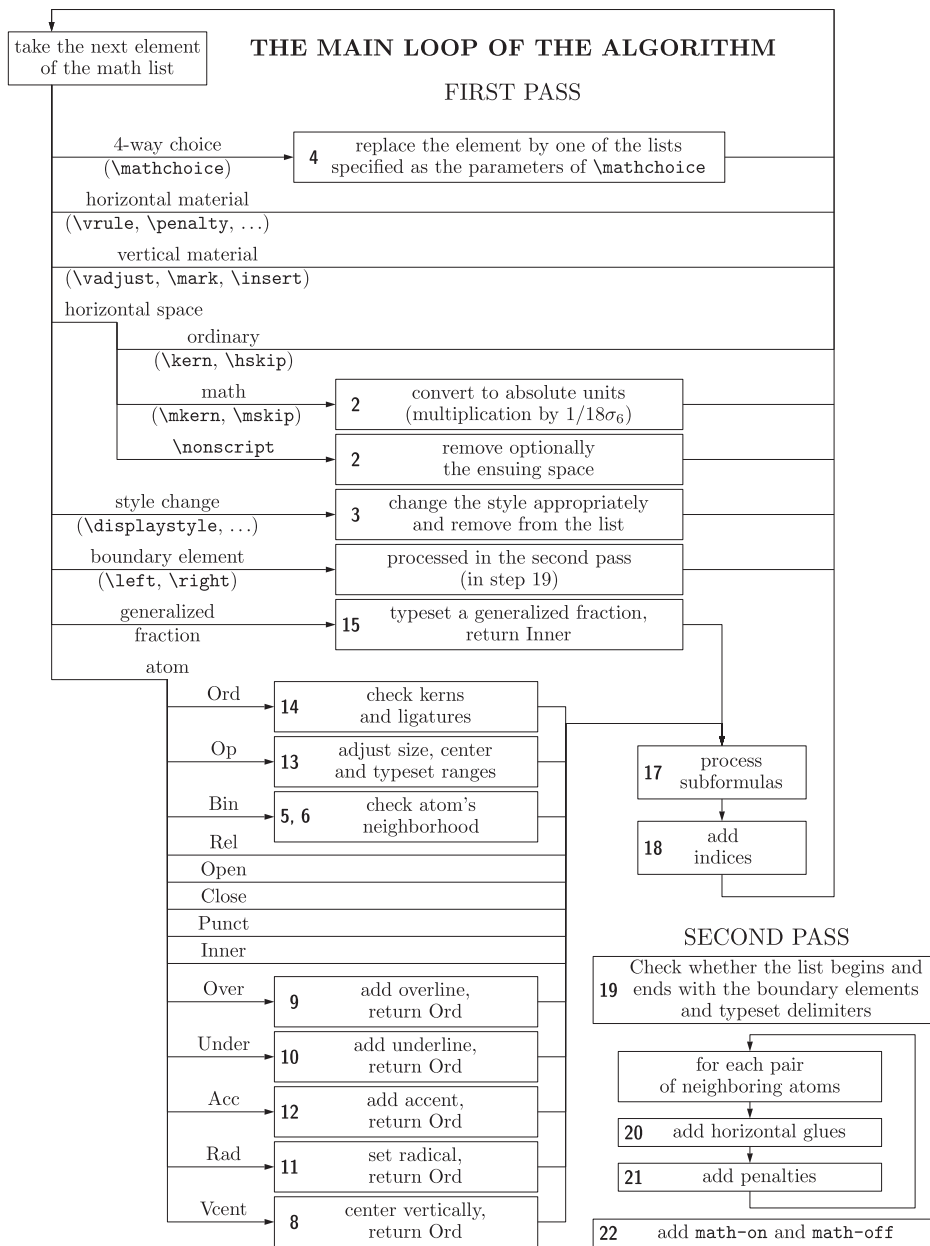
THE MAIN LOOP OF THE ALGORITHM

take the next element
of the math list

FIRST PASS

4-way choice
(\mathchoice)

| 4 | replace the element by one of the lists specified as the parameters of \mathchoice |

horizontal material
(\vrule, \penalty, ...)

vertical material
(\vadjust, \mark, \insert)

horizontal space

ordinary
(\kern, \hskip)

math
(\mkern, \mskip)

| 2 | convert to absolute units (multiplication by $1/18\sigma_6$) |

\nonscript

| 2 | remove optionally the ensuing space |

style change
(\displaystyle, ...)

| 3 | change the style appropriately and remove from the list |

boundary element
(\left, \right)

processed in the second pass
(in step 19)

generalized
fraction

| 15 | typeset a generalized fraction, return Inner |

atom

Ord

| 14 | check kerns and ligatures |

Op

| 13 | adjust size, center and typeset ranges |

Bin

| 5, 6 | check atom's neighborhood |

Rel

Open

Close

Punct

Inner

| 17 | process subformulas |

| 18 | add indices |

Over

| 9 | add overline, return Ord |

Under

| 10 | add underline, return Ord |

Acc

| 12 | add accent, return Ord |

Rad

| 11 | set radical, return Ord |

Vcent

| 8 | center vertically, return Ord |

SECOND PASS

| 19 | Check whether the list begins and ends with the boundary elements and typeset delimiters |

for each pair
of neighboring atoms

| 20 | add horizontal glues |

| 21 | add penalties |

| 22 | add math-on and math-off |

Figure 1: The scheme of the algorithm of the math list processing; the numbers at the left of the boxes refer to the steps of the algorithm, as described in *Appendix G*

The quantity Δ is computed in such a way that the the radical symbol is vertically centered with respect to the radicand: $\Delta = \frac{1}{2}\big(d_y - (h_x + d_x + \psi)\big)$.

The baseline of the resulting formula coincides with the baseline of the radicand.

Typesetting mathematical accents

The typesetting of an accented formula is simpler than the typesetting of a radical. Nevertheless, Figures 3 and 4 reveal non-trivial subtleties of the routine.

As before, let w_x, h_x, d_x denote respectively the width, the height and the depth of the accentee (typeset in the style C'), and w_y, h_y, d_y those of the accenter. Actually, these are the widths of the respective boxes; if the accentee is a symbol, its width, w_x, is computed as the sum $wd + ic$, where wd is the nominal (metric) width of the accentee, and ic the italic correction.

Both the accenter and accentee boxes are put into a *vbox* one above the other, and a negative vertical
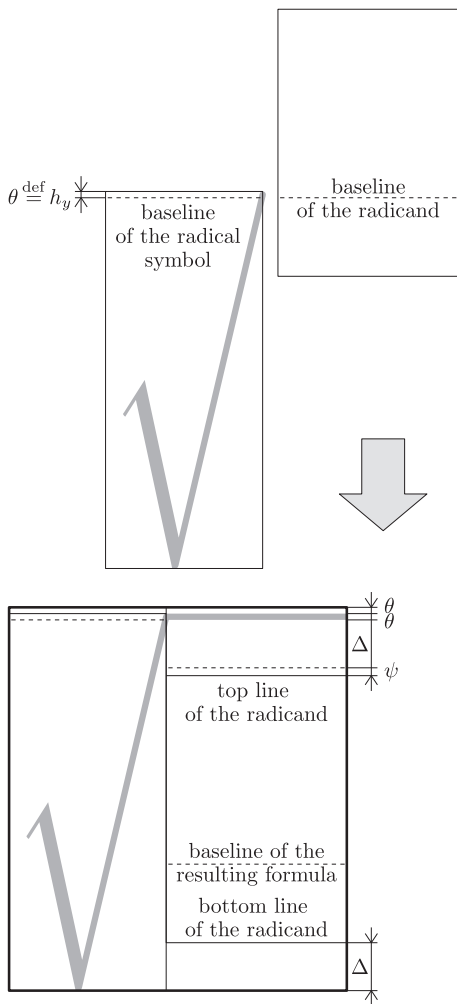
Figure 2: Assembling a radical; symbols explained in the text

kern, $-\delta$, is inserted between the boxes, where $\delta = \min(x\text{-}height, h_x)$. The *x-height* is defined by the fifth dimen parameter (`\fontdimen5`) of the accenter font.

The horizontal shift of the accenter, s, is equal to the implicit kern between the accentee and the special character, *skewchar* (defined by the command `\skewchar`; in the plain format, it is the character of code 127 (*tie after*) for family 1, and the character of code 48 (*prime*) for family 2. The kern has nothing to do with the shape of the `\skewchar`, but is intended to provide an appropriate correction due to the skewness of the accentee. If the accentee is already a boxed formula, TEX assumes that $s = 0$.

The width of the resulting formula is always equal to the width of the accentee, w_x; the baseline of the resulting formula coincides with the baseline of the accentee.
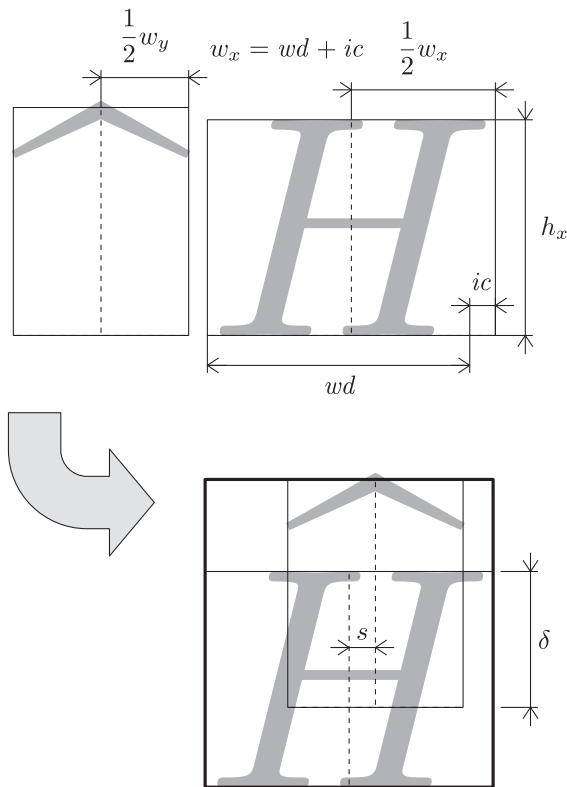
Figure 3: Assembling an accented formula, $w_y \leq w_x$; symbols are explained in the text
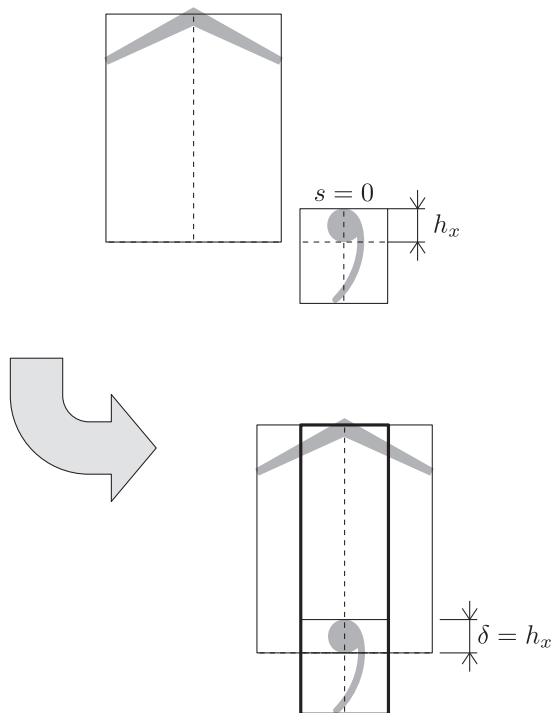
Figure 4: Assembling an accented formula, $w_y > w_x$; symbols have the same meaning as in Figure 3
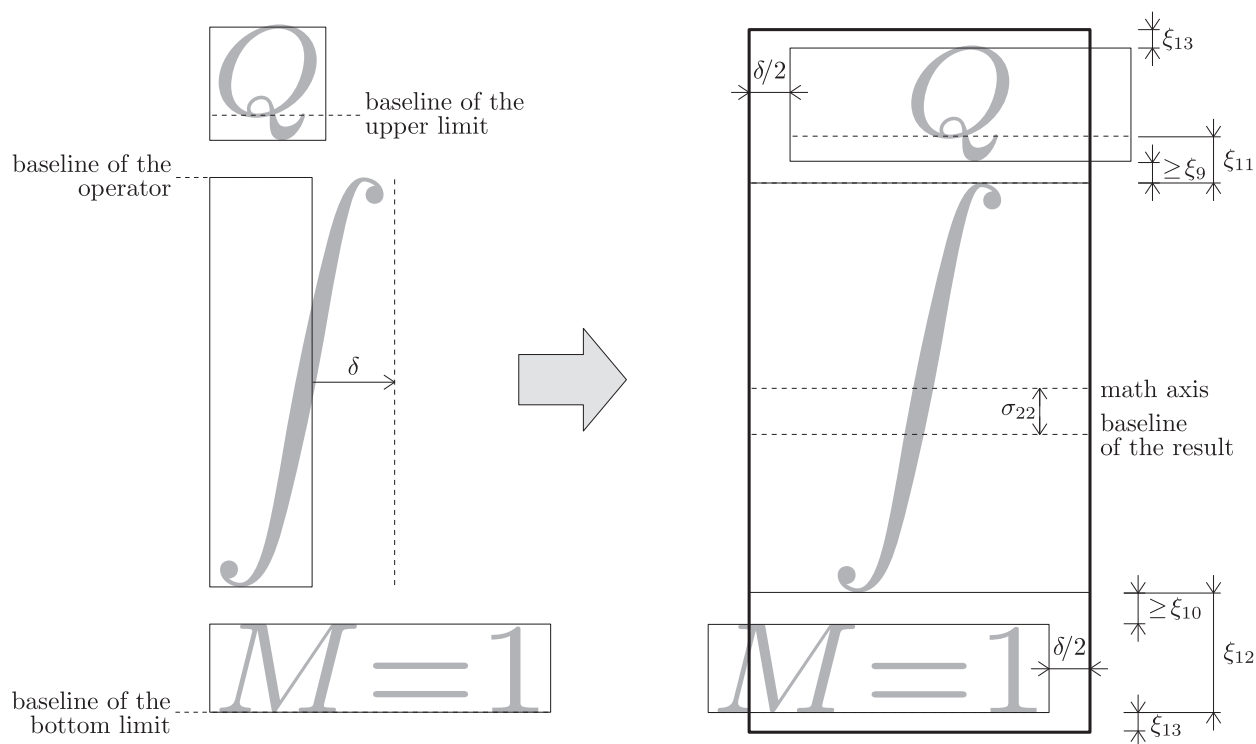
Figure 5: Assembling an operator with limits placed above and below; δ denotes the italic correction of the operator symbol

Typesetting operators with limits

The placement of the limits of an operator depends on the current style and the usage of `\limits` and `\nolimits` commands. If the style is D or D' and the operator `\nolimits` was not applied, the limits are placed above and below the operator, as displayed in Figure 5; otherwise, unless the operator `\limits` was used, the limits are processed as fractions (see following section about fractions).

The operator symbol is centered vertically with respect to the math axis (σ_{22}). TeX tries to place the upper formula in such a way that its baseline is distant by ξ_{11} from the top of the operator; however, if the distance between the bottom of the upper subformula and the top of the operator would be less than ξ_9, the distance ξ_9 is forced. Similarly, the baseline of the lower subformula is distant by ξ_{12} from the bottom of the operator, unless the distance between the top of the lower subformula and the bottom of the operator would be less than ξ_{10}, in which case the distance ξ_{10} is forced.

For the correction of the horizontal placement of the limits, the value of the italic correction of the operator symbol (denoted by δ in Figure 5), is used.

Typesetting generalized fractions

There are two kinds of fractions implemented in TeX: with or without a bar between the numerator and denominator. They are typeset using different rules, as shows Figure 6. These rules, however, do not suffice, as the numerator and denominator are likely to collide. TeX cleverly avoids collisions, as is shown in Figures 7 and 8.

For a fraction with a bar, the numerator and denominator are shifted independently in order to provide a minimal gap, φ, between the formulas and the bar. The position of the bar remains intact—it coincides with the math axis (see Figure 7). For a fraction without a bar, a different strategy is used to avoid the collision, namely, both the numerator and denominator are shifted apart so that the gap between them is equal to φ (see Figure 8). Note that φ has a different meaning in the two cases.
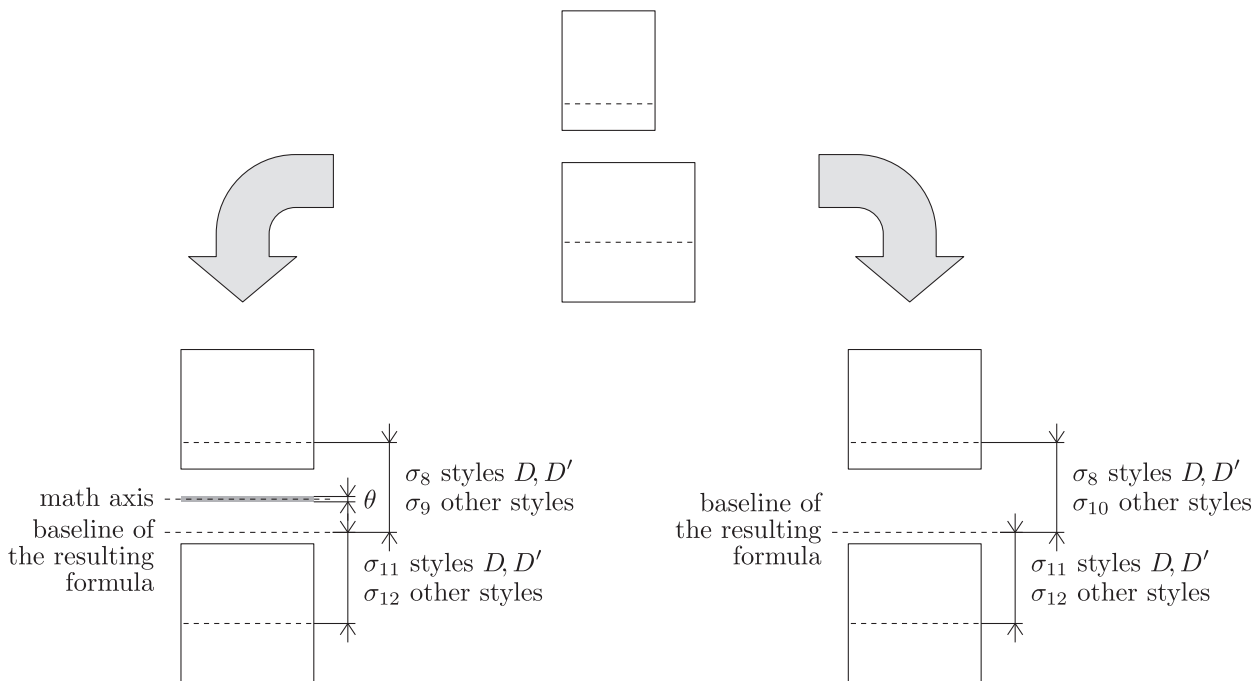
Figure 6: The placement of numerators and denominators in generalized fractions; the thickness of the rule, θ, is given either by the value of ξ_8 or explicitly; the latter possibility is provided by the `\abovewithdelims` command; observe that σ_9 is used for the formula with a bar, while σ_{10} for the formula without a bar
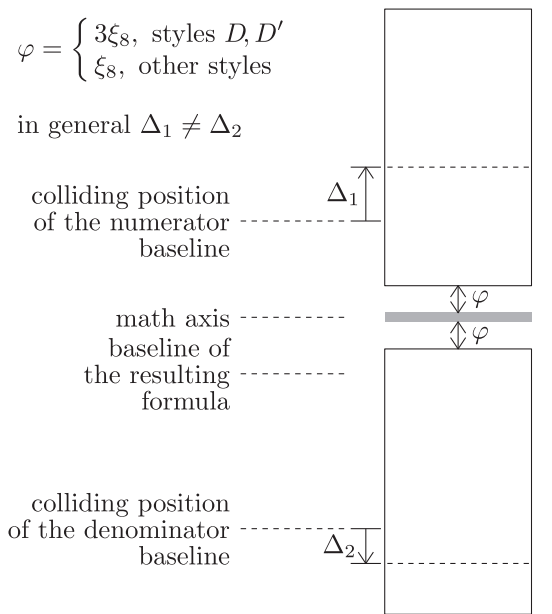

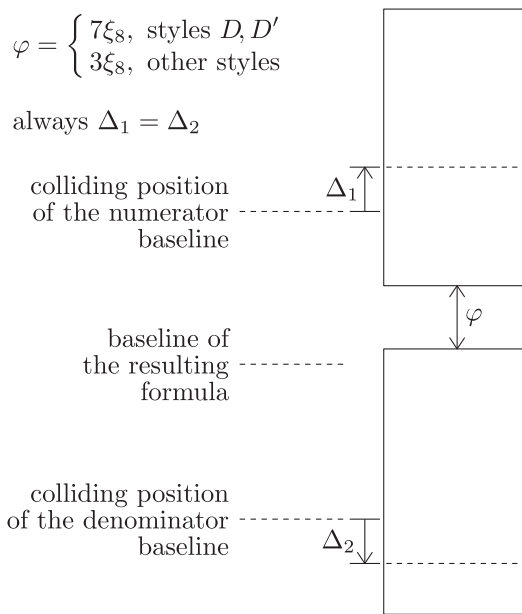*Figure 7*: Resolving a collision between the numerator and denominator in the case of a fraction with a bar

Figure 8: Resolving a collision between the numerator and denominator in the case of a fraction without a bar
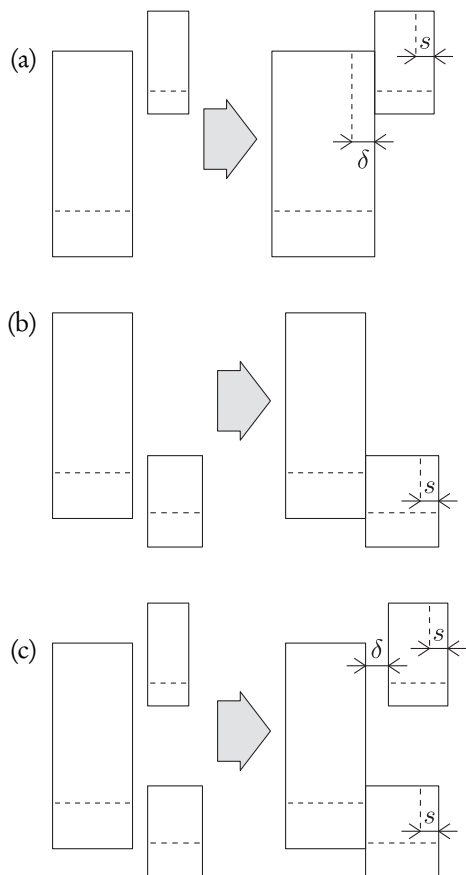
Figure 9: The horizontal placement of indices: (a) the placement of a lone superscript, (b) the placement of a lone subscript, (c) the placement of both superscript and subscript
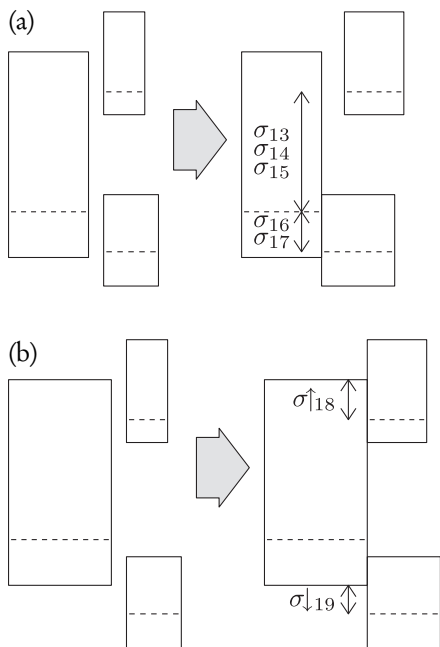

Figure 10: The vertical placement of indices: (a) the placement when the kernel is a symbol, (b) the placement when the kernel is a boxed formula
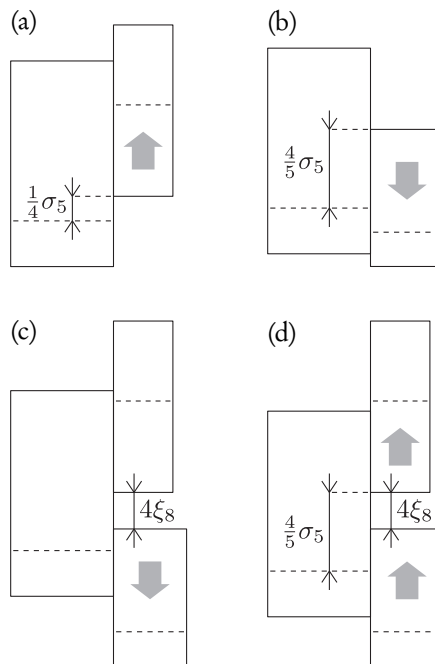

Figure 11: Resolving collisions of indices (further explanations in the text)

Typesetting formulas with indices

The placement of indices is a fairly complex task due to the variety of situations that may occur.

Figures 9a – 9c show the horizontal placement of indices. If the kernel is a single symbol, the superscript, if present, is shifted to the right by the amount of the italic correction of the kernel symbol. Technically, a slightly different procedure is involved in the presence of a subscript, as stated in the description of step 17 in *Appendix G*. If the kernel is already a boxed formula, TeX assumes that $\delta = 0$. A kern of the value `\scriptspace` (s in the figure) is always appended to index formulas.

The procedure for the vertical placement (see Figures 10a – 10b) makes use of 7 parameters: from σ_{13} to σ_{19}. Again, different procedures are employed depending on the structure of the kernel. If it is a symbol, σ_{13} for the style D, σ_{14} for other uncramped styles, and σ_{15} for cramped styles are used for the placement of a superscript; for the placement of a subscript, σ_{16} is used if a superscript is absent and σ_{17} otherwise. If the kernel is a boxed formula, σ_{18} is used for the positioning of the superscript and σ_{19} for the positioning of the subscript. Moreover, the respective values are

not taken from the current font: $\sigma\uparrow$ and $\sigma\downarrow$ mean that the parameters refer to the fonts corresponding to the styles $C\uparrow$ and $C\downarrow$, respectively.

Indices, as one can expect, are also subject to potential collisions. The actions for such a case are depicted in Figures 11a – 11d: (a) the bottom of the superscript formula cannot be placed lower than $\frac{1}{4}\sigma_5$ above the baseline; (b) the top of the subscript formula cannot be placed higher than $\frac{4}{5}\sigma_5$ above the baseline; (c) the gap between the bottom of the superscript and the top of the subscript cannot be smaller than $4\xi_8$ (recall that ξ_8 stores the math rule thickness) — the subscript is shifted if required; (d) finally, if the latter situation occurs, both indices can be shifted up so that the bottom of the superscript is not lower than $\frac{4}{5}\sigma_5$ above the baseline.

Conclusions

As originally mentioned, I prepared the illustrations initially for myself and only later decided to publish them in hope that somebody else may also benefit. Therefore, I eagerly welcome any feedback.

Acknowledgements

References

[1] Donald E. Knuth, *The TEXbook*, Computers & Typesetting: Volume A, Addison Wesley, 1986

[2] Davide P. Cervone, *jsMath: A Method of Including Mathematics in Web Pages*, `http://www.math.union.edu/ dpvc/jsMath/`

[3] Murray Sargent III, *Unicode Nearly Plain-Text Encoding of Mathematics*, `http://www.unicode.org/notes/tn28/`