Philology

TEX and Russian Traditions of Typesetting

Mikhail Ivanovich Grinchuk

Abstract

It is well-known that the Russian style of mathematical typesetting requires one to repeat the signs of operation and relation when a line-break occurs in a formula. Standard TEX styles have no such capability. In attempting to solve this problem, a TEX program was created such that we can use, practically, standard typesetting and generate automatic repetition of all signs. Several difficulties which arose during the realization of this program are discussed and solutions proposed.

Problems

It is well-known that the remarkable system TEX was created in the tradition of English typesetting, which is different from that accepted in Russia, in particular, as concerns the typesetting of mathematical formulae. We shall specify some, more-or-less essential, differences between these two systems. For brevity we shall refer to the systems as E (English) and R (Russian).

1. The "proclaim" of theorems in the E system nowadays are typeset by a slanted font (`\sl`), whereas in the R system the standard is italic (`\it`).
2. Even in the text, typed by slanted or italic font, the R system requires punctuation marks and digits to be in the "orthogonal" shape (among punctuation marks the distinction is essential for braces, brackets, parentheses, and the marks ':', ';', '?', '!'). This rule is not the case for fictional literature.
3. The R system often requires the use of l e t t e r - s p a c e d typesetting.
4. The E system has two kinds of quotation marks: "xxxx" and 'xxxx'. The R system has two kinds too, but these are different: «xxxx» and „xxxx“.
5. Use of dashes. It is possible to illustrate four distinct situations:
   a) ...we shall take 2—3 litres of water...
   b) ...we shall take two—three litres of water...
   c) ...the Newton—Leibnitz formula...
   d) ...and other—they do not want...

   The E system distinguishes between short (–) and long (—) dashes; the first one is used for cases (a) and (b). In case (c), the hyphen char (-) is usually used, though it leads to confusion between two surnames and one double surname. In all four cases, the dash is used without spaces at either end, and a possible line break may occur following the dash.

   In contrast with the E system, the R system has only the long dash, surrounded by spaces (half the width of a usual space). [Though it may appear rather strange, in my opinion, it would be more regular to act in the following manner: in case (a) to use a short dash without spaces; in case (b)—the long dash without spaces; in case (c)—the long dash with half-spaces; and in case (d)—the long dash with normal spaces.] The possible line-break in all cases would occur after a dash.
6. The R system does not recommend increasing space between sentences (i.e., we use "`\frenchspacing`").
7. There are some differences in punctuation of headings and captions: the E system accepts "1.1 Introduction" and "Fig. 1.", but the R system uses "1.1. Introduction" and "Fig. 1".
8. There is a short list of non-equal "mathematical words": `tan/tg`, `cosec/csc`, `sinh/sh`, etc.
9. The R system uses `\varphi`, `\varkappa`, `\varepsilon`, `\varnothing`, `\leqslant`, and `\geqslant` instead of `\phi`, `\kappa`, `\epsilon`, `\emptyset`, `\le`, and `\ge`.
10. With regard to ellipses, the R system uses `\cdots` only in products of the form $x_1 \cdots x_n$. All other cases requires `\ldots`. The R system does not use 4-dot ellipses, and in combination with '?' or '!' uses "?.." and "!.." (i.e., two dots).
11. And now the most essential: the E system permits one to break a formula after a binary operation or binary relation, but the R system also requires repetition of the mark at the beginning of the second line. This is true also for `\dots`.

Some Solutions

For almost all of the listed differences, it is very easy to add to or modify TEX. Corresponding to the previously listed items:

1. If the fonts are switched "manually", there is no problem; otherwise, it is necessary to make slight corrections to the style file.
2. The optimum solution is to have slanted and italic fonts with non-slanted punctuation and digits. If we have no such fonts, then when necessary, it is possible either to use something like `{\rm(}` or `$($`, or to define a command like `\def\({{\rm(}}`.
3. If you require much letterspaced text, it is possible to insert spaces manually (but perhaps, it is better not to use spaces, but to use the `\~`). Here we need to increase interword spacing (up to `\quad`), and where appropriate to insert discretionary hyphens `\-`.

   It is not very difficult to write a command for the call "`\spaced{Proof}`" to generate "P r o o f", but there are possible difficulties in a situations like

   ```
   \spaced{Lemma 9$'$ \cite{17} (the main)}
   ```
4. Either you have «such» quotes, or not. Russian TEX manuals try to standardize commands (written by Cyrillic letters) — `\lk` and `\pk` (or

\l and \p) — as shortcuts for "Lyevaya Kavychka" and "Pravaya Kavychka" (Left Quote and Right Quote). But it is more natural to use the ligatures « and » (there will be no difficulties associated with "eating" the following spaces).

The „ and " can be made of existing signs; they are the same as used in `german.sty` commands `\glqq` and `\grqq`. However, it is more natural to type "`,,`" and "`‘‘`".

5. If we do not use "half-spaces", it is enough to use a combination "`~---`" with a space added at the end. The half-space is produced as follows:
```
\newskip \thehalfspace
\def \halfspace{\unskip
  \thehalfspace=
     \the \fontdimen2 \the \font
     plus \the \fontdimen3 \the \font
     minus \the \fontdimen4 \the \font
   \divide\thehalfspace by 2
   \hskip \thehalfspace
   \ignorespaces }
```
6. For this item, we use `\frenchspacing`.
7. Requires either manual correction at each point, or it is necessary to correct the style files (for LaTeX, such work was done by Lwowsky).
8. This is also relatively simple — find how `\tan` is defined, and define the `\tg` in the same way.
9. AMS-TeX has all necessary marks. In other cases you need to load AMS-TeX fonts `msam` and `msbm`.
10. The easiest solution is to use only the commands `\cdots` and `\ldots`.
11. See the next section.

The Choices

In one approach, we can make, for example, a breakable "+" sign like this:
```
\def\plus{\discretionary{+}{}{}+}
```
or
```
\def\plus{+\discretionary{}{+}{}}
```
(the more natural variant
```
\def\plus{\discretionary{+}{+}{+}}
```
is impossible, because in mathematics mode, the `\discretionary` third field should be empty). The second variant provides a more correct spacing.

Naturally, if you are using such commands, it is necessary to forbid "built-in" math breaks, making the parameters `\relpenalty` and `\binoppenalty` more than 10000.

The offered choices are not too successful for two reasons — first, the `\discretionary` mark will always be large (`\scriptstyle` is ignored); and secondly, the object inside `\discretionary` is textual, and the transition here to mathematics is impossible.

A more reliable solution is a pair of commands
```
\def\brokenrel#1{%
    \mathrel{#1}\selector{#1}}
\def\brokenbin#1{%
     \mathbin{#1}\selector{#1}}
```
(which means that to have a breakable command like "`\ne`" the user must type `\brokenrel\ne`, and for the breakable operation "`\times`" typesetting is `\brokenbin\times`). Here the command `\selector` is defined as:
```
\def\selector#1{\mathchoice
   {\discretionary {}%
     {\hbox{$\displaystyle#1$}}{}}%
   {\discretionary {}%
     {\hbox{$\textstyle#1$}}{}}%
   {\discretionary {}%
     {\hbox{$\scriptstyle#1$}}{}}%
   {\discretionary {}%
     {\hbox{$\scriptscriptstyle#1$}}{}}%
              }
```
It is more convenient, however, rather than input the commands `\brokenrel` and `\brokenbin` manually, to redefine the "breakable" operation as:
```
\let\originaltimes=\times
\def\times{\brokenbin\originaltimes}

\let\originalleqslant=\leqslant
\def\leqslant{\brokenrel
         \originalleqslant}
```
One must not overlook the command `\not`:
```
\let\originalnot=\not
\def\not#1{\brokenrel{\originalnot#1}}
```
In a similar way it is possible to make breakable marks "+", "−", "<", ">", and "=" (as well as "∗"):
```
\mathcode`\+="8000
{\catcode`\+=\active
 \gdef+{\brokenbin{\mathchar"202B}}}

\mathcode`\>="8000
{\catcode`\>=\active
  \gdef>{\brokenrel{\mathchar"313E}}}
```
(codes `"202B` and `"313E`, etc., are used in the beginning of format files like `plain.tex` or `lplain.tex`, where the `\mathcode` table is set).

But the methodical realization of all such redefinitions may lead to some unexpectedly strange breaks:

...... system of $\geq$
    $\geq n$ vectors......
...... has the value of $-$
    -1 or -2 ......
...... $x = -$
    $-y$......
...... $x = (-$

$-y+z)$ ......

Fortunately, these situations are simple to correct. The idea is to provide, at the beginning of each formula and after each "redefined" operation, some large additional penalty (for example, 13131). Then each potentially breakable mark should at first check the previous penalty and should not be broken if it is 13131. A possible realization:

```
\relpenalty=13131
\binoppenalty=14141
\everymath={\penalty\relpenalty}

\def\brokenbin#1{%
 \ifnum\lastpenalty=\relpenalty
   \mathbin\{#1} \else
   \mathbin\{#1} \selector{#1} \fi
     \penalty\binoppenalty}

\def\brokenrel#1{%
  \ifnum\lastpenalty=\relpenalty
     \mathrel{#1}\else
     \mathrel{#1}\selector{#1}\fi
     \penalty\relpenalty}
```

It is thus possible to forbid breaks after `\bigl`-signs (`\Bigl`, `\biggl`, etc.):

```
\def\bigl#1{\mathopen\big#1%
  \penalty\relpenalty}
```

(The `\left`...`\right`-constructs give no problems because breaks are prohibited).

To forbid breaks after left parentheses, braces, and brackets, etc., we can redefine their signs in math. For example,

```
\mathcode`\(="8000
{\catcode`\(=\active
 \gdef({\delimiter"0283000
        \penalty\relpenalty}}
```

(the magic number is the `\delcode` of "(").

Another possible difficulty may occur. After such total redefinitions, expressions with indices and expressions of the type `\mathord\rightarrow` (e.g., this appears in the standard definition of the command `\arrowfill`) can "deteriorate". A solution here is to specify the index in braces (e.g., to replace `\mathord\rightarrow` with `\mathord{\rightarrow}`). If you do *not* want to place indices in braces, it is possible to redefine the signs for indices, defining them as active symbols with parameters:

```
\let\upperindex=^
\let\lowerindex=_
\catcode`\^=\active
   \def^#1{\upperindex{#1}}
\catcode`\_=\active
   \def_#1{\lowerindex{#1}}
```

Acknowledgements

This work was supported in part by Russian Foundation for Basic Research, grant no. 96–7–89406.

◇ Mikhail Ivanovich Grinchuk
Department of Mechanics and Mathematics, Moscow State University, Moscow, Russia