This is a list of all corrections made to Computers & Typesetting, Volumes A, B, C, and D, between 1 January 1991 and 15 March 1992. Corrections made to the softcover version of The \textsc{TeX}book are the same as corrections to Volume A. Corrections to the softcover version of The \textsc{MetaFont}book are the same as corrections to Volume C. Some of the corrections below have already been made in reprintings of the books. Changes to Volume B refer to the fourth printing (1991), which differs markedly from earlier printings because it includes all the revisions for \textsc{TeX}3.0. Changes to Volume D refer to the third printing (1991), which differs markedly from earlier printings because it includes all the revisions for \textsc{MetaFont}2.0. Changes to the mini-indexes and master indexes of Volumes B and D are not shown here unless they are not obviously derivable from what has been shown. Dozens of changes, too many to list here, have been made to Volume E because of recent upgrades to the Computer Modern font source files. Those changes, which affect only the digitization at low resolution and the appearance of lowercase delta and some characters in the math symbols fonts (but not the \textsc{TFM} files), are documented at the end of file \texttt{cm85.bug}.

Page A96, lines 9–11 (9/18/91)

Some German words traditionally change their spelling when they are split between lines. For example, ‘backen’ becomes ‘bak-ken’ and ‘Bettuch’ becomes ‘Bett-tuch’. How can you instruct \TeX to produce such effects?

Page A178, line 17 (11/19/91)

If you say `$\phantom{\langle\text{subformula}\rangle}$' in any formula, plain \TeX will do its

Page A286, bottom two lines and continuing into A287 (11/21/91)

stands for zero or more (assignment) commands other than \texttt{\setbox}. If the assignments are not followed by a (character), where (character) stands for any of the commands just discussed in the previous paragraph, \TeX treats \texttt{\accent} as if it were \texttt{\char}, except that the space factor is set to 1000. Otherwise the character that follows the assignment is accented by the character that corresponds to the (8-bit number). (The purpose of the intervening assignments is to allow the accenter and accentee to be in different fonts.) If the accent must be moved up or down, it is put into an hbox that is raised or lowered. Then the accent is effectively superposed on the character by means of kerns, in such a way that the width of the accent does not influence the width of the resulting horizontal list. Finally, \TeX sets \texttt{\spacefactor=1000}.

Page A291, lines 6–8 (11/21/91)

'\}' may be followed by optional (assignment) commands other than \texttt{\setbox}, after which '$$' must conclude the display. \TeX will insert the \texttt{\abovedisplayskip} and \texttt{\belowdisplayskip} glue before and after the result of the alignment.
explained in Appendix G. \TeX{} scans (one optional space) after completing a displayed formula; this is usually the implicit space at the end of a line in the input file.

Page A311, bottom four lines (9/18/91)

12.7. 1000, except: 999 after O, B, S, D, and J; 1250 after the comma; 3000 after the exclamation point, the right-quote marks, and the periods. If a period had come just after the B (i.e., if the text had said ‘B. Sally’), the space factor after that period would have been 1000, not 3000.

Page A314, lines 16–18 from the bottom (1/10/92)

14.8. back/en and Be\ttt/uch, where the macros \ck/ and \ttt/ are defined by
\def\ck/{\discretionary{k-}{k}{ck}}
\def\ttt/{tt\discretionary{-}{t}{}
\def\lgroup{\delimiter"462833A } \def\rgroup{\delimiter"562933B }
\def\lmoustache{\delimiter"437A340 } \def\rmoustache{\delimiter"537B341 }
\def\uparrow{\delimiter"3222378 } \def\Uparrow{\delimiter"322A37E }
\def\downarrow{\delimiter"3223379 } \def\Downarrow{\delimiter"322B37F }
\def\updownarrow{\delimiter"326C33F } \def\arrowvert{\delimiter"026A33C }
\def\Updownarrow{\delimiter"326D377 } \def\Arrowvert{\delimiter"026B33D }
\def\vert{\delimiter"026A30C } \def\Vert{\delimiter"026B30D }
\def\backslash{\delimiter"026E30F } \def\bracevert{\delimiter"077C33E }

\phantom, \smash, \root, and other operations. (Actually \phantom and \smash are not perfect: They assume that the current style is uncramped.)

\def\c@ncel#1#2{\m@th\ooalign{$\hfil#1\mkern1mu/\hfil$}\crcr$#1#2$}

\def\rlh@#1{\vcenter{\m@th\hbox{\ooalign{\raise2pt\hfill#1\hfill\crcr}}}}

\def\fmtname{plain}\def\fmtversion{3.141}

story: Macro \texttt{\stest} decides whether or not a given token list begins with a \texttt{space token} as defined in Chapter 24. If so, the macro decides whether the token is explicit and/or funny and/or active.

\long\def\ssss#1#2\stest{\funnytrue{\escapechar=\if*#1'?'\else'*\fi\relax
\if#1\string#1\uccode'#1='~ % we assume that ~ is an active character
\uppercase{\ifcat\noexpand#1}\noexpand~\global\activetrue
\else\global\explicittrue\fi
\else\testactive#1\s\fi}
\long\def\tact#1#2\act{\def\next{#2}\ifx\next\sx\global\activetrue
\else\ifx\next\empty\global\activetrue\fi\fi}
\def\sx{\s}
\long\def\ss#1\stest{\spacefalse}
If the current item is an Ord atom, go directly to Rule 17 unless all of the following are true: The nucleus is a symbol; the subscript and superscript are both empty; the very next item in the math list is an atom of type Ord, Op, Bin, Rel, Open, Close, or Punct; and the nucleus of the next item is a symbol whose family is the same as the family in the present Ord atom. In such cases the present symbol is marked as a text symbol. If the font information shows a ligature between this symbol and the following one, using the specified family and the current size, then insert the ligature character and continue as specified by the font; in this process, two characters may collapse into a single Ord text symbol, and/or new Ord text characters may appear. If the font information shows a kern between the current symbol and the next, insert a kern item following the current atom. As soon as an Ord atom has been fully processed for ligatures and kerns, go to Rule 17.

are used to change the current style just as in the first pass, so that both passes have the same value of \textit{C} when they work on any particular atom.

[Delete the entry for ‘\sigma_2’; the entry for ‘\sigma_{17}’ moves down to the bottom of the left column.] to parameters in arbitrary families: Rule 17 uses \texttt{fontdimen} parameter 2 (space) to de-


define \texttt{banner} \texttt{≡ "This is TeX, Version 3.141" \{} printed when \TeX \ starts \}
Page B26, new line before fourth line from bottom (1/24/92)

\[ nl: \text{integer}; \quad \{ \text{new-line character to restore} \} \]

Page B26, bottom line and top 3 lines of B27 (1/24/92)

```
else begin if selector > pseudo then
    begin print_char(s); return; \{ internal strings are not expanded \}
    end;
    if ((Character s is the current new-line character 244)) then
        if selector < pseudo then
            begin print_line; return; end;
    nl ← new_line_char; new_line_char ← −1; \{ temporarily disable new-line character \}
    j ← str_start[s];
    while j < str_start[s + 1] do
        begin print_char(so(str_pool[j])); incr(j); end;
    new_line_char ← nl; return;
end;
```

Page B27, lines 9 and 10 (9/19/91)

```
60. Control sequence names, file names, and strings constructed with \string might contain ASCII code values that can’t be printed using print_char. Therefore we use slow_print for them:
```

Page B27, lines 13–26 (1/24/92)

```
var j: pool_pointer; \{ current character code position \}
begin if (s ≥ str_ptr) ∨ (s < 256) then print(s)
else begin j ← str_start[s];
    while j < str_start[s + 1] do
        begin print(so(str_pool[j])); incr(j);
    end;
end;
```

Page B28, line 8 (9/19/91)

```
else begin slow_print(format_ident); print_line;
```

Page B33, line 3 (1/11/92)

```
recursively. A similar interlock is provided by set_box_allowed.
```

Page B33, new line to come after line 14 (1/11/92)

```
set_box_allowed: boolean; \{ is it safe to do a \setbox assignment? \}
```

Page B33, new line to come after line 20 (1/11/92)

```
set_box_allowed ← true;
```
begin print_nl("You want to edit file "); slow_print(input_stack[base_ptr].name_field);

arithmetic; see TUGboat 3,1 (March 1982), 10–27. (But the routines cited there must be modified to allow negative glue ratios.)

structures on a memory word, which contains either a (signed) integer, possibly scaled, or a (signed) glue_ratio, or a small number of fields that are one half or one quarter of the size used

begin print_err("Bad mathchar");
help2("A mathchar number must be between 0 and 32767.")

if align_state < 1000000 then 
begin repeat get_token; until cur_tok = 0;
align_state ← 1000000; goto done;
end;

begin slow_print(a); slow_print(n); slow_print(e);

begin wlog(banner); slow_print(format_ident); print("\&"); print_int(day); print_char("\*");

print_char("*"); incr(open_parens); slow_print(name); update_terminal; state ← new_line;

print("in font "); slow_print(font_name[f]); print_char("!"); end_diagnostic(false);

print_nl("Output written on "); slow_print(output_file_name);
print("*"); print_int(total_pages); print("page");

if \ f < 0 then 
begin decr(n); \ f ← f + \ `200000; 
end;
if \( f < 0 \) then
  begin\( \text{decr}(n); \quad f \leftarrow f + '200000; \)
  end;

Up to three passes might be made through the paragraph in an attempt to find at least one set of feasible breakpoints. On the first pass, we have \( \text{threshold} = \text{pretolerance} \) and \( \text{second\_pass} = \)

863. The ‘loop’ in the following code is performed at most thrice per call of \text{line\_break}, since

\begin{verbatim}
hyf_bchar: halfword; \{ boundary character after \( c_n \} \)
hyf_bchar ← character(s); \quad c ← \text{qo(hyf\_bchar)};
\end{verbatim}

\begin{verbatim}
hb ← s; \quad \text{incr(hn)}; \quad \text{hu[hn]} ← c; \quad \text{hc[hn]} ← \text{lc\_code(c)}; \quad \text{hyf\_bchar} ← \text{non\_char};
\end{verbatim}

\begin{verbatim}
else if (\text{type(s) = kern\_node}) \land (\text{subtype(s) = normal}) \text{then} \quad hb ← s
  else \quad \text{goto done3};
\end{verbatim}

\begin{verbatim}
j ← hn; \quad q ← \text{hyf\_ptr(s)}; \quad \text{if} \quad q > \text{null} \quad \text{then} \quad \text{hyf\_bchar} ← \text{character(q)};
\end{verbatim}

\begin{verbatim}
if \text{odd(subtype(s))} \text{then} \quad \text{hyf\_bchar} ← \text{font\_bchar[hf]} \quad \text{else} \quad \text{hyf\_bchar} ← \text{non\_char};
\end{verbatim}

\begin{verbatim}
if \text{hn} < \text{L\_hyf} + \text{r\_hyf} \quad \text{then} \quad \text{goto done1}; \quad \{ \text{L\_hyf} \text{ and r\_hyf} \text{ are always} \geq 1 \}
\end{verbatim}

\begin{verbatim}
q ← \text{link(hb)}; \quad \text{link(hb)} ← \text{null}; \quad r ← \text{link(\text{ha})}; \quad \text{link(\text{ha})} ← \text{null}; \quad \text{bchar} ← \text{hyf\_bchar};
\end{verbatim}
cur,r = {character(lig_stack), if lig_stack > null; font_false_bchar[cur_font], otherwise;}

except when character(lig_stack) = font_false_bchar[cur_font]. Several additional global variables are needed.

cur,q ← tail; cur,l ← character(lig_stack); (3/15/92)

scan_optional_equals;
if set_box_allowed then scan_box(box_flag + n)
else begin print_err("Improper "); print_esc("setbox");
help2("Sorry, \setbox is not allowed after \halign in a display,");
("or between \accent and an accented character."); error;
end; (1/11/92)
flushable_string: str_number; { string not yet referenced } (1/24/92)
flushable_string ← str_ptr − 1;

begin if cur_name = flushable_string then
    begin flush_string; cur_name ← font_name[f]; end;
    if s > 0 then

set_font: begin print("select_font"); slow_print(font_name[chr_code]);

set_box_allowed ← false; prefixed_command; set_box_allowed ← true;

slow_print(s); update_terminal;

begin print_err("*"); slow_print(s); (9/19/91)
Page B531, lines 19 and 20

\begin{verbatim}
print_nl("Beginning_to_dump_on_file."); slow_print(w_make_name_string(fmt_file)); flush_string;
print_nl(""); slow_print(format_ident);
\end{verbatim}

Page B533, line 29

\begin{verbatim}
begin print_nl("Transcript_written_on."); slow_print(log_name); print_char(".");
\end{verbatim}

Page B538, line 13

\begin{verbatim}
10: slow_print(n);
\end{verbatim}

Page B577, left column

[Add 798 to the index entries for 'system dependencies'.]

Page C262, line 15

\begin{verbatim}
string base_name, base_version; base_name="plain"; base_version="2.7";
\end{verbatim}

Page C271, line 17 from the bottom

\begin{verbatim}
currentpen_path shifted (z.t_) withpen penspeck enddef;
\end{verbatim}

Page C347, Brontë entry

[The accent was clobbered; her name should, of course, be Brontë. Fix the entries for Dürer,
Möbius, and Stravinsky in the same way.]

Page C348, left column

compound statement, 155, 217.

Page C353, right column

\begin{verbatim}
*numeric, 55, 56, 65, 88.
\end{verbatim}

Page C354, miscellaneous entries in both columns

\begin{verbatim}
*openwindow, 191–193, 220, 277, 312–313.
*or, 65, 170, 210, 237, 288–289.
*pair, 55, 56, 65.
*path, 55, 56, 171.
*pen, 55, 56, 65, 170.
*picture, 55, 56, 114.
\end{verbatim}

Page C356, right column

\begin{verbatim}
*string, 55, 56, 69.
\end{verbatim}

Page C357, right column

\begin{verbatim}
*transform, 55, 56, 57, 141–143, 160, 266.
\end{verbatim}
Page D2, last line of section 2 (1/24/92)

\texttt{define banner \equiv \text{`This is METAFONT, Version 2.71'} \quad \{\text{printed when METAFONT starts}\}}

Page D102, line 15 from the bottom (11/1/91)

Then \(eq_{\text{type}}(h(x)) = tag_{\text{token}}\) and \(eq_{\text{ev}}(h(x)) = p\), where \(p\) is a two-word value node with

Page D188, lines 16 and 17 (1/24/92)

errors. Our subroutines also obey the identity \(t[a, b] + t[b, a] = a + b\).

Page D190, new copy before bottom four lines (1/24/92)

\begin{verbatim}
if \(x_{\text{coord}}(r) < x_{\text{coord}}(pp)\) then \(x_{\text{coord}}(r) \leftarrow x_{\text{coord}}(pp)\)
else if \(x_{\text{coord}}(r) > dest_{\text{x}}\) then \(x_{\text{coord}}(r) \leftarrow dest_{\text{x}}\);
if \(left_{\text{x}}(r) > x_{\text{coord}}(r)\) then
  \begin{verbatim}
beg\text{begin left}_{\text{x}}(r) \leftarrow x_{\text{coord}}(r); if right_{\text{x}}(pp) > x_{\text{coord}}(r) then right_{\text{x}}(pp) \leftarrow x_{\text{coord}}(r); end;\end{verbatim}
if \(right_{\text{x}}(r) < x_{\text{coord}}(r)\) then
  \begin{verbatim}
beg\text{begin right}_{\text{x}}(r) \leftarrow x_{\text{coord}}(r); if left_{\text{x}}(qq) < x_{\text{coord}}(r) then left_{\text{x}}(qq) \leftarrow x_{\text{coord}}(r); end;\end{verbatim}
\end{verbatim}
\end{verbatim}

Page D191, new copy before bottom two lines of section 416 (1/24/92)

\begin{verbatim}
if \(x_{\text{coord}}(s) < x_{\text{coord}}(r)\) then \(x_{\text{coord}}(s) \leftarrow x_{\text{coord}}(r)\)
else if \(x_{\text{coord}}(s) > dest_{\text{x}}\) then \(x_{\text{coord}}(s) \leftarrow dest_{\text{x}}\);
if \(left_{\text{x}}(s) > x_{\text{coord}}(s)\) then
  \begin{verbatim}
beg\text{begin left}_{\text{x}}(s) \leftarrow x_{\text{coord}}(s); if right_{\text{x}}(r) > x_{\text{coord}}(s) then right_{\text{x}}(r) \leftarrow x_{\text{coord}}(s); end;\end{verbatim}
if \(right_{\text{x}}(s) < x_{\text{coord}}(s)\) then
  \begin{verbatim}
beg\text{begin right}_{\text{x}}(s) \leftarrow x_{\text{coord}}(s); if left_{\text{x}}(qq) < x_{\text{coord}}(s) then left_{\text{x}}(qq) \leftarrow x_{\text{coord}}(s); end;\end{verbatim}
\end{verbatim}
\end{verbatim}

Page D194, lines 4 and 5 (1/24/92)

[Delete those two lines; I no longer believe that the assertion has been proved (although it might be true).]

Page D194, lines 7–13 of section 424 (1/24/92)

\begin{verbatim}
if \(y_{\text{coord}}(r) < y_{\text{coord}}(p)\) then \(y_{\text{coord}}(r) \leftarrow y_{\text{coord}}(p)\)
else if \(y_{\text{coord}}(r) > dest_{\text{y}}\) then \(y_{\text{coord}}(r) \leftarrow dest_{\text{y}}\);
if \(x_{\text{coord}}(p) + y_{\text{coord}}(r) > dest_{\text{x}} + dest_{\text{y}}\) then \(y_{\text{coord}}(r) \leftarrow dest_{\text{x}} + dest_{\text{y}} - x_{\text{coord}}(p)\);
if \(left_{\text{y}}(r) > y_{\text{coord}}(r)\) then
  \begin{verbatim}
beg\text{begin left}_{\text{y}}(r) \leftarrow y_{\text{coord}}(r); if right_{\text{y}}(p) > y_{\text{coord}}(r) then right_{\text{y}}(p) \leftarrow y_{\text{coord}}(r); end;\end{verbatim}
if \(right_{\text{y}}(r) < y_{\text{coord}}(r)\) then
  \begin{verbatim}
beg\text{begin right}_{\text{y}}(r) \leftarrow y_{\text{coord}}(r); if left_{\text{y}}(q) < y_{\text{coord}}(r) then left_{\text{y}}(q) \leftarrow y_{\text{coord}}(r); end;\end{verbatim}
\end{verbatim}
\end{verbatim}

Page D194, lines 8–11 from the bottom (1/24/92)

\begin{verbatim}
if \(right_{\text{y}}(r) < y_{\text{coord}}(r)\) then
  \begin{verbatim}
beg\text{begin right}_{\text{y}}(r) \leftarrow y_{\text{coord}}(r); if left_{\text{y}}(q) < y_{\text{coord}}(r) then left_{\text{y}}(q) \leftarrow y_{\text{coord}}(r); end;\end{verbatim}
\end{verbatim}
if $y_{\text{coord}}(s) < y_{\text{coord}}(r)$ then $y_{\text{coord}}(s) \leftarrow y_{\text{coord}}(r)$
else if $y_{\text{coord}}(s) > dest_y$ then $y_{\text{coord}}(s) \leftarrow dest_y$
if $x_{\text{coord}}(r) + y_{\text{coord}}(s) > dest_x + dest_y$ then $y_{\text{coord}}(s) \leftarrow dest_x + dest_y - x_{\text{coord}}(r)$
if left_y(s) > $y_{\text{coord}}(s)$ then
  begin left_y(s) \leftarrow $y_{\text{coord}}(s)$;
  if right_y(r) > $y_{\text{coord}}(s)$ then right_y(r) \leftarrow $y_{\text{coord}}(s)$;  end;
if right_y(s) < $y_{\text{coord}}(s)$ then
  begin right_y(s) \leftarrow $y_{\text{coord}}(s)$;
  if left_y(q) < $y_{\text{coord}}(s)$ then left_y(q) \leftarrow $y_{\text{coord}}(s)$;  end;

if right_y(s) < $y_{\text{coord}}(s)$ then
  begin right_y(s) \leftarrow $y_{\text{coord}}(s)$;
  if left_y(q) < $y_{\text{coord}}(s)$ then left_y(q) \leftarrow $y_{\text{coord}}(s)$;  end;

$p \leftarrow \text{dep_list}(p)$;  $r \leftarrow \text{inf_val}$;
repeat if $\text{value(info}(p)) \geq \text{value}(r)$ then

The $\text{label_loc}$ and $\text{label_char}$ arrays have been set up to record all the starting addresses; we have