```
%%
%% This is file `expl3-code.tex',
%% generated with the docstrip utility.
%%
%% The original source files were:
%%
%% expl3.dtx  (with options: `code')
%% l3bootstrap.dtx  (with options: `code')
%% l3names.dtx  (with options: `code')
%% l3basics.dtx  (with options: `code')
%% l3expan.dtx  (with options: `code')
%% l3quark.dtx  (with options: `code')
%% l3tl.dtx  (with options: `code')
%% l3tl-build.dtx  (with options: `code')
%% l3str.dtx  (with options: `code')
%% l3seq.dtx  (with options: `code')
%% l3int.dtx  (with options: `code')
%% l3flag.dtx  (with options: `code')
%% l3prg.dtx  (with options: `code')
%% l3sys.dtx  (with options: `code')
%% l3clist.dtx  (with options: `code')
%% l3token.dtx  (with options: `code')
%% l3prop.dtx  (with options: `code')
%% l3msg.dtx  (with options: `code')
%% l3file.dtx  (with options: `code')
%% l3skip.dtx  (with options: `code')
%% l3keys.dtx  (with options: `code')
%% l3intarray.dtx  (with options: `code')
%% l3fp.dtx  (with options: `code')
%% l3fp-aux.dtx  (with options: `code')
%% l3fp-traps.dtx  (with options: `code')
%% l3fp-round.dtx  (with options: `code')
%% l3fp-parse.dtx  (with options: `code')
%% l3fp-assign.dtx  (with options: `code')
%% l3fp-logic.dtx  (with options: `code')
%% l3fp-basics.dtx  (with options: `code')
%% l3fp-extended.dtx  (with options: `code')
%% l3fp-expo.dtx  (with options: `code')
%% l3fp-trig.dtx  (with options: `code')
%% l3fp-convert.dtx  (with options: `code')
%% l3fp-random.dtx  (with options: `code')
%% l3fp-types.dtx  (with options: `code')
%% l3fp-symbolic.dtx  (with options: `code')
%% l3fp-functions.dtx  (with options: `code')
%% l3fparray.dtx  (with options: `code')
%% l3bitset.dtx  (with options: `code')
%% l3cctab.dtx  (with options: `code')
%% l3sort.dtx  (with options: `code')
%% l3str-convert.dtx  (with options: `code')
%% l3tl-analysis.dtx  (with options: `code')
%% l3benchmark.dtx  (with options: `code')
%% l3regex.dtx  (with options: `code')
%% l3box.dtx  (with options: `code')
%% l3color.dtx  (with options: `code')
%% l3graphics.dtx  (with options: `code')
%% l3opacity.dtx  (with options: `code')
%% l3pdf.dtx  (with options: `code')
%% l3coffins.dtx  (with options: `code')
%% l3luatex.dtx  (with options: `code')
%% l3unicode.dtx  (with options: `code')
%% l3text.dtx  (with options: `code')
%% l3text-case.dtx  (with options: `code')
%% l3text-map.dtx  (with options: `code')
%% l3text-purify.dtx  (with options: `code')
%% l3legacy.dtx  (with options: `code')
%% l3deprecation.dtx  (with options: `code')
%%
%% Copyright (C) 1990-2025 The LaTeX Project
%%
%% It may be distributed and/or modified under the conditions of
%% the LaTeX Project Public License (LPPL), either version 1.3c of
%% this license or (at your option) any later version.  The latest
%% version of this license is in the file:
%%
%%    https://www.latex-project.org/lppl.txt
%%
%% This file is part of the "l3kernel bundle" (The Work in LPPL)
%% and all files in that bundle must be distributed together.
%%
%% File: expl3.dtx
\def\ExplFileDate{2025-04-29}%
\begingroup
  \def\next{\endgroup}%
  \expandafter\ifx\csname PackageError\endcsname\relax
    \begingroup
      \def\next{\endgroup\endgroup}%
      \def\PackageError#1#2#3%
        {%
          \endgroup
          \errhelp{#3}%
          \errmessage{#1 Error: #2!}%
        }%
  \fi
  \expandafter\ifx\csname ExplLoaderFileDate\endcsname\relax
    \def\next
      {%
        \PackageError{expl3}{No expl3 loader detected}
          {%
            You have attempted to use the expl3 code directly rather than using
            the correct loader. Loading of expl3 will abort.
          }%
        \endgroup
        \endinput
      }
  \else
    \ifx\ExplLoaderFileDate\ExplFileDate
    \else
      \def\next
        {%
          \PackageError{expl3}{Mismatched expl3 files detected}
            {%
              You have attempted to load expl3 with mismatched files:
              probably you have one or more files 'locally installed' which
              are in conflict. Loading of expl3 will abort.
            }%
          \endgroup
          \endinput
        }%
    \fi
\fi
\next
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname ver@expl3-code.tex\endcsname\relax
  \expandafter\edef\csname ver@expl3-code.tex\endcsname
    {%
      \ExplFileDate\space
      L3 programming layer
    }%
\else
  \expandafter\endinput
\fi
\immediate\write-1 %
  {%
    Package: expl3
      \ExplFileDate\space
      L3 programming layer (code)%
  }%
%% File: l3bootstrap.dtx
```

```
\begingroup\expandafter\expandafter\expandafter\endgroup
  \expandafter\ifx\csname pdfstrcmp\endcsname\relax
  \let\pdfstrcmp\strcmp
\fi
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname directlua\endcsname\relax
\else
  \ifnum\luatexversion<110 %
  \else
    \begingroup\expandafter\expandafter\expandafter\endgroup
    \expandafter\ifx\csname newcatcodetable\endcsname\relax
      \input{ltluatex}%
    \fi
    \begingroup\expandafter\expandafter\expandafter\endgroup
    \expandafter\ifx\csname newluabytecode\endcsname\relax
    \else
      \newluabytecode\@expl@luadata@bytecode
    \fi
    \directlua{require("expl3")}%
    \ifnum 0%
      \directlua{
        if status.ini_version then
          tex.write("1")
        end
      }>0 %
      \everyjob\expandafter{%
        \the\expandafter\everyjob
        \csname\detokenize{lua_now:n}\endcsname{require("expl3")}%
      }%
    \fi
  \fi
\fi
\begingroup
  \def\next{\endgroup}%
  \def\ShortText{Required primitives not found}%
  \def\LongText%
    {%
      The L3 programming layer requires the e-TeX primitives and the
      \LineBreak 'pdfTeX utilities' as described in the README file.
      \LineBreak
      These are available in the engines\LineBreak
      - pdfTeX v1.40.20\LineBreak
      - XeTeX v0.999991\LineBreak
      - LuaTeX v1.10\LineBreak
      - e-(u)pTeX v3.8.2\LineBreak
      - Prote (2021)\LineBreak
      or later.\LineBreak
      \LineBreak
    }%
  \ifnum0%
    \expandafter\ifx\csname luatexversion\endcsname\relax
      \expandafter\ifx\csname expanded\endcsname\relax\else 1\fi
    \else
      \ifnum\luatexversion<110 \else 1\fi
    \fi
    =0 %
      \newlinechar`\^^J %
      \def\LineBreak{\noexpand\MessageBreak}%
      \expandafter\ifx\csname PackageError\endcsname\relax
        \def\LineBreak{^^J}%
        \begingroup
          \lccode`\~=`\ \lccode`\}=`\ %
          \lccode`\T=`\T\lccode`\H=`\H%
          \catcode`\ =11 %
\lowercase{\endgroup\def\PackageError#1#2#3{%
\begingroup\errorcontextlines-1\immediate\write0{}\errhelp{#3}\def%
\                                                   {#1 Error: #2.^^J^^J
Type  H <return>  for immediate help}\def~{\errmessage{%
\                                                   }}~\endgroup}}%
      \fi
      \edef\next
        {%
          \noexpand\PackageError{expl3}{\ShortText}
            {\LongText Loading of expl3 will abort!}%
          \endgroup
          \noexpand\endinput
        }%
  \fi
\next
\protected\edef\ExplSyntaxOff
  {%
    \protected\def\noexpand\ExplSyntaxOff{}%
    \catcode   9 = \the\catcode   9\relax
    \catcode  32 = \the\catcode  32\relax
    \catcode  34 = \the\catcode  34\relax
    \catcode  58 = \the\catcode  58\relax
    \catcode  94 = \the\catcode  94\relax
    \catcode  95 = \the\catcode  95\relax
    \catcode 124 = \the\catcode 124\relax
    \catcode 126 = \the\catcode 126\relax
    \endlinechar = \the\endlinechar\relax
    \chardef\csname\detokenize{l__kernel_expl_bool}\endcsname = 0\relax
  }%
\catcode 9   = 9\relax
\catcode 32  = 9\relax
\catcode 34  = 12\relax
\catcode 58  = 11\relax
\catcode 94  = 7\relax
\catcode 95  = 11\relax
\catcode 124 = 12\relax
\catcode 126 = 10\relax
\endlinechar = 32\relax
\global\chardef\l__kernel_expl_bool = 1\relax
\protected \def \ExplSyntaxOn
  {
    \bool_if:NF \l__kernel_expl_bool
      {
        \cs_set_protected:Npe \ExplSyntaxOff
          {
            \char_set_catcode:nn { 9 }   { \char_value_catcode:n { 9 } }
            \char_set_catcode:nn { 32 }  { \char_value_catcode:n { 32 } }
            \char_set_catcode:nn { 34 }  { \char_value_catcode:n { 34 } }
            \char_set_catcode:nn { 58 }  { \char_value_catcode:n { 58 } }
            \char_set_catcode:nn { 94 }  { \char_value_catcode:n { 94 } }
            \char_set_catcode:nn { 95 }  { \char_value_catcode:n { 95 } }
            \char_set_catcode:nn { 124 } { \char_value_catcode:n { 124 } }
            \char_set_catcode:nn { 126 } { \char_value_catcode:n { 126 } }
            \tex_endlinechar:D =
              \tex_the:D \tex_endlinechar:D \scan_stop:
            \bool_set_false:N \l__kernel_expl_bool
            \cs_set_protected:Npn \ExplSyntaxOff { }
          }
      }
    \char_set_catcode_ignore:n           { 9 }   % tab
    \char_set_catcode_ignore:n           { 32 }  % space
    \char_set_catcode_other:n            { 34 }  % double quote
    \char_set_catcode_letter:n           { 58 }  % colon
    \char_set_catcode_math_superscript:n { 94 }  % circumflex
    \char_set_catcode_letter:n           { 95 }  % underscore
    \char_set_catcode_other:n            { 124 } % pipe
    \char_set_catcode_space:n            { 126 } % tilde
    \tex_endlinechar:D = 32 \scan_stop:
    \bool_set_true:N \l__kernel_expl_bool
  }
%% File: l3names.dtx
\let \tex_global:D \global
\let \tex_let:D    \let
\begingroup
```

```
\long \def \__kernel_primitive:NN #1#2
  { \tex_global:D \tex_let:D #2 #1 }
\__kernel_primitive:NN \                      \tex_space:D
\__kernel_primitive:NN \/                     \tex_italiccorrection:D
\__kernel_primitive:NN \-                     \tex_hyphen:D
\__kernel_primitive:NN \above                 \tex_above:D
\__kernel_primitive:NN \abovedisplayshortskip \tex_abovedisplayshortskip:D
\__kernel_primitive:NN \abovedisplayskip      \tex_abovedisplayskip:D
\__kernel_primitive:NN \abovewithdelims       \tex_abovewithdelims:D
\__kernel_primitive:NN \accent                \tex_accent:D
\__kernel_primitive:NN \adjdemerits           \tex_adjdemerits:D
\__kernel_primitive:NN \advance               \tex_advance:D
\__kernel_primitive:NN \afterassignment       \tex_afterassignment:D
\__kernel_primitive:NN \aftergroup            \tex_aftergroup:D
\__kernel_primitive:NN \atop                  \tex_atop:D
\__kernel_primitive:NN \atopwithdelims        \tex_atopwithdelims:D
\__kernel_primitive:NN \badness               \tex_badness:D
\__kernel_primitive:NN \baselineskip          \tex_baselineskip:D
\__kernel_primitive:NN \batchmode             \tex_batchmode:D
\__kernel_primitive:NN \begingroup            \tex_begingroup:D
\__kernel_primitive:NN \belowdisplayshortskip \tex_belowdisplayshortskip:D
\__kernel_primitive:NN \belowdisplayskip      \tex_belowdisplayskip:D
\__kernel_primitive:NN \binoppenalty          \tex_binoppenalty:D
\__kernel_primitive:NN \botmark               \tex_botmark:D
\__kernel_primitive:NN \box                   \tex_box:D
\__kernel_primitive:NN \boxmaxdepth           \tex_boxmaxdepth:D
\__kernel_primitive:NN \brokenpenalty         \tex_brokenpenalty:D
\__kernel_primitive:NN \catcode               \tex_catcode:D
\__kernel_primitive:NN \char                  \tex_char:D
\__kernel_primitive:NN \chardef               \tex_chardef:D
\__kernel_primitive:NN \cleaders              \tex_cleaders:D
\__kernel_primitive:NN \closein               \tex_closein:D
\__kernel_primitive:NN \closeout              \tex_closeout:D
\__kernel_primitive:NN \clubpenalty           \tex_clubpenalty:D
\__kernel_primitive:NN \copy                  \tex_copy:D
\__kernel_primitive:NN \count                 \tex_count:D
\__kernel_primitive:NN \countdef              \tex_countdef:D
\__kernel_primitive:NN \cr                    \tex_cr:D
\__kernel_primitive:NN \crcr                  \tex_crcr:D
\__kernel_primitive:NN \csname                \tex_csname:D
\__kernel_primitive:NN \day                   \tex_day:D
\__kernel_primitive:NN \deadcycles            \tex_deadcycles:D
\__kernel_primitive:NN \def                   \tex_def:D
\__kernel_primitive:NN \defaulthyphenchar     \tex_defaulthyphenchar:D
\__kernel_primitive:NN \defaultskewchar       \tex_defaultskewchar:D
\__kernel_primitive:NN \delcode               \tex_delcode:D
```

```
\__kernel_primitive:NN \delimiter              \tex_delimiter:D
\__kernel_primitive:NN \delimiterfactor        \tex_delimiterfactor:D
\__kernel_primitive:NN \delimitershortfall     \tex_delimitershortfall:D
\__kernel_primitive:NN \dimen                  \tex_dimen:D
\__kernel_primitive:NN \dimendef               \tex_dimendef:D
\__kernel_primitive:NN \discretionary          \tex_discretionary:D
\__kernel_primitive:NN \displayindent          \tex_displayindent:D
\__kernel_primitive:NN \displaylimits          \tex_displaylimits:D
\__kernel_primitive:NN \displaystyle           \tex_displaystyle:D
\__kernel_primitive:NN \displaywidowpenalty    \tex_displaywidowpenalty:D
\__kernel_primitive:NN \displaywidth           \tex_displaywidth:D
\__kernel_primitive:NN \divide                 \tex_divide:D
\__kernel_primitive:NN \doublehyphendemerits   \tex_doublehyphendemerits:D
\__kernel_primitive:NN \dp                     \tex_dp:D
\__kernel_primitive:NN \dump                   \tex_dump:D
\__kernel_primitive:NN \edef                   \tex_edef:D
\__kernel_primitive:NN \else                   \tex_else:D
\__kernel_primitive:NN \emergencystretch       \tex_emergencystretch:D
\__kernel_primitive:NN \end                    \tex_end:D
\__kernel_primitive:NN \endcsname              \tex_endcsname:D
\__kernel_primitive:NN \endgroup               \tex_endgroup:D
\__kernel_primitive:NN \endinput               \tex_endinput:D
\__kernel_primitive:NN \endlinechar            \tex_endlinechar:D
\__kernel_primitive:NN \eqno                   \tex_eqno:D
\__kernel_primitive:NN \errhelp                \tex_errhelp:D
\__kernel_primitive:NN \errmessage             \tex_errmessage:D
\__kernel_primitive:NN \errorcontextlines      \tex_errorcontextlines:D
\__kernel_primitive:NN \errorstopmode          \tex_errorstopmode:D
\__kernel_primitive:NN \escapechar             \tex_escapechar:D
\__kernel_primitive:NN \everycr                \tex_everycr:D
\__kernel_primitive:NN \everydisplay           \tex_everydisplay:D
\__kernel_primitive:NN \everyhbox              \tex_everyhbox:D
\__kernel_primitive:NN \everyjob               \tex_everyjob:D
\__kernel_primitive:NN \everymath              \tex_everymath:D
\__kernel_primitive:NN \everypar               \tex_everypar:D
\__kernel_primitive:NN \everyvbox              \tex_everyvbox:D
\__kernel_primitive:NN \exhyphenpenalty        \tex_exhyphenpenalty:D
\__kernel_primitive:NN \expandafter            \tex_expandafter:D
\__kernel_primitive:NN \fam                    \tex_fam:D
\__kernel_primitive:NN \fi                     \tex_fi:D
\__kernel_primitive:NN \finalhyphendemerits    \tex_finalhyphendemerits:D
\__kernel_primitive:NN \firstmark              \tex_firstmark:D
\__kernel_primitive:NN \floatingpenalty        \tex_floatingpenalty:D
\__kernel_primitive:NN \font                   \tex_font:D
\__kernel_primitive:NN \fontdimen              \tex_fontdimen:D
\__kernel_primitive:NN \fontname               \tex_fontname:D
```

```
\__kernel_primitive:NN \futurelet             \tex_futurelet:D
\__kernel_primitive:NN \gdef                  \tex_gdef:D
\__kernel_primitive:NN \global                \tex_global:D
\__kernel_primitive:NN \globaldefs            \tex_globaldefs:D
\__kernel_primitive:NN \halign                \tex_halign:D
\__kernel_primitive:NN \hangafter             \tex_hangafter:D
\__kernel_primitive:NN \hangindent            \tex_hangindent:D
\__kernel_primitive:NN \hbadness              \tex_hbadness:D
\__kernel_primitive:NN \hbox                  \tex_hbox:D
\__kernel_primitive:NN \hfil                  \tex_hfil:D
\__kernel_primitive:NN \hfill                 \tex_hfill:D
\__kernel_primitive:NN \hfilneg               \tex_hfilneg:D
\__kernel_primitive:NN \hfuzz                 \tex_hfuzz:D
\__kernel_primitive:NN \hoffset               \tex_hoffset:D
\__kernel_primitive:NN \holdinginserts        \tex_holdinginserts:D
\__kernel_primitive:NN \hrule                 \tex_hrule:D
\__kernel_primitive:NN \hsize                 \tex_hsize:D
\__kernel_primitive:NN \hskip                 \tex_hskip:D
\__kernel_primitive:NN \hss                   \tex_hss:D
\__kernel_primitive:NN \ht                    \tex_ht:D
\__kernel_primitive:NN \hyphenation           \tex_hyphenation:D
\__kernel_primitive:NN \hyphenchar            \tex_hyphenchar:D
\__kernel_primitive:NN \hyphenpenalty         \tex_hyphenpenalty:D
\__kernel_primitive:NN \if                    \tex_if:D
\__kernel_primitive:NN \ifcase                \tex_ifcase:D
\__kernel_primitive:NN \ifcat                 \tex_ifcat:D
\__kernel_primitive:NN \ifdim                 \tex_ifdim:D
\__kernel_primitive:NN \ifeof                 \tex_ifeof:D
\__kernel_primitive:NN \iffalse               \tex_iffalse:D
\__kernel_primitive:NN \ifhbox                \tex_ifhbox:D
\__kernel_primitive:NN \ifhmode               \tex_ifhmode:D
\__kernel_primitive:NN \ifinner               \tex_ifinner:D
\__kernel_primitive:NN \ifmmode               \tex_ifmmode:D
\__kernel_primitive:NN \ifnum                 \tex_ifnum:D
\__kernel_primitive:NN \ifodd                 \tex_ifodd:D
\__kernel_primitive:NN \iftrue                \tex_iftrue:D
\__kernel_primitive:NN \ifvbox                \tex_ifvbox:D
\__kernel_primitive:NN \ifvmode               \tex_ifvmode:D
\__kernel_primitive:NN \ifvoid                \tex_ifvoid:D
\__kernel_primitive:NN \ifx                   \tex_ifx:D
\__kernel_primitive:NN \ignorespaces          \tex_ignorespaces:D
\__kernel_primitive:NN \immediate             \tex_immediate:D
\__kernel_primitive:NN \indent                \tex_indent:D
\__kernel_primitive:NN \input                 \tex_input:D
\__kernel_primitive:NN \inputlineno           \tex_inputlineno:D
\__kernel_primitive:NN \insert                \tex_insert:D
```

```
\__kernel_primitive:NN \insertpenalties       \tex_insertpenalties:D
\__kernel_primitive:NN \interlinepenalty      \tex_interlinepenalty:D
\__kernel_primitive:NN \jobname               \tex_jobname:D
\__kernel_primitive:NN \kern                  \tex_kern:D
\__kernel_primitive:NN \language              \tex_language:D
\__kernel_primitive:NN \lastbox               \tex_lastbox:D
\__kernel_primitive:NN \lastkern              \tex_lastkern:D
\__kernel_primitive:NN \lastpenalty           \tex_lastpenalty:D
\__kernel_primitive:NN \lastskip              \tex_lastskip:D
\__kernel_primitive:NN \lccode                \tex_lccode:D
\__kernel_primitive:NN \leaders               \tex_leaders:D
\__kernel_primitive:NN \left                  \tex_left:D
\__kernel_primitive:NN \lefthyphenmin         \tex_lefthyphenmin:D
\__kernel_primitive:NN \leftskip              \tex_leftskip:D
\__kernel_primitive:NN \leqno                 \tex_leqno:D
\__kernel_primitive:NN \let                   \tex_let:D
\__kernel_primitive:NN \limits                \tex_limits:D
\__kernel_primitive:NN \linepenalty           \tex_linepenalty:D
\__kernel_primitive:NN \lineskip              \tex_lineskip:D
\__kernel_primitive:NN \lineskiplimit         \tex_lineskiplimit:D
\__kernel_primitive:NN \long                  \tex_long:D
\__kernel_primitive:NN \looseness             \tex_looseness:D
\__kernel_primitive:NN \lower                 \tex_lower:D
\__kernel_primitive:NN \lowercase             \tex_lowercase:D
\__kernel_primitive:NN \mag                   \tex_mag:D
\__kernel_primitive:NN \mark                  \tex_mark:D
\__kernel_primitive:NN \mathaccent            \tex_mathaccent:D
\__kernel_primitive:NN \mathbin               \tex_mathbin:D
\__kernel_primitive:NN \mathchar              \tex_mathchar:D
\__kernel_primitive:NN \mathchardef           \tex_mathchardef:D
\__kernel_primitive:NN \mathchoice            \tex_mathchoice:D
\__kernel_primitive:NN \mathclose             \tex_mathclose:D
\__kernel_primitive:NN \mathcode              \tex_mathcode:D
\__kernel_primitive:NN \mathinner             \tex_mathinner:D
\__kernel_primitive:NN \mathop                \tex_mathop:D
\__kernel_primitive:NN \mathopen              \tex_mathopen:D
\__kernel_primitive:NN \mathord               \tex_mathord:D
\__kernel_primitive:NN \mathpunct             \tex_mathpunct:D
\__kernel_primitive:NN \mathrel               \tex_mathrel:D
\__kernel_primitive:NN \mathsurround          \tex_mathsurround:D
\__kernel_primitive:NN \maxdeadcycles         \tex_maxdeadcycles:D
\__kernel_primitive:NN \maxdepth              \tex_maxdepth:D
\__kernel_primitive:NN \meaning               \tex_meaning:D
\__kernel_primitive:NN \medmuskip             \tex_medmuskip:D
\__kernel_primitive:NN \message               \tex_message:D
\__kernel_primitive:NN \mkern                 \tex_mkern:D
```

```
\__kernel_primitive:NN \month                 \tex_month:D
\__kernel_primitive:NN \moveleft              \tex_moveleft:D
\__kernel_primitive:NN \moveright             \tex_moveright:D
\__kernel_primitive:NN \mskip                 \tex_mskip:D
\__kernel_primitive:NN \multiply              \tex_multiply:D
\__kernel_primitive:NN \muskip                \tex_muskip:D
\__kernel_primitive:NN \muskipdef             \tex_muskipdef:D
\__kernel_primitive:NN \newlinechar           \tex_newlinechar:D
\__kernel_primitive:NN \noalign               \tex_noalign:D
\__kernel_primitive:NN \noboundary            \tex_noboundary:D
\__kernel_primitive:NN \noexpand              \tex_noexpand:D
\__kernel_primitive:NN \noindent              \tex_noindent:D
\__kernel_primitive:NN \nolimits              \tex_nolimits:D
\__kernel_primitive:NN \nonscript             \tex_nonscript:D
\__kernel_primitive:NN \nonstopmode           \tex_nonstopmode:D
\__kernel_primitive:NN \nulldelimiterspace    \tex_nulldelimiterspace:D
\__kernel_primitive:NN \nullfont              \tex_nullfont:D
\__kernel_primitive:NN \number                \tex_number:D
\__kernel_primitive:NN \omit                  \tex_omit:D
\__kernel_primitive:NN \openin                \tex_openin:D
\__kernel_primitive:NN \openout               \tex_openout:D
\__kernel_primitive:NN \or                    \tex_or:D
\__kernel_primitive:NN \outer                 \tex_outer:D
\__kernel_primitive:NN \output                \tex_output:D
\__kernel_primitive:NN \outputpenalty         \tex_outputpenalty:D
\__kernel_primitive:NN \over                  \tex_over:D
\__kernel_primitive:NN \overfullrule          \tex_overfullrule:D
\__kernel_primitive:NN \overline              \tex_overline:D
\__kernel_primitive:NN \overwithdelims        \tex_overwithdelims:D
\__kernel_primitive:NN \pagedepth             \tex_pagedepth:D
\__kernel_primitive:NN \pagefilllstretch      \tex_pagefilllstretch:D
\__kernel_primitive:NN \pagefillstretch       \tex_pagefillstretch:D
\__kernel_primitive:NN \pagefilstretch        \tex_pagefilstretch:D
\__kernel_primitive:NN \pagegoal              \tex_pagegoal:D
\__kernel_primitive:NN \pageshrink            \tex_pageshrink:D
\__kernel_primitive:NN \pagestretch           \tex_pagestretch:D
\__kernel_primitive:NN \pagetotal             \tex_pagetotal:D
\__kernel_primitive:NN \par                   \tex_par:D
\__kernel_primitive:NN \parfillskip           \tex_parfillskip:D
\__kernel_primitive:NN \parindent             \tex_parindent:D
\__kernel_primitive:NN \parshape              \tex_parshape:D
\__kernel_primitive:NN \parskip               \tex_parskip:D
\__kernel_primitive:NN \patterns              \tex_patterns:D
\__kernel_primitive:NN \pausing               \tex_pausing:D
\__kernel_primitive:NN \penalty               \tex_penalty:D
\__kernel_primitive:NN \postdisplaypenalty    \tex_postdisplaypenalty:D
```

```
\__kernel_primitive:NN \predisplaypenalty      \tex_predisplaypenalty:D
\__kernel_primitive:NN \predisplaysize         \tex_predisplaysize:D
\__kernel_primitive:NN \pretolerance           \tex_pretolerance:D
\__kernel_primitive:NN \prevdepth              \tex_prevdepth:D
\__kernel_primitive:NN \prevgraf               \tex_prevgraf:D
\__kernel_primitive:NN \radical                \tex_radical:D
\__kernel_primitive:NN \raise                  \tex_raise:D
\__kernel_primitive:NN \read                   \tex_read:D
\__kernel_primitive:NN \relax                  \tex_relax:D
\__kernel_primitive:NN \relpenalty             \tex_relpenalty:D
\__kernel_primitive:NN \right                  \tex_right:D
\__kernel_primitive:NN \righthyphenmin         \tex_righthyphenmin:D
\__kernel_primitive:NN \rightskip              \tex_rightskip:D
\__kernel_primitive:NN \romannumeral           \tex_romannumeral:D
\__kernel_primitive:NN \scriptfont             \tex_scriptfont:D
\__kernel_primitive:NN \scriptscriptfont       \tex_scriptscriptfont:D
\__kernel_primitive:NN \scriptscriptstyle      \tex_scriptscriptstyle:D
\__kernel_primitive:NN \scriptspace            \tex_scriptspace:D
\__kernel_primitive:NN \scriptstyle            \tex_scriptstyle:D
\__kernel_primitive:NN \scrollmode             \tex_scrollmode:D
\__kernel_primitive:NN \setbox                 \tex_setbox:D
\__kernel_primitive:NN \setlanguage            \tex_setlanguage:D
\__kernel_primitive:NN \sfcode                 \tex_sfcode:D
\__kernel_primitive:NN \shipout                \tex_shipout:D
\__kernel_primitive:NN \show                   \tex_show:D
\__kernel_primitive:NN \showbox                \tex_showbox:D
\__kernel_primitive:NN \showboxbreadth         \tex_showboxbreadth:D
\__kernel_primitive:NN \showboxdepth           \tex_showboxdepth:D
\__kernel_primitive:NN \showlists              \tex_showlists:D
\__kernel_primitive:NN \showthe                \tex_showthe:D
\__kernel_primitive:NN \skewchar               \tex_skewchar:D
\__kernel_primitive:NN \skip                   \tex_skip:D
\__kernel_primitive:NN \skipdef                \tex_skipdef:D
\__kernel_primitive:NN \spacefactor            \tex_spacefactor:D
\__kernel_primitive:NN \spaceskip              \tex_spaceskip:D
\__kernel_primitive:NN \span                   \tex_span:D
\__kernel_primitive:NN \special                \tex_special:D
\__kernel_primitive:NN \splitbotmark           \tex_splitbotmark:D
\__kernel_primitive:NN \splitfirstmark         \tex_splitfirstmark:D
\__kernel_primitive:NN \splitmaxdepth          \tex_splitmaxdepth:D
\__kernel_primitive:NN \splittopskip           \tex_splittopskip:D
\__kernel_primitive:NN \string                 \tex_string:D
\__kernel_primitive:NN \tabskip                \tex_tabskip:D
\__kernel_primitive:NN \textfont               \tex_textfont:D
\__kernel_primitive:NN \textstyle              \tex_textstyle:D
\__kernel_primitive:NN \the                    \tex_the:D
```

```
\__kernel_primitive:NN \thickmuskip           \tex_thickmuskip:D
\__kernel_primitive:NN \thinmuskip            \tex_thinmuskip:D
\__kernel_primitive:NN \time                  \tex_time:D
\__kernel_primitive:NN \toks                  \tex_toks:D
\__kernel_primitive:NN \toksdef               \tex_toksdef:D
\__kernel_primitive:NN \tolerance             \tex_tolerance:D
\__kernel_primitive:NN \topmark               \tex_topmark:D
\__kernel_primitive:NN \topskip               \tex_topskip:D
\__kernel_primitive:NN \tracingcommands       \tex_tracingcommands:D
\__kernel_primitive:NN \tracinglostchars      \tex_tracinglostchars:D
\__kernel_primitive:NN \tracingmacros         \tex_tracingmacros:D
\__kernel_primitive:NN \tracingonline         \tex_tracingonline:D
\__kernel_primitive:NN \tracingoutput         \tex_tracingoutput:D
\__kernel_primitive:NN \tracingpages          \tex_tracingpages:D
\__kernel_primitive:NN \tracingparagraphs     \tex_tracingparagraphs:D
\__kernel_primitive:NN \tracingrestores       \tex_tracingrestores:D
\__kernel_primitive:NN \tracingstats          \tex_tracingstats:D
\__kernel_primitive:NN \uccode                \tex_uccode:D
\__kernel_primitive:NN \uchyph                \tex_uchyph:D
\__kernel_primitive:NN \underline             \tex_underline:D
\__kernel_primitive:NN \unhbox                \tex_unhbox:D
\__kernel_primitive:NN \unhcopy               \tex_unhcopy:D
\__kernel_primitive:NN \unkern                \tex_unkern:D
\__kernel_primitive:NN \unpenalty             \tex_unpenalty:D
\__kernel_primitive:NN \unskip                \tex_unskip:D
\__kernel_primitive:NN \unvbox                \tex_unvbox:D
\__kernel_primitive:NN \unvcopy               \tex_unvcopy:D
\__kernel_primitive:NN \uppercase             \tex_uppercase:D
\__kernel_primitive:NN \vadjust               \tex_vadjust:D
\__kernel_primitive:NN \valign                \tex_valign:D
\__kernel_primitive:NN \vbadness              \tex_vbadness:D
\__kernel_primitive:NN \vbox                  \tex_vbox:D
\__kernel_primitive:NN \vcenter               \tex_vcenter:D
\__kernel_primitive:NN \vfil                  \tex_vfil:D
\__kernel_primitive:NN \vfill                 \tex_vfill:D
\__kernel_primitive:NN \vfilneg               \tex_vfilneg:D
\__kernel_primitive:NN \vfuzz                 \tex_vfuzz:D
\__kernel_primitive:NN \voffset               \tex_voffset:D
\__kernel_primitive:NN \vrule                 \tex_vrule:D
\__kernel_primitive:NN \vsize                 \tex_vsize:D
\__kernel_primitive:NN \vskip                 \tex_vskip:D
\__kernel_primitive:NN \vsplit                \tex_vsplit:D
\__kernel_primitive:NN \vss                   \tex_vss:D
\__kernel_primitive:NN \vtop                  \tex_vtop:D
\__kernel_primitive:NN \wd                    \tex_wd:D
\__kernel_primitive:NN \widowpenalty          \tex_widowpenalty:D
```

```
\__kernel_primitive:NN \write                  \tex_write:D
\__kernel_primitive:NN \xdef                   \tex_xdef:D
\__kernel_primitive:NN \xleaders               \tex_xleaders:D
\__kernel_primitive:NN \xspaceskip             \tex_xspaceskip:D
\__kernel_primitive:NN \year                   \tex_year:D
\__kernel_primitive:NN \beginL                 \tex_beginL:D
\__kernel_primitive:NN \beginR                 \tex_beginR:D
\__kernel_primitive:NN \botmarks               \tex_botmarks:D
\__kernel_primitive:NN \clubpenalties          \tex_clubpenalties:D
\__kernel_primitive:NN \currentgrouplevel      \tex_currentgrouplevel:D
\__kernel_primitive:NN \currentgrouptype       \tex_currentgrouptype:D
\__kernel_primitive:NN \currentifbranch        \tex_currentifbranch:D
\__kernel_primitive:NN \currentiflevel         \tex_currentiflevel:D
\__kernel_primitive:NN \currentiftype          \tex_currentiftype:D
\__kernel_primitive:NN \detokenize             \tex_detokenize:D
\__kernel_primitive:NN \dimexpr                \tex_dimexpr:D
\__kernel_primitive:NN \displaywidowpenalties  \tex_displaywidowpenalties:D
\__kernel_primitive:NN \endL                   \tex_endL:D
\__kernel_primitive:NN \endR                   \tex_endR:D
\__kernel_primitive:NN \eTeXrevision           \tex_eTeXrevision:D
\__kernel_primitive:NN \eTeXversion            \tex_eTeXversion:D
\__kernel_primitive:NN \everyeof               \tex_everyeof:D
\__kernel_primitive:NN \firstmarks             \tex_firstmarks:D
\__kernel_primitive:NN \fontchardp             \tex_fontchardp:D
\__kernel_primitive:NN \fontcharht             \tex_fontcharht:D
\__kernel_primitive:NN \fontcharic             \tex_fontcharic:D
\__kernel_primitive:NN \fontcharwd             \tex_fontcharwd:D
\__kernel_primitive:NN \glueexpr               \tex_glueexpr:D
\__kernel_primitive:NN \glueshrink             \tex_glueshrink:D
\__kernel_primitive:NN \glueshrinkorder        \tex_glueshrinkorder:D
\__kernel_primitive:NN \gluestretch            \tex_gluestretch:D
\__kernel_primitive:NN \gluestretchorder       \tex_gluestretchorder:D
\__kernel_primitive:NN \gluetomu               \tex_gluetomu:D
\__kernel_primitive:NN \ifcsname               \tex_ifcsname:D
\__kernel_primitive:NN \ifdefined              \tex_ifdefined:D
\__kernel_primitive:NN \iffontchar             \tex_iffontchar:D
\__kernel_primitive:NN \interactionmode        \tex_interactionmode:D
\__kernel_primitive:NN \interlinepenalties     \tex_interlinepenalties:D
\__kernel_primitive:NN \lastlinefit            \tex_lastlinefit:D
\__kernel_primitive:NN \lastnodetype           \tex_lastnodetype:D
\__kernel_primitive:NN \marks                  \tex_marks:D
\__kernel_primitive:NN \middle                 \tex_middle:D
\__kernel_primitive:NN \muexpr                 \tex_muexpr:D
\__kernel_primitive:NN \mutoglue               \tex_mutoglue:D
\__kernel_primitive:NN \numexpr                \tex_numexpr:D
\__kernel_primitive:NN \pagediscards           \tex_pagediscards:D
```

```
\__kernel_primitive:NN \parshapedimen         \tex_parshapedimen:D
\__kernel_primitive:NN \parshapeindent        \tex_parshapeindent:D
\__kernel_primitive:NN \parshapelength        \tex_parshapelength:D
\__kernel_primitive:NN \predisplaydirection   \tex_predisplaydirection:D
\__kernel_primitive:NN \protected             \tex_protected:D
\__kernel_primitive:NN \readline              \tex_readline:D
\__kernel_primitive:NN \savinghyphcodes       \tex_savinghyphcodes:D
\__kernel_primitive:NN \savingvdiscards       \tex_savingvdiscards:D
\__kernel_primitive:NN \scantokens            \tex_scantokens:D
\__kernel_primitive:NN \showgroups            \tex_showgroups:D
\__kernel_primitive:NN \showifs               \tex_showifs:D
\__kernel_primitive:NN \showtokens            \tex_showtokens:D
\__kernel_primitive:NN \splitbotmarks         \tex_splitbotmarks:D
\__kernel_primitive:NN \splitdiscards         \tex_splitdiscards:D
\__kernel_primitive:NN \splitfirstmarks       \tex_splitfirstmarks:D
\__kernel_primitive:NN \TeXXeTstate           \tex_TeXXeTstate:D
\__kernel_primitive:NN \topmarks              \tex_topmarks:D
\__kernel_primitive:NN \tracingassigns        \tex_tracingassigns:D
\__kernel_primitive:NN \tracinggroups         \tex_tracinggroups:D
\__kernel_primitive:NN \tracingifs            \tex_tracingifs:D
\__kernel_primitive:NN \tracingnesting        \tex_tracingnesting:D
\__kernel_primitive:NN \tracingscantokens     \tex_tracingscantokens:D
\__kernel_primitive:NN \unexpanded            \tex_unexpanded:D
\__kernel_primitive:NN \unless                \tex_unless:D
\__kernel_primitive:NN \widowpenalties        \tex_widowpenalties:D
\__kernel_primitive:NN \pdfannot              \tex_pdfannot:D
\__kernel_primitive:NN \pdfcatalog            \tex_pdfcatalog:D
\__kernel_primitive:NN \pdfcompresslevel      \tex_pdfcompresslevel:D
\__kernel_primitive:NN \pdfcolorstack         \tex_pdfcolorstack:D
\__kernel_primitive:NN \pdfcolorstackinit     \tex_pdfcolorstackinit:D
\__kernel_primitive:NN \pdfdecimaldigits      \tex_pdfdecimaldigits:D
\__kernel_primitive:NN \pdfdest               \tex_pdfdest:D
\__kernel_primitive:NN \pdfdestmargin         \tex_pdfdestmargin:D
\__kernel_primitive:NN \pdfendlink            \tex_pdfendlink:D
\__kernel_primitive:NN \pdfendthread          \tex_pdfendthread:D
\__kernel_primitive:NN \pdffakespace          \tex_pdffakespace:D
\__kernel_primitive:NN \pdffontattr           \tex_pdffontattr:D
\__kernel_primitive:NN \pdffontname           \tex_pdffontname:D
\__kernel_primitive:NN \pdffontobjnum         \tex_pdffontobjnum:D
\__kernel_primitive:NN \pdfgamma              \tex_pdfgamma:D
\__kernel_primitive:NN \pdfgentounicode       \tex_pdfgentounicode:D
\__kernel_primitive:NN \pdfglyphtounicode     \tex_pdfglyphtounicode:D
\__kernel_primitive:NN \pdfhorigin            \tex_pdfhorigin:D
\__kernel_primitive:NN \pdfimageapplygamma    \tex_pdfimageapplygamma:D
\__kernel_primitive:NN \pdfimagegamma         \tex_pdfimagegamma:D
\__kernel_primitive:NN \pdfimagehicolor       \tex_pdfimagehicolor:D
```

```
\__kernel_primitive:NN \pdfimageresolution    \tex_pdfimageresolution:D
\__kernel_primitive:NN \pdfincludechars       \tex_pdfincludechars:D
\__kernel_primitive:NN \pdfinclusioncopyfonts \tex_pdfinclusioncopyfonts:D
\__kernel_primitive:NN \pdfinclusionerrorlevel
  \tex_pdfinclusionerrorlevel:D
\__kernel_primitive:NN \pdfinfo               \tex_pdfinfo:D
\__kernel_primitive:NN \pdfinfoomitdate       \tex_pdfinfoomitdate:D
\__kernel_primitive:NN \pdfinterwordspaceoff  \tex_pdfinterwordspaceoff:D
\__kernel_primitive:NN \pdfinterwordspaceon   \tex_pdfinterwordspaceon:D
\__kernel_primitive:NN \pdflastannot          \tex_pdflastannot:D
\__kernel_primitive:NN \pdflastlink           \tex_pdflastlink:D
\__kernel_primitive:NN \pdflastobj            \tex_pdflastobj:D
\__kernel_primitive:NN \pdflastxform          \tex_pdflastxform:D
\__kernel_primitive:NN \pdflastximage         \tex_pdflastximage:D
\__kernel_primitive:NN \pdflastximagecolordepth
  \tex_pdflastximagecolordepth:D
\__kernel_primitive:NN \pdflastximagepages    \tex_pdflastximagepages:D
\__kernel_primitive:NN \pdflinkmargin         \tex_pdflinkmargin:D
\__kernel_primitive:NN \pdfliteral            \tex_pdfliteral:D
\__kernel_primitive:NN \pdfmapfile            \tex_pdfmapfile:D
\__kernel_primitive:NN \pdfmapline            \tex_pdfmapline:D
\__kernel_primitive:NN \pdfmajorversion       \tex_pdfmajorversion:D
\__kernel_primitive:NN \pdfminorversion       \tex_pdfminorversion:D
\__kernel_primitive:NN \pdfnames              \tex_pdfnames:D
\__kernel_primitive:NN \pdfnobuiltintounicode \tex_pdfnobuiltintounicode:D
\__kernel_primitive:NN \pdfobj                \tex_pdfobj:D
\__kernel_primitive:NN \pdfobjcompresslevel   \tex_pdfobjcompresslevel:D
\__kernel_primitive:NN \pdfomitcharset        \tex_pdfomitcharset:D
\__kernel_primitive:NN \pdfoutline            \tex_pdfoutline:D
\__kernel_primitive:NN \pdfoutput             \tex_pdfoutput:D
\__kernel_primitive:NN \pdfpageattr           \tex_pdfpageattr:D
\__kernel_primitive:NN \pdfpagebox            \tex_pdfpagebox:D
\__kernel_primitive:NN \pdfpageref            \tex_pdfpageref:D
\__kernel_primitive:NN \pdfpageresources      \tex_pdfpageresources:D
\__kernel_primitive:NN \pdfpagesattr          \tex_pdfpagesattr:D
\__kernel_primitive:NN \pdfptexuseunderscore  \tex_pdfptexuseunderscore:D
\__kernel_primitive:NN \pdfrefobj             \tex_pdfrefobj:D
\__kernel_primitive:NN \pdfrefxform           \tex_pdfrefxform:D
\__kernel_primitive:NN \pdfrefximage          \tex_pdfrefximage:D
\__kernel_primitive:NN \pdfrestore            \tex_pdfrestore:D
\__kernel_primitive:NN \pdfretval             \tex_pdfretval:D
\__kernel_primitive:NN \pdfrunninglinkoff     \tex_pdfrunninglinkoff:D
\__kernel_primitive:NN \pdfrunninglinkon      \tex_pdfrunninglinkon:D
\__kernel_primitive:NN \pdfsave               \tex_pdfsave:D
\__kernel_primitive:NN \pdfsetmatrix          \tex_pdfsetmatrix:D
\__kernel_primitive:NN \pdfstartlink          \tex_pdfstartlink:D
```

```
\__kernel_primitive:NN \pdfstartthread        \tex_pdfstartthread:D
\__kernel_primitive:NN \pdfsuppressptexinfo   \tex_pdfsuppressptexinfo:D
\__kernel_primitive:NN \pdfsuppresswarningdupdest
  \tex_pdfsuppresswarningdupdest:D
\__kernel_primitive:NN \pdfsuppresswarningdupmap
  \tex_pdfsuppresswarningdupmap:D
\__kernel_primitive:NN \pdfsuppresswarningpagegroup
  \tex_pdfsuppresswarningpagegroup:D
\__kernel_primitive:NN \pdfthread             \tex_pdfthread:D
\__kernel_primitive:NN \pdfthreadmargin       \tex_pdfthreadmargin:D
\__kernel_primitive:NN \pdftrailer            \tex_pdftrailer:D
\__kernel_primitive:NN \pdftrailerid          \tex_pdftrailerid:D
\__kernel_primitive:NN \pdfuniqueresname      \tex_pdfuniqueresname:D
\__kernel_primitive:NN \pdfvorigin            \tex_pdfvorigin:D
\__kernel_primitive:NN \pdfxform              \tex_pdfxform:D
\__kernel_primitive:NN \pdfxformname          \tex_pdfxformname:D
\__kernel_primitive:NN \pdfximage             \tex_pdfximage:D
\__kernel_primitive:NN \pdfximagebbox         \tex_pdfximagebbox:D
\__kernel_primitive:NN \ifpdfabsdim           \tex_ifabsdim:D
\__kernel_primitive:NN \ifpdfabsnum           \tex_ifabsnum:D
\__kernel_primitive:NN \ifpdfprimitive        \tex_ifprimitive:D
\__kernel_primitive:NN \pdfadjustinterwordglue
  \tex_adjustinterwordglue:D
\__kernel_primitive:NN \pdfadjustspacing      \tex_adjustspacing:D
\__kernel_primitive:NN \pdfappendkern         \tex_appendkern:D
\__kernel_primitive:NN \pdfcopyfont           \tex_copyfont:D
\__kernel_primitive:NN \pdfcreationdate       \tex_creationdate:D
\__kernel_primitive:NN \pdfdraftmode          \tex_draftmode:D
\__kernel_primitive:NN \pdfeachlinedepth      \tex_eachlinedepth:D
\__kernel_primitive:NN \pdfeachlineheight     \tex_eachlineheight:D
\__kernel_primitive:NN \pdfelapsedtime        \tex_elapsedtime:D
\__kernel_primitive:NN \pdfescapehex          \tex_escapehex:D
\__kernel_primitive:NN \pdfescapename         \tex_escapename:D
\__kernel_primitive:NN \pdfescapestring       \tex_escapestring:D
\__kernel_primitive:NN \pdffirstlineheight    \tex_firstlineheight:D
\__kernel_primitive:NN \pdffontexpand         \tex_fontexpand:D
\__kernel_primitive:NN \pdffontsize           \tex_fontsize:D
\__kernel_primitive:NN \pdfignoreddimen       \tex_ignoreddimen:D
\__kernel_primitive:NN \pdfinsertht           \tex_insertht:D
\__kernel_primitive:NN \pdflastlinedepth      \tex_lastlinedepth:D
\__kernel_primitive:NN \pdflastmatch          \tex_lastmatch:D
\__kernel_primitive:NN \pdflastxpos           \tex_lastxpos:D
\__kernel_primitive:NN \pdflastypos           \tex_lastypos:D
\__kernel_primitive:NN \pdfmatch              \tex_match:D
\__kernel_primitive:NN \pdfnoligatures        \tex_noligatures:D
\__kernel_primitive:NN \pdfnormaldeviate      \tex_normaldeviate:D
```

```
\__kernel_primitive:NN \pdfpageheight         \tex_pageheight:D
\__kernel_primitive:NN \pdfpagewidth          \tex_pagewidth:D
\__kernel_primitive:NN \pdfpkmode             \tex_pkmode:D
\__kernel_primitive:NN \pdfpkresolution       \tex_pkresolution:D
\__kernel_primitive:NN \pdfprimitive          \tex_primitive:D
\__kernel_primitive:NN \pdfprependkern        \tex_prependkern:D
\__kernel_primitive:NN \pdfprotrudechars      \tex_protrudechars:D
\__kernel_primitive:NN \pdfpxdimen            \tex_pxdimen:D
\__kernel_primitive:NN \pdfrandomseed         \tex_randomseed:D
\__kernel_primitive:NN \pdfresettimer         \tex_resettimer:D
\__kernel_primitive:NN \pdfsavepos            \tex_savepos:D
\__kernel_primitive:NN \pdfsetrandomseed      \tex_setrandomseed:D
\__kernel_primitive:NN \pdfshellescape        \tex_shellescape:D
\__kernel_primitive:NN \pdftracingfonts       \tex_tracingfonts:D
\__kernel_primitive:NN \pdfunescapehex        \tex_unescapehex:D
\__kernel_primitive:NN \pdfuniformdeviate     \tex_uniformdeviate:D
\__kernel_primitive:NN \pdftexbanner          \tex_pdftexbanner:D
\__kernel_primitive:NN \pdftexrevision        \tex_pdftexrevision:D
\__kernel_primitive:NN \pdftexversion         \tex_pdftexversion:D
\__kernel_primitive:NN \efcode                \tex_efcode:D
\__kernel_primitive:NN \ifincsname            \tex_ifincsname:D
\__kernel_primitive:NN \knaccode              \tex_knaccode:D
\__kernel_primitive:NN \knbccode              \tex_knbccode:D
\__kernel_primitive:NN \knbscode              \tex_knbscode:D
\__kernel_primitive:NN \leftmarginkern        \tex_leftmarginkern:D
\__kernel_primitive:NN \letterspacefont       \tex_letterspacefont:D
\__kernel_primitive:NN \lpcode                \tex_lpcode:D
\__kernel_primitive:NN \quitvmode             \tex_quitvmode:D
\__kernel_primitive:NN \rightmarginkern       \tex_rightmarginkern:D
\__kernel_primitive:NN \rpcode                \tex_rpcode:D
\__kernel_primitive:NN \shbscode              \tex_shbscode:D
\__kernel_primitive:NN \stbscode              \tex_stbscode:D
\__kernel_primitive:NN \synctex               \tex_synctex:D
\__kernel_primitive:NN \tagcode               \tex_tagcode:D
\tex_long:D \tex_def:D \use_ii:nn #1#2 {#2}
\tex_long:D \tex_def:D \use_none:n #1 { }
\tex_long:D \tex_def:D \__kernel_primitive:NN #1#2
  {
    \tex_ifdefined:D #1
      \tex_expandafter:D \use_ii:nn
    \tex_fi:D
      \use_none:n { \tex_global:D \tex_let:D #2 #1 }
  }
\__kernel_primitive:NN \pdfstrcmp             \tex_strcmp:D
\__kernel_primitive:NN \pdffilesize           \tex_filesize:D
\__kernel_primitive:NN \pdfmdfivesum          \tex_mdfivesum:D
```

```
\__kernel_primitive:NN \pdffilemoddate        \tex_filemoddate:D
\__kernel_primitive:NN \pdffiledump           \tex_filedump:D
\__kernel_primitive:NN \suppressfontnotfounderror
  \tex_suppressfontnotfounderror:D
\__kernel_primitive:NN \XeTeXcharclass        \tex_XeTeXcharclass:D
\__kernel_primitive:NN \XeTeXcharglyph        \tex_XeTeXcharglyph:D
\__kernel_primitive:NN \XeTeXcountfeatures    \tex_XeTeXcountfeatures:D
\__kernel_primitive:NN \XeTeXcountglyphs      \tex_XeTeXcountglyphs:D
\__kernel_primitive:NN \XeTeXcountselectors   \tex_XeTeXcountselectors:D
\__kernel_primitive:NN \XeTeXcountvariations  \tex_XeTeXcountvariations:D
\__kernel_primitive:NN \XeTeXdefaultencoding  \tex_XeTeXdefaultencoding:D
\__kernel_primitive:NN \XeTeXdashbreakstate   \tex_XeTeXdashbreakstate:D
\__kernel_primitive:NN \XeTeXfeaturecode      \tex_XeTeXfeaturecode:D
\__kernel_primitive:NN \XeTeXfeaturename      \tex_XeTeXfeaturename:D
\__kernel_primitive:NN \XeTeXfindfeaturebyname
  \tex_XeTeXfindfeaturebyname:D
\__kernel_primitive:NN \XeTeXfindselectorbyname
  \tex_XeTeXfindselectorbyname:D
\__kernel_primitive:NN \XeTeXfindvariationbyname
  \tex_XeTeXfindvariationbyname:D
\__kernel_primitive:NN \XeTeXfirstfontchar    \tex_XeTeXfirstfontchar:D
\__kernel_primitive:NN \XeTeXfonttype         \tex_XeTeXfonttype:D
\__kernel_primitive:NN \XeTeXgenerateactualtext
  \tex_XeTeXgenerateactualtext:D
\__kernel_primitive:NN \XeTeXglyph            \tex_XeTeXglyph:D
\__kernel_primitive:NN \XeTeXglyphbounds      \tex_XeTeXglyphbounds:D
\__kernel_primitive:NN \XeTeXglyphindex       \tex_XeTeXglyphindex:D
\__kernel_primitive:NN \XeTeXglyphname        \tex_XeTeXglyphname:D
\__kernel_primitive:NN \XeTeXinputencoding    \tex_XeTeXinputencoding:D
\__kernel_primitive:NN \XeTeXinputnormalization
  \tex_XeTeXinputnormalization:D
\__kernel_primitive:NN \XeTeXinterchartokenstate
  \tex_XeTeXinterchartokenstate:D
\__kernel_primitive:NN \XeTeXinterchartoks    \tex_XeTeXinterchartoks:D
\__kernel_primitive:NN \XeTeXisdefaultselector
  \tex_XeTeXisdefaultselector:D
\__kernel_primitive:NN \XeTeXisexclusivefeature
  \tex_XeTeXisexclusivefeature:D
\__kernel_primitive:NN \XeTeXlastfontchar     \tex_XeTeXlastfontchar:D
\__kernel_primitive:NN \XeTeXlinebreakskip    \tex_XeTeXlinebreakskip:D
\__kernel_primitive:NN \XeTeXlinebreaklocale  \tex_XeTeXlinebreaklocale:D
\__kernel_primitive:NN \XeTeXlinebreakpenalty \tex_XeTeXlinebreakpenalty:D
\__kernel_primitive:NN \XeTeXOTcountfeatures  \tex_XeTeXOTcountfeatures:D
\__kernel_primitive:NN \XeTeXOTcountlanguages \tex_XeTeXOTcountlanguages:D
\__kernel_primitive:NN \XeTeXOTcountscripts   \tex_XeTeXOTcountscripts:D
\__kernel_primitive:NN \XeTeXOTfeaturetag     \tex_XeTeXOTfeaturetag:D
```

```
  \__kernel_primitive:NN \XeTeXOTlanguagetag    \tex_XeTeXOTlanguagetag:D
  \__kernel_primitive:NN \XeTeXOTscripttag      \tex_XeTeXOTscripttag:D
  \__kernel_primitive:NN \XeTeXpdffile          \tex_XeTeXpdffile:D
  \__kernel_primitive:NN \XeTeXpdfpagecount     \tex_XeTeXpdfpagecount:D
  \__kernel_primitive:NN \XeTeXpicfile          \tex_XeTeXpicfile:D
  \__kernel_primitive:NN \XeTeXrevision         \tex_XeTeXrevision:D
  \__kernel_primitive:NN \XeTeXselectorname     \tex_XeTeXselectorname:D
  \__kernel_primitive:NN \XeTeXtracingfonts     \tex_XeTeXtracingfonts:D
  \__kernel_primitive:NN \XeTeXupwardsmode      \tex_XeTeXupwardsmode:D
  \__kernel_primitive:NN \XeTeXuseglyphmetrics  \tex_XeTeXuseglyphmetrics:D
  \__kernel_primitive:NN \XeTeXvariation        \tex_XeTeXvariation:D
  \__kernel_primitive:NN \XeTeXvariationdefault \tex_XeTeXvariationdefault:D
  \__kernel_primitive:NN \XeTeXvariationmax     \tex_XeTeXvariationmax:D
  \__kernel_primitive:NN \XeTeXvariationmin     \tex_XeTeXvariationmin:D
  \__kernel_primitive:NN \XeTeXvariationname    \tex_XeTeXvariationname:D
  \__kernel_primitive:NN \XeTeXversion          \tex_XeTeXversion:D
  \__kernel_primitive:NN \XeTeXselectorcode     \tex_XeTeXselectorcode:D
  \__kernel_primitive:NN \XeTeXinterwordspaceshaping
                   \tex_XeTeXinterwordspaceshaping:D
  \__kernel_primitive:NN \XeTeXhyphenatablelength
                   \tex_XeTeXhyphenatablelength:D
  \__kernel_primitive:NN \creationdate          \tex_creationdate:D
  \__kernel_primitive:NN \elapsedtime           \tex_elapsedtime:D
  \__kernel_primitive:NN \filedump              \tex_filedump:D
  \__kernel_primitive:NN \filemoddate           \tex_filemoddate:D
  \__kernel_primitive:NN \filesize              \tex_filesize:D
  \__kernel_primitive:NN \mdfivesum             \tex_mdfivesum:D
  \__kernel_primitive:NN \ifprimitive           \tex_ifprimitive:D
  \__kernel_primitive:NN \primitive             \tex_primitive:D
  \__kernel_primitive:NN \resettimer            \tex_resettimer:D
  \__kernel_primitive:NN \shellescape           \tex_shellescape:D
  \__kernel_primitive:NN \XeTeXprotrudechars    \tex_protrudechars:D
  \__kernel_primitive:NN \alignmark             \tex_alignmark:D
  \__kernel_primitive:NN \aligntab              \tex_aligntab:D
  \__kernel_primitive:NN \attribute             \tex_attribute:D
  \__kernel_primitive:NN \attributedef          \tex_attributedef:D
  \__kernel_primitive:NN \automaticdiscretionary
    \tex_automaticdiscretionary:D
  \__kernel_primitive:NN \automatichyphenmode   \tex_automatichyphenmode:D
  \__kernel_primitive:NN \automatichyphenpenalty
    \tex_automatichyphenpenalty:D
  \__kernel_primitive:NN \begincsname           \tex_begincsname:D
  \__kernel_primitive:NN \bodydir               \tex_bodydir:D
  \__kernel_primitive:NN \bodydirection         \tex_bodydirection:D
  \__kernel_primitive:NN \boundary              \tex_boundary:D
  \__kernel_primitive:NN \boxdir                \tex_boxdir:D
```

```
\__kernel_primitive:NN \boxdirection          \tex_boxdirection:D
\__kernel_primitive:NN \breakafterdirmode     \tex_breakafterdirmode:D
\__kernel_primitive:NN \catcodetable          \tex_catcodetable:D
\__kernel_primitive:NN \clearmarks            \tex_clearmarks:D
\__kernel_primitive:NN \crampeddisplaystyle   \tex_crampeddisplaystyle:D
\__kernel_primitive:NN \crampedscriptscriptstyle
  \tex_crampedscriptscriptstyle:D
\__kernel_primitive:NN \crampedscriptstyle    \tex_crampedscriptstyle:D
\__kernel_primitive:NN \crampedtextstyle      \tex_crampedtextstyle:D
\__kernel_primitive:NN \csstring              \tex_csstring:D
\__kernel_primitive:NN \deferred              \tex_deferred:D
\__kernel_primitive:NN \discretionaryligaturemode
  \tex_discretionaryligaturemode:D
\__kernel_primitive:NN \directlua             \tex_directlua:D
\__kernel_primitive:NN \dviextension          \tex_dviextension:D
\__kernel_primitive:NN \dvifeedback           \tex_dvifeedback:D
\__kernel_primitive:NN \dvivariable           \tex_dvivariable:D
\__kernel_primitive:NN \eTeXglueshrinkorder   \tex_eTeXglueshrinkorder:D
\__kernel_primitive:NN \eTeXgluestretchorder  \tex_eTeXgluestretchorder:D
\__kernel_primitive:NN \endlocalcontrol       \tex_endlocalcontrol:D
\__kernel_primitive:NN \etoksapp              \tex_etoksapp:D
\__kernel_primitive:NN \etokspre              \tex_etokspre:D
\__kernel_primitive:NN \exceptionpenalty      \tex_exceptionpenalty:D
\__kernel_primitive:NN \exhyphenchar          \tex_exhyphenchar:D
\__kernel_primitive:NN \explicithyphenpenalty \tex_explicithyphenpenalty:D
\__kernel_primitive:NN \expanded              \tex_expanded:D
\__kernel_primitive:NN \explicitdiscretionary \tex_explicitdiscretionary:D
\__kernel_primitive:NN \firstvalidlanguage    \tex_firstvalidlanguage:D
\__kernel_primitive:NN \fontid                \tex_fontid:D
\__kernel_primitive:NN \formatname            \tex_formatname:D
\__kernel_primitive:NN \hjcode                \tex_hjcode:D
\__kernel_primitive:NN \hpack                 \tex_hpack:D
\__kernel_primitive:NN \hyphenationbounds     \tex_hyphenationbounds:D
\__kernel_primitive:NN \hyphenationmin        \tex_hyphenationmin:D
\__kernel_primitive:NN \hyphenpenaltymode     \tex_hyphenpenaltymode:D
\__kernel_primitive:NN \gleaders              \tex_gleaders:D
\__kernel_primitive:NN \glet                  \tex_glet:D
\__kernel_primitive:NN \glyphdimensionsmode   \tex_glyphdimensionsmode:D
\__kernel_primitive:NN \gtoksapp              \tex_gtoksapp:D
\__kernel_primitive:NN \gtokspre              \tex_gtokspre:D
\__kernel_primitive:NN \ifcondition           \tex_ifcondition:D
\__kernel_primitive:NN \immediateassigned     \tex_immediateassigned:D
\__kernel_primitive:NN \immediateassignment   \tex_immediateassignment:D
\__kernel_primitive:NN \initcatcodetable      \tex_initcatcodetable:D
\__kernel_primitive:NN \lastnamedcs           \tex_lastnamedcs:D
\__kernel_primitive:NN \latelua               \tex_latelua:D
```

```
\__kernel_primitive:NN \lateluafunction       \tex_lateluafunction:D
\__kernel_primitive:NN \leftghost             \tex_leftghost:D
\__kernel_primitive:NN \letcharcode           \tex_letcharcode:D
\__kernel_primitive:NN \linedir               \tex_linedir:D
\__kernel_primitive:NN \linedirection         \tex_linedirection:D
\__kernel_primitive:NN \localbrokenpenalty    \tex_localbrokenpenalty:D
\__kernel_primitive:NN \localinterlinepenalty \tex_localinterlinepenalty:D
\__kernel_primitive:NN \luabytecode           \tex_luabytecode:D
\__kernel_primitive:NN \luabytecodecall       \tex_luabytecodecall:D
\__kernel_primitive:NN \luacopyinputnodes     \tex_luacopyinputnodes:D
\__kernel_primitive:NN \luadef                \tex_luadef:D
\__kernel_primitive:NN \localleftbox          \tex_localleftbox:D
\__kernel_primitive:NN \localrightbox         \tex_localrightbox:D
\__kernel_primitive:NN \luaescapestring       \tex_luaescapestring:D
\__kernel_primitive:NN \luafunction           \tex_luafunction:D
\__kernel_primitive:NN \luafunctioncall       \tex_luafunctioncall:D
\__kernel_primitive:NN \luatexbanner          \tex_luatexbanner:D
\__kernel_primitive:NN \luatexrevision        \tex_luatexrevision:D
\__kernel_primitive:NN \luatexversion         \tex_luatexversion:D
\__kernel_primitive:NN \mathdefaultsmode      \tex_mathdefaultsmode:D
\__kernel_primitive:NN \mathdelimitersmode    \tex_mathdelimitersmode:D
\__kernel_primitive:NN \mathdir               \tex_mathdir:D
\__kernel_primitive:NN \mathdirection         \tex_mathdirection:D
\__kernel_primitive:NN \mathdisplayskipmode   \tex_mathdisplayskipmode:D
\__kernel_primitive:NN \matheqdirmode         \tex_matheqdirmode:D
\__kernel_primitive:NN \matheqnogapstep       \tex_matheqnogapstep:D
\__kernel_primitive:NN \mathemptydisplaymode  \tex_mathemptydisplaymode:D
\__kernel_primitive:NN \mathflattenmode       \tex_mathflattenmode:D
\__kernel_primitive:NN \mathitalicsmode       \tex_mathitalicsmode:D
\__kernel_primitive:NN \mathnolimitsmode      \tex_mathnolimitsmode:D
\__kernel_primitive:NN \mathoption            \tex_mathoption:D
\__kernel_primitive:NN \mathpenaltiesmode     \tex_mathpenaltiesmode:D
\__kernel_primitive:NN \mathrulesfam          \tex_mathrulesfam:D
\__kernel_primitive:NN \mathscriptsmode       \tex_mathscriptsmode:D
\__kernel_primitive:NN \mathscriptboxmode     \tex_mathscriptboxmode:D
\__kernel_primitive:NN \mathscriptcharmode    \tex_mathscriptcharmode:D
\__kernel_primitive:NN \mathstyle             \tex_mathstyle:D
\__kernel_primitive:NN \mathsurroundmode      \tex_mathsurroundmode:D
\__kernel_primitive:NN \mathsurroundskip      \tex_mathsurroundskip:D
\__kernel_primitive:NN \nohrule               \tex_nohrule:D
\__kernel_primitive:NN \nokerns               \tex_nokerns:D
\__kernel_primitive:NN \noligs                \tex_noligs:D
\__kernel_primitive:NN \nospaces              \tex_nospaces:D
\__kernel_primitive:NN \novrule               \tex_novrule:D
\__kernel_primitive:NN \outputbox             \tex_outputbox:D
\__kernel_primitive:NN \pagebottomoffset      \tex_pagebottomoffset:D
```

```
\__kernel_primitive:NN \pagedir               \tex_pagedir:D
\__kernel_primitive:NN \pagedirection         \tex_pagedirection:D
\__kernel_primitive:NN \pageleftoffset        \tex_pageleftoffset:D
\__kernel_primitive:NN \pagerightoffset       \tex_pagerightoffset:D
\__kernel_primitive:NN \pagetopoffset         \tex_pagetopoffset:D
\__kernel_primitive:NN \pardir                \tex_pardir:D
\__kernel_primitive:NN \pardirection          \tex_pardirection:D
\__kernel_primitive:NN \pdfextension          \tex_pdfextension:D
\__kernel_primitive:NN \pdffeedback           \tex_pdffeedback:D
\__kernel_primitive:NN \pdfvariable           \tex_pdfvariable:D
\__kernel_primitive:NN \postexhyphenchar      \tex_postexhyphenchar:D
\__kernel_primitive:NN \posthyphenchar        \tex_posthyphenchar:D
\__kernel_primitive:NN \prebinoppenalty       \tex_prebinoppenalty:D
\__kernel_primitive:NN \predisplaygapfactor   \tex_predisplaygapfactor:D
\__kernel_primitive:NN \preexhyphenchar       \tex_preexhyphenchar:D
\__kernel_primitive:NN \prehyphenchar         \tex_prehyphenchar:D
\__kernel_primitive:NN \prerelpenalty         \tex_prerelpenalty:D
\__kernel_primitive:NN \protrusionboundary    \tex_protrusionboundary:D
\__kernel_primitive:NN \rightghost            \tex_rightghost:D
\__kernel_primitive:NN \savecatcodetable      \tex_savecatcodetable:D
\__kernel_primitive:NN \scantextokens         \tex_scantextokens:D
\__kernel_primitive:NN \setfontid             \tex_setfontid:D
\__kernel_primitive:NN \shapemode             \tex_shapemode:D
\__kernel_primitive:NN \suppressifcsnameerror \tex_suppressifcsnameerror:D
\__kernel_primitive:NN \suppresslongerror     \tex_suppresslongerror:D
\__kernel_primitive:NN \suppressmathparerror  \tex_suppressmathparerror:D
\__kernel_primitive:NN \suppressoutererror    \tex_suppressoutererror:D
\__kernel_primitive:NN \suppressprimitiveerror
  \tex_suppressprimitiveerror:D
\__kernel_primitive:NN \textdir               \tex_textdir:D
\__kernel_primitive:NN \textdirection         \tex_textdirection:D
\__kernel_primitive:NN \toksapp               \tex_toksapp:D
\__kernel_primitive:NN \tokspre               \tex_tokspre:D
\__kernel_primitive:NN \tpack                 \tex_tpack:D
\__kernel_primitive:NN \variablefam           \tex_variablefam:D
\__kernel_primitive:NN \vpack                 \tex_vpack:D
\__kernel_primitive:NN \wordboundary          \tex_wordboundary:D
\__kernel_primitive:NN \xtoksapp              \tex_xtoksapp:D
\__kernel_primitive:NN \xtokspre              \tex_xtokspre:D
\__kernel_primitive:NN \adjustspacing         \tex_adjustspacing:D
\__kernel_primitive:NN \copyfont              \tex_copyfont:D
\__kernel_primitive:NN \draftmode             \tex_draftmode:D
\__kernel_primitive:NN \expandglyphsinfont    \tex_fontexpand:D
\__kernel_primitive:NN \ifabsdim              \tex_ifabsdim:D
\__kernel_primitive:NN \ifabsnum              \tex_ifabsnum:D
\__kernel_primitive:NN \ignoreligaturesinfont \tex_ignoreligaturesinfont:D
```

```
  \__kernel_primitive:NN \insertht              \tex_insertht:D
  \__kernel_primitive:NN \lastsavedboxresourceindex
    \tex_pdflastxform:D
  \__kernel_primitive:NN \lastsavedimageresourceindex
    \tex_pdflastximage:D
  \__kernel_primitive:NN \lastsavedimageresourcepages
    \tex_pdflastximagepages:D
  \__kernel_primitive:NN \lastxpos              \tex_lastxpos:D
  \__kernel_primitive:NN \lastypos              \tex_lastypos:D
  \__kernel_primitive:NN \normaldeviate         \tex_normaldeviate:D
  \__kernel_primitive:NN \outputmode            \tex_pdfoutput:D
  \__kernel_primitive:NN \pageheight            \tex_pageheight:D
  \__kernel_primitive:NN \pagewidth             \tex_pagewidth:D
  \__kernel_primitive:NN \protrudechars         \tex_protrudechars:D
  \__kernel_primitive:NN \pxdimen               \tex_pxdimen:D
  \__kernel_primitive:NN \randomseed            \tex_randomseed:D
  \__kernel_primitive:NN \useboxresource        \tex_pdfrefxform:D
  \__kernel_primitive:NN \useimageresource      \tex_pdfrefximage:D
  \__kernel_primitive:NN \savepos               \tex_savepos:D
  \__kernel_primitive:NN \saveboxresource       \tex_pdfxform:D
  \__kernel_primitive:NN \saveimageresource     \tex_pdfximage:D
  \__kernel_primitive:NN \setrandomseed         \tex_setrandomseed:D
  \__kernel_primitive:NN \tracingfonts          \tex_tracingfonts:D
  \__kernel_primitive:NN \uniformdeviate        \tex_uniformdeviate:D
  \__kernel_primitive:NN \Uchar                 \tex_Uchar:D
  \__kernel_primitive:NN \Ucharcat              \tex_Ucharcat:D
  \__kernel_primitive:NN \Udelcode              \tex_Udelcode:D
  \__kernel_primitive:NN \Udelcodenum           \tex_Udelcodenum:D
  \__kernel_primitive:NN \Udelimiter            \tex_Udelimiter:D
  \__kernel_primitive:NN \Udelimiterover        \tex_Udelimiterover:D
  \__kernel_primitive:NN \Udelimiterunder       \tex_Udelimiterunder:D
  \__kernel_primitive:NN \Uhextensible          \tex_Uhextensible:D
  \__kernel_primitive:NN \Uleft                 \tex_Uleft:D
  \__kernel_primitive:NN \Umathaccent           \tex_Umathaccent:D
  \__kernel_primitive:NN \Umathaxis             \tex_Umathaxis:D
  \__kernel_primitive:NN \Umathbinbinspacing    \tex_Umathbinbinspacing:D
  \__kernel_primitive:NN \Umathbinclosespacing  \tex_Umathbinclosespacing:D
  \__kernel_primitive:NN \Umathbininnerspacing  \tex_Umathbininnerspacing:D
  \__kernel_primitive:NN \Umathbinopenspacing   \tex_Umathbinopenspacing:D
  \__kernel_primitive:NN \Umathbinopspacing     \tex_Umathbinopspacing:D
  \__kernel_primitive:NN \Umathbinordspacing    \tex_Umathbinordspacing:D
  \__kernel_primitive:NN \Umathbinpunctspacing  \tex_Umathbinpunctspacing:D
  \__kernel_primitive:NN \Umathbinrelspacing    \tex_Umathbinrelspacing:D
  \__kernel_primitive:NN \Umathchar             \tex_Umathchar:D
  \__kernel_primitive:NN \Umathcharclass        \tex_Umathcharclass:D
  \__kernel_primitive:NN \Umathchardef          \tex_Umathchardef:D
  \__kernel_primitive:NN \Umathcharfam          \tex_Umathcharfam:D
  \__kernel_primitive:NN \Umathcharnum          \tex_Umathcharnum:D
  \__kernel_primitive:NN \Umathcharnumdef       \tex_Umathcharnumdef:D
  \__kernel_primitive:NN \Umathcharslot         \tex_Umathcharslot:D
  \__kernel_primitive:NN \Umathclosebinspacing  \tex_Umathclosebinspacing:D
  \__kernel_primitive:NN \Umathcloseclosespacing
    \tex_Umathcloseclosespacing:D
  \__kernel_primitive:NN \Umathcloseinnerspacing
    \tex_Umathcloseinnerspacing:D
  \__kernel_primitive:NN \Umathcloseopenspacing \tex_Umathcloseopenspacing:D
  \__kernel_primitive:NN \Umathcloseopspacing   \tex_Umathcloseopspacing:D
  \__kernel_primitive:NN \Umathcloseordspacing  \tex_Umathcloseordspacing:D
  \__kernel_primitive:NN \Umathclosepunctspacing
    \tex_Umathclosepunctspacing:D
  \__kernel_primitive:NN \Umathcloserelspacing  \tex_Umathcloserelspacing:D
  \__kernel_primitive:NN \Umathcode             \tex_Umathcode:D
  \__kernel_primitive:NN \Umathcodenum          \tex_Umathcodenum:D
  \__kernel_primitive:NN \Umathconnectoroverlapmin
    \tex_Umathconnectoroverlapmin:D
  \__kernel_primitive:NN \Umathfractiondelsize  \tex_Umathfractiondelsize:D
  \__kernel_primitive:NN \Umathfractiondenomdown
    \tex_Umathfractiondenomdown:D
  \__kernel_primitive:NN \Umathfractiondenomvgap
    \tex_Umathfractiondenomvgap:D
  \__kernel_primitive:NN \Umathfractionnumup    \tex_Umathfractionnumup:D
  \__kernel_primitive:NN \Umathfractionnumvgap  \tex_Umathfractionnumvgap:D
  \__kernel_primitive:NN \Umathfractionrule     \tex_Umathfractionrule:D
  \__kernel_primitive:NN \Umathinnerbinspacing  \tex_Umathinnerbinspacing:D
  \__kernel_primitive:NN \Umathinnerclosespacing
    \tex_Umathinnerclosespacing:D
  \__kernel_primitive:NN \Umathinnerinnerspacing
    \tex_Umathinnerinnerspacing:D
  \__kernel_primitive:NN \Umathinneropenspacing \tex_Umathinneropenspacing:D
  \__kernel_primitive:NN \Umathinneropspacing   \tex_Umathinneropspacing:D
  \__kernel_primitive:NN \Umathinnerordspacing  \tex_Umathinnerordspacing:D
  \__kernel_primitive:NN \Umathinnerpunctspacing
    \tex_Umathinnerpunctspacing:D
  \__kernel_primitive:NN \Umathinnerrelspacing  \tex_Umathinnerrelspacing:D
  \__kernel_primitive:NN \Umathlimitabovebgap   \tex_Umathlimitabovebgap:D
  \__kernel_primitive:NN \Umathlimitabovekern   \tex_Umathlimitabovekern:D
  \__kernel_primitive:NN \Umathlimitabovevgap   \tex_Umathlimitabovevgap:D
  \__kernel_primitive:NN \Umathlimitbelowbgap   \tex_Umathlimitbelowbgap:D
  \__kernel_primitive:NN \Umathlimitbelowkern   \tex_Umathlimitbelowkern:D
  \__kernel_primitive:NN \Umathlimitbelowvgap   \tex_Umathlimitbelowvgap:D
  \__kernel_primitive:NN \Umathnolimitsubfactor \tex_Umathnolimitsubfactor:D
  \__kernel_primitive:NN \Umathnolimitsupfactor \tex_Umathnolimitsupfactor:D
```

```
\__kernel_primitive:NN \Umathopbinspacing     \tex_Umathopbinspacing:D
\__kernel_primitive:NN \Umathopclosespacing   \tex_Umathopclosespacing:D
\__kernel_primitive:NN \Umathopenbinspacing   \tex_Umathopenbinspacing:D
\__kernel_primitive:NN \Umathopenclosespacing \tex_Umathopenclosespacing:D
\__kernel_primitive:NN \Umathopeninnerspacing \tex_Umathopeninnerspacing:D
\__kernel_primitive:NN \Umathopenopenspacing  \tex_Umathopenopenspacing:D
\__kernel_primitive:NN \Umathopenopspacing    \tex_Umathopenopspacing:D
\__kernel_primitive:NN \Umathopenordspacing   \tex_Umathopenordspacing:D
\__kernel_primitive:NN \Umathopenpunctspacing \tex_Umathopenpunctspacing:D
\__kernel_primitive:NN \Umathopenrelspacing   \tex_Umathopenrelspacing:D
\__kernel_primitive:NN \Umathoperatorsize     \tex_Umathoperatorsize:D
\__kernel_primitive:NN \Umathopinnerspacing   \tex_Umathopinnerspacing:D
\__kernel_primitive:NN \Umathopopenspacing    \tex_Umathopopenspacing:D
\__kernel_primitive:NN \Umathopopspacing      \tex_Umathopopspacing:D
\__kernel_primitive:NN \Umathopordspacing     \tex_Umathopordspacing:D
\__kernel_primitive:NN \Umathoppunctspacing   \tex_Umathoppunctspacing:D
\__kernel_primitive:NN \Umathoprelspacing     \tex_Umathoprelspacing:D
\__kernel_primitive:NN \Umathordbinspacing    \tex_Umathordbinspacing:D
\__kernel_primitive:NN \Umathordclosespacing  \tex_Umathordclosespacing:D
\__kernel_primitive:NN \Umathordinnerspacing  \tex_Umathordinnerspacing:D
\__kernel_primitive:NN \Umathordopenspacing   \tex_Umathordopenspacing:D
\__kernel_primitive:NN \Umathordopspacing     \tex_Umathordopspacing:D
\__kernel_primitive:NN \Umathordordspacing    \tex_Umathordordspacing:D
\__kernel_primitive:NN \Umathordpunctspacing  \tex_Umathordpunctspacing:D
\__kernel_primitive:NN \Umathordrelspacing    \tex_Umathordrelspacing:D
\__kernel_primitive:NN \Umathoverbarkern      \tex_Umathoverbarkern:D
\__kernel_primitive:NN \Umathoverbarrule      \tex_Umathoverbarrule:D
\__kernel_primitive:NN \Umathoverbarvgap      \tex_Umathoverbarvgap:D
\__kernel_primitive:NN \Umathoverdelimiterbgap
  \tex_Umathoverdelimiterbgap:D
\__kernel_primitive:NN \Umathoverdelimitervgap
  \tex_Umathoverdelimitervgap:D
\__kernel_primitive:NN \Umathpunctbinspacing  \tex_Umathpunctbinspacing:D
\__kernel_primitive:NN \Umathpunctclosespacing
  \tex_Umathpunctclosespacing:D
\__kernel_primitive:NN \Umathpunctinnerspacing
  \tex_Umathpunctinnerspacing:D
\__kernel_primitive:NN \Umathpunctopenspacing \tex_Umathpunctopenspacing:D
\__kernel_primitive:NN \Umathpunctopspacing   \tex_Umathpunctopspacing:D
\__kernel_primitive:NN \Umathpunctordspacing  \tex_Umathpunctordspacing:D
\__kernel_primitive:NN \Umathpunctpunctspacing
  \tex_Umathpunctpunctspacing:D
\__kernel_primitive:NN \Umathpunctrelspacing  \tex_Umathpunctrelspacing:D
\__kernel_primitive:NN \Umathquad             \tex_Umathquad:D
\__kernel_primitive:NN \Umathradicaldegreeafter
  \tex_Umathradicaldegreeafter:D
```

```
\__kernel_primitive:NN \Umathradicaldegreebefore
  \tex_Umathradicaldegreebefore:D
\__kernel_primitive:NN \Umathradicaldegreeraise
  \tex_Umathradicaldegreeraise:D
\__kernel_primitive:NN \Umathradicalkern      \tex_Umathradicalkern:D
\__kernel_primitive:NN \Umathradicalrule      \tex_Umathradicalrule:D
\__kernel_primitive:NN \Umathradicalvgap      \tex_Umathradicalvgap:D
\__kernel_primitive:NN \Umathrelbinspacing    \tex_Umathrelbinspacing:D
\__kernel_primitive:NN \Umathrelclosespacing  \tex_Umathrelclosespacing:D
\__kernel_primitive:NN \Umathrelinnerspacing  \tex_Umathrelinnerspacing:D
\__kernel_primitive:NN \Umathrelopenspacing   \tex_Umathrelopenspacing:D
\__kernel_primitive:NN \Umathrelopspacing     \tex_Umathrelopspacing:D
\__kernel_primitive:NN \Umathrelordspacing    \tex_Umathrelordspacing:D
\__kernel_primitive:NN \Umathrelpunctspacing  \tex_Umathrelpunctspacing:D
\__kernel_primitive:NN \Umathrelrelspacing    \tex_Umathrelrelspacing:D
\__kernel_primitive:NN \Umathskewedfractionhgap
  \tex_Umathskewedfractionhgap:D
\__kernel_primitive:NN \Umathskewedfractionvgap
  \tex_Umathskewedfractionvgap:D
\__kernel_primitive:NN \Umathspaceafterscript \tex_Umathspaceafterscript:D
\__kernel_primitive:NN \Umathstackdenomdown   \tex_Umathstackdenomdown:D
\__kernel_primitive:NN \Umathstacknumup       \tex_Umathstacknumup:D
\__kernel_primitive:NN \Umathstackvgap        \tex_Umathstackvgap:D
\__kernel_primitive:NN \Umathsubshiftdown     \tex_Umathsubshiftdown:D
\__kernel_primitive:NN \Umathsubshiftdrop     \tex_Umathsubshiftdrop:D
\__kernel_primitive:NN \Umathsubsupshiftdown  \tex_Umathsubsupshiftdown:D
\__kernel_primitive:NN \Umathsubsupvgap       \tex_Umathsubsupvgap:D
\__kernel_primitive:NN \Umathsubtopmax        \tex_Umathsubtopmax:D
\__kernel_primitive:NN \Umathsupbottommin     \tex_Umathsupbottommin:D
\__kernel_primitive:NN \Umathsupshiftdrop     \tex_Umathsupshiftdrop:D
\__kernel_primitive:NN \Umathsupshiftup       \tex_Umathsupshiftup:D
\__kernel_primitive:NN \Umathsupsubbottommax  \tex_Umathsupsubbottommax:D
\__kernel_primitive:NN \Umathunderbarkern     \tex_Umathunderbarkern:D
\__kernel_primitive:NN \Umathunderbarrule     \tex_Umathunderbarrule:D
\__kernel_primitive:NN \Umathunderbarvgap     \tex_Umathunderbarvgap:D
\__kernel_primitive:NN \Umathunderdelimiterbgap
  \tex_Umathunderdelimiterbgap:D
\__kernel_primitive:NN \Umathunderdelimitervgap
  \tex_Umathunderdelimitervgap:D
\__kernel_primitive:NN \Umiddle               \tex_Umiddle:D
\__kernel_primitive:NN \Unosubscript          \tex_Unosubscript:D
\__kernel_primitive:NN \Unosuperscript        \tex_Unosuperscript:D
\__kernel_primitive:NN \Uoverdelimiter        \tex_Uoverdelimiter:D
\__kernel_primitive:NN \Uradical              \tex_Uradical:D
\__kernel_primitive:NN \Uright                \tex_Uright:D
\__kernel_primitive:NN \Uroot                 \tex_Uroot:D
```

```
\__kernel_primitive:NN \Uskewed                 \tex_Uskewed:D
\__kernel_primitive:NN \Uskewedwithdelims       \tex_Uskewedwithdelims:D
\__kernel_primitive:NN \Ustack                  \tex_Ustack:D
\__kernel_primitive:NN \Ustartdisplaymath       \tex_Ustartdisplaymath:D
\__kernel_primitive:NN \Ustartmath              \tex_Ustartmath:D
\__kernel_primitive:NN \Ustopdisplaymath        \tex_Ustopdisplaymath:D
\__kernel_primitive:NN \Ustopmath               \tex_Ustopmath:D
\__kernel_primitive:NN \Usubscript              \tex_Usubscript:D
\__kernel_primitive:NN \Usuperscript            \tex_Usuperscript:D
\__kernel_primitive:NN \Uunderdelimiter         \tex_Uunderdelimiter:D
\__kernel_primitive:NN \Uvextensible            \tex_Uvextensible:D
\__kernel_primitive:NN \autospacing             \tex_autospacing:D
\__kernel_primitive:NN \autoxspacing            \tex_autoxspacing:D
\__kernel_primitive:NN \currentcjktoken         \tex_currentcjktoken:D
\__kernel_primitive:NN \currentspacingmode      \tex_currentspacingmode:D
\__kernel_primitive:NN \currentxspacingmode     \tex_currentxspacingmode:D
\__kernel_primitive:NN \disinhibitglue          \tex_disinhibitglue:D
\__kernel_primitive:NN \dtou                    \tex_dtou:D
\__kernel_primitive:NN \epTeXinputencoding      \tex_epTeXinputencoding:D
\__kernel_primitive:NN \epTeXversion            \tex_epTeXversion:D
\__kernel_primitive:NN \euc                     \tex_euc:D
\__kernel_primitive:NN \hfi                     \tex_hfi:D
\__kernel_primitive:NN \ifdbox                  \tex_ifdbox:D
\__kernel_primitive:NN \ifddir                  \tex_ifddir:D
\__kernel_primitive:NN \ifjfont                 \tex_ifjfont:D
\__kernel_primitive:NN \ifmbox                  \tex_ifmbox:D
\__kernel_primitive:NN \ifmdir                  \tex_ifmdir:D
\__kernel_primitive:NN \iftbox                  \tex_iftbox:D
\__kernel_primitive:NN \iftfont                 \tex_iftfont:D
\__kernel_primitive:NN \iftdir                  \tex_iftdir:D
\__kernel_primitive:NN \ifybox                  \tex_ifybox:D
\__kernel_primitive:NN \ifydir                  \tex_ifydir:D
\__kernel_primitive:NN \inhibitglue             \tex_inhibitglue:D
\__kernel_primitive:NN \inhibitxspcode          \tex_inhibitxspcode:D
\__kernel_primitive:NN \jcharwidowpenalty       \tex_jcharwidowpenalty:D
\__kernel_primitive:NN \jfam                    \tex_jfam:D
\__kernel_primitive:NN \jfont                   \tex_jfont:D
\__kernel_primitive:NN \jis                     \tex_jis:D
\__kernel_primitive:NN \kanjiskip               \tex_kanjiskip:D
\__kernel_primitive:NN \kansuji                 \tex_kansuji:D
\__kernel_primitive:NN \kansujichar             \tex_kansujichar:D
\__kernel_primitive:NN \kcatcode                \tex_kcatcode:D
\__kernel_primitive:NN \kuten                   \tex_kuten:D
\__kernel_primitive:NN \lastnodechar            \tex_lastnodechar:D
\__kernel_primitive:NN \lastnodefont            \tex_lastnodefont:D
\__kernel_primitive:NN \lastnodesubtype         \tex_lastnodesubtype:D
```

```
\__kernel_primitive:NN \noautospacing        \tex_noautospacing:D
\__kernel_primitive:NN \noautoxspacing       \tex_noautoxspacing:D
\__kernel_primitive:NN \pagefistretch        \tex_pagefistretch:D
\__kernel_primitive:NN \postbreakpenalty     \tex_postbreakpenalty:D
\__kernel_primitive:NN \prebreakpenalty      \tex_prebreakpenalty:D
\__kernel_primitive:NN \ptexfontname         \tex_ptexfontname:D
\__kernel_primitive:NN \ptexlineendmode      \tex_lineendmode:D
\__kernel_primitive:NN \ptexminorversion     \tex_ptexminorversion:D
\__kernel_primitive:NN \ptexrevision         \tex_ptexrevision:D
\__kernel_primitive:NN \ptextracingfonts     \tex_ptextracingfonts:D
\__kernel_primitive:NN \ptexversion          \tex_ptexversion:D
\__kernel_primitive:NN \readpapersizespecial \tex_readpapersizespecial:D
\__kernel_primitive:NN \scriptbaselineshiftfactor
  \tex_scriptbaselineshiftfactor:D
\__kernel_primitive:NN \scriptscriptbaselineshiftfactor
  \tex_scriptscriptbaselineshiftfactor:D
\__kernel_primitive:NN \showmode             \tex_showmode:D
\__kernel_primitive:NN \sjis                 \tex_sjis:D
\__kernel_primitive:NN \tate                 \tex_tate:D
\__kernel_primitive:NN \tbaselineshift       \tex_tbaselineshift:D
\__kernel_primitive:NN \textbaselineshiftfactor
  \tex_textbaselineshiftfactor:D
\__kernel_primitive:NN \tfont                \tex_tfont:D
\__kernel_primitive:NN \tojis                \tex_tojis:D
\__kernel_primitive:NN \toucs                \tex_toucs:D
\__kernel_primitive:NN \ucs                  \tex_ucs:D
\__kernel_primitive:NN \xkanjiskip           \tex_xkanjiskip:D
\__kernel_primitive:NN \xspcode              \tex_xspcode:D
\__kernel_primitive:NN \ybaselineshift       \tex_ybaselineshift:D
\__kernel_primitive:NN \yoko                 \tex_yoko:D
\__kernel_primitive:NN \vfi                  \tex_vfi:D
\__kernel_primitive:NN \disablecjktoken      \tex_disablecjktoken:D
\__kernel_primitive:NN \enablecjktoken       \tex_enablecjktoken:D
\__kernel_primitive:NN \forcecjktoken        \tex_forcecjktoken:D
\__kernel_primitive:NN \kchar                \tex_kchar:D
\__kernel_primitive:NN \kchardef             \tex_kchardef:D
\__kernel_primitive:NN \uptexrevision        \tex_uptexrevision:D
\__kernel_primitive:NN \uptexversion         \tex_uptexversion:D
\__kernel_primitive:NN \odelcode             \tex_odelcode:D
\__kernel_primitive:NN \odelimiter           \tex_odelimiter:D
\__kernel_primitive:NN \omathaccent          \tex_omathaccent:D
\__kernel_primitive:NN \omathchar            \tex_omathchar:D
\__kernel_primitive:NN \omathchardef         \tex_omathchardef:D
\__kernel_primitive:NN \omathcode            \tex_omathcode:D
\__kernel_primitive:NN \oradical             \tex_oradical:D
\__kernel_primitive:NN \ignoreprimitiveerror \tex_ignoreprimitiveerror:D
```

```
  \__kernel_primitive:NN \partokencontext       \tex_partokencontext:D
  \__kernel_primitive:NN \partokenname          \tex_partokenname:D
  \__kernel_primitive:NN \showstream            \tex_showstream:D
  \__kernel_primitive:NN \tracingstacklevels    \tex_tracingstacklevels:D
\tex_endgroup:D
\tex_ifdefined:D \@@end
  \tex_let:D \tex_end:D                   \@@end
  \tex_let:D \tex_input:D                 \@@input
\tex_fi:D
\tex_ifdefined:D \@@hyph
  \tex_let:D \tex_everydisplay:D          \frozen@everydisplay
  \tex_let:D \tex_everymath:D             \frozen@everymath
  \tex_let:D \tex_hyphen:D                \@@hyph
  \tex_let:D \tex_italiccorrection:D      \@@italiccorr
  \tex_let:D \tex_underline:D             \@@underline
  \tex_ifdefined:D \@@shipout
    \tex_let:D \tex_shipout:D \@@shipout
  \tex_fi:D
  \tex_begingroup:D
    \tex_edef:D \l_tmpa_tl { \tex_string:D \shipout }
    \tex_edef:D \l_tmpb_tl { \tex_meaning:D \shipout }
    \tex_ifx:D \l_tmpa_tl \l_tmpb_tl
    \tex_else:D
      \tex_expandafter:D \@tfor \tex_expandafter:D \@tempa \tex_string:D :=
        \CROP@shipout
        \dup@shipout
        \GPTorg@shipout
        \LL@shipout
        \mem@oldshipout
        \opem@shipout
        \pgfpages@originalshipout
        \pr@shipout
        \Shipout
        \verso@orig@shipout
        \do
          {
            \tex_edef:D \l_tmpb_tl
              { \tex_expandafter:D \tex_meaning:D \@tempa }
            \tex_ifx:D \l_tmpa_tl \l_tmpb_tl
              \tex_global:D \tex_expandafter:D \tex_let:D
                \tex_expandafter:D \tex_shipout:D \@tempa
            \tex_fi:D
          }
    \tex_fi:D
  \tex_endgroup:D
  \tex_let:D \tex_tracingfonts:D \tex_undefined:D
  \tex_ifdefined:D \pdftracingfonts
    \tex_let:D \tex_tracingfonts:D \pdftracingfonts
  \tex_else:D
    \tex_ifdefined:D \tex_directlua:D
      \tex_directlua:D { tex.enableprimitives("@@", {"tracingfonts"}) }
      \tex_let:D \tex_tracingfonts:D \@@tracingfonts
    \tex_fi:D
  \tex_fi:D
\tex_fi:D
\tex_ifnum:D 0
  \tex_ifdefined:D \tex_pdftexversion:D 1 \tex_fi:D
  \tex_ifdefined:D \tex_luatexversion:D 1 \tex_fi:D
    = 0 %
  \tex_let:D \tex_pdfmapfile:D \tex_undefined:D
  \tex_let:D \tex_pdfmapline:D \tex_undefined:D
\tex_fi:D
\tex_begingroup:D
  \tex_edef:D \l_tmpa_tl { \tex_meaning:D \tex_time:D }
  \tex_edef:D \l_tmpb_tl { \tex_string:D \time }
  \tex_ifx:D \l_tmpa_tl \l_tmpb_tl
  \tex_else:D
    \tex_global:D \tex_let:D \tex_time:D \tex_undefined:D
  \tex_fi:D
  \tex_edef:D \l_tmpa_tl { \tex_meaning:D \tex_day:D }
  \tex_edef:D \l_tmpb_tl { \tex_string:D \day }
  \tex_ifx:D \l_tmpa_tl \l_tmpb_tl
  \tex_else:D
    \tex_global:D \tex_let:D \tex_day:D \tex_undefined:D
  \tex_fi:D
  \tex_edef:D \l_tmpa_tl { \tex_meaning:D \tex_month:D }
  \tex_edef:D \l_tmpb_tl { \tex_string:D \month }
  \tex_ifx:D \l_tmpa_tl \l_tmpb_tl
  \tex_else:D
    \tex_global:D \tex_let:D \tex_month:D \tex_undefined:D
  \tex_fi:D
  \tex_edef:D \l_tmpa_tl { \tex_meaning:D \tex_year:D }
  \tex_edef:D \l_tmpb_tl { \tex_string:D \year }
  \tex_ifx:D \l_tmpa_tl \l_tmpb_tl
  \tex_else:D
    \tex_global:D \tex_let:D \tex_year:D \tex_undefined:D
  \tex_fi:D
\tex_endgroup:D
\tex_ifdefined:D \orieveryjob
  \tex_let:D \tex_everyjob:D \orieveryjob
\tex_fi:D
\tex_ifdefined:D \oripdfoutput
```

```
  \tex_let:D \tex_pdfoutput:D \oripdfoutput
\tex_fi:D
\tex_ifdefined:D \normalend
  \tex_let:D \tex_end:D         \normalend
  \tex_let:D \tex_everyjob:D    \normaleveryjob
  \tex_let:D \tex_input:D       \normalinput
  \tex_let:D \tex_language:D    \normallanguage
  \tex_let:D \tex_mathop:D      \normalmathop
  \tex_let:D \tex_month:D       \normalmonth
  \tex_let:D \tex_outer:D       \normalouter
  \tex_let:D \tex_over:D        \normalover
  \tex_let:D \tex_vcenter:D     \normalvcenter
  \tex_let:D \tex_unexpanded:D  \normalunexpanded
  \tex_let:D \tex_expanded:D    \normalexpanded
\tex_fi:D
\tex_ifdefined:D \normalitaliccorrection
  \tex_let:D \tex_hoffset:D          \normalhoffset
  \tex_let:D \tex_italiccorrection:D \normalitaliccorrection
  \tex_let:D \tex_voffset:D          \normalvoffset
  \tex_let:D \tex_showtokens:D       \normalshowtokens
  \tex_let:D \tex_bodydir:D          \spac_directions_normal_body_dir
  \tex_let:D \tex_pagedir:D          \spac_directions_normal_page_dir
\tex_fi:D
\tex_ifdefined:D \normalleft
  \tex_let:D \tex_left:D   \normalleft
  \tex_let:D \tex_middle:D \normalmiddle
  \tex_let:D \tex_right:D  \normalright
\tex_fi:D
%% File: l3basics.dtx
\tex_global:D \tex_let:D \if_true:           \tex_iftrue:D
\tex_global:D \tex_let:D \if_false:          \tex_iffalse:D
\tex_global:D \tex_let:D \or:                \tex_or:D
\tex_global:D \tex_let:D \else:              \tex_else:D
\tex_global:D \tex_let:D \fi:                \tex_fi:D
\tex_global:D \tex_let:D \reverse_if:N       \tex_unless:D
\tex_global:D \tex_let:D \if:w               \tex_if:D
\tex_global:D \tex_let:D \if_charcode:w      \tex_if:D
\tex_global:D \tex_let:D \if_catcode:w       \tex_ifcat:D
\tex_global:D \tex_let:D \if_meaning:w       \tex_ifx:D
\tex_global:D \tex_let:D \if_bool:N          \tex_ifodd:D
\tex_global:D \tex_let:D \if_mode_math:       \tex_ifmmode:D
\tex_global:D \tex_let:D \if_mode_horizontal: \tex_ifhmode:D
\tex_global:D \tex_let:D \if_mode_vertical:   \tex_ifvmode:D
\tex_global:D \tex_let:D \if_mode_inner:      \tex_ifinner:D
\tex_global:D \tex_let:D \if_cs_exist:N      \tex_ifdefined:D
\tex_global:D \tex_let:D \if_cs_exist:w      \tex_ifcsname:D
```

```
\tex_global:D \tex_let:D \cs:w               \tex_csname:D
\tex_global:D \tex_let:D \cs_end:            \tex_endcsname:D
\tex_global:D \tex_let:D \exp_after:wN       \tex_expandafter:D
\tex_global:D \tex_let:D \exp_not:N          \tex_noexpand:D
\tex_global:D \tex_let:D \exp_not:n          \tex_unexpanded:D
\tex_global:D \tex_let:D \exp:w              \tex_romannumeral:D
\tex_global:D \tex_chardef:D \exp_end:  = 0 ~
\tex_global:D \tex_let:D \token_to_meaning:N \tex_meaning:D
\tex_global:D \tex_let:D \cs_meaning:N       \tex_meaning:D
\tex_global:D \tex_let:D \tl_to_str:n          \tex_detokenize:D
\tex_global:D \tex_let:D \token_to_str:N       \tex_string:D
\tex_global:D \tex_let:D \__kernel_tl_to_str:w \tex_detokenize:D
\tex_global:D \tex_let:D \scan_stop:         \tex_relax:D
\tex_global:D \tex_let:D \group_begin:       \tex_begingroup:D
\tex_global:D \tex_let:D \group_end:         \tex_endgroup:D
\tex_global:D \tex_let:D \if_int_compare:w   \tex_ifnum:D
\tex_global:D \tex_let:D \__int_to_roman:w     \tex_romannumeral:D
\tex_global:D \tex_let:D \group_insert_after:N \tex_aftergroup:D
\tex_long:D \tex_gdef:D \exp_args:Nc #1#2
  { \exp_after:wN #1 \cs:w #2 \cs_end: }
\tex_long:D \tex_gdef:D \exp_args:cc #1#2
  { \cs:w #1 \exp_after:wN \cs_end: \cs:w #2 \cs_end: }
\tex_gdef:D \token_to_str:c { \exp_args:Nc \token_to_str:N }
\tex_long:D \tex_gdef:D \cs_meaning:c #1
  {
    \if_cs_exist:w #1 \cs_end:
      \exp_after:wN \use_i:nn
    \else:
      \exp_after:wN \use_ii:nn
    \fi:
    { \exp_args:Nc \cs_meaning:N {#1} }
    { \tl_to_str:n {undefined} }
  }
\tex_global:D \tex_let:D \token_to_meaning:c = \cs_meaning:c
\tex_global:D \tex_chardef:D \c_zero_int    = 0 ~
\tex_ifdefined:D \tex_luatexversion:D
  \tex_global:D \tex_chardef:D \c_max_register_int = 65 535 ~
\tex_else:D
  \tex_ifdefined:D \tex_omathchardef:D
    \tex_global:D \tex_omathchardef:D \c_max_register_int = 65535 ~
  \tex_else:D
    \tex_global:D \tex_mathchardef:D \c_max_register_int = 32767 ~
  \tex_fi:D
\tex_fi:D
\tex_global:D \tex_let:D \cs_gset_nopar:Npn           \tex_gdef:D
\tex_global:D \tex_let:D \cs_gset_nopar:Npe           \tex_xdef:D
```

```
\tex_global:D \tex_let:D \cs_gset_nopar:Npx            \tex_xdef:D
\tex_protected:D \tex_long:D \tex_gdef:D \cs_gset:Npn
  { \tex_long:D \tex_gdef:D }
\tex_protected:D \tex_long:D \tex_gdef:D \cs_gset:Npe
  { \tex_long:D \tex_xdef:D }
\tex_global:D \tex_let:D \cs_gset:Npx \cs_gset:Npe
\tex_protected:D \tex_long:D \tex_gdef:D \cs_gset_protected_nopar:Npn
  { \tex_protected:D \tex_gdef:D }
\tex_protected:D \tex_long:D \tex_gdef:D \cs_gset_protected_nopar:Npe
  { \tex_protected:D \tex_xdef:D }
\tex_global:D \tex_let:D \cs_gset_protected_nopar:Npx \cs_gset_protected_nopar:Npe
\tex_protected:D \tex_long:D \tex_gdef:D \cs_gset_protected:Npn
  { \tex_protected:D \tex_long:D \tex_gdef:D }
\tex_protected:D \tex_long:D \tex_gdef:D \cs_gset_protected:Npe
  { \tex_protected:D \tex_long:D \tex_xdef:D }
\tex_global:D \tex_let:D \cs_gset_protected:Npx \cs_gset_protected:Npe
\tex_global:D \tex_let:D \cs_set_nopar:Npn            \tex_def:D
\tex_global:D \tex_let:D \cs_set_nopar:Npe            \tex_edef:D
\tex_global:D \tex_let:D \cs_set_nopar:Npx            \tex_edef:D
\cs_gset_protected:Npn \cs_set:Npn
  { \tex_long:D \tex_def:D }
\cs_gset_protected:Npn \cs_set:Npe
  { \tex_long:D \tex_edef:D }
\tex_global:D \tex_let:D \cs_set:Npx \cs_set:Npe
\cs_gset_protected:Npn \cs_set_protected_nopar:Npn
  { \tex_protected:D \tex_def:D }
\cs_gset_protected:Npn \cs_set_protected_nopar:Npe
  { \tex_protected:D \tex_edef:D }
\tex_global:D \tex_let:D \cs_set_protected_nopar:Npx \cs_set_protected_nopar:Npe
\cs_gset_protected:Npn \cs_set_protected:Npn
  { \tex_protected:D \tex_long:D \tex_def:D }
\cs_gset_protected:Npn \cs_set_protected:Npe
  { \tex_protected:D \tex_long:D \tex_edef:D }
\tex_global:D \tex_let:D \cs_set_protected:Npx \cs_set_protected:Npe
\cs_gset_nopar:Npn \l__exp_internal_tl { }
\cs_gset:Npn \use:c #1 { \cs:w #1 \cs_end: }
\cs_gset_protected:Npn \use:x #1
  {
    \cs_set_nopar:Npx \l__exp_internal_tl {#1}
    \l__exp_internal_tl
  }
\cs_gset:Npn \use:e #1 { \tex_expanded:D {#1} }
\cs_gset:Npn \use:n    #1       {#1}
\cs_gset:Npn \use:nn   #1#2     {#1#2}
\cs_gset:Npn \use:nnn  #1#2#3   {#1#2#3}
\cs_gset:Npn \use:nnnn #1#2#3#4 {#1#2#3#4}
```

```
\cs_gset:Npn \use_i:nn  #1#2 {#1}
\cs_gset:Npn \use_ii:nn #1#2 {#2}
\cs_gset:Npn \use_i:nnn    #1#2#3 {#1}
\cs_gset:Npn \use_ii:nnn   #1#2#3 {#2}
\cs_gset:Npn \use_iii:nnn  #1#2#3 {#3}
\cs_gset:Npn \use_i:nnnn   #1#2#3#4 {#1}
\cs_gset:Npn \use_ii:nnnn  #1#2#3#4 {#2}
\cs_gset:Npn \use_iii:nnnn #1#2#3#4 {#3}
\cs_gset:Npn \use_iv:nnnn  #1#2#3#4 {#4}
\cs_gset:Npn \use_i:nnnnn   #1#2#3#4#5 {#1}
\cs_gset:Npn \use_ii:nnnnn  #1#2#3#4#5 {#2}
\cs_gset:Npn \use_iii:nnnnn #1#2#3#4#5 {#3}
\cs_gset:Npn \use_iv:nnnnn  #1#2#3#4#5 {#4}
\cs_gset:Npn \use_v:nnnnn   #1#2#3#4#5 {#5}
\cs_gset:Npn \use_i:nnnnnn   #1#2#3#4#5#6 {#1}
\cs_gset:Npn \use_ii:nnnnnn  #1#2#3#4#5#6 {#2}
\cs_gset:Npn \use_iii:nnnnnn #1#2#3#4#5#6 {#3}
\cs_gset:Npn \use_iv:nnnnnn  #1#2#3#4#5#6 {#4}
\cs_gset:Npn \use_v:nnnnnn   #1#2#3#4#5#6 {#5}
\cs_gset:Npn \use_vi:nnnnnn  #1#2#3#4#5#6 {#6}
\cs_gset:Npn \use_i:nnnnnnn   #1#2#3#4#5#6#7 {#1}
\cs_gset:Npn \use_ii:nnnnnnn  #1#2#3#4#5#6#7 {#2}
\cs_gset:Npn \use_iii:nnnnnnn #1#2#3#4#5#6#7 {#3}
\cs_gset:Npn \use_iv:nnnnnnn  #1#2#3#4#5#6#7 {#4}
\cs_gset:Npn \use_v:nnnnnnn   #1#2#3#4#5#6#7 {#5}
\cs_gset:Npn \use_vi:nnnnnnn  #1#2#3#4#5#6#7 {#6}
\cs_gset:Npn \use_vii:nnnnnnn #1#2#3#4#5#6#7 {#7}
\cs_gset:Npn \use_i:nnnnnnnn    #1#2#3#4#5#6#7#8 {#1}
\cs_gset:Npn \use_ii:nnnnnnnn   #1#2#3#4#5#6#7#8 {#2}
\cs_gset:Npn \use_iii:nnnnnnnn  #1#2#3#4#5#6#7#8 {#3}
\cs_gset:Npn \use_iv:nnnnnnnn   #1#2#3#4#5#6#7#8 {#4}
\cs_gset:Npn \use_v:nnnnnnnn    #1#2#3#4#5#6#7#8 {#5}
\cs_gset:Npn \use_vi:nnnnnnnn   #1#2#3#4#5#6#7#8 {#6}
\cs_gset:Npn \use_vii:nnnnnnnn  #1#2#3#4#5#6#7#8 {#7}
\cs_gset:Npn \use_viii:nnnnnnnn #1#2#3#4#5#6#7#8 {#8}
\cs_gset:Npn \use_i:nnnnnnnnn    #1#2#3#4#5#6#7#8#9 {#1}
\cs_gset:Npn \use_ii:nnnnnnnnn   #1#2#3#4#5#6#7#8#9 {#2}
\cs_gset:Npn \use_iii:nnnnnnnnn  #1#2#3#4#5#6#7#8#9 {#3}
\cs_gset:Npn \use_iv:nnnnnnnnn   #1#2#3#4#5#6#7#8#9 {#4}
\cs_gset:Npn \use_v:nnnnnnnnn    #1#2#3#4#5#6#7#8#9 {#5}
\cs_gset:Npn \use_vi:nnnnnnnnn   #1#2#3#4#5#6#7#8#9 {#6}
\cs_gset:Npn \use_vii:nnnnnnnnn  #1#2#3#4#5#6#7#8#9 {#7}
\cs_gset:Npn \use_viii:nnnnnnnnn #1#2#3#4#5#6#7#8#9 {#8}
\cs_gset:Npn \use_ix:nnnnnnnnn   #1#2#3#4#5#6#7#8#9 {#9}
\cs_gset:Npn \use_i_ii:nnn #1#2#3 {#1#2}
\cs_gset:Npn \use_ii_i:nn #1#2 { #2 #1 }
```

```
\cs_gset:Npn \use_none_delimit_by_q_nil:w  #1 \q_nil  { }
\cs_gset:Npn \use_none_delimit_by_q_stop:w #1 \q_stop { }
\cs_gset:Npn \use_none_delimit_by_q_recursion_stop:w #1 \q_recursion_stop { }
\cs_gset:Npn \use_i_delimit_by_q_nil:nw  #1#2 \q_nil  {#1}
\cs_gset:Npn \use_i_delimit_by_q_stop:nw #1#2 \q_stop {#1}
\cs_gset:Npn \use_i_delimit_by_q_recursion_stop:nw
  #1#2 \q_recursion_stop {#1}
\cs_gset:Npn \use_none:n         #1                 { }
\cs_gset:Npn \use_none:nn        #1#2               { }
\cs_gset:Npn \use_none:nnn       #1#2#3             { }
\cs_gset:Npn \use_none:nnnn      #1#2#3#4           { }
\cs_gset:Npn \use_none:nnnnn     #1#2#3#4#5         { }
\cs_gset:Npn \use_none:nnnnnn    #1#2#3#4#5#6       { }
\cs_gset:Npn \use_none:nnnnnnn   #1#2#3#4#5#6#7     { }
\cs_gset:Npn \use_none:nnnnnnnn  #1#2#3#4#5#6#7#8   { }
\cs_gset:Npn \use_none:nnnnnnnnn #1#2#3#4#5#6#7#8#9 { }
\cs_gset_protected:Npn \__kernel_if_debug:TF #1#2 {#2}
\cs_gset_protected:Npn \debug_on:n #1
  {
    \sys_load_debug:
    \cs_if_exist:NT \__debug_all_on:
      { \debug_on:n {#1} }
  }
\cs_gset_protected:Npn \debug_off:n #1
  {
    \sys_load_debug:
    \cs_if_exist:NT \__debug_all_on:
      { \debug_off:n {#1} }
  }
\cs_gset_protected:Npn \debug_suspend: { }
\cs_gset_protected:Npn \debug_resume: { }
\cs_gset_nopar:Npn \g__debug_deprecation_on_tl { }
\cs_gset_nopar:Npn \g__debug_deprecation_off_tl { }
\cs_gset_protected:Npn \__kernel_deprecation_code:nn #1#2
  {
    \tl_gput_right:Nn \g__debug_deprecation_on_tl {#1}
    \tl_gput_right:Nn \g__debug_deprecation_off_tl {#2}
  }
\cs_gset:Npn \prg_return_true:
  { \exp_after:wN \use_i:nn  \exp:w }
\cs_gset:Npn \prg_return_false:
  { \exp_after:wN \use_ii:nn \exp:w}
\cs_gset:Npn \__prg_use_none_delimit_by_q_recursion_stop:w
  #1 \q__prg_recursion_stop { }
\cs_gset_protected:Npn \prg_set_conditional:Npnn
  { \__prg_generate_conditional_parm:NNNpnn \cs_set:Npn e }
```

```
\cs_gset_protected:Npn \prg_gset_conditional:Npnn
  { \__prg_generate_conditional_parm:NNNpnn \cs_gset:Npn e }
\cs_gset_protected:Npn \prg_new_conditional:Npnn
  { \__prg_generate_conditional_parm:NNNpnn \cs_new:Npn e }
\cs_gset_protected:Npn \prg_set_protected_conditional:Npnn
  { \__prg_generate_conditional_parm:NNNpnn \cs_set_protected:Npn p }
\cs_gset_protected:Npn \prg_gset_protected_conditional:Npnn
  { \__prg_generate_conditional_parm:NNNpnn \cs_gset_protected:Npn p }
\cs_gset_protected:Npn \prg_new_protected_conditional:Npnn
  { \__prg_generate_conditional_parm:NNNpnn \cs_new_protected:Npn p }
\cs_gset_protected:Npn \__prg_generate_conditional_parm:NNNpnn #1#2#3#4#
  {
    \use:e
      {
        \__prg_generate_conditional:nnNNNnnn
          \cs_split_function:N #3
      }
      #1 #2 {#4}
  }
\cs_gset_protected:Npn \prg_set_conditional:Nnn
  { \__prg_generate_conditional_count:NNNnn \cs_set:Npn e }
\cs_gset_protected:Npn \prg_gset_conditional:Nnn
  { \__prg_generate_conditional_count:NNNnn \cs_gset:Npn e }
\cs_gset_protected:Npn \prg_new_conditional:Nnn
  { \__prg_generate_conditional_count:NNNnn \cs_new:Npn e }
\cs_gset_protected:Npn \prg_set_protected_conditional:Nnn
  { \__prg_generate_conditional_count:NNNnn \cs_set_protected:Npn p }
\cs_gset_protected:Npn \prg_gset_protected_conditional:Nnn
  { \__prg_generate_conditional_count:NNNnn \cs_gset_protected:Npn p }
\cs_gset_protected:Npn \prg_new_protected_conditional:Nnn
  { \__prg_generate_conditional_count:NNNnn \cs_new_protected:Npn p }
\cs_gset_protected:Npn \__prg_generate_conditional_count:NNNnn #1#2#3
  {
    \use:e
      {
        \__prg_generate_conditional_count:nnNNNnn
        \cs_split_function:N #3
      }
      #1 #2
  }
\cs_gset_protected:Npn \__prg_generate_conditional_count:nnNNNnn #1#2#3#4#5
  {
    \__kernel_cs_parm_from_arg_count:nnF
      { \__prg_generate_conditional:nnNNNnnn {#1} {#2} #3 #4 #5 }
      { \tl_count:n {#2} }
      {
        \msg_error:nnee { kernel } { bad-number-of-arguments }
          { \token_to_str:c { #1 : #2 } }
          { \tl_count:n {#2} }
        \use_none:nn
      }
  }
\cs_gset_protected:Npn \__prg_generate_conditional:nnNNNnnn #1#2#3#4#5#6#7#8
  {
    \if_meaning:w \c_false_bool #3
      \msg_error:nne { kernel } { missing-colon }
        { \token_to_str:c {#1} }
      \exp_after:wN \use_none:nn
    \fi:
    \use:e
      {
        \exp_not:N \__prg_generate_conditional:NNnnnnNw
        \exp_not:n { #4 #5 {#1} {#2} {#6} }
        \__prg_generate_conditional_test:w
          #8 \s__prg_mark
            \__prg_generate_conditional_fast:nw
          \prg_return_true: \else: \prg_return_false: \fi: \s__prg_mark
            \use_none:n
        \exp_not:n { {#8} \use_i_ii:nnn }
        \tl_to_str:n {#7}
        \exp_not:n { , \q__prg_recursion_tail , \q__prg_recursion_stop }
      }
  }
\cs_gset:Npn \__prg_generate_conditional_test:w
    #1 \prg_return_true: \else: \prg_return_false: \fi: \s__prg_mark #2
  { #2 {#1} }
\cs_gset:Npn \__prg_generate_conditional_fast:nw #1#2 \exp_not:n #3
  { \exp_not:n { {#1} \use_i:nn } }
\cs_gset_protected:Npn \__prg_generate_conditional:NNnnnnNw #1#2#3#4#5#6#7#8 ,
  {
    \if_meaning:w \q__prg_recursion_tail #8
      \exp_after:wN \__prg_use_none_delimit_by_q_recursion_stop:w
    \fi:
    \use:c { __prg_generate_ #8 _form:wNNnnnnN }
        \tl_if_empty:nF {#8}
          {
            \msg_error:nnee
              { kernel } { conditional-form-unknown }
              {#8} { \token_to_str:c { #3 : #4 } }
          }
        \use_none:nnnnnnnn
      \s__prg_stop
      #1 #2 {#3} {#4} {#5} {#6} #7
    \__prg_generate_conditional:NNnnnnNw #1 #2 {#3} {#4} {#5} {#6} #7
  }
\cs_gset_protected:Npn \__prg_generate_p_form:wNNnnnnN
    #1 \s__prg_stop #2#3#4#5#6#7#8
  {
    \if_meaning:w e #3
      \exp_after:wN \use_i:nn
    \else:
      \exp_after:wN \use_ii:nn
    \fi:
      {
        #8
          { \exp_args:Nc #2 { #4 _p: #5 } #6 }
          { { #7 \exp_end: \c_true_bool \c_false_bool } }
          { #7 \__prg_p_true:w \fi: \c_false_bool }
      }
      {
        \msg_error:nne { kernel } { protected-predicate }
          { \token_to_str:c { #4 _p: #5 } }
      }
  }
\cs_gset_protected:Npn \__prg_generate_T_form:wNNnnnnN
    #1 \s__prg_stop #2#3#4#5#6#7#8
  {
    #8
      { \exp_args:Nc #2 { #4 : #5 T } #6 }
      { { #7 \exp_end: \use:n \use_none:n } }
      { #7 \__prg_T_true:w \fi: \use_none:n }
  }
\cs_gset_protected:Npn \__prg_generate_F_form:wNNnnnnN
    #1 \s__prg_stop #2#3#4#5#6#7#8
  {
    #8
      { \exp_args:Nc #2 { #4 : #5 F } #6 }
      { { #7 \exp_end: { } } }
      { #7 \__prg_F_true:w \fi: \use:n }
  }
\cs_gset_protected:Npn \__prg_generate_TF_form:wNNnnnnN
    #1 \s__prg_stop #2#3#4#5#6#7#8
  {
    #8
      { \exp_args:Nc #2 { #4 : #5 TF } #6 }
      { { #7 \exp_end: } }
      { #7 \__prg_TF_true:w \fi: \use_ii:nn }
  }
```

```
\cs_gset:Npn \__prg_p_true:w  \fi: \c_false_bool { \fi: \c_true_bool }
\cs_gset:Npn \__prg_T_true:w  \fi: \use_none:n   { \fi: \use:n }
\cs_gset:Npn \__prg_F_true:w  \fi: \use:n        { \fi: \use_none:n }
\cs_gset:Npn \__prg_TF_true:w \fi: \use_ii:nn    { \fi: \use_i:nn }
\cs_gset_protected:Npn \prg_set_eq_conditional:NNn
  { \__prg_set_eq_conditional:NNNn \cs_set_eq:cc }
\cs_gset_protected:Npn \prg_gset_eq_conditional:NNn
  { \__prg_set_eq_conditional:NNNn \cs_gset_eq:cc }
\cs_gset_protected:Npn \prg_new_eq_conditional:NNn
  { \__prg_set_eq_conditional:NNNn \cs_new_eq:cc }
\cs_gset_protected:Npn \__prg_set_eq_conditional:NNNn #1#2#3#4
  {
    \use:e
      {
        \exp_not:N \__prg_set_eq_conditional:nnNnnNNw
          \cs_split_function:N #2
          \cs_split_function:N #3
          \exp_not:N #1
          \tl_to_str:n {#4}
          \exp_not:n { , \q__prg_recursion_tail , \q__prg_recursion_stop }
      }
  }
\cs_gset_protected:Npn \__prg_set_eq_conditional:nnNnnNNw #1#2#3#4#5#6
  {
    \if_meaning:w \c_false_bool #3
      \msg_error:nne { kernel } { missing-colon }
        { \token_to_str:c {#1} }
      \exp_after:wN \__prg_use_none_delimit_by_q_recursion_stop:w
    \fi:
    \if_meaning:w \c_false_bool #6
      \msg_error:nne { kernel } { missing-colon }
        { \token_to_str:c {#4} }
      \exp_after:wN \__prg_use_none_delimit_by_q_recursion_stop:w
    \fi:
    \__prg_set_eq_conditional_loop:nnnnNw {#1} {#2} {#4} {#5}
  }
\cs_gset_protected:Npn \__prg_set_eq_conditional_loop:nnnnNw #1#2#3#4#5#6 ,
  {
    \if_meaning:w \q__prg_recursion_tail #6
      \exp_after:wN \__prg_use_none_delimit_by_q_recursion_stop:w
    \fi:
    \use:c { __prg_set_eq_conditional_ #6 _form:wNnnnn }
        \tl_if_empty:nF {#6}
          {
            \msg_error:nnee
              { kernel } { conditional-form-unknown }
              {#6} { \token_to_str:c { #1 : #2 } }
          }
        \use_none:nnnnnn
      \s__prg_stop
      #5 {#1} {#2} {#3} {#4}
    \__prg_set_eq_conditional_loop:nnnnNw {#1} {#2} {#3} {#4} #5
  }
\cs_gset:Npn \__prg_set_eq_conditional_p_form:wNnnnn #1 \s__prg_stop #2#3#4#5#6
  { #2 { #3 _p : #4    }    { #5 _p : #6    } }
\cs_gset:Npn \__prg_set_eq_conditional_TF_form:wNnnnn #1 \s__prg_stop #2#3#4#5#6
  { #2 { #3    : #4 TF }    { #5    : #6 TF } }
\cs_gset:Npn \__prg_set_eq_conditional_T_form:wNnnnn #1 \s__prg_stop #2#3#4#5#6
  { #2 { #3    : #4 T  }    { #5    : #6 T  } }
\cs_gset:Npn \__prg_set_eq_conditional_F_form:wNnnnn #1 \s__prg_stop #2#3#4#5#6
  { #2 { #3    : #4  F }    { #5    : #6  F } }
\tex_global:D \tex_chardef:D \c_true_bool  = 1 ~
\tex_global:D \tex_chardef:D \c_false_bool = 0 ~
\cs_gset:Npn \cs_to_str:N
  {
    \tex_romannumeral:D
      \if:w \token_to_str:N \ \__cs_to_str:w \fi:
      \exp_after:wN \__cs_to_str:N \token_to_str:N
  }
\cs_gset:Npn \__cs_to_str:N #1 { \c_zero_int }
\cs_gset:Npn \__cs_to_str:w #1 \__cs_to_str:N
  { - \int_value:w \fi: \exp_after:wN \c_zero_int }
\cs_gset_protected:Npn \__cs_tmp:w #1
  {
    \cs_gset:Npn \cs_split_function:N ##1
      {
        \exp_after:wN \exp_after:wN \exp_after:wN
        \__cs_split_function_auxi:w
          \cs_to_str:N ##1 \s__cs_mark \c_true_bool
          #1 \s__cs_mark \c_false_bool \s__cs_stop
      }
    \cs_gset:Npn \__cs_split_function_auxi:w
        ##1 #1 ##2 \s__cs_mark ##3##4 \s__cs_stop
      { \__cs_split_function_auxii:w ##1 \s__cs_mark \s__cs_stop {##2} ##3 }
    \cs_gset:Npn \__cs_split_function_auxii:w ##1 \s__cs_mark ##2 \s__cs_stop
      { {##1} }
  }
\exp_after:wN \__cs_tmp:w \token_to_str:N :
\prg_gset_conditional:Npnn \cs_if_exist:N #1 { p , T , F , TF }
  {
    \if_meaning:w #1 \scan_stop:
      \use_i:nnnn
    \else:
    \fi:
    \if_cs_exist:N #1
      \prg_return_true:
    \else:
      \prg_return_false:
    \fi:
  }
\cs_if_exist:NTF \tex_lastnamedcs:D
  {
    \prg_gset_conditional:Npnn \cs_if_exist:c #1 { p , T , F , TF }
      {
        \if_cs_exist:w #1 \cs_end:
          \__cs_if_exist_c_aux:
          \prg_return_true:
        \else:
          \prg_return_false:
        \fi:
      }
    \cs_gset:Npn \__cs_if_exist_c_aux:
      { \fi: \exp_after:wN \if_meaning:w \tex_lastnamedcs:D \scan_stop: \else: }
  }
  {
    \prg_gset_conditional:Npnn \cs_if_exist:c #1 { p , T , F , TF }
      {
        \if_cs_exist:w #1 \cs_end:
          \__cs_if_exist_c_aux:w
        \fi:
        \use_none:n {#1}
        \if_false:
          \prg_return_true:
        \else:
          \prg_return_false:
        \fi:
      }
    \cs_gset:Npn \__cs_if_exist_c_aux:w \fi: \use_none:n #1 \if_false:
      { \fi: \exp_after:wN \if_meaning:w \cs:w #1 \cs_end: \scan_stop: \else: }
  }
\prg_gset_conditional:Npnn \cs_if_free:N #1 { p , T , F , TF }
  {
    \if_cs_exist:N #1
    \else:
      \use_none:nnnn
    \fi:
    \if_meaning:w #1 \scan_stop:
      \prg_return_true:
```

```
    \else:
      \prg_return_false:
    \fi:
  }
\cs_if_exist:NTF \tex_lastnamedcs:D
  {
    \prg_gset_conditional:Npnn \cs_if_free:c #1 { p , T , F , TF }
      {
        \if_cs_exist:w #1 \cs_end:
          \__cs_if_free_c_aux:w
        \fi:
        \if_true:
          \prg_return_true:
        \else:
          \prg_return_false:
        \fi:
      }
    \cs_gset:Npn \__cs_if_free_c_aux:w \fi: \if_true:
      { \fi: \exp_after:wN \if_meaning:w \tex_lastnamedcs:D \scan_stop: }
  }
  {
    \prg_gset_conditional:Npnn \cs_if_free:c #1 { p , T , F , TF }
      {
        \if_cs_exist:w #1 \cs_end:
          \__cs_if_free_c_aux:w
        \fi:
        \use_none:n {#1}
        \if_true:
          \prg_return_true:
        \else:
          \prg_return_false:
        \fi:
      }
    \cs_gset:Npn \__cs_if_free_c_aux:w \fi: \use_none:n #1 \if_true:
      { \fi: \exp_after:wN \if_meaning:w \cs:w #1 \cs_end: \scan_stop: }
  }
\cs_gset:Npn \cs_if_exist_use:NTF #1#2
  { \cs_if_exist:NTF #1 { #1 #2 } }
\cs_gset:Npn \cs_if_exist_use:NF #1
  { \cs_if_exist:NTF #1 #1 }
\cs_gset:Npn \cs_if_exist_use:NT #1 #2
  { \cs_if_exist:NT #1 { #1 #2 } }
\cs_gset:Npn \cs_if_exist_use:N #1
  { \cs_if_exist:NT #1 #1 }
\cs_if_exist:NTF \tex_lastnamedcs:D
  {
    \cs_gset:Npn \cs_if_exist_use:cTF #1
      {
        \if_cs_exist:w #1 \cs_end:
          \__cs_if_exist_use_aux:w
        \fi:
        \use_ii:nn
      }
    \cs_gset:Npn \__cs_if_exist_use_aux:w \fi: \use_ii:nn
      { \fi: \exp_after:wN \__cs_if_exist_use_aux:Nnn \tex_lastnamedcs:D }
  }
  {
    \cs_gset:Npn \cs_if_exist_use:cTF #1
      {
        \if_cs_exist:w #1 \cs_end:
          \__cs_if_exist_use_aux:w
        \fi:
        \use_iii:nnn {#1}
      }
    \cs_gset:Npn \__cs_if_exist_use_aux:w \fi: \use_iii:nnn #1
      { \fi: \exp_after:wN \__cs_if_exist_use_aux:Nnn \cs:w #1 \cs_end: }
  }
\cs_gset:Npn \__cs_if_exist_use_aux:Nnn #1#2
  {
    \if_meaning:w #1 \scan_stop:
      \exp_after:wN \use_iii:nnn
    \fi:
    \use_i:nn { #1 #2 }
  }
\cs_gset:Npn \cs_if_exist_use:cF #1
  { \cs_if_exist_use:cTF {#1} {} }
\cs_gset:Npn \cs_if_exist_use:cT #1#2
  { \cs_if_exist_use:cTF {#1} {#2} {} }
\cs_gset:Npn \cs_if_exist_use:c #1
  { \cs_if_exist_use:cTF {#1} {} {} }
\cs_gset_protected:Npn \msg_error:nnee #1#2#3#4
  {
    \tex_newlinechar:D = `\^^J \scan_stop:
    \tex_errmessage:D
      {
        !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!~! ^^J
        Argh,~internal~LaTeX3~error! ^^J ^^J
        Module ~ #1 , ~ message~name~"#2": ^^J
        Arguments~'#3'~and~'#4' ^^J ^^J
        This~is~one~for~The~LaTeX3~Project:~bailing~out
      }
    \tex_end:D
  }
\cs_gset_protected:Npn \msg_error:nne #1#2#3
  { \msg_error:nnee {#1} {#2} {#3} { } }
\cs_gset_protected:Npn \msg_error:nn #1#2
  { \msg_error:nnee {#1} {#2} { } { } }
\cs_gset:Npn \msg_line_context:
  { on~line~ \tex_the:D \tex_inputlineno:D }
\cs_gset_protected:Npn \iow_log:e
  { \tex_immediate:D \tex_write:D -1 }
\cs_gset_protected:Npn \iow_term:e
  { \tex_immediate:D \tex_write:D 16 }
\cs_gset_protected:Npn \__kernel_chk_if_free_cs:N #1
  {
    \cs_if_free:NF #1
      {
        \msg_error:nnee { kernel } { command-already-defined }
          { \token_to_str:N #1 } { \token_to_meaning:N #1 }
      }
  }
\cs_gset_protected:Npn \__kernel_chk_if_free_cs:c
  { \exp_args:Nc \__kernel_chk_if_free_cs:N }
\cs_set:Npn \__cs_tmp:w #1#2
  {
    \cs_gset_protected:Npn #1 ##1
      {
        \__kernel_chk_if_free_cs:N ##1
        #2 ##1
      }
  }
\__cs_tmp:w \cs_new_nopar:Npn           \cs_gset_nopar:Npn
\__cs_tmp:w \cs_new_nopar:Npe           \cs_gset_nopar:Npe
\__cs_tmp:w \cs_new_nopar:Npx           \cs_gset_nopar:Npx
\__cs_tmp:w \cs_new:Npn                 \cs_gset:Npn
\__cs_tmp:w \cs_new:Npe                 \cs_gset:Npe
\__cs_tmp:w \cs_new:Npx                 \cs_gset:Npx
\__cs_tmp:w \cs_new_protected_nopar:Npn \cs_gset_protected_nopar:Npn
\__cs_tmp:w \cs_new_protected_nopar:Npe \cs_gset_protected_nopar:Npe
\__cs_tmp:w \cs_new_protected_nopar:Npx \cs_gset_protected_nopar:Npx
\__cs_tmp:w \cs_new_protected:Npn       \cs_gset_protected:Npn
\__cs_tmp:w \cs_new_protected:Npe       \cs_gset_protected:Npe
\__cs_tmp:w \cs_new_protected:Npx       \cs_gset_protected:Npx
\cs_set:Npn \__cs_tmp:w #1#2
  { \cs_new_protected_nopar:Npn #1 { \exp_args:Nc #2 } }
\__cs_tmp:w \cs_set_nopar:cpn  \cs_set_nopar:Npn
\__cs_tmp:w \cs_set_nopar:cpe  \cs_set_nopar:Npe
\__cs_tmp:w \cs_set_nopar:cpx  \cs_set_nopar:Npx
```

```
\__cs_tmp:w \cs_gset_nopar:cpn \cs_gset_nopar:Npn
\__cs_tmp:w \cs_gset_nopar:cpe \cs_gset_nopar:Npe
\__cs_tmp:w \cs_gset_nopar:cpx \cs_gset_nopar:Npx
\__cs_tmp:w \cs_new_nopar:cpn  \cs_new_nopar:Npn
\__cs_tmp:w \cs_new_nopar:cpe  \cs_new_nopar:Npe
\__cs_tmp:w \cs_new_nopar:cpx  \cs_new_nopar:Npx
\__cs_tmp:w \cs_set:cpn  \cs_set:Npn
\__cs_tmp:w \cs_set:cpe  \cs_set:Npe
\__cs_tmp:w \cs_set:cpx  \cs_set:Npx
\__cs_tmp:w \cs_gset:cpn \cs_gset:Npn
\__cs_tmp:w \cs_gset:cpe \cs_gset:Npe
\__cs_tmp:w \cs_gset:cpx \cs_gset:Npx
\__cs_tmp:w \cs_new:cpn  \cs_new:Npn
\__cs_tmp:w \cs_new:cpe  \cs_new:Npe
\__cs_tmp:w \cs_new:cpx  \cs_new:Npx
\__cs_tmp:w \cs_set_protected_nopar:cpn  \cs_set_protected_nopar:Npn
\__cs_tmp:w \cs_set_protected_nopar:cpe  \cs_set_protected_nopar:Npe
\__cs_tmp:w \cs_set_protected_nopar:cpx  \cs_set_protected_nopar:Npx
\__cs_tmp:w \cs_gset_protected_nopar:cpn \cs_gset_protected_nopar:Npn
\__cs_tmp:w \cs_gset_protected_nopar:cpe \cs_gset_protected_nopar:Npe
\__cs_tmp:w \cs_gset_protected_nopar:cpx \cs_gset_protected_nopar:Npx
\__cs_tmp:w \cs_new_protected_nopar:cpn  \cs_new_protected_nopar:Npn
\__cs_tmp:w \cs_new_protected_nopar:cpe  \cs_new_protected_nopar:Npe
\__cs_tmp:w \cs_new_protected_nopar:cpx  \cs_new_protected_nopar:Npx
\__cs_tmp:w \cs_set_protected:cpn  \cs_set_protected:Npn
\__cs_tmp:w \cs_set_protected:cpe  \cs_set_protected:Npe
\__cs_tmp:w \cs_set_protected:cpx  \cs_set_protected:Npx
\__cs_tmp:w \cs_gset_protected:cpn \cs_gset_protected:Npn
\__cs_tmp:w \cs_gset_protected:cpe \cs_gset_protected:Npe
\__cs_tmp:w \cs_gset_protected:cpx \cs_gset_protected:Npx
\__cs_tmp:w \cs_new_protected:cpn  \cs_new_protected:Npn
\__cs_tmp:w \cs_new_protected:cpe  \cs_new_protected:Npe
\__cs_tmp:w \cs_new_protected:cpx  \cs_new_protected:Npx
\cs_new_protected:Npn \cs_set_eq:NN #1 { \tex_let:D #1 =~ }
\cs_new_protected:Npn \cs_set_eq:cN { \exp_args:Nc  \cs_set_eq:NN }
\cs_new_protected:Npn \cs_set_eq:Nc { \exp_args:NNc \cs_set_eq:NN }
\cs_new_protected:Npn \cs_set_eq:cc { \exp_args:Ncc \cs_set_eq:NN }
\cs_new_protected:Npn \cs_gset_eq:NN { \tex_global:D  \cs_set_eq:NN }
\cs_new_protected:Npn \cs_gset_eq:Nc { \exp_args:NNc  \cs_gset_eq:NN }
\cs_new_protected:Npn \cs_gset_eq:cN { \exp_args:Nc   \cs_gset_eq:NN }
\cs_new_protected:Npn \cs_gset_eq:cc { \exp_args:Ncc  \cs_gset_eq:NN }
\cs_new_protected:Npn \cs_new_eq:NN #1
  {
    \__kernel_chk_if_free_cs:N #1
    \tex_global:D \cs_set_eq:NN #1
  }
```

```
\cs_new_protected:Npn \cs_new_eq:cN { \exp_args:Nc  \cs_new_eq:NN }
\cs_new_protected:Npn \cs_new_eq:Nc { \exp_args:NNc \cs_new_eq:NN }
\cs_new_protected:Npn \cs_new_eq:cc { \exp_args:Ncc \cs_new_eq:NN }
\cs_new_protected:Npn \cs_undefine:N #1
  { \cs_gset_eq:NN #1 \tex_undefined:D }
\cs_new_protected:Npn \cs_undefine:c #1
  {
    \if_cs_exist:w #1 \cs_end:
    \else:
      \use_i:nnnn
    \fi:
    \exp_args:Nc \cs_undefine:N {#1}
  }
\cs_new_protected:Npn \__kernel_cs_parm_from_arg_count:nnF #1#2
  {
    \exp_args:Ne \__cs_parm_from_arg_count_test:nnF
      {
        \exp_after:wN \exp_not:n
        \if_case:w \int_eval:n {#2}
             { }
        \or: { ##1 }
        \or: { ##1##2 }
        \or: { ##1##2##3 }
        \or: { ##1##2##3##4 }
        \or: { ##1##2##3##4##5 }
        \or: { ##1##2##3##4##5##6 }
        \or: { ##1##2##3##4##5##6##7 }
        \or: { ##1##2##3##4##5##6##7##8 }
        \or: { ##1##2##3##4##5##6##7##8##9 }
        \else: { \c_false_bool }
        \fi:
      }
      {#1}
  }
\cs_new_protected:Npn \__cs_parm_from_arg_count_test:nnF #1#2
  {
    \if_meaning:w \c_false_bool #1
      \exp_after:wN \use_ii:nn
    \else:
      \exp_after:wN \use_i:nn
    \fi:
    { #2 {#1} }
  }
\cs_new:Npn \__cs_count_signature:N #1
  { \exp_args:Nf \__cs_count_signature:n { \cs_split_function:N #1 } }
\cs_new:Npn \__cs_count_signature:n #1
```

```
  { \int_eval:n { \__cs_count_signature:nnN #1 } }
\cs_new:Npn \__cs_count_signature:nnN #1#2#3
  {
    \if_meaning:w \c_true_bool #3
      \tl_count:n {#2}
    \else:
      -1
    \fi:
  }
\cs_new:Npn \__cs_count_signature:c
  { \exp_args:Nc \__cs_count_signature:N }
\cs_new_protected:Npn \cs_generate_from_arg_count:NNnn #1#2#3#4
  {
    \__kernel_cs_parm_from_arg_count:nnF { \use:nnn #2 #1 } {#3}
      {
        \msg_error:nnee { kernel } { bad-number-of-arguments }
          { \token_to_str:N #1 } { \int_eval:n {#3} }
        \use_none:n
      }
      {#4}
  }
\cs_new_protected:Npn \cs_generate_from_arg_count:cNnn
  { \exp_args:Nc \cs_generate_from_arg_count:NNnn }
\cs_new_protected:Npn \cs_generate_from_arg_count:Ncnn
  { \exp_args:NNc \cs_generate_from_arg_count:NNnn }
\cs_set:Npn \__cs_tmp:w #1#2#3
  {
    \cs_new_protected:cpx { cs_ #1 : #2 }
      {
        \exp_not:N \__cs_generate_from_signature:NNn
        \exp_after:wN \exp_not:N \cs:w cs_ #1 : #3 \cs_end:
      }
  }
\cs_new_protected:Npn \__cs_generate_from_signature:NNn #1#2
  {
    \use:e
      {
        \__cs_generate_from_signature:nnNNNn
        \cs_split_function:N #2
      }
      #1 #2
  }
\cs_new_protected:Npn \__cs_generate_from_signature:nnNNNn #1#2#3#4#5#6
  {
    \bool_if:NTF #3
      {
        \cs_set_nopar:Npx \__cs_tmp:w
          { \tl_map_function:nN {#2} \__cs_generate_from_signature:n }
        \tl_if_empty:oF \__cs_tmp:w
          {
            \msg_error:nneee { kernel } { non-base-function }
              { \token_to_str:N #5 } {#2} { \__cs_tmp:w }
          }
        \cs_generate_from_arg_count:NNnn
          #5 #4 { \tl_count:n {#2} } {#6}
      }
      {
        \msg_error:nne { kernel } { missing-colon }
          { \token_to_str:N #5 }
      }
  }
\cs_new:Npn \__cs_generate_from_signature:n #1
  {
    \if:w n #1 \else: \if:w N #1 \else:
    \if:w T #1 \else: \if:w F #1 \else: #1 \fi: \fi: \fi: \fi:
  }
\__cs_tmp:w { set }                  { Nn } { Npn }
\__cs_tmp:w { set }                  { Ne } { Npe }
\__cs_tmp:w { set }                  { Nx } { Npx }
\__cs_tmp:w { set_nopar }            { Nn } { Npn }
\__cs_tmp:w { set_nopar }            { Ne } { Npe }
\__cs_tmp:w { set_nopar }            { Nx } { Npx }
\__cs_tmp:w { set_protected }        { Nn } { Npn }
\__cs_tmp:w { set_protected }        { Ne } { Npe }
\__cs_tmp:w { set_protected }        { Nx } { Npx }
\__cs_tmp:w { set_protected_nopar }  { Nn } { Npn }
\__cs_tmp:w { set_protected_nopar }  { Ne } { Npe }
\__cs_tmp:w { set_protected_nopar }  { Nx } { Npx }
\__cs_tmp:w { gset }                 { Nn } { Npn }
\__cs_tmp:w { gset }                 { Ne } { Npe }
\__cs_tmp:w { gset }                 { Nx } { Npx }
\__cs_tmp:w { gset_nopar }           { Nn } { Npn }
\__cs_tmp:w { gset_nopar }           { Ne } { Npe }
\__cs_tmp:w { gset_nopar }           { Nx } { Npx }
\__cs_tmp:w { gset_protected }       { Nn } { Npn }
\__cs_tmp:w { gset_protected }       { Ne } { Npe }
\__cs_tmp:w { gset_protected }       { Nx } { Npx }
\__cs_tmp:w { gset_protected_nopar } { Nn } { Npn }
\__cs_tmp:w { gset_protected_nopar } { Ne } { Npe }
\__cs_tmp:w { gset_protected_nopar } { Nx } { Npx }
\__cs_tmp:w { new }                  { Nn } { Npn }
\__cs_tmp:w { new }                  { Ne } { Npe }
```

```
\__cs_tmp:w { new }                  { Nx } { Npx }
\__cs_tmp:w { new_nopar }            { Nn } { Npn }
\__cs_tmp:w { new_nopar }            { Ne } { Npe }
\__cs_tmp:w { new_nopar }            { Nx } { Npx }
\__cs_tmp:w { new_protected }        { Nn } { Npn }
\__cs_tmp:w { new_protected }        { Ne } { Npe }
\__cs_tmp:w { new_protected }        { Nx } { Npx }
\__cs_tmp:w { new_protected_nopar }  { Nn } { Npn }
\__cs_tmp:w { new_protected_nopar }  { Ne } { Npe }
\__cs_tmp:w { new_protected_nopar }  { Nx } { Npx }
\cs_set:Npn \__cs_tmp:w #1#2
  {
    \cs_new_protected:cpx { cs_ #1 : c #2 }
      {
        \exp_not:N \exp_args:Nc
        \exp_after:wN \exp_not:N \cs:w cs_ #1 : N #2 \cs_end:
      }
  }
\__cs_tmp:w { set }                  { n }
\__cs_tmp:w { set }                  { e }
\__cs_tmp:w { set }                  { x }
\__cs_tmp:w { set_nopar }            { n }
\__cs_tmp:w { set_nopar }            { e }
\__cs_tmp:w { set_nopar }            { x }
\__cs_tmp:w { set_protected }        { n }
\__cs_tmp:w { set_protected }        { e }
\__cs_tmp:w { set_protected }        { x }
\__cs_tmp:w { set_protected_nopar }  { n }
\__cs_tmp:w { set_protected_nopar }  { e }
\__cs_tmp:w { set_protected_nopar }  { x }
\__cs_tmp:w { gset }                 { n }
\__cs_tmp:w { gset }                 { e }
\__cs_tmp:w { gset }                 { x }
\__cs_tmp:w { gset_nopar }           { n }
\__cs_tmp:w { gset_nopar }           { e }
\__cs_tmp:w { gset_nopar }           { x }
\__cs_tmp:w { gset_protected }       { n }
\__cs_tmp:w { gset_protected }       { e }
\__cs_tmp:w { gset_protected }       { x }
\__cs_tmp:w { gset_protected_nopar } { n }
\__cs_tmp:w { gset_protected_nopar } { e }
\__cs_tmp:w { gset_protected_nopar } { x }
\__cs_tmp:w { new }                  { n }
\__cs_tmp:w { new }                  { e }
\__cs_tmp:w { new }                  { x }
\__cs_tmp:w { new_nopar }            { n }
```

```
\__cs_tmp:w { new_nopar }            { e }
\__cs_tmp:w { new_nopar }            { x }
\__cs_tmp:w { new_protected }        { n }
\__cs_tmp:w { new_protected }        { e }
\__cs_tmp:w { new_protected }        { x }
\__cs_tmp:w { new_protected_nopar }  { n }
\__cs_tmp:w { new_protected_nopar }  { e }
\__cs_tmp:w { new_protected_nopar }  { x }
\prg_new_conditional:Npnn \cs_if_eq:NN #1#2 { p , T , F , TF }
  {
    \if_meaning:w #1#2
      \prg_return_true: \else: \prg_return_false: \fi:
  }
\cs_new:Npn \cs_if_eq_p:cN { \exp_args:Nc  \cs_if_eq_p:NN }
\cs_new:Npn \cs_if_eq:cNTF { \exp_args:Nc  \cs_if_eq:NNTF }
\cs_new:Npn \cs_if_eq:cNT  { \exp_args:Nc  \cs_if_eq:NNT }
\cs_new:Npn \cs_if_eq:cNF  { \exp_args:Nc  \cs_if_eq:NNF }
\cs_new:Npn \cs_if_eq_p:Nc { \exp_args:NNc \cs_if_eq_p:NN }
\cs_new:Npn \cs_if_eq:NcTF { \exp_args:NNc \cs_if_eq:NNTF }
\cs_new:Npn \cs_if_eq:NcT  { \exp_args:NNc \cs_if_eq:NNT }
\cs_new:Npn \cs_if_eq:NcF  { \exp_args:NNc \cs_if_eq:NNF }
\cs_new:Npn \cs_if_eq_p:cc { \exp_args:Ncc \cs_if_eq_p:NN }
\cs_new:Npn \cs_if_eq:ccTF { \exp_args:Ncc \cs_if_eq:NNTF }
\cs_new:Npn \cs_if_eq:ccT  { \exp_args:Ncc \cs_if_eq:NNT }
\cs_new:Npn \cs_if_eq:ccF  { \exp_args:Ncc \cs_if_eq:NNF }
\cs_new_protected:Npn \__kernel_chk_defined:NT #1#2
  {
    \cs_if_exist:NTF #1
      {#2}
      {
        \msg_error:nne { kernel } { variable-not-defined }
          { \token_to_str:N #1 }
      }
  }
\cs_new_protected:Npn \__kernel_register_show:N
  { \__kernel_register_show_aux:NN \tl_show:n }
\cs_new_protected:Npn \__kernel_register_show:c
  { \exp_args:Nc \__kernel_register_show:N }
\cs_new_protected:Npn \__kernel_register_log:N
  { \__kernel_register_show_aux:NN \tl_log:n }
\cs_new_protected:Npn \__kernel_register_log:c
  { \exp_args:Nc \__kernel_register_log:N }
\cs_new_protected:Npn \__kernel_register_show_aux:NN #1#2
  {
    \__kernel_chk_defined:NT #2
      {
        \exp_args:No \__kernel_register_show_aux:nNN
          { \tex_the:D #2 } #2 #1
      }
  }
\cs_new_protected:Npn \__kernel_register_show_aux:nNN #1#2#3
  { \exp_args:No #3 { \token_to_str:N #2 = #1 } }
\cs_new_protected:Npn \cs_show:N { \__kernel_show:NN \tl_show:n }
\cs_new_protected:Npn \cs_show:c
  { \group_begin: \exp_args:NNc \group_end: \cs_show:N }
\cs_new_protected:Npn \cs_log:N { \__kernel_show:NN \tl_log:n }
\cs_new_protected:Npn \cs_log:c
  { \group_begin: \exp_args:NNc \group_end: \cs_log:N }
\cs_new_protected:Npn \__kernel_show:NN #1#2
  {
    \group_begin:
      \int_set:Nn \tex_escapechar:D { `\\ }
      \exp_args:NNe
    \group_end:
    #1 { \token_to_str:N #2 = \cs_meaning:N #2 }
  }
\cs_new_protected:Npn \group_show_list:
  { \__kernel_group_show:NN \use_none:n 1 }
\cs_new_protected:Npn \group_log_list:
  { \__kernel_group_show:NN \int_gzero:N 0 }
\cs_new_protected:Npn \__kernel_group_show:NN #1#2
  {
    \use:e
      {
        #1 \tex_interactionmode:D
        \int_set:Nn \tex_tracingonline:D  {#2}
        \int_set:Nn \tex_errorcontextlines:D { -1 }
        \exp_not:N \exp_after:wN \scan_stop:
        \tex_showgroups:D
        \int_gset:Nn \tex_interactionmode:D
          { \int_use:N \tex_interactionmode:D }
        \int_set:Nn \tex_tracingonline:D
          { \int_use:N \tex_tracingonline:D }
        \int_set:Nn \tex_errorcontextlines:D
          { \int_use:N \tex_errorcontextlines:D }
      }
  }
\use:e
  {
    \exp_not:n { \cs_new:Npn \__kernel_prefix_arg_replacement:wN #1 }
    \tl_to_str:n { macro : } \exp_not:n { #2 -> #3 \s__kernel_stop #4 }
  }
  { #4 {#1} {#2} {#3} }
\cs_new:Npn \cs_prefix_spec:N #1
  {
    \token_if_macro:NTF #1
      {
        \exp_after:wN \__kernel_prefix_arg_replacement:wN
          \token_to_meaning:N #1 \s__kernel_stop \use_i:nnn
      }
      { \scan_stop: }
  }
\cs_new:Npn \cs_parameter_spec:N #1
  {
    \token_if_macro:NTF #1
      {
        \exp_after:wN \__kernel_prefix_arg_replacement:wN
          \token_to_meaning:N #1 \s__kernel_stop \use_ii:nnn
      }
      { \scan_stop: }
  }
\cs_new:Npn \cs_replacement_spec:N #1
  {
    \token_if_macro:NTF #1
      {
        \exp_after:wN \__kernel_prefix_arg_replacement:wN
          \token_to_meaning:N #1 \s__kernel_stop \use_iii:nnn
      }
      { \scan_stop: }
  }
\cs_new:Npn \prg_do_nothing: { }
\cs_new_eq:NN \prg_break_point:Nn \use_ii:nn
\cs_new:Npn \prg_map_break:Nn #1#2#3 \prg_break_point:Nn #4#5
  {
    #5
    \if_meaning:w #1 #4
      \exp_after:wN \use_iii:nnn
    \fi:
    \prg_map_break:Nn #1 {#2}
  }
\cs_new_eq:NN \prg_break_point: \prg_do_nothing:
\cs_new:Npn \prg_break: #1 \prg_break_point: { }
\cs_new:Npn \prg_break:n #1#2 \prg_break_point: {#1}
\cs_new_protected:Npn \mode_leave_vertical:
  {
    \if_mode_vertical:
      \exp_after:wN \tex_indent:D
    \fi:
  }
%% File: l3expan.dtx
\cs_new:Npn \__exp_arg_next:nnn #1#2#3 { #2 \::: { #3 {#1} } }
\cs_new:Npn \__exp_arg_next:Nnn #1#2#3 { #2 \::: { #3 #1 } }
\cs_new:Npn \::: #1 {#1}
\cs_new:Npn \::n #1 \::: #2#3 { #1 \::: { #2 {#3} } }
\cs_new:Npn \::N #1 \::: #2#3 { #1 \::: {#2#3} }
\cs_new:Npn \::p #1 \::: #2#3# { #1 \::: {#2#3} }
\cs_new:Npn \::c #1 \::: #2#3
  { \exp_after:wN \__exp_arg_next:Nnn \cs:w #3 \cs_end: {#1} {#2} }
\cs_new:Npn \::o #1 \::: #2#3
  { \exp_after:wN \__exp_arg_next:nnn \exp_after:wN {#3} {#1} {#2} }
\cs_new:Npn \::e #1 \::: #2#3
  { \tex_expanded:D { \exp_not:n { #1 \::: } { \exp_not:n {#2} {#3} } } }
\cs_new:Npn \::f #1 \::: #2#3
  {
    \exp_after:wN \__exp_arg_next:nnn
      \exp_after:wN { \exp:w \exp_end_continue_f:w #3 }
      {#1} {#2}
  }
\use:nn { \cs_new_eq:NN \exp_stop_f: } { ~ }
\cs_new_protected:Npn \::x #1 \::: #2#3
  {
    \cs_set_nopar:Npe \l__exp_internal_tl
      { \exp_not:n { #1 \::: } { \exp_not:n {#2} {#3} } }
    \l__exp_internal_tl
  }
\cs_new:Npn \::V #1 \::: #2#3
  {
    \exp_after:wN \__exp_arg_next:nnn
      \exp_after:wN { \exp:w \__exp_eval_register:N #3 }
      {#1} {#2}
}
\cs_new:Npn \::v #1 \::: #2#3
  {
    \exp_after:wN \__exp_arg_next:nnn
      \exp_after:wN { \exp:w \__exp_eval_register:c {#3} }
      {#1} {#2}
  }
\cs_new:Npn \__exp_eval_register:N #1
  {
    \exp_after:wN \if_meaning:w \exp_not:N #1 #1
      \if_meaning:w \scan_stop: #1
        \__exp_eval_error_msg:w
      \fi:
    \else:
      \exp_after:wN \use_i_ii:nnn
    \fi:
    \exp_after:wN \exp_end: \tex_the:D #1
  }
\cs_new:Npn \__exp_eval_register:c #1
  { \exp_after:wN \__exp_eval_register:N \cs:w #1 \cs_end: }
\cs_new:Npn \__exp_eval_error_msg:w #1 \tex_the:D #2
  {
      \fi:
    \fi:
    \msg_expandable_error:nnn { kernel } { bad-variable } {#2}
    \exp_end:
  }
\cs_new:Npn \exp_args:NNc #1#2#3
  { \exp_after:wN #1 \exp_after:wN #2 \cs:w # 3\cs_end: }
\cs_new:Npn \exp_args:Ncc #1#2#3
  { \exp_after:wN #1 \cs:w #2 \exp_after:wN \cs_end: \cs:w #3 \cs_end: }
\cs_new:Npn \exp_args:Nccc #1#2#3#4
  {
    \exp_after:wN #1
      \cs:w #2 \exp_after:wN \cs_end:
      \cs:w #3 \exp_after:wN \cs_end:
      \cs:w #4 \cs_end:
  }
\cs_new:Npn \exp_args:No #1#2 { \exp_after:wN #1 \exp_after:wN {#2} }
\cs_new:Npn \exp_args:NNo #1#2#3
  { \exp_after:wN #1 \exp_after:wN #2 \exp_after:wN {#3} }
\cs_new:Npn \exp_args:NNNo #1#2#3#4
  { \exp_after:wN #1 \exp_after:wN#2 \exp_after:wN #3 \exp_after:wN {#4} }
\cs_new:Npn \exp_args:Ne #1#2
  { \exp_after:wN #1 \tex_expanded:D { {#2} } }
\cs_new:Npn \exp_args:Nf #1#2
  { \exp_after:wN #1 \exp_after:wN { \exp:w \exp_end_continue_f:w #2 } }
\cs_new:Npn \exp_args:Nv #1#2
  {
    \exp_after:wN #1 \exp_after:wN
      { \exp:w \__exp_eval_register:c {#2} }
  }
\cs_new:Npn \exp_args:NV #1#2
  {
    \exp_after:wN #1 \exp_after:wN
      { \exp:w \__exp_eval_register:N #2 }
  }
\cs_new:Npn \exp_args:NNV #1#2#3
  {
    \exp_after:wN #1
    \exp_after:wN #2
    \exp_after:wN { \exp:w \__exp_eval_register:N #3 }
  }
\cs_new:Npn \exp_args:NNv #1#2#3
  {
    \exp_after:wN #1
    \exp_after:wN #2
    \exp_after:wN { \exp:w \__exp_eval_register:c {#3} }
  }
\cs_new:Npn \exp_args:NNe #1#2#3
  {
    \exp_after:wN #1
    \exp_after:wN #2
    \tex_expanded:D { {#3} }
  }
\cs_new:Npn \exp_args:NNf #1#2#3
  {
    \exp_after:wN #1
    \exp_after:wN #2
    \exp_after:wN { \exp:w \exp_end_continue_f:w #3 }
  }
\cs_new:Npn \exp_args:Nco #1#2#3
  {
    \exp_after:wN #1
    \cs:w #2 \exp_after:wN \cs_end:
    \exp_after:wN {#3}
  }
\cs_new:Npn \exp_args:NcV #1#2#3
  {
    \exp_after:wN #1
    \cs:w #2 \exp_after:wN \cs_end:
    \exp_after:wN { \exp:w \__exp_eval_register:N #3 }
  }
\cs_new:Npn \exp_args:Ncv #1#2#3
  {
    \exp_after:wN #1
    \cs:w #2 \exp_after:wN \cs_end:
    \exp_after:wN { \exp:w \__exp_eval_register:c {#3} }
  }
\cs_new:Npn \exp_args:Ncf #1#2#3
  {
    \exp_after:wN #1
    \cs:w #2 \exp_after:wN \cs_end:
    \exp_after:wN { \exp:w \exp_end_continue_f:w #3 }
  }
\cs_new:Npn \exp_args:NVV #1#2#3
```

```
  {
    \exp_after:wN #1
    \exp_after:wN { \exp:w \exp_after:wN
      \__exp_eval_register:N \exp_after:wN #2 \exp_after:wN }
    \exp_after:wN { \exp:w \__exp_eval_register:N #3 }
  }
\cs_new:Npn \exp_args:NNNV #1#2#3#4
  {
    \exp_after:wN #1
    \exp_after:wN #2
    \exp_after:wN #3
    \exp_after:wN { \exp:w \__exp_eval_register:N #4 }
  }
\cs_new:Npn \exp_args:NNNv #1#2#3#4
  {
    \exp_after:wN #1
    \exp_after:wN #2
    \exp_after:wN #3
    \exp_after:wN { \exp:w \__exp_eval_register:c {#4} }
  }
\cs_new:Npn \exp_args:NNNe #1#2#3#4
  {
    \exp_after:wN #1
    \exp_after:wN #2
    \exp_after:wN #3
    \tex_expanded:D { {#4} }
  }
\cs_new:Npn \exp_args:NcNc #1#2#3#4
  {
    \exp_after:wN #1
    \cs:w #2 \exp_after:wN \cs_end:
    \exp_after:wN #3
    \cs:w #4 \cs_end:
  }
\cs_new:Npn \exp_args:NcNo #1#2#3#4
  {
    \exp_after:wN #1
    \cs:w #2 \exp_after:wN \cs_end:
    \exp_after:wN #3
    \exp_after:wN {#4}
  }
\cs_new:Npn \exp_args:Ncco #1#2#3#4
  {
    \exp_after:wN #1
    \cs:w #2 \exp_after:wN \cs_end:
    \cs:w #3 \exp_after:wN \cs_end:
```

```
    \exp_after:wN {#4}
  }
\cs_new_protected:Npn \exp_args:Nx #1#2
  { \use:x { \exp_not:N #1 {#2} } }
\cs_new:Npn \__exp_arg_last_unbraced:nn #1#2 { #2#1 }
\cs_new:Npn \::o_unbraced \::: #1#2
  { \exp_after:wN \__exp_arg_last_unbraced:nn \exp_after:wN {#2} {#1} }
\cs_new:Npn \::V_unbraced \::: #1#2
  {
    \exp_after:wN \__exp_arg_last_unbraced:nn
      \exp_after:wN { \exp:w \__exp_eval_register:N #2 } {#1}
  }
\cs_new:Npn \::v_unbraced \::: #1#2
  {
    \exp_after:wN \__exp_arg_last_unbraced:nn
      \exp_after:wN { \exp:w \__exp_eval_register:c {#2} } {#1}
  }
\cs_new:Npn \::e_unbraced \::: #1#2
  { \tex_expanded:D { \exp_not:n {#1} #2 } }
\cs_new:Npn \::f_unbraced \::: #1#2
  {
    \exp_after:wN \__exp_arg_last_unbraced:nn
      \exp_after:wN { \exp:w \exp_end_continue_f:w #2 } {#1}
  }
\cs_new_protected:Npn \::x_unbraced \::: #1#2
  {
    \cs_set_nopar:Npe \l__exp_internal_tl { \exp_not:n {#1} #2 }
    \l__exp_internal_tl
  }
\cs_new:Npn \exp_last_unbraced:No #1#2 { \exp_after:wN #1 #2 }
\cs_new:Npn \exp_last_unbraced:NV #1#2
  { \exp_after:wN #1 \exp:w \__exp_eval_register:N #2 }
\cs_new:Npn \exp_last_unbraced:Nv #1#2
  { \exp_after:wN #1 \exp:w \__exp_eval_register:c {#2} }
\cs_new:Npn \exp_last_unbraced:Ne #1#2
  { \exp_after:wN #1 \tex_expanded:D {#2} }
\cs_new:Npn \exp_last_unbraced:Nf #1#2
  { \exp_after:wN #1 \exp:w \exp_end_continue_f:w #2 }
\cs_new:Npn \exp_last_unbraced:NNo #1#2#3
  { \exp_after:wN #1 \exp_after:wN #2 #3 }
\cs_new:Npn \exp_last_unbraced:NNV #1#2#3
  {
    \exp_after:wN #1
    \exp_after:wN #2
    \exp:w \__exp_eval_register:N #3
  }
```

```
\cs_new:Npn \exp_last_unbraced:NNf #1#2#3
  {
    \exp_after:wN #1
    \exp_after:wN #2
    \exp:w \exp_end_continue_f:w #3
  }
\cs_new:Npn \exp_last_unbraced:Nco #1#2#3
  { \exp_after:wN #1 \cs:w #2 \exp_after:wN \cs_end: #3 }
\cs_new:Npn \exp_last_unbraced:NcV #1#2#3
  {
    \exp_after:wN #1
    \cs:w #2 \exp_after:wN \cs_end:
    \exp:w \__exp_eval_register:N #3
  }
\cs_new:Npn \exp_last_unbraced:NNNo #1#2#3#4
  { \exp_after:wN #1 \exp_after:wN #2 \exp_after:wN #3 #4 }
\cs_new:Npn \exp_last_unbraced:NNNV #1#2#3#4
  {
    \exp_after:wN #1
    \exp_after:wN #2
    \exp_after:wN #3
    \exp:w \__exp_eval_register:N #4
  }
\cs_new:Npn \exp_last_unbraced:NNNf #1#2#3#4
  {
    \exp_after:wN #1
    \exp_after:wN #2
    \exp_after:wN #3
    \exp:w \exp_end_continue_f:w #4
  }
\cs_new:Npn \exp_last_unbraced:Nno { \::n \::o_unbraced \::: }
\cs_new:Npn \exp_last_unbraced:Nnf { \::n \::f_unbraced \::: }
\cs_new:Npn \exp_last_unbraced:Noo { \::o \::o_unbraced \::: }
\cs_new:Npn \exp_last_unbraced:Nfo { \::f \::o_unbraced \::: }
\cs_new:Npn \exp_last_unbraced:NnNo { \::n \::N \::o_unbraced \::: }
\cs_new:Npn \exp_last_unbraced:NNNNo #1#2#3#4#5
  { \exp_after:wN #1 \exp_after:wN #2 \exp_after:wN #3 \exp_after:wN #4 #5 }
\cs_new:Npn \exp_last_unbraced:NNNNf #1#2#3#4#5
  {
    \exp_after:wN #1
    \exp_after:wN #2
    \exp_after:wN #3
    \exp_after:wN #4
    \exp:w \exp_end_continue_f:w #5
  }
\cs_new_protected:Npn \exp_last_unbraced:Nx { \::x_unbraced \::: }
```

```
\cs_new:Npn \exp_last_two_unbraced:Noo #1#2#3
  { \exp_after:wN \__exp_last_two_unbraced:noN \exp_after:wN {#3} {#2} #1 }
\cs_new:Npn \__exp_last_two_unbraced:noN #1#2#3
  { \exp_after:wN #3 #2 #1 }
\cs_new_eq:NN \__kernel_exp_not:w \tex_unexpanded:D
\cs_new:Npn \exp_not:c #1 { \exp_after:wN \exp_not:N \cs:w #1 \cs_end: }
\cs_new:Npn \exp_not:o #1 { \__kernel_exp_not:w \exp_after:wN {#1} }
\cs_new:Npn \exp_not:e #1
  { \__kernel_exp_not:w \tex_expanded:D { {#1} } }
\cs_new:Npn \exp_not:f #1
  { \__kernel_exp_not:w \exp_after:wN { \exp:w \exp_end_continue_f:w #1 } }
\cs_new:Npn \exp_not:V #1
  {
    \__kernel_exp_not:w \exp_after:wN
      { \exp:w \__exp_eval_register:N #1 }
  }
\cs_new:Npn \exp_not:v #1
  {
    \__kernel_exp_not:w \exp_after:wN
      { \exp:w \__exp_eval_register:c {#1} }
  }
\group_begin:
  \tex_catcode:D `\^^@ = 13
  \cs_new_protected:Npn \exp_end_continue_f:w { `^^@ }
  \if_cs_exist:N ^^@
  \else:
    \cs_new:Npn ^^@
      { \msg_expandable_error:nn { kernel } { bad-exp-end-f } }
  \fi:
  \cs_new:Npn \exp_end_continue_f:nw #1 { `^^@ #1 }
\group_end:
\cs_new_eq:NN \s__cs_mark \scan_stop:
\cs_new_eq:NN \s__cs_stop \scan_stop:
\cs_new:Npn \q__cs_recursion_stop { \q__cs_recursion_stop }
\cs_new:Npn \__cs_use_none_delimit_by_s_stop:w #1 \s__cs_stop { }
\cs_new:Npn \__cs_use_i_delimit_by_s_stop:nw #1 #2 \s__cs_stop {#1}
\cs_new:Npn \__cs_use_none_delimit_by_q_recursion_stop:w
  #1 \q__cs_recursion_stop { }
\cs_new_protected:Npn \cs_generate_variant:Nn #1#2
  {
    \__cs_generate_variant:N #1
    \use:e
      {
        \__cs_generate_variant:nnNN
          \cs_split_function:N #1
          \exp_not:N #1
          \tl_to_str:n {#2} ,
            \exp_not:N \scan_stop: ,
            \exp_not:N \q__cs_recursion_stop
      }
  }
\cs_new_protected:Npn \cs_generate_variant:cn
  { \exp_args:Nc \cs_generate_variant:Nn }
\cs_new_protected:Npe \__cs_generate_variant:N #1
  {
    \exp_not:N \exp_after:wN \exp_not:N \if_meaning:w
      \exp_not:N \exp_not:N #1 #1
      \cs_set_eq:NN \exp_not:N \__cs_tmp:w \cs_new_protected:Npe
    \exp_not:N \else:
      \exp_not:N \exp_after:wN \exp_not:N \__cs_generate_variant:ww
        \exp_not:N \token_to_meaning:N #1 \tl_to_str:n { ma }
          \s__cs_mark
        \s__cs_mark \cs_new_protected:Npe
        \tl_to_str:n { pr }
        \s__cs_mark \cs_new:Npe
        \s__cs_stop
    \exp_not:N \fi:
  }
\exp_last_unbraced:NNNNo
  \cs_new_protected:Npn \__cs_generate_variant:ww
    #1 { \tl_to_str:n { ma } } #2 \s__cs_mark
    { \__cs_generate_variant:wwNw #1 }
\exp_last_unbraced:NNNNo
  \cs_new_protected:Npn \__cs_generate_variant:wwNw
    #1 { \tl_to_str:n { pr } } #2 \s__cs_mark #3 #4 \s__cs_stop
    { \cs_set_eq:NN \__cs_tmp:w #3 }
\cs_new_protected:Npn \__cs_generate_variant:nnNN #1#2#3#4
  {
    \if_meaning:w \c_false_bool #3
      \msg_error:nne { kernel } { missing-colon }
        { \token_to_str:c {#1} }
      \exp_after:wN \__cs_use_none_delimit_by_q_recursion_stop:w
    \fi:
    \__cs_generate_variant:Nnnw #4 {#1}{#2}
  }
\cs_new_protected:Npn \__cs_generate_variant:Nnnw #1#2#3#4 ,
  {
    \if_meaning:w \scan_stop: #4
      \exp_after:wN \__cs_use_none_delimit_by_q_recursion_stop:w
    \fi:
    \use:e
      {
        \exp_not:N \__cs_generate_variant:wwNN
        \__cs_generate_variant_loop:nNwN { }
          #4
          \__cs_generate_variant_loop_end:nwwwNNnn
          \s__cs_mark
          #3 ~
          { ~ { } \fi: \__cs_generate_variant_loop_long:wNNnn } ~
          { }
          \s__cs_stop
        \exp_not:N #1 {#2} {#4}
      }
    \__cs_generate_variant:Nnnw #1 {#2} {#3}
  }
\cs_new:Npn \__cs_generate_variant_loop:nNwN #1#2#3 \s__cs_mark #4
  {
    \if:w #2 #4
      \exp_after:wN \__cs_generate_variant_loop_same:w
    \else:
      \if:w #4 \__cs_generate_variant_loop_base:N #2 \else:
        \if:w 0
          \if:w N #4 \else: \if:w n #4 \else: 1 \fi: \fi:
          \if:w \scan_stop: \__cs_generate_variant_loop_base:N #2 1 \fi:
          0
          \__cs_generate_variant_loop_special:NNwNNnn #4#2
        \else:
          \__cs_generate_variant_loop_invalid:NNwNNnn #4#2
        \fi:
      \fi:
    \fi:
    #1
    \prg_do_nothing:
    #2
    \__cs_generate_variant_loop:nNwN { } #3 \s__cs_mark
  }
\cs_new:Npn \__cs_generate_variant_loop_base:N #1
  {
    \if:w c #1 N \else:
      \if:w o #1 n \else:
        \if:w V #1 n \else:
          \if:w v #1 n \else:
            \if:w f #1 n \else:
              \if:w e #1 n \else:
                \if:w x #1 n \else:
                  \if:w n #1 n \else:
                    \if:w N #1 N \else:
                      \scan_stop:
```

```
                    \fi:
                  \fi:
                \fi:
              \fi:
            \fi:
          \fi:
        \fi:
      \fi:
    \fi:
  }
\cs_new:Npn \__cs_generate_variant_loop_same:w
    #1 \prg_do_nothing: #2#3#4
  { #3 { #1 \__cs_generate_variant_same:N #2 } }
\cs_new:Npn \__cs_generate_variant_loop_end:nwwwNNnn
    #1#2 \s__cs_mark #3 ~ #4 \s__cs_stop #5#6#7#8
  {
    \scan_stop: \scan_stop: \fi:
    \s__cs_mark \s__cs_stop
    \exp_not:N #6
    \exp_not:c { #7 : #8 #1 #3 }
  }
\cs_new:Npn \__cs_generate_variant_loop_long:wNNnn #1 \s__cs_stop #2#3#4#5
  {
    \exp_not:n
      {
        \s__cs_mark
        \msg_error:nnee { kernel } { variant-too-long }
          {#5} { \token_to_str:N #3 }
        \use_none:nnn
        \s__cs_stop
        #3
        #3
      }
  }
\cs_new:Npn \__cs_generate_variant_loop_invalid:NNwNNnn
    #1#2 \fi: \fi: \fi: #3 \s__cs_stop #4#5#6#7
  {
    \fi: \fi: \fi:
    \exp_not:n
      {
        \s__cs_mark
        \msg_error:nneeee { kernel } { invalid-variant }
          {#7} { \token_to_str:N #5 } {#1} {#2}
        \use_none:nnn
        \s__cs_stop
        #5
        #5
      }
  }
\cs_new:Npn \__cs_generate_variant_loop_special:NNwNNnn
  #1#2#3 \s__cs_stop #4#5#6#7
  {
    #3 \s__cs_stop #4 #5 {#6} {#7}
    \exp_not:n
      {
        \msg_error:nneeee
          { kernel } { deprecated-variant }
          {#7} { \token_to_str:N #5 } {#1} {#2}
      }
  }
\cs_new:Npn \__cs_generate_variant_same:N #1
  {
    \if:w N #1 #1 \else:
      \if:w p #1 #1 \else:
        \token_to_str:N n
        \if:w n #1 \else:
          \__cs_generate_variant_loop_special:NNwNNnn #1#1
        \fi:
      \fi:
    \fi:
  }
\cs_new_protected:Npn \__cs_generate_variant:wwNN
    #1 \s__cs_mark #2 \s__cs_stop #3#4
  {
    #2
    \cs_if_free:NT #4
      {
        \group_begin:
          \__cs_generate_internal_variant:n {#1}
          \__cs_tmp:w #4 { \exp_not:c { exp_args:N #1 } \exp_not:N #3 }
        \group_end:
      }
  }
\cs_new_protected:Npe \__cs_generate_internal_variant:n #1
  {
    \exp_not:N \__cs_generate_internal_variant:wwnNwn
      #1 \s__cs_mark
        { \cs_set_eq:NN \exp_not:N \__cs_tmp:w \cs_new_protected:Npe }
        \cs_new_protected:cpn
        \use:x
      \token_to_str:N x \s__cs_mark
        { }
        \cs_new:cpn
        \exp_not:N \tex_expanded:D
    \s__cs_stop
      {#1}
  }
\exp_last_unbraced:NNNNo
  \cs_new_protected:Npn \__cs_generate_internal_variant:wwnNwn #1
    { \token_to_str:N x } #2 \s__cs_mark #3#4#5#6 \s__cs_stop #7
  {
    #3
    \cs_if_free:cT { exp_args:N #7 }
      { \__cs_generate_internal_variant:NNn #4 #5 {#7} }
  }
\cs_set_protected:Npn \__cs_tmp:w #1
  {
    \cs_new_protected:Npn \__cs_generate_internal_variant:NNn ##1##2##3
      {
        \if_catcode:w X \use_none:nnnnnnnn ##3
            \prg_do_nothing: \prg_do_nothing: \prg_do_nothing:
            \prg_do_nothing: \prg_do_nothing: \prg_do_nothing:
            \prg_do_nothing: \prg_do_nothing: X
          \exp_after:wN \__cs_generate_internal_test:Nw \exp_after:wN ##2
        \else:
          \exp_after:wN \__cs_generate_internal_test_aux:w \exp_after:wN #1
        \fi:
        ##3
        \s__cs_mark
        {
          \use:e
            {
              ##1 { exp_args:N ##3 }
                { \__cs_generate_internal_variant_loop:n ##3 { : \use_i:nn } }
            }
        }
        #1
        \s__cs_mark
        { \exp_not:n { \__cs_generate_internal_one_go:NNn ##1 ##2 {##3} } }
        \s__cs_stop
      }
    \cs_new_protected:Npn \__cs_generate_internal_test_aux:w
        ##1 #1 ##2 \s__cs_mark ##3 ##4 \s__cs_stop {##3}
    \cs_new_eq:NN \__cs_generate_internal_test:Nw
      \__cs_generate_internal_test_aux:w
  }
\exp_args:No \__cs_tmp:w { \token_to_str:N p }
\cs_new_protected:Npn \__cs_generate_internal_one_go:NNn #1#2#3
```

```
  {
    \__cs_generate_internal_loop:nwnnw
      { \exp_not:N ##1 } 1 . { } { }
      #3 { ? \__cs_generate_internal_end:w } X ;
      23456789 { ? \__cs_generate_internal_long:w } ;
    #1 #2 {#3}
  }
\cs_new_protected:Npn \__cs_generate_internal_loop:nwnnw #1#2 . #3#4#5#6 ; #7
  {
    \use_none:n #5
    \use_none:n #7
    \cs_if_exist_use:cF { __cs_generate_internal_#5:NN }
      { \__cs_generate_internal_other:NN }
        #5 #7
    #7 .
    { #3 #1 } { #4 ## #2 }
    #6 ;
  }
\cs_new_protected:Npn \__cs_generate_internal_N:NN #1#2
  { \__cs_generate_internal_loop:nwnnw { \exp_not:N ###2 } }
\cs_new_protected:Npn \__cs_generate_internal_c:NN #1#2
  { \exp_args:No \__cs_generate_internal_loop:nwnnw { \exp_not:c {###2} } }
\cs_new_protected:Npn \__cs_generate_internal_n:NN #1#2
  { \__cs_generate_internal_loop:nwnnw { { \exp_not:n {###2} } } }
\cs_new_protected:Npn \__cs_generate_internal_x:NN #1#2
  { \__cs_generate_internal_loop:nwnnw { {###2} } }
\cs_new_protected:Npn \__cs_generate_internal_other:NN #1#2
  {
    \exp_args:No \__cs_generate_internal_loop:nwnnw
      {
        \exp_after:wN
        {
          \exp:w \exp_args:NNc \exp_after:wN \exp_end:
          { exp_not:#1 } {###2}
        }
      }
  }
\cs_new_protected:Npn \__cs_generate_internal_end:w #1 . #2#3#4 ; #5 ; #6#7#8
  { #6 { exp_args:N #8 } #3 { #7 {#2} } }
\cs_new_protected:Npn \__cs_generate_internal_long:w #1 N #2#3 . #4#5#6#
  {
    \exp_args:Nx \__cs_generate_internal_long:nnnNNn
      { \__cs_generate_internal_variant_loop:n #2 #6 { : \use_i:nn } }
      {#4} {#5}
  }
\cs_new:Npn \__cs_generate_internal_long:nnnNNn #1#2#3#4 ; ; #5#6#7
```

```
  { #5 { exp_args:N #7 } #3 { #6 { \exp_not:n {#1} {#2} } } }
\cs_new:Npn \__cs_generate_internal_variant_loop:n #1
  {
    \exp_after:wN \exp_not:N \cs:w :: #1 \cs_end:
    \__cs_generate_internal_variant_loop:n
  }
\cs_new_protected:Npn \prg_generate_conditional_variant:Nnn #1
  {
    \use:e
      {
        \__cs_generate_variant:nnNnn
          \cs_split_function:N #1
      }
  }
\cs_new_protected:Npn \__cs_generate_variant:nnNnn #1#2#3#4#5
  {
    \if_meaning:w \c_false_bool #3
      \msg_error:nne { kernel } { missing-colon }
        { \token_to_str:c {#1} }
      \__cs_use_i_delimit_by_s_stop:nw
    \fi:
    \exp_after:wN \__cs_generate_variant:w
    \tl_to_str:n {#5} , \scan_stop: , \q__cs_recursion_stop
    \__cs_use_none_delimit_by_s_stop:w \s__cs_mark {#1} {#2} {#4} \s__cs_stop
  }
\cs_new_protected:Npn \__cs_generate_variant:w
    #1 , #2 \s__cs_mark #3#4#5
  {
    \if_meaning:w \scan_stop: #1 \scan_stop:
      \if_meaning:w \q__cs_nil #1 \q__cs_nil
        \use_i:nnn
      \fi:
      \exp_after:wN \__cs_use_none_delimit_by_q_recursion_stop:w
    \else:
      \cs_if_exist_use:cTF { __cs_generate_variant_#1_form:nnn }
        { {#3} {#4} {#5} }
        {
          \msg_error:nnee
            { kernel } { conditional-form-unknown }
            {#1} { \token_to_str:c { #3 : #4 } }
        }
    \fi:
    \__cs_generate_variant:w #2 \s__cs_mark {#3} {#4} {#5}
  }
\cs_new_protected:Npn \__cs_generate_variant_p_form:nnn #1#2
  { \__cs_generate_variant_check:nn { #1 _p : #2 } }
```

```
\cs_new_protected:Npn \__cs_generate_variant_T_form:nnn #1#2
  { \__cs_generate_variant_check:nn { #1 : #2 T } }
\cs_new_protected:Npn \__cs_generate_variant_F_form:nnn #1#2
  { \__cs_generate_variant_check:nn { #1 : #2 F } }
\cs_new_protected:Npn \__cs_generate_variant_TF_form:nnn #1#2
  { \__cs_generate_variant_check:nn { #1 : #2 TF } }
\cs_new_protected:Npn \__cs_generate_variant_check:nn #1#2
  {
    \cs_if_exist:cTF {#1}
      { \cs_generate_variant:cn {#1} {#2} }
      {
        \msg_error:nne
          { kernel } { conditional-base-undefined }
            { \token_to_str:c {#1} }
      }
  }
\cs_new_protected:Npn \exp_args_generate:n #1
  {
    \exp_args:No \clist_map_inline:nn { \tl_to_str:n {#1} }
      {
        \str_map_inline:nn {##1}
          {
            \str_if_in:nnF { NnpcofeVvx } {####1}
              {
                \msg_error:nnnn { kernel } { invalid-exp-args }
                  {####1} {##1}
                \str_map_break:n { \use_none:nn }
              }
          }
        \__cs_generate_internal_variant:n {##1}
      }
  }
\cs_set_protected:Npn \__cs_tmp:w #1
  {
    \group_begin:
      \exp_args:No \__cs_generate_internal_variant:n
        { \tl_to_str:n {#1} }
    \group_end:
  }
\__cs_tmp:w { nc }
\__cs_tmp:w { no }
\__cs_tmp:w { nV }
\__cs_tmp:w { nv }
\__cs_tmp:w { ne }
\__cs_tmp:w { nf }
\__cs_tmp:w { oc }
```

```
\__cs_tmp:w { oo }
\__cs_tmp:w { of }
\__cs_tmp:w { Vo }
\__cs_tmp:w { fo }
\__cs_tmp:w { ff }
\__cs_tmp:w { ee }
\__cs_tmp:w { Nx }
\__cs_tmp:w { cx }
\__cs_tmp:w { nx }
\__cs_tmp:w { ox }
\__cs_tmp:w { xo }
\__cs_tmp:w { xx }
\__cs_tmp:w { Ncf }
\__cs_tmp:w { Nno }
\__cs_tmp:w { NnV }
\__cs_tmp:w { Noo }
\__cs_tmp:w { NVV }
\__cs_tmp:w { cno }
\__cs_tmp:w { cnV }
\__cs_tmp:w { coo }
\__cs_tmp:w { cVV }
\__cs_tmp:w { nnc }
\__cs_tmp:w { nno }
\__cs_tmp:w { nnf }
\__cs_tmp:w { nff }
\__cs_tmp:w { ooo }
\__cs_tmp:w { oof }
\__cs_tmp:w { ffo }
\__cs_tmp:w { eee }
\__cs_tmp:w { NNx }
\__cs_tmp:w { Nnx }
\__cs_tmp:w { Nox }
\__cs_tmp:w { nnx }
\__cs_tmp:w { nox }
\__cs_tmp:w { ccx }
\__cs_tmp:w { cnx }
\__cs_tmp:w { oox }
\cs_generate_variant:Nn \cs_generate_from_arg_count:NNnn { NNno }
\cs_generate_variant:Nn \cs_replacement_spec:N { c }
%% File: l3quark.dtx
\cs_new_protected:Npn \quark_new:N #1
  {
    \__kernel_chk_if_free_cs:N #1
    \cs_gset_nopar:Npn #1 {#1}
  }
\quark_new:N \q_nil
```

```
\quark_new:N \q_mark
\quark_new:N \q_no_value
\quark_new:N \q_stop
\quark_new:N \q_recursion_tail
\quark_new:N \q_recursion_stop
\cs_new_eq:NN \s__quark \scan_stop:
\quark_new:N \q__quark_nil
\cs_new:Npn \quark_if_recursion_tail_stop:N #1
  {
    \if_meaning:w \q_recursion_tail #1
      \exp_after:wN \use_none_delimit_by_q_recursion_stop:w
    \fi:
  }
\cs_new:Npn \quark_if_recursion_tail_stop_do:Nn #1
  {
    \if_meaning:w \q_recursion_tail #1
      \exp_after:wN \use_i_delimit_by_q_recursion_stop:nw
    \else:
      \exp_after:wN \use_none:n
    \fi:
  }
\cs_new:Npn \quark_if_recursion_tail_stop:n #1
  {
    \tl_if_empty:oTF
      { \__quark_if_recursion_tail:w {} #1 {} ?! \q_recursion_tail ??! }
      { \use_none_delimit_by_q_recursion_stop:w }
      { }
  }
\cs_new:Npn \quark_if_recursion_tail_stop_do:nn #1
  {
    \tl_if_empty:oTF
      { \__quark_if_recursion_tail:w {} #1 {} ?! \q_recursion_tail ??! }
      { \use_i_delimit_by_q_recursion_stop:nw }
      { \use_none:n }
  }
\cs_new:Npn \__quark_if_recursion_tail:w
    #1 \q_recursion_tail #2 ? #3 ?! { #1 #2 }
\cs_generate_variant:Nn \quark_if_recursion_tail_stop:n { o }
\cs_generate_variant:Nn \quark_if_recursion_tail_stop_do:nn { o }
\cs_new:Npn \quark_if_recursion_tail_break:NN #1#2
  {
    \if_meaning:w \q_recursion_tail #1
      \exp_after:wN #2
    \fi:
  }
\cs_new:Npn \quark_if_recursion_tail_break:nN #1#2
```

```
  {
    \tl_if_empty:oT
      { \__quark_if_recursion_tail:w {} #1 {} ?! \q_recursion_tail ??! }
      {#2}
  }
\prg_new_conditional:Npnn \quark_if_nil:N #1 { p, T , F , TF }
  {
    \if_meaning:w \q_nil #1
      \prg_return_true:
    \else:
      \prg_return_false:
    \fi:
  }
\prg_new_conditional:Npnn \quark_if_no_value:N #1 { p, T , F , TF }
  {
    \if_meaning:w \q_no_value #1
      \prg_return_true:
    \else:
      \prg_return_false:
    \fi:
  }
\prg_generate_conditional_variant:Nnn \quark_if_no_value:N
  { c } { p , T , F , TF }
\prg_new_conditional:Npnn \quark_if_nil:n #1 { p, T , F , TF }
  {
    \__quark_if_empty_if:o
      { \__quark_if_nil:w {} #1 {} ? ! \q_nil ? ? ! }
      \prg_return_true:
    \else:
      \prg_return_false:
    \fi:
  }
\cs_new:Npn \__quark_if_nil:w #1 \q_nil #2 ? #3 ? ! { #1 #2 }
\prg_new_conditional:Npnn \quark_if_no_value:n #1 { p, T , F , TF }
  {
    \__quark_if_empty_if:o
      { \__quark_if_no_value:w {} #1 {} ? ! \q_no_value ? ? ! }
      \prg_return_true:
    \else:
      \prg_return_false:
    \fi:
  }
\cs_new:Npn \__quark_if_no_value:w #1 \q_no_value #2 ? #3 ? ! { #1 #2 }
\prg_generate_conditional_variant:Nnn \quark_if_nil:n
  { V , o } { p , TF , T , F }
\cs_new:Npn \__quark_if_empty_if:o #1
```

```
  {
    \exp_after:wN \if_meaning:w \exp_after:wN \q_nil
      \__kernel_tl_to_str:w \exp_after:wN {#1} \q_nil
  }
\cs_new_protected:Npn \__kernel_quark_new_test:N #1
  { \__quark_new_test_aux:Ne #1 { \__quark_module_name:N #1 } }
\cs_new_protected:Npn \__quark_new_test_aux:Nn #1 #2
  {
    \if_meaning:w \q_nil #2 \q_nil
      \msg_error:nne { quark } { invalid-function }
        { \token_to_str:N #1 }
    \else:
      \__quark_new_test:Nccn #1
        { q__#2_recursion_tail } { q__#2_recursion_stop } { __#2 }
    \fi:
  }
\cs_generate_variant:Nn \__quark_new_test_aux:Nn { Ne }
\cs_new_protected:Npn \__quark_new_test:NNNn #1
  {
    \exp_last_unbraced:Nf \__quark_new_test_aux:nnNNnnnn
      { \cs_split_function:N #1 }
      #1 { test }
  }
\cs_generate_variant:Nn \__quark_new_test:NNNn { Ncc }
\cs_new_protected:Npn \__kernel_quark_new_conditional:Nn #1
  {
    \__quark_new_conditional:Neen #1
      { \__quark_quark_conditional_name:N #1 }
      { \__quark_module_name:N #1 }
  }
\cs_new_protected:Npn \__quark_new_conditional:Nnnn #1#2#3#4
  {
    \if_meaning:w \q_nil #2 \q_nil
      \msg_error:nne { quark } { invalid-function }
        { \token_to_str:N #1 }
    \else:
      \if_meaning:w \q_nil #3 \q_nil
        \msg_error:nne { quark } { invalid-function }
          { \token_to_str:N #1 }
      \else:
        \exp_last_unbraced:Nf \__quark_new_test_aux:nnNNnnnn
          { \cs_split_function:N #1 }
          #1 { conditional }
          {#2} {#3} {#4}
      \fi:
    \fi:
  }
\cs_generate_variant:Nn \__quark_new_conditional:Nnnn { Nee }
\cs_new_protected:Npn \__quark_new_test_aux:nnNNnnnn #1 #2 #3 #4 #5
  {
    \cs_if_exist_use:cTF { __quark_new_#5_#2:Nnnn } { #4 }
      {
        \msg_error:nnee { quark } { invalid-function }
          { \token_to_str:N #4 } {#2}
        \use_none:nnn
      }
  }
\cs_new_protected:Npn \__quark_new_test_n:Nnnn #1 #2 #3 #4
  {
    \__quark_new_test_aux_do:nNNnnnnNNn {#4} #2 #3 { none } { } { } { }
      \__quark_new_test_define_tl:nNnNNn #1 { }
  }
\cs_new_protected:Npn \__quark_new_test_nn:Nnnn #1 #2 #3 #4
  {
    \__quark_new_test_aux_do:nNNnnnnNNn {#4} #2 #3 { i } { n } {##1} {##2}
      \__quark_new_test_define_tl:nNnNNn #1 { \use_none:n }
  }
\cs_new_protected:Npn \__quark_new_test_nN:Nnnn #1 #2 #3 #4
  {
    \__quark_new_test_aux_do:nNNnnnnNNn {#4} #2 #3 { i } { n } {##1} {##2}
      \__quark_new_test_define_break_tl:nNNNNn #1 { }
  }
\cs_new_protected:Npn \__quark_new_test_N:Nnnn #1 #2 #3 #4
  {
    \__quark_new_test_aux_do:nNNnnnnNNn {#4} #2 #3 { none } { } { } { }
      \__quark_new_test_define_ifx:nNnNNn #1 { }
  }
\cs_new_protected:Npn \__quark_new_test_Nn:Nnnn #1 #2 #3 #4
  {
    \__quark_new_test_aux_do:nNNnnnnNNn {#4} #2 #3 { i } { n } {##1} {##2}
      \__quark_new_test_define_ifx:nNnNNn #1
      { \else: \exp_after:wN \use_none:n }
  }
\cs_new_protected:Npn \__quark_new_test_NN:Nnnn #1 #2 #3 #4
  {
    \__quark_new_test_aux_do:nNNnnnnNNn {#4} #2 #3 { i } { n } {##1} {##2}
      \__quark_new_test_define_break_ifx:nNNNNn #1 { }
  }
\cs_new_protected:Npn \__quark_new_test_aux_do:nNNnnnnNNn #1 #2 #3 #4 #5
  {
    \exp_args:Ncc \__quark_test_define_aux:NNNNnnNNn
      { #1 _quark_recursion_tail:w }
      { #1 _use_ #4 _delimit_by_q_recursion_stop: #5 w }
      #2 #3
  }
\cs_new_protected:Npn \__quark_test_define_aux:NNNNnnNNn #1 #2 #3 #4 #5 #6 #7
  {
    \cs_gset:Npn #1  ##1 #3 ##2 ? ##3 ?! { ##1 ##2 }
    \cs_gset:Npn #2  ##1 #6 #4 {#5}
    #7 {##1} #1 #2 #3
  }
\cs_new_protected:Npn \__quark_new_test_define_tl:nNnNNn #1 #2 #3 #4 #5 #6
  {
    \cs_new:Npn #5 #1
      {
        \tl_if_empty:oTF
          { #2 {} ##1 {} ?! #4 ??! }
          {#3} {#6}
      }
  }
\cs_new_protected:Npn \__quark_new_test_define_ifx:nNnNNn #1 #2 #3 #4 #5 #6
  {
    \cs_new:Npn #5 #1
      {
        \if_meaning:w #4 ##1
          \exp_after:wN #3
          #6
        \fi:
      }
  }
\cs_new_protected:Npn \__quark_new_test_define_break_tl:nNNNNn #1 #2 #3
  { \__quark_new_test_define_tl:nNnNNn {##1##2} #2 {##2} }
\cs_new_protected:Npn \__quark_new_test_define_break_ifx:nNNNNn #1 #2 #3
  { \__quark_new_test_define_ifx:nNnNNn {##1##2} #2 {##2} }
\cs_new_protected:Npn \__quark_new_conditional_n:Nnnn #1 #2 #3
  {
    \exp_args:Ncc \__quark_new_conditional_n_aux:NNNn
      { __ #3 _if_quark_ #2 :w } { q__ #3 _ #2 } #1
  }
\cs_new_protected:Npn \__quark_new_conditional_N:Nnnn #1 #2 #3
  {
    \exp_args:NNc \__quark_new_conditional_N_aux:NNNn
      \prg_do_nothing: { q__ #3 _ #2 } #1
  }
\cs_new_protected:Npn \__quark_new_conditional_n_aux:NNNn #1 #2 #3 #4
  {
    \cs_gset:Npn #1 ##1 #2 ##2 ? ##3 ?! { ##1##2 }
    \prg_new_conditional:Npnn #3 ##1 {#4}
```

```
      {
        \__quark_if_empty_if:o { #1 {} ##1 {} ?! #2 ??! }
          \prg_return_true:
        \else:
          \prg_return_false:
        \fi:
      }
  }
\cs_new_protected:Npn \__quark_new_conditional_N_aux:NNNn #1 #2 #3 #4
  {
    \prg_new_conditional:Npnn #3 ##1 {#4}
      {
        \if_meaning:w #2 ##1
          \prg_return_true:
        \else:
          \prg_return_false:
        \fi:
      }
  }
\cs_set:Npn \__quark_tmp:w #1#2
  {
    \cs_new:Npn \__quark_module_name:N ##1
      {
        \exp_last_unbraced:Nf \__quark_module_name:w
          { \cs_to_str:N ##1 } #1 \s__quark
      }
    \cs_new:Npn \__quark_module_name:w ##1 #1 ##2 \s__quark
      { \__quark_module_name_loop:w ##1 #2 \use_none:n { } #2 \s__quark }
    \cs_new:Npn \__quark_module_name_loop:w ##1 #2
      {
        \use_i_ii:nnn \if_meaning:w \prg_do_nothing:
            ##1 \prg_do_nothing: \prg_do_nothing:
          \exp_after:wN \__quark_module_name_loop:w
        \else:
          \__quark_module_name_end:w ##1
        \fi:
      }
    \cs_new:Npn \__quark_module_name_end:w
      ##1 \fi: ##2 \s__quark { \fi: ##1 }
  }
\exp_after:wN \__quark_tmp:w \tl_to_str:n { : _ }
\cs_set:Npn \__quark_tmp:w #1 #2 \s__quark
  {
    \cs_new:Npn \__quark_quark_conditional_name:N ##1
      {
        \exp_last_unbraced:Nf \__quark_quark_conditional_name:w
```

```
          { \cs_to_str:N ##1 } #1 #2 #1 \s__quark
      }
    \cs_new:Npn \__quark_quark_conditional_name:w
      ##1 #2 ##2 #1 ##3 \s__quark {##2}
  }
\exp_after:wN \__quark_tmp:w \tl_to_str:n { : _quark_if_ } \s__quark
\cs_new_protected:Npn \scan_new:N #1
  {
    \tl_if_in:NnTF \g__scan_marks_tl { #1 }
      {
        \msg_error:nne { scanmark } { already-defined }
          { \token_to_str:N #1 }
      }
      {
        \tl_gput_right:Nn \g__scan_marks_tl {#1}
        \cs_new_eq:NN #1 \scan_stop:
      }
  }
\cs_new_eq:NN \s_stop \scan_stop:
\cs_gset_nopar:Npn \g__scan_marks_tl
  {
    \s_stop
    \s__quark
    \s__cs_mark
    \s__cs_stop
  }
\cs_new:Npn \use_none_delimit_by_s_stop:w #1 \s_stop { }
%% File: l3tl.dtx
\cs_new_eq:NN \__kernel_tl_set:Nx  \cs_set_nopar:Npe
\cs_new_eq:NN \__kernel_tl_gset:Nx \cs_gset_nopar:Npe
\cs_new_protected:Npn \tl_new:N #1
  {
    \__kernel_chk_if_free_cs:N #1
    \cs_gset_eq:NN #1 \c_empty_tl
  }
\cs_generate_variant:Nn \tl_new:N { c }
\cs_new_protected:Npn \tl_const:Nn #1#2
  {
    \__kernel_chk_if_free_cs:N #1
    \cs_gset_nopar:Npe #1 { \__kernel_exp_not:w {#2} }
  }
\cs_generate_variant:Nn \tl_const:Nn { NV , Ne , c , cV , ce }
\cs_generate_variant:Nn \tl_const:Nn { Nx , cx }
\cs_new_protected:Npn \tl_clear:N  #1
  { \tex_let:D #1 = ~ \c_empty_tl }
\cs_new_protected:Npn \tl_gclear:N #1
```

```
  { \tex_global:D \tex_let:D #1 ~ \c_empty_tl }
\cs_generate_variant:Nn \tl_clear:N  { c }
\cs_generate_variant:Nn \tl_gclear:N { c }
\cs_new_protected:Npn \tl_clear_new:N  #1
  { \tl_if_exist:NTF #1 { \tl_clear:N #1 } { \tl_new:N #1 } }
\cs_new_protected:Npn \tl_gclear_new:N #1
  { \tl_if_exist:NTF #1 { \tl_gclear:N #1 } { \tl_new:N #1 } }
\cs_generate_variant:Nn \tl_clear_new:N  { c }
\cs_generate_variant:Nn \tl_gclear_new:N { c }
\cs_new_protected:Npn \tl_set_eq:NN  #1#2
  { \tex_let:D #1 = ~ #2 }
\cs_new_protected:Npn \tl_gset_eq:NN #1#2
  { \tex_global:D \tex_let:D #1 = ~ #2 }
\cs_generate_variant:Nn \tl_set_eq:NN { cN, Nc, cc }
\cs_generate_variant:Nn \tl_gset_eq:NN { cN, Nc, cc }
\cs_new_protected:Npn \tl_concat:NNN #1#2#3
  {
    \__kernel_tl_set:Nx #1
      {
        \__kernel_exp_not:w \exp_after:wN {#2}
        \__kernel_exp_not:w \exp_after:wN {#3}
      }
  }
\cs_new_protected:Npn \tl_gconcat:NNN #1#2#3
  {
    \__kernel_tl_gset:Nx #1
      {
        \__kernel_exp_not:w \exp_after:wN {#2}
        \__kernel_exp_not:w \exp_after:wN {#3}
      }
  }
\cs_generate_variant:Nn \tl_concat:NNN  { ccc }
\cs_generate_variant:Nn \tl_gconcat:NNN { ccc }
\prg_new_eq_conditional:NNn \tl_if_exist:N \cs_if_exist:N { TF , T , F , p }
\prg_new_eq_conditional:NNn \tl_if_exist:c \cs_if_exist:c { TF , T , F , p }
\tl_const:Nn \c_empty_tl { }
\group_begin:
\tex_catcode:D `- = 11 ~
\tl_const:Ne \c_novalue_tl { - NoValue \token_to_str:N - }
\group_end:
\tl_const:Nn \c_space_tl { ~ }
\cs_new_protected:Npn \tl_set:Nn #1#2
  { \__kernel_tl_set:Nx #1 { \__kernel_exp_not:w {#2} } }
\cs_new_protected:Npn \tl_set:No #1#2
  { \__kernel_tl_set:Nx #1 { \__kernel_exp_not:w \exp_after:wN {#2} } }
\cs_new_protected:Npn \tl_gset:Nn #1#2
```

```
  { \__kernel_tl_gset:Nx #1 { \__kernel_exp_not:w {#2} } }
\cs_new_protected:Npn \tl_gset:No #1#2
  { \__kernel_tl_gset:Nx #1 { \__kernel_exp_not:w \exp_after:wN {#2} } }
\cs_generate_variant:Nn \tl_set:Nn  {    NV , Nv , Ne , Nf }
\cs_generate_variant:Nn \tl_set:Nn  { c, cV , cv , ce , cf }
\cs_generate_variant:Nn \tl_set:No  { c }
\cs_generate_variant:Nn \tl_set:Nn  { Nx , cx }
\cs_generate_variant:Nn \tl_gset:Nn {    NV , Nv , Ne , Nf }
\cs_generate_variant:Nn \tl_gset:Nn { c, cV , cv , ce , cf }
\cs_generate_variant:Nn \tl_gset:No { c }
\cs_generate_variant:Nn \tl_gset:Nn { Nx , cx }
\cs_new_protected:Npn \tl_put_left:Nn #1#2
  {
    \__kernel_tl_set:Nx #1
      { \__kernel_exp_not:w {#2} \__kernel_exp_not:w \exp_after:wN {#1} }
  }
\cs_new_protected:Npn \tl_put_left:NV #1#2
  {
    \__kernel_tl_set:Nx #1
      { \exp_not:V #2 \__kernel_exp_not:w \exp_after:wN {#1} }
  }
\cs_new_protected:Npn \tl_put_left:Nv #1#2
  {
    \__kernel_tl_set:Nx #1
      { \exp_not:v {#2} \__kernel_exp_not:w \exp_after:wN {#1} }
  }
\cs_new_protected:Npn \tl_put_left:Ne #1#2
  {
    \__kernel_tl_set:Nx #1
      {
        \__kernel_exp_not:w \tex_expanded:D { {#2} }
        \__kernel_exp_not:w \exp_after:wN {#1}
      }
  }
\cs_new_protected:Npn \tl_put_left:No #1#2
  {
    \__kernel_tl_set:Nx #1
      {
        \__kernel_exp_not:w \exp_after:wN {#2}
        \__kernel_exp_not:w \exp_after:wN {#1}
      }
  }
\cs_new_protected:Npn \tl_gput_left:Nn #1#2
  {
    \__kernel_tl_gset:Nx #1
      { \__kernel_exp_not:w {#2} \__kernel_exp_not:w \exp_after:wN {#1} }
  }
\cs_new_protected:Npn \tl_gput_left:NV #1#2
  {
    \__kernel_tl_gset:Nx #1
      { \exp_not:V #2 \__kernel_exp_not:w \exp_after:wN {#1} }
  }
\cs_new_protected:Npn \tl_gput_left:Nv #1#2
  {
    \__kernel_tl_gset:Nx #1
      { \exp_not:v {#2} \__kernel_exp_not:w \exp_after:wN {#1} }
  }
\cs_new_protected:Npn \tl_gput_left:Ne #1#2
  {
    \__kernel_tl_gset:Nx #1
      {
        \__kernel_exp_not:w \tex_expanded:D { {#2} }
        \__kernel_exp_not:w \exp_after:wN {#1}
      }
  }
\cs_new_protected:Npn \tl_gput_left:No #1#2
  {
    \__kernel_tl_gset:Nx #1
      {
        \__kernel_exp_not:w \exp_after:wN {#2}
        \__kernel_exp_not:w \exp_after:wN {#1}
      }
  }
\cs_generate_variant:Nn \tl_put_left:Nn  { c }
\cs_generate_variant:Nn \tl_put_left:NV  { c }
\cs_generate_variant:Nn \tl_put_left:Nv  { c }
\cs_generate_variant:Nn \tl_put_left:Ne  { c }
\cs_generate_variant:Nn \tl_put_left:No  { c }
\cs_generate_variant:Nn \tl_put_left:Nn  { Nx, cx }
\cs_generate_variant:Nn \tl_gput_left:Nn { c }
\cs_generate_variant:Nn \tl_gput_left:NV { c }
\cs_generate_variant:Nn \tl_gput_left:Nv { c }
\cs_generate_variant:Nn \tl_gput_left:Ne { c }
\cs_generate_variant:Nn \tl_gput_left:No { c }
\cs_generate_variant:Nn \tl_gput_left:Nn { Nx , cx }
\cs_new_protected:Npn \tl_put_right:Nn #1#2
  { \__kernel_tl_set:Nx #1 { \__kernel_exp_not:w \exp_after:wN { #1 #2 } } }
\cs_new_protected:Npn \tl_put_right:NV #1#2
  {
    \__kernel_tl_set:Nx #1
      { \__kernel_exp_not:w \exp_after:wN {#1} \exp_not:V #2 }
  }
```

```
\cs_new_protected:Npn \tl_put_right:Nv #1#2
  {
    \__kernel_tl_set:Nx #1
      { \__kernel_exp_not:w \exp_after:wN {#1} \exp_not:v {#2} }
  }
\cs_new_protected:Npn \tl_put_right:Ne #1#2
  {
    \__kernel_tl_set:Nx #1
      {
        \__kernel_exp_not:w \exp_after:wN {#1}
        \__kernel_exp_not:w \tex_expanded:D { {#2} }
      }
  }
\cs_new_protected:Npn \tl_put_right:No #1#2
  {
    \__kernel_tl_set:Nx #1
      {
        \__kernel_exp_not:w \exp_after:wN {#1}
        \__kernel_exp_not:w \exp_after:wN {#2}
      }
  }
\cs_new_protected:Npn \tl_gput_right:Nn #1#2
  { \__kernel_tl_gset:Nx #1 { \__kernel_exp_not:w \exp_after:wN { #1 #2 } } }
\cs_new_protected:Npn \tl_gput_right:NV #1#2
  {
    \__kernel_tl_gset:Nx #1
      { \__kernel_exp_not:w \exp_after:wN {#1} \exp_not:V #2 }
  }
\cs_new_protected:Npn \tl_gput_right:Nv #1#2
  {
    \__kernel_tl_gset:Nx #1
      { \__kernel_exp_not:w \exp_after:wN {#1} \exp_not:v {#2} }
  }
\cs_new_protected:Npn \tl_gput_right:Ne #1#2
  {
    \__kernel_tl_gset:Nx #1
      {
        \__kernel_exp_not:w \exp_after:wN {#1}
        \__kernel_exp_not:w \tex_expanded:D { {#2} }
      }
  }
\cs_new_protected:Npn \tl_gput_right:No #1#2
  {
    \__kernel_tl_gset:Nx #1
      {
        \__kernel_exp_not:w \exp_after:wN {#1}
        \__kernel_exp_not:w \exp_after:wN {#2}
      }
  }
\cs_generate_variant:Nn \tl_put_right:Nn  { c }
\cs_generate_variant:Nn \tl_put_right:NV  { c }
\cs_generate_variant:Nn \tl_put_right:Nv  { c }
\cs_generate_variant:Nn \tl_put_right:Ne  { c }
\cs_generate_variant:Nn \tl_put_right:No  { c }
\cs_generate_variant:Nn \tl_put_right:Nn  { Nx , cx }
\cs_generate_variant:Nn \tl_gput_right:Nn { c }
\cs_generate_variant:Nn \tl_gput_right:NV { c }
\cs_generate_variant:Nn \tl_gput_right:Nv { c }
\cs_generate_variant:Nn \tl_gput_right:Ne { c }
\cs_generate_variant:Nn \tl_gput_right:No { c }
\cs_generate_variant:Nn \tl_gput_right:Nn { Nx, cx }
\quark_new:N \q__tl_nil
\quark_new:N \q__tl_mark
\quark_new:N \q__tl_stop
\quark_new:N \q__tl_recursion_tail
\quark_new:N \q__tl_recursion_stop
\__kernel_quark_new_test:N \__tl_if_recursion_tail_break:nN
\__kernel_quark_new_conditional:Nn \__tl_quark_if_nil:n { TF }
\tl_const:Ne \c__tl_rescan_marker_tl { : \token_to_str:N : }
\cs_new_protected:Npn \tl_rescan:nn #1#2
  {
    \tl_set_rescan:Nnn \l__tl_internal_a_tl {#1} {#2}
    \exp_after:wN \__tl_rescan_aux:
    \l__tl_internal_a_tl
  }
\cs_generate_variant:Nn \tl_rescan:nn { nV }
\exp_args:NNo \cs_new_protected:Npn \__tl_rescan_aux:
  { \tl_clear:N \l__tl_internal_a_tl }
\cs_new_protected:Npn \tl_set_rescan:Nnn
  { \__tl_set_rescan:NNnn \tl_set:No }
\cs_new_protected:Npn \tl_gset_rescan:Nnn
  { \__tl_set_rescan:NNnn \tl_gset:No }
\cs_new_protected:Npn \__tl_set_rescan:NNnn #1#2#3#4
  {
    \group_begin:
      \if_false: { \fi:
      \int_set_eq:NN \tex_tracingnesting:D \c_zero_int
      \int_compare:nNnT \tex_endlinechar:D = { 32 }
        { \int_set:Nn \tex_endlinechar:D { -1 } }
      \int_set_eq:NN \tex_newlinechar:D \tex_endlinechar:D
      #3 \scan_stop:
      \exp_args:No \__tl_set_rescan:nNN { \tl_to_str:n {#4} } #1 #2
    \if_false: } \fi:
  }
\cs_new_protected:Npn \__tl_set_rescan_multi:nNN #1#2#3
  {
    \tex_everyeof:D \exp_after:wN { \c__tl_rescan_marker_tl }
    \exp_after:wN \__tl_rescan:NNw
    \exp_after:wN #2
    \exp_after:wN #3
    \exp_after:wN \prg_do_nothing:
    \tex_scantokens:D {#1}
  }
\exp_args:Nno \use:nn
  { \cs_new:Npn \__tl_rescan:NNw #1#2#3 } \c__tl_rescan_marker_tl
  {
    \group_end:
    #1 #2 {#3}
  }
\cs_generate_variant:Nn \tl_set_rescan:Nnn  { NnV , Nne , c , cnV , cne }
\cs_generate_variant:Nn \tl_set_rescan:Nnn  { Nno , Nnx , cno , cnx }
\cs_generate_variant:Nn \tl_gset_rescan:Nnn { NnV , Nne , c , cnV , cne }
\cs_generate_variant:Nn \tl_gset_rescan:Nnn { Nno , Nnx , cno , cnx }
\cs_new_protected:Npn \__tl_set_rescan:nNN #1
  {
    \int_compare:nNnTF \tex_newlinechar:D < 0
      { \use_ii:nn }
      {
        \exp_args:Nnf \tl_if_in:nnTF {#1}
          { \char_generate:nn { \tex_newlinechar:D } { 12 } }
      }
        { \__tl_set_rescan_multi:nNN }
        {
          \int_set:Nn \tex_endlinechar:D { -1 }
          \__tl_set_rescan_single:nnNN { `' }
        }
    {#1}
  }
\cs_new_protected:Npn \__tl_set_rescan_single:nnNN #1
  {
    \int_compare:nNnTF
      { \char_value_catcode:n {#1} / 2 } = 6
      {
        \exp_args:Nof \__tl_set_rescan_single_aux:nnnNN
          \c__tl_rescan_marker_tl
          { \char_generate:nn {#1} { \char_value_catcode:n {#1} } }
      }
      {
        \int_compare:nNnTF {#1} < { `\~ }
          {
            \exp_args:Nf \__tl_set_rescan_single:nnNN
              { \int_eval:n { #1 + 1 } }
          }
          { \__tl_set_rescan_multi:nNN }
      }
  }
\cs_new_protected:Npn \__tl_set_rescan_single_aux:nnnNN #1#2#3#4#5
  {
    \tex_everyeof:D
      {
        #1 \use_none:n
        #2 #1 { \exp:w \__tl_set_rescan_single_aux:w }
        \s__tl_stop
      }
    \cs_set:Npn \__tl_rescan:NNw ##1##2##3 #2 #1 ##4 ##5 \s__tl_stop
      {
        \group_end:
        ##1 ##2 { ##4 ##3 }
      }
    \exp_after:wN \__tl_rescan:NNw
    \exp_after:wN #4
    \exp_after:wN #5
    \tex_scantokens:D { #2 #3 #2 }
  }
\exp_args:Nno \use:nn
  { \cs_new:Npn \__tl_set_rescan_single_aux:w #1 }
  \c__tl_rescan_marker_tl #2
  { \use_i:nn \exp_end: #1 }
\cs_new_protected:Npn \tl_replace_once:Nnn
  { \__tl_replace:NnNNNnn \q__tl_mark ? \__tl_replace_wrap:w \__kernel_tl_set:Nx  }
\cs_new_protected:Npn \tl_greplace_once:Nnn
  { \__tl_replace:NnNNNnn \q__tl_mark ? \__tl_replace_wrap:w \__kernel_tl_gset:Nx }
\cs_new_protected:Npn \tl_replace_all:Nnn
  { \__tl_replace:NnNNNnn \q__tl_mark ? \__tl_replace_next:w \__kernel_tl_set:Nx  }
\cs_new_protected:Npn \tl_greplace_all:Nnn
  { \__tl_replace:NnNNNnn \q__tl_mark ? \__tl_replace_next:w \__kernel_tl_gset:Nx }
\cs_generate_variant:Nn \tl_replace_once:Nnn
  { NnV , Nne , NV , Ne , Nee , c , cnV , cne , cV , ce , cee }
\cs_generate_variant:Nn \tl_replace_once:Nnn
  { Nx , Nnx , Nxx , cxn , cnx , cxx }
\cs_generate_variant:Nn \tl_greplace_once:Nnn
  { NnV , Nne , NV , Ne , Nee , c , cnV , cne , cV , ce , cee }
\cs_generate_variant:Nn \tl_greplace_once:Nnn
  { Nx , Nnx , Nxx , cxn , cnx , cxx }
```

```
\cs_generate_variant:Nn \tl_replace_all:Nnn
  { NnV , Nne , NV , Ne , Nee , c , cnV , cne , cV , ce , cee }
\cs_generate_variant:Nn \tl_replace_all:Nnn
  { Nx , Nnx , Nxx , cxn , cnx , cxx }
\cs_generate_variant:Nn \tl_greplace_all:Nnn
  { NnV , Nne , NV , Ne , Nee , c , cnV , cne , cV , ce , cee }
\cs_generate_variant:Nn \tl_greplace_all:Nnn
  { Nx , Nnx , Nxx , cxn , cnx , cxx }
\cs_new_protected:Npn \__tl_replace:NnNNNnn #1#2#3#4#5#6#7
  {
    \tl_if_empty:nTF {#6}
      {
        \msg_error:nne { kernel } { empty-search-pattern }
          { \tl_to_str:n {#7} }
      }
      {
        \tl_if_in:onTF { #5 #6 } {#1}
          {
            \tl_if_in:nnTF {#6} {#1}
              { \exp_args:Nc \__tl_replace:NnNNNnn {#2} {#2?} }
              {
                \__tl_quark_if_nil:nTF {#6}
                  { \__tl_replace_auxi:NnnNNNnn #5 {#1} { #1 \q__tl_stop } }
                  { \__tl_replace_auxi:NnnNNNnn #5 {#1} { #1 \q__tl_nil  } }
              }
          }
          { \__tl_replace_auxii:nNNNnn {#1} }
          #3#4#5 {#6} {#7}
      }
  }
\cs_new_protected:Npn \__tl_replace_auxi:NnnNNNnn #1#2#3
  {
    \tl_if_in:NnTF #1 { #2 #3 #3 }
      { \__tl_replace_auxi:NnnNNNnn #1 { #2 #3 } {#2} }
      { \__tl_replace_auxii:nNNNnn { #2 #3 #3 } }
  }
\cs_new_protected:Npn \__tl_replace_auxii:nNNNnn #1#2#3#4#5#6
  {
    \group_align_safe_begin:
    \cs_set:Npn \__tl_replace_wrap:w ##1 #1 ##2
      { \__kernel_exp_not:w \exp_after:wN { \use_none:nn ##1 } ##2 }
    \cs_set:Npe \__tl_replace_next:w ##1 #5
      {
        \exp_not:N \__tl_replace_wrap:w ##1
        \exp_not:n { #1 }
        \exp_not:n { \exp_not:n {#6} }
        \exp_not:n { #2 { } { } }
      }
    #3 #4
      {
        \exp_after:wN \__tl_replace_next_aux:w
        #4
        #1
        {
          \if_false: { \fi: }
          \exp_after:wN \use_none:n \exp_after:wN { \if_false: } \fi:
        }
        #5
      }
    \group_align_safe_end:
  }
\cs_new:Npn \__tl_replace_next_aux:w { \__tl_replace_next:w { } { } }
\cs_new_eq:NN \__tl_replace_wrap:w ?
\cs_new_eq:NN \__tl_replace_next:w ?
\cs_new_protected:Npn \tl_regex_replace_once:Nnn #1#2#3
  { \regex_replace_once:nnN {#2} {#3} #1 }
\cs_generate_variant:Nn \tl_regex_replace_once:Nnn { c }
\cs_new_protected:Npn \tl_regex_replace_once:NNn #1#2#3
  { \regex_replace_once:NnN #2 {#3} #1 }
\cs_generate_variant:Nn \tl_regex_replace_once:NNn { c }
\cs_new_protected:Npn \tl_regex_replace_all:Nnn #1#2#3
  { \regex_replace_all:nnN {#2} {#3} #1 }
\cs_generate_variant:Nn \tl_regex_replace_all:Nnn { c }
\cs_new_protected:Npn \tl_regex_replace_all:NNn #1#2#3
  { \regex_replace_all:NnN #2 {#3} #1 }
\cs_generate_variant:Nn \tl_regex_replace_all:NNn { c }
\group_begin:
  \cs_set_protected:Npn \__tl_tmp:w #1#2#3
    {
      \cs_new_protected:cpe { tl_regex_greplace_ #1 :N #2 n } ##1##2##3
        {
          \group_begin:
            \tl_set_eq:NN \exp_not:N \l__tl_internal_a_tl ##1
            \exp_not:c { regex_replace_ #1 :Nn #2 }
              #3 {##2} {##3} \exp_not:N \l__tl_internal_a_tl
            \tl_gset_eq:NN ##1 \exp_not:N \l__tl_internal_a_tl
          \group_end:
        }
      \cs_generate_variant:cn { tl_regex_greplace_ #1 :N #2 n } { c }
    }
  \__tl_tmp:w { once } n { }
  \__tl_tmp:w { once } N \use:n
  \__tl_tmp:w { all } n { }
  \__tl_tmp:w { all } N \use:n
\group_end:
\cs_new_protected:Npn \tl_remove_once:Nn #1#2
  { \tl_replace_once:Nnn #1 {#2} { } }
\cs_new_protected:Npn \tl_gremove_once:Nn #1#2
  { \tl_greplace_once:Nnn #1 {#2} { } }
\cs_generate_variant:Nn \tl_remove_once:Nn  { NV , Ne , c , cV , ce }
\cs_generate_variant:Nn \tl_gremove_once:Nn { NV , Ne , c , cV , ce }
\cs_new_protected:Npn \tl_remove_all:Nn #1#2
  { \tl_replace_all:Nnn #1 {#2} { } }
\cs_new_protected:Npn \tl_gremove_all:Nn #1#2
  { \tl_greplace_all:Nnn #1 {#2} { } }
\cs_generate_variant:Nn \tl_remove_all:Nn  { NV , Ne , c , cV , ce }
\cs_generate_variant:Nn \tl_remove_all:Nn  { Nx , cx }
\cs_generate_variant:Nn \tl_gremove_all:Nn { NV , Ne , c , cV , ce }
\cs_generate_variant:Nn \tl_gremove_all:Nn { Nx , cx }
\prg_new_conditional:Npnn \tl_if_empty:N #1 { p , T , F , TF }
  {
    \if_meaning:w #1 \c_empty_tl
      \prg_return_true:
    \else:
      \prg_return_false:
    \fi:
  }
\prg_generate_conditional_variant:Nnn \tl_if_empty:N
  { c } { p , T , F , TF }
\prg_new_conditional:Npnn \tl_if_empty:n #1 { p , TF , T , F }
  {
    \if:w \scan_stop: \tl_to_str:n {#1} \scan_stop:
      \prg_return_true:
    \else:
      \prg_return_false:
    \fi:
  }
\prg_generate_conditional_variant:Nnn \tl_if_empty:n
  { V , e } { p , TF , T , F }
\cs_new:Npn \__tl_if_empty_if:o #1
  {
    \if:w \scan_stop: \__kernel_tl_to_str:w \exp_after:wN {#1} \scan_stop:
  }
\exp_args:Nno \use:n
  { \prg_new_conditional:Npnn \tl_if_empty:o #1 { p , TF , T , F } }
  {
    \__tl_if_empty_if:o {#1}
      \prg_return_true:
```

```
    \else:
      \prg_return_false:
    \fi:
  }
\exp_args:Nno \use:n
  { \prg_new_conditional:Npnn \tl_if_blank:n #1 { p , T , F , TF } }
  {
    \__tl_if_empty_if:o { \use_none:n #1 ? }
      \prg_return_true:
    \else:
      \prg_return_false:
    \fi:
  }
\prg_generate_conditional_variant:Nnn \tl_if_blank:n
  { e , V , o } { p , T , F , TF }
\prg_new_eq_conditional:NNn \tl_if_eq:NN \cs_if_eq:NN { p , T , F , TF }
\prg_generate_conditional_variant:Nnn \tl_if_eq:NN
  { Nc , c , cc } { p , TF , T , F }
\tl_new:N \l__tl_internal_a_tl
\tl_new:N \l__tl_internal_b_tl
\prg_new_protected_conditional:Npnn \tl_if_eq:Nn #1#2 { T , F , TF }
  {
    \group_begin:
      \tl_set:Nn \l__tl_internal_b_tl {#2}
      \exp_after:wN
    \group_end:
    \if_meaning:w #1 \l__tl_internal_b_tl
      \prg_return_true:
    \else:
      \prg_return_false:
    \fi:
  }
\prg_generate_conditional_variant:Nnn \tl_if_eq:Nn { c } { TF , T , F }
\prg_new_protected_conditional:Npnn \tl_if_eq:nn #1#2 { T , F ,  TF }
  {
    \group_begin:
      \tl_set:Nn \l__tl_internal_a_tl {#1}
      \tl_set:Nn \l__tl_internal_b_tl {#2}
      \exp_after:wN
    \group_end:
    \if_meaning:w \l__tl_internal_a_tl \l__tl_internal_b_tl
      \prg_return_true:
    \else:
      \prg_return_false:
    \fi:
  }
```

```
\prg_generate_conditional_variant:Nnn \tl_if_eq:nn
  { nV , ne , nx , V, e , ee , x , xx }
  { TF , T , F }
\cs_new_protected:Npn \tl_if_in:NnT  { \exp_args:No \tl_if_in:nnT  }
\cs_new_protected:Npn \tl_if_in:NnF  { \exp_args:No \tl_if_in:nnF  }
\cs_new_protected:Npn \tl_if_in:NnTF { \exp_args:No \tl_if_in:nnTF }
\prg_generate_conditional_variant:Nnn \tl_if_in:Nn
  { NV , No , c , cV , co } { T , F , TF }
\prg_new_protected_conditional:Npnn \tl_if_in:nn #1#2 { T  , F , TF }
  {
    \scan_stop:
    \if_false: { \fi:
    \cs_set:Npn \__tl_tmp:w ##1 #2 { }
    \tl_if_empty:oTF { \__tl_tmp:w #1 {} {} #2 }
      { \prg_return_false: } { \prg_return_true: }
    \if_false: } \fi:
  }
\prg_generate_conditional_variant:Nnn \tl_if_in:nn
  { V , VV , o , oo , nV , no } { T , F , TF }
\cs_set_protected:Npn \__tl_tmp:w #1
  {
    \prg_new_conditional:Npnn \tl_if_novalue:n ##1
      { p , T ,  F , TF }
      {
        \__tl_if_empty_if:o { \__tl_if_novalue:w {} ##1 {} ? ! #1 ? ? ! }
          \prg_return_true:
        \else:
          \prg_return_false:
        \fi:
      }
    \cs_new:Npn \__tl_if_novalue:w ##1 #1 ##2 ? ##3 ? ! { ##1 ##2 }
  }
\exp_args:No \__tl_tmp:w { \c_novalue_tl }
\cs_new:Npn \tl_if_single_p:N { \exp_args:No \tl_if_single_p:n }
\cs_new:Npn \tl_if_single:NT  { \exp_args:No \tl_if_single:nT  }
\cs_new:Npn \tl_if_single:NF  { \exp_args:No \tl_if_single:nF  }
\cs_new:Npn \tl_if_single:NTF { \exp_args:No \tl_if_single:nTF }
\prg_generate_conditional_variant:Nnn \tl_if_single:N {c} { p , T , F , TF }
\prg_new_conditional:Npnn \tl_if_single:n #1 { p , T , F , TF }
  {
    \if:w \scan_stop: \exp_after:wN \__tl_if_single:nnw
        \__kernel_tl_to_str:w
          \exp_after:wN { \use_none:nn #1 ?? } \scan_stop: ? \s__tl_stop
      \prg_return_true:
    \else:
      \prg_return_false:
```

```
    \fi:
  }
\cs_new:Npn \__tl_if_single:nnw #1#2#3 \s__tl_stop {#2}
\prg_new_conditional:Npnn \tl_if_single_token:n #1 { p , T , F , TF }
  {
    \tl_if_head_is_N_type:nTF {#1}
      { \__tl_if_empty_if:o { \use_none:n #1 } }
      {
        \tl_if_empty:nTF {#1}
          { \if_false: }
          { \__tl_if_empty_if:o { \exp:w \exp_end_continue_f:w #1 } }
      }
      \prg_return_true:
    \else:
      \prg_return_false:
    \fi:
  }
\prg_new_protected_conditional:Npnn \tl_if_regex_match:nn #1#2 { TF , T , F }
  {
    \regex_match:nnTF {#2} {#1}
      \prg_return_true: \prg_return_false:
  }
\prg_generate_conditional_variant:Nnn \tl_if_regex_match:nn
  { V } { TF , T , F }
\prg_new_protected_conditional:Npnn \tl_if_regex_match:nN #1#2 { TF , T , F }
  {
    \regex_match:nNTF {#2} #1
      \prg_return_true: \prg_return_false:
  }
\prg_generate_conditional_variant:Nnn \tl_if_regex_match:nN
  { V } { TF , T , F }
\cs_new:Npn \tl_map_function:nN #1#2
  {
    \__tl_map_function:Nnnnnnnnn #2 #1
      \s__tl_stop \s__tl_stop \s__tl_stop \s__tl_stop
      \s__tl_stop \s__tl_stop \s__tl_stop \s__tl_stop
    \prg_break_point:Nn \tl_map_break: { }
  }
\cs_generate_variant:Nn \tl_map_function:nN { e }
\cs_new:Npn \tl_map_function:NN
  { \exp_args:No \tl_map_function:nN }
\cs_generate_variant:Nn \tl_map_function:NN { c }
\cs_new:Npn \__tl_map_function:Nnnnnnnnn #1#2#3#4#5#6#7#8#9
  {
    \__tl_use_none_delimit_by_s_stop:w
      #9 \__tl_map_function_end:w \s__tl_stop
    #1 {#2} #1 {#3} #1 {#4} #1 {#5} #1 {#6} #1 {#7} #1 {#8} #1 {#9}
    \__tl_map_function:Nnnnnnnnn #1
  }
\cs_new:Npn \__tl_map_function_end:w \s__tl_stop #1#2
  {
    \__tl_use_none_delimit_by_s_stop:w #2 \tl_map_break: \s__tl_stop
    #1 {#2}
    \__tl_map_function_end:w \s__tl_stop
  }
\cs_new:Npn \__tl_use_none_delimit_by_s_stop:w #1 \s__tl_stop { }
\cs_new_protected:Npn \tl_map_inline:nn #1#2
  {
    \int_gincr:N \g__kernel_prg_map_int
    \cs_gset_protected:cpn
      { __tl_map_ \int_use:N \g__kernel_prg_map_int :w } ##1 {#2}
    \exp_args:Nc \__tl_map_function:Nnnnnnnnn
      { __tl_map_ \int_use:N \g__kernel_prg_map_int :w }
      #1
      \s__tl_stop \s__tl_stop \s__tl_stop \s__tl_stop
      \s__tl_stop \s__tl_stop \s__tl_stop \s__tl_stop
    \prg_break_point:Nn \tl_map_break:
      { \int_gdecr:N \g__kernel_prg_map_int }
  }
\cs_new_protected:Npn \tl_map_inline:Nn
  { \exp_args:No \tl_map_inline:nn }
\cs_generate_variant:Nn \tl_map_inline:Nn { c }
\cs_new:Npn \tl_map_tokens:nn #1#2
  {
    \__tl_map_tokens:nnnnnnnnn {#2} #1
      \s__tl_stop \s__tl_stop \s__tl_stop \s__tl_stop
      \s__tl_stop \s__tl_stop \s__tl_stop \s__tl_stop
    \prg_break_point:Nn \tl_map_break: { }
  }
\cs_new:Npn \tl_map_tokens:Nn
  { \exp_args:No \tl_map_tokens:nn }
\cs_generate_variant:Nn \tl_map_tokens:Nn { c }
\cs_new:Npn \__tl_map_tokens:nnnnnnnnn #1#2#3#4#5#6#7#8#9
  {
    \__tl_use_none_delimit_by_s_stop:w
      #9 \__tl_map_tokens_end:w \s__tl_stop
    \use:n {#1} {#2} \use:n {#1} {#3} \use:n {#1} {#4} \use:n {#1} {#5}
    \use:n {#1} {#6} \use:n {#1} {#7} \use:n {#1} {#8} \use:n {#1} {#9}
    \__tl_map_tokens:nnnnnnnnn {#1}
  }
\cs_new:Npn \__tl_map_tokens_end:w \s__tl_stop \use:n #1#2
  {
    \__tl_use_none_delimit_by_s_stop:w #2 \tl_map_break: \s__tl_stop
    #1 {#2}
    \__tl_map_tokens_end:w \s__tl_stop
  }
\cs_new_protected:Npn \tl_map_variable:nNn #1#2#3
  { \tl_map_tokens:nn {#1} { \__tl_map_variable:Nnn #2 {#3} } }
\cs_new_protected:Npn \__tl_map_variable:Nnn #1#2#3
  { \tl_set:Nn #1 {#3} #2 }
\cs_new_protected:Npn \tl_map_variable:NNn
  { \exp_args:No \tl_map_variable:nNn }
\cs_generate_variant:Nn \tl_map_variable:NNn { c }
\cs_new:Npn \tl_map_break:
  { \prg_map_break:Nn \tl_map_break: { } }
\cs_new:Npn \tl_map_break:n
  { \prg_map_break:Nn \tl_map_break: }
\cs_generate_variant:Nn \tl_to_str:n { o , V , v , e }
\cs_new:Npn \tl_to_str:N #1 { \__kernel_tl_to_str:w \exp_after:wN {#1} }
\cs_generate_variant:Nn \tl_to_str:N { c }
\cs_new:Npn \tl_use:N #1
  {
    \tl_if_exist:NTF #1 {#1}
      {
        \msg_expandable_error:nnn
          { kernel } { bad-variable } {#1}
      }
  }
\cs_generate_variant:Nn \tl_use:N { c }
\cs_new:Npn \tl_count:n #1
  {
    \int_eval:n
      { 0 \tl_map_function:nN {#1} \__tl_count:n }
  }
\cs_new:Npn \tl_count:N #1
  {
    \int_eval:n
      { 0 \tl_map_function:NN #1 \__tl_count:n }
  }
\cs_new:Npn \__tl_count:n #1 { + 1 }
\cs_generate_variant:Nn \tl_count:n { V , v , e , o }
\cs_generate_variant:Nn \tl_count:N { c }
\cs_new:Npn \tl_count_tokens:n #1
  {
    \int_eval:n
      {
        \__tl_act:NNNn
          \__tl_act_count_normal:N
          \__tl_act_count_group:n
          \__tl_act_count_space:
          {#1}
      }
  }
\cs_new:Npn \__tl_act_count_normal:N #1 { 1 + }
\cs_new:Npn \__tl_act_count_space: { 1 + }
\cs_new:Npn \__tl_act_count_group:n #1 { 2 + \tl_count_tokens:n {#1} + }
\cs_new:Npn \tl_reverse_items:n #1
  {
    \__tl_reverse_items:nwNwn #1 ?
      \s__tl_mark \__tl_reverse_items:nwNwn
      \s__tl_mark \__tl_reverse_items:wn
      \s__tl_stop { }
  }
\cs_new:Npn \__tl_reverse_items:nwNwn #1 #2 \s__tl_mark #3 #4 \s__tl_stop #5
  {
    #3 #2
      \s__tl_mark \__tl_reverse_items:nwNwn
      \s__tl_mark \__tl_reverse_items:wn
      \s__tl_stop { {#1} #5 }
  }
\cs_new:Npn \__tl_reverse_items:wn #1 \s__tl_stop #2
  { \__kernel_exp_not:w \exp_after:wN { \use_none:nn #2 } }
\cs_new:Npn \tl_trim_spaces:n
  { \__tl_trim_spaces:nn { \__kernel_exp_not:w \exp_after:wN } }
\cs_new:Npn \tl_trim_left_spaces:n
  { \__tl_trim_left_spaces:nn { \__kernel_exp_not:w \exp_after:wN } }
\cs_new:Npn \tl_trim_right_spaces:n
  { \__tl_trim_right_spaces:nn { \__kernel_exp_not:w \exp_after:wN } }
\cs_generate_variant:Nn \tl_trim_spaces:n       { V , v , e , o }
\cs_generate_variant:Nn \tl_trim_left_spaces:n  { V , v , e , o }
\cs_generate_variant:Nn \tl_trim_right_spaces:n { V , v , e , o }
\cs_new:Npn \tl_trim_spaces_apply:nN #1#2
  { \__tl_trim_spaces:nn { \exp_args:No #2 } { #1 } }
\cs_new:Npn \tl_trim_left_spaces_apply:nN #1#2
  { \__tl_trim_left_spaces:nn { \exp_args:No #2 } { #1 } }
\cs_new:Npn \tl_trim_right_spaces_apply:nN #1#2
  { \__tl_trim_right_spaces:nn { \exp_args:No #2 } { #1 } }
\cs_generate_variant:Nn \tl_trim_spaces_apply:nN { o }
\cs_generate_variant:Nn \tl_trim_left_spaces_apply:nN { o }
\cs_generate_variant:Nn \tl_trim_right_spaces_apply:nN { o }
\cs_new_protected:Npn \tl_trim_spaces:N #1
  { \__kernel_tl_set:Nx #1 { \exp_args:No \tl_trim_spaces:n {#1} } }
\cs_new_protected:Npn \tl_trim_left_spaces:N #1
  { \__kernel_tl_set:Nx #1 { \exp_args:No \tl_trim_left_spaces:n {#1} } }
```

```
\cs_new_protected:Npn \tl_trim_right_spaces:N #1
  { \__kernel_tl_set:Nx #1 { \exp_args:No \tl_trim_right_spaces:n {#1} } }
\cs_new_protected:Npn \tl_gtrim_spaces:N #1
  { \__kernel_tl_gset:Nx #1 { \exp_args:No \tl_trim_spaces:n {#1} } }
\cs_new_protected:Npn \tl_gtrim_left_spaces:N #1
  { \__kernel_tl_gset:Nx #1 { \exp_args:No \tl_trim_left_spaces:n {#1} } }
\cs_new_protected:Npn \tl_gtrim_right_spaces:N #1
  { \__kernel_tl_gset:Nx #1 { \exp_args:No \tl_trim_right_spaces:n {#1} } }
\cs_generate_variant:Nn \tl_trim_spaces:N  { c }
\cs_generate_variant:Nn \tl_trim_left_spaces:N  { c }
\cs_generate_variant:Nn \tl_trim_right_spaces:N  { c }
\cs_generate_variant:Nn \tl_gtrim_spaces:N { c }
\cs_generate_variant:Nn \tl_gtrim_left_spaces:N { c }
\cs_generate_variant:Nn \tl_gtrim_right_spaces:N { c }
\cs_set_protected:Npn \__tl_tmp:w #1
  {
    \cs_new:Npn \__tl_trim_spaces:nn ##1##2
      {
        \__tl_trim_spaces_auxi:w
          \__tl_trim_mark: ##2 \s__tl_nil
          \__tl_trim_mark: \__tl_trim_spaces_auxi:w
          \__tl_trim_mark: #1
          \__tl_trim_mark: \__tl_trim_spaces_auxii:w
        {##1}
      }
    \cs_new:Npn \__tl_trim_left_spaces:nn ##1##2
      {
        \__tl_trim_spaces_auxi:w
          \__tl_trim_mark: ##2 \s__tl_nil
          \__tl_trim_mark: \__tl_trim_spaces_auxi:w
          \__tl_trim_mark: #1
          \__tl_trim_mark: \__tl_trim_spaces_auxv:w
        {##1}
      }
    \cs_new:Npn \__tl_trim_right_spaces:nn ##1##2
      {
        \__tl_trim_spaces_auxiii:w
          \__tl_trim_mark: ##2 \s__tl_nil \__tl_trim_spaces_auxiii:w
          #1 \s__tl_nil \__tl_trim_spaces_auxiv:w
        {##1}
      }
    \cs_new:Npn \__tl_trim_spaces_auxi:w
        ##1 \__tl_trim_mark: #1 ##2 \__tl_trim_mark: ##3
      { ##3 ##1 \__tl_trim_mark: ##2 \__tl_trim_mark: \__tl_trim_spaces_auxi:w }
    \cs_new:Npn \__tl_trim_spaces_auxii:w
        \__tl_trim_mark: ##1 \__tl_trim_mark: ##2 \__tl_trim_spaces_auxi:w
        \__tl_trim_mark: \__tl_trim_mark: \__tl_trim_spaces_auxi:w
      {
        \__tl_trim_spaces_auxiii:w
          \__tl_trim_mark: ##1 \__tl_trim_spaces_auxiii:w
          #1 \s__tl_nil \__tl_trim_spaces_auxiv:w
      }
    \cs_new:Npn \__tl_trim_spaces_auxiii:w ##1 #1 \s__tl_nil ##2
      { ##2 ##1 \s__tl_nil \__tl_trim_spaces_auxiii:w }
    \cs_new:Npn \__tl_trim_spaces_auxiv:w
        ##1 \s__tl_nil
        \__tl_trim_spaces_auxiii:w \s__tl_nil \__tl_trim_spaces_auxiii:w
        ##2
      { ##2 {##1} }
    \cs_new:Npn \__tl_trim_spaces_auxv:w
        ##1 \s__tl_nil
        \__tl_trim_mark: \__tl_trim_spaces_auxi:w \__tl_trim_mark:
        \__tl_trim_mark: \__tl_trim_spaces_auxi:w ##2
      { ##2 {##1} }
    \cs_new:Npn \__tl_trim_mark: {}
  }
\__tl_tmp:w { ~ }
\cs_new:Npn \tl_head:n #1
  {
    \__kernel_exp_not:w \tex_expanded:D
      { { \if_false: { \fi: \__tl_head_aux:n #1 { } } } }
  }
\cs_new:Npn \__tl_head_aux:n #1
  {
    \__kernel_exp_not:w {#1}
    \exp_after:wN \use_none:n \exp_after:wN { \if_false: } \fi:
  }
\cs_generate_variant:Nn \tl_head:n { V , v , f , e }
\cs_new:Npn \tl_head:w #1#2 \q_stop {#1}
\cs_new:Npn \__tl_tl_head:w #1#2 \s__tl_stop {#1}
\cs_new:Npn \tl_head:N { \exp_args:No \tl_head:n }
\exp_args:Nno \use:n { \cs_new:Npn \tl_tail:n #1 }
  {
    \exp_after:wN \__kernel_exp_not:w
      \tl_if_blank:nTF {#1}
        { { } }
        { \exp_after:wN { \use_none:n #1 } }
  }
\cs_generate_variant:Nn \tl_tail:n { V , v , f , e }
\cs_new:Npn \tl_tail:N { \exp_args:No \tl_tail:n }
\prg_new_conditional:Npnn \tl_if_head_eq_charcode:nN #1#2 { p , T , F , TF }
  {
    \if_charcode:w
        \tl_if_head_is_N_type:nTF { #1 ? }
          { \__tl_head_exp_not:w #1 { ^ \__tl_if_head_eq_empty_arg:w } \s__tl_stop }
          { \str_head:n {#1} }
        \exp_not:N #2
      \prg_return_true:
    \else:
      \prg_return_false:
    \fi:
  }
\prg_generate_conditional_variant:Nnn \tl_if_head_eq_charcode:nN
  { V , e , f } { p , TF , T , F }
\prg_new_conditional:Npnn \tl_if_head_eq_catcode:nN #1 #2 { p , T , F , TF }
  {
    \if_catcode:w
        \tl_if_head_is_N_type:nTF { #1 ? }
          { \__tl_head_exp_not:w #1 { ^ \__tl_if_head_eq_empty_arg:w } \s__tl_stop }
          {
            \tl_if_head_is_group:nTF {#1}
              \c_group_begin_token
              \c_space_token
          }
        \exp_not:N #2
      \prg_return_true:
    \else:
      \prg_return_false:
    \fi:
  }
\prg_generate_conditional_variant:Nnn \tl_if_head_eq_catcode:nN
  { V , e , o } { p , TF , T , F }
\prg_new_conditional:Npnn \tl_if_head_eq_meaning:nN #1#2 { p , T , F , TF }
  {
    \tl_if_head_is_N_type:nTF { #1 ? }
      \__tl_if_head_eq_meaning_normal:nN
      \__tl_if_head_eq_meaning_special:nN
    {#1} #2
  }
\prg_generate_conditional_variant:Nnn \tl_if_head_eq_meaning:nN
  { V , e } { p , TF , T , F }
\cs_new:Npn \__tl_if_head_eq_meaning_normal:nN #1 #2
  {
    \exp_after:wN \if_meaning:w
        \__tl_tl_head:w #1 { ?? \use_none:nnn } \s__tl_stop #2
      \prg_return_true:
    \else:
      \prg_return_false:
```

```
    \fi:
  }
\cs_new:Npn \__tl_if_head_eq_meaning_special:nN #1 #2
  {
    \if_charcode:w \str_head:n {#1} \exp_not:N #2
      \exp_after:wN \use_ii:nn
    \else:
      \prg_return_false:
    \fi:
    \use_none:n
    {
      \if_catcode:w \exp_not:N #2
                    \tl_if_head_is_group:nTF {#1}
                      { \c_group_begin_token }
                      { \c_space_token }
        \prg_return_true:
      \else:
        \prg_return_false:
      \fi:
    }
  }
\cs_new:Npn \__tl_head_exp_not:w #1 #2 \s__tl_stop
  { \exp_not:N #1 }
\cs_new:Npn \__tl_if_head_eq_empty_arg:w \exp_not:N #1
  { ? }
\prg_new_conditional:Npnn \tl_if_head_is_N_type:n #1 { p , T , F , TF }
  {
    \if:w
        \if_false: { \fi: \__tl_if_head_is_N_type_auxi:w #1 ~ }
        { \exp_after:wN { \token_to_str:N #1 } }
        \scan_stop: \scan_stop:
      \prg_return_true:
    \else:
      \prg_return_false:
    \fi:
  }
\exp_args:Nno \use:n { \cs_new:Npn \__tl_if_head_is_N_type_auxi:w #1 ~ }
  {
    \tl_if_empty:nTF {#1}
      { f \exp_after:wN \use_none:nn }
      { \exp_after:wN \__tl_if_head_is_N_type_auxii:n }
    \exp_after:wN { \if_false: } \fi:
  }
\cs_new:Npn \__tl_if_head_is_N_type_auxii:n #1
  { \exp_after:wN \use_none:n \exp_after:wN }
\prg_new_conditional:Npnn \tl_if_head_is_group:n #1 { p , T , F , TF }
```

```
  {
    \if:w
        \exp_after:wN \use_none:n
          \exp_after:wN { \exp_after:wN { \token_to_str:N #1 ? } }
        \scan_stop: \scan_stop:
      \__tl_if_head_is_group_fi_false:w
    \fi:
    \if_true:
      \prg_return_true:
    \else:
      \prg_return_false:
    \fi:
  }
\cs_new:Npn \__tl_if_head_is_group_fi_false:w \fi: \if_true: { \fi: \if_false: }
\prg_new_conditional:Npnn \tl_if_head_is_space:n #1 { p , T , F , TF }
  {
    \if:w
        \if_false: { \fi: \__tl_if_head_is_space:w \prg_do_nothing: #1 ? ~ }
        \scan_stop: \scan_stop:
      \prg_return_true:
    \else:
      \prg_return_false:
    \fi:
  }
\exp_args:Nno \use:n { \cs_new:Npn \__tl_if_head_is_space:w #1 ~ }
  {
    \__tl_if_empty_if:o {#1} \else: f \fi:
    \exp_after:wN \use_none:n \exp_after:wN { \if_false: } \fi:
  }
\scan_new:N \s__tl_act_stop
\cs_set_protected:Npn \__tl_tmp:w #1
  {
    \cs_new:Npn \__tl_act_if_head_is_space:nTF ##1
      {
        \__tl_act_if_head_is_space:w
          \s__tl_act_stop ##1 \s__tl_act_stop \__tl_act_if_head_is_space_true:w
          \s__tl_act_stop #1  \s__tl_act_stop \use_ii:nn
      }
    \cs_new:Npn \__tl_act_if_head_is_space:w
        ##1 \s__tl_act_stop #1 ##2 \s__tl_act_stop
      {}
    \cs_new:Npn \__tl_act_if_head_is_space_true:w
        \s__tl_act_stop #1 \s__tl_act_stop \use_ii:nn ##1 ##2
      {##1}
  }
\__tl_tmp:w { ~ }
```

```
\exp_args:Nne \use:n { \cs_new:Npn \__tl_act_loop:w #1 \s__tl_act_stop }
  {
    \exp_not:o { \__tl_act_if_head_is_space:nTF {#1} }
      \exp_not:N \__tl_act_space:wwNNN
      {
        \exp_not:o { \tl_if_head_is_group:nTF {#1} }
          \exp_not:N \__tl_act_group:nwNNN
          \exp_not:N \__tl_act_normal:NwNNN
      }
    \exp_not:n {#1} \s__tl_act_stop
  }
\cs_undefine:N \__tl_act_if_head_is_space:nTF
\cs_new:Npn \__tl_act_normal:NwNNN #1 #2 \s__tl_act_stop #3
  {
    #3 #1
    \__tl_act_loop:w #2 \s__tl_act_stop
    #3
  }
\cs_new:Npn \__tl_use_none_delimit_by_s_act_stop:w #1 \s__tl_act_stop { }
\cs_new:Npn \__tl_act_end:wn #1 \__tl_act_result:n #2
  { \group_align_safe_end: \exp_end: #2 }
\cs_new:Npn \__tl_act_group:nwNNN #1 #2 \s__tl_act_stop #3#4#5
  {
    \__tl_use_none_delimit_by_s_act_stop:w #1 \__tl_act_end:wn \s__tl_act_stop
    #5 {#1}
    \__tl_act_loop:w #2 \s__tl_act_stop
    #3 #4 #5
  }
\exp_last_unbraced:NNo
  \cs_new:Npn \__tl_act_space:wwNNN \c_space_tl #1 \s__tl_act_stop #2#3
  {
    #3
    \__tl_act_loop:w #1 \s__tl_act_stop
    #2 #3
  }
\cs_new:Npn \__tl_act:NNNn #1#2#3#4
  {
    \group_align_safe_begin:
    \__tl_act_loop:w #4 { \s__tl_act_stop } ? \s__tl_act_stop
    #1 #3 #2
    \__tl_act_result:n { }
  }
\cs_new:Npn \__tl_act_output:n #1 #2 \__tl_act_result:n #3
  { #2 \__tl_act_result:n { #3 #1 } }
\cs_new:Npn \__tl_act_reverse_output:n #1 #2 \__tl_act_result:n #3
  { #2 \__tl_act_result:n { #1 #3 } }
```

```
\cs_new:Npn \tl_reverse:n #1
  {
    \__kernel_exp_not:w \exp_after:wN
      {
        \exp:w
        \__tl_act:NNNn
          \__tl_reverse_normal:N
          \__tl_reverse_group_preserve:n
          \__tl_reverse_space:
          {#1}
      }
  }
\cs_generate_variant:Nn \tl_reverse:n { o , V , f , e }
\cs_new:Npn \__tl_reverse_normal:N
  { \__tl_act_reverse_output:n }
\cs_new:Npn \__tl_reverse_group_preserve:n #1
  { \__tl_act_reverse_output:n { {#1} } }
\cs_new:Npn \__tl_reverse_space:
  { \__tl_act_reverse_output:n { ~ } }
\cs_new_protected:Npn \tl_reverse:N #1
  { \__kernel_tl_set:Nx #1 { \exp_args:No \tl_reverse:n { #1 } } }
\cs_new_protected:Npn \tl_greverse:N #1
  { \__kernel_tl_gset:Nx #1 { \exp_args:No \tl_reverse:n { #1 } } }
\cs_generate_variant:Nn \tl_reverse:N  { c }
\cs_generate_variant:Nn \tl_greverse:N { c }
\cs_new:Npn \tl_item:nn #1#2
  {
    \exp_args:Nf \__tl_item:nn
      { \exp_args:Nf \__tl_item_aux:nn { \int_eval:n {#2} } {#1} }
    #1
    \q__tl_recursion_tail
    \prg_break_point:
  }
\cs_new:Npn \__tl_item_aux:nn #1#2
  {
    \int_compare:nNnTF {#1} < 0
      { \int_eval:n { \tl_count:n {#2} + 1 + #1 } }
      {#1}
  }
\cs_new:Npn \__tl_item:nn #1#2
  {
    \__tl_if_recursion_tail_break:nN {#2} \prg_break:
    \int_compare:nNnTF {#1} = 1
      { \prg_break:n { \exp_not:n {#2} } }
      { \exp_args:Nf \__tl_item:nn { \int_eval:n { #1 - 1 } } }
  }
```

```
\cs_new:Npn \tl_item:Nn { \exp_args:No \tl_item:nn }
\cs_generate_variant:Nn \tl_item:Nn { c }
\cs_new:Npn \tl_rand_item:n #1
  {
    \tl_if_blank:nF {#1}
      { \tl_item:nn {#1} { \int_rand:nn { 1 } { \tl_count:n {#1} } } }
  }
\cs_new:Npn \tl_rand_item:N { \exp_args:No \tl_rand_item:n }
\cs_generate_variant:Nn \tl_rand_item:N { c }
\cs_new_eq:NN \__kernel_int_sep: \tex_right:D
\cs_new_eq:NN \__tl_sep: \__kernel_int_sep:
\cs_new:Npn \tl_range:Nnn { \exp_args:No \tl_range:nnn }
\cs_generate_variant:Nn \tl_range:Nnn { c }
\cs_new:Npn \tl_range:nnn { \__tl_range:Nnnn \__tl_range:w }
\cs_new:Npn \__tl_range:Nnnn #1#2#3#4
  {
    \tl_head:f
      {
        \exp_args:Nf \__tl_range:nnnNn
          { \tl_count:n {#2} } {#3} {#4} #1 {#2}
      }
  }
\cs_new:Npn \__tl_range:nnnNn #1#2#3
  {
    \exp_args:Nff \__tl_range:nnNn
      {
        \exp_args:Nf \__tl_range_normalize:nn
          { \int_eval:n { #2 - 1 } } {#1}
      }
      {
        \exp_args:Nf \__tl_range_normalize:nn
          { \int_eval:n {#3} } {#1}
      }
  }
\cs_new:Npn \__tl_range:nnNn #1#2#3#4
  {
    \if_int_compare:w #2 > #1 \exp_stop_f: \else:
      \exp_after:wN { \exp_after:wN }
    \fi:
    \exp_after:wN #3
    \int_value:w \int_eval:n { #2 - #1 } \exp_after:wN \__tl_sep:
    \exp_after:wN { \exp:w \__tl_range_skip:w #1 \__tl_sep: { } #4 }
  }
\cs_new:Npn \__tl_range_skip:w #1 \__tl_sep: #2
  {
    \if_int_compare:w #1 > \c_zero_int
```

```
      \exp_after:wN \__tl_range_skip:w
      \int_value:w \int_eval:n { #1 - 1 } \exp_after:wN \__tl_sep:
    \else:
      \exp_after:wN \exp_end:
    \fi:
  }
\cs_new:Npn \__tl_range:w #1 \__tl_sep: #2
  {
    \exp_args:Nf \__tl_range_collect:nn
      { \__tl_range_skip_spaces:n {#2} } {#1}
  }
\cs_new:Npn \__tl_range_skip_spaces:n #1
  {
    \tl_if_head_is_space:nTF {#1}
      { \exp_args:Nf \__tl_range_skip_spaces:n {#1} }
      { { } #1 }
  }
\cs_new:Npn \__tl_range_collect:nn #1#2
  {
    \int_compare:nNnTF {#2} = 0
      {#1}
      {
        \exp_args:No \tl_if_head_is_space:nTF { \use_none:n #1 }
          {
            \exp_args:Nf \__tl_range_collect:nn
              { \__tl_range_collect_space:nw #1 }
              {#2}
          }
          {
            \__tl_range_collect:ff
              {
                \exp_args:No \tl_if_head_is_N_type:nTF { \use_none:n #1 }
                  { \__tl_range_collect_N:nN }
                  { \__tl_range_collect_group:nn }
                #1
              }
              { \int_eval:n { #2 - 1 } }
          }
      }
  }
\cs_new:Npn \__tl_range_collect_space:nw #1 ~ { { #1 ~ } }
\cs_new:Npn \__tl_range_collect_N:nN #1#2 { { #1 #2 } }
\cs_new:Npn \__tl_range_collect_group:nn #1#2 { { #1 {#2} } }
\cs_generate_variant:Nn \__tl_range_collect:nn { ff }
\cs_new:Npn \__tl_range_normalize:nn #1#2
  {
    \int_eval:n
      {
        \if_int_compare:w #1 < \c_zero_int
          \if_int_compare:w #1 < -#2 \exp_stop_f:
            0
          \else:
            #1 + #2 + 1
          \fi:
        \else:
          \if_int_compare:w #1 < #2 \exp_stop_f:
            #1
          \else:
            #2
          \fi:
        \fi:
      }
  }
\cs_new_protected:Npn \tl_show:N { \__tl_show:NN \tl_show:n }
\cs_generate_variant:Nn \tl_show:N { c }
\cs_new_protected:Npn \tl_log:N { \__tl_show:NN \tl_log:n }
\cs_generate_variant:Nn \tl_log:N { c }
\cs_new_protected:Npn \__tl_show:NN #1#2
  {
    \__kernel_chk_defined:NT #2
      {
        \exp_args:Nf \tl_if_empty:nTF
          { \cs_prefix_spec:N #2 \cs_parameter_spec:N #2 }
          {
            \exp_args:Ne #1
              { \token_to_str:N #2 = \__kernel_exp_not:w \exp_after:wN {#2} }
          }
          {
            \msg_error:nneee { kernel } { bad-type }
              { \token_to_str:N #2 } { \token_to_meaning:N #2 } { tl }
          }
      }
  }
\cs_new_protected:Npn \tl_show:n #1
  { \iow_wrap:nnnN { >~ \tl_to_str:n {#1} . } { } { } \__tl_show:n }
\cs_generate_variant:Nn \tl_show:n { e , x }
\cs_new_protected:Npn \__tl_show:n #1
  {
    \tl_set:Nf \l__tl_internal_a_tl { \__tl_show:w #1 \s__tl_stop }
    \__kernel_iow_with:Nnn \tex_newlinechar:D { 10 }
      {
        \__kernel_iow_with:Nnn \tex_errorcontextlines:D { -1 }
```

```
          {
            \tex_showtokens:D \exp_after:wN \exp_after:wN \exp_after:wN
              { \exp_after:wN \l__tl_internal_a_tl }
          }
      }
  }
\cs_new:Npn \__tl_show:w #1 > #2 . \s__tl_stop {#2}
\cs_new_protected:Npn \tl_log:n #1
  { \iow_wrap:nnnN { > ~ \tl_to_str:n {#1} . } { } { } \iow_log:n }
\cs_generate_variant:Nn \tl_log:n { e , x }
\cs_new_protected:Npn \__kernel_chk_tl_type:NnnT #1#2#3#4
  {
    \__kernel_chk_defined:NT #1
      {
        \exp_args:Nf \tl_if_empty:nTF
          { \cs_prefix_spec:N #1 \cs_parameter_spec:N #1 }
          {
            \tl_set:Ne \l__tl_internal_a_tl {#3}
            \tl_if_eq:NNTF #1 \l__tl_internal_a_tl
              {#4}
              {
                \msg_error:nneeee { kernel } { bad-type }
                  { \token_to_str:N #1 } { \tl_to_str:N #1 }
                  {#2} { \tl_to_str:N \l__tl_internal_a_tl }
              }
          }
          {
            \msg_error:nneee { kernel } { bad-type }
              { \token_to_str:N #1 } { \token_to_meaning:N #1 } {#2}
          }
      }
  }
\scan_new:N \s__tl_nil
\scan_new:N \s__tl_mark
\scan_new:N \s__tl_stop
\tl_new:N \g_tmpa_tl
\tl_new:N \g_tmpb_tl
\tl_new:N \l_tmpa_tl
\tl_new:N \l_tmpb_tl
\cs_undefine:N \__tl_tmp:w
%% File: l3tl-build.dtx
\cs_new_protected:Npn \tl_build_begin:N #1
  { \__tl_build_begin:NN \cs_set_nopar:Npe #1 }
\cs_new_protected:Npn \tl_build_gbegin:N #1
  { \__tl_build_begin:NN \cs_gset_nopar:Npe #1 }
\cs_new_protected:Npn \__tl_build_begin:NN #1#2
```

```
  { \exp_args:Nc \__tl_build_begin:NNN { \cs_to_str:N #2 ' } #2 #1 }
\cs_new_protected:Npn \__tl_build_begin:NNN #1#2#3
  {
    #3 #1 { }
    #3 #2
      {
        \exp_not:n { \exp_end: \exp_end: \exp_end: \exp_end: }
        \exp_not:n { \__tl_build_last:NNn #3 #1 { } }
      }
  }
\cs_new_protected:Npn \tl_build_put_right:Nn #1#2
  {
    \cs_set_nopar:Npe #1
      { \__kernel_exp_not:w \exp_after:wN { \exp:w #1 #2 } }
  }
\cs_generate_variant:Nn \tl_build_put_right:Nn { Ne , Nx }
\cs_new_protected:Npn \tl_build_gput_right:Nn #1#2
  {
    \cs_gset_nopar:Npe #1
      { \__kernel_exp_not:w \exp_after:wN { \exp:w #1 #2 } }
  }
\cs_generate_variant:Nn \tl_build_gput_right:Nn { Ne , Nx }
\cs_new_protected:Npn \__tl_build_last:NNn #1#2
  {
    \if_false: { { \fi:
          \exp_end: \exp_end: \exp_end: \exp_end: \exp_end:
          \__tl_build_last:NNn #1 #2 { }
        }
      }
    \if_meaning:w \c_empty_tl #2
      \__tl_build_begin:NN #1 #2
    \fi:
    #1 #2
      {
        \__kernel_exp_not:w \exp_after:wN
          {
            \exp:w \if_false: } } \fi:
            \exp_after:wN \__tl_build_put:nn \exp_after:wN {#2}
  }
\cs_new_protected:Npn \__tl_build_put:nn #1#2 { \__tl_build_put:nw {#2} #1 }
\cs_new_protected:Npn \__tl_build_put:nw #1#2 \__tl_build_last:NNn #3#4#5
  { #2 \__tl_build_last:NNn #3 #4 { #1 #5 } }
\cs_new_protected:Npn \tl_build_put_left:Nn #1
  { \__tl_build_put_left:NNn \cs_set_nopar:Npe #1 }
\cs_generate_variant:Nn \tl_build_put_left:Nn { Ne , Nx }
\cs_new_protected:Npn \tl_build_gput_left:Nn #1
```

```
  { \__tl_build_put_left:NNn \cs_gset_nopar:Npe #1 }
\cs_generate_variant:Nn \tl_build_gput_left:Nn { Ne , Nx }
\cs_new_protected:Npn \__tl_build_put_left:NNn #1#2#3
  {
    #1 #2
      {
        \__kernel_exp_not:w \exp_after:wN
          {
            \exp:w \exp_after:wN \__tl_build_put:nn
              \exp_after:wN {#2} {#3}
          }
      }
  }
\cs_new_protected:Npn \tl_build_end:N #1
  {
    \__tl_build_get:NNN \__kernel_tl_set:Nx #1 #1
    \exp_args:Nc \__tl_build_end_loop:NN { \cs_to_str:N #1 ' } \tl_clear:N
  }
\cs_new_protected:Npn \tl_build_gend:N #1
  {
    \__tl_build_get:NNN \__kernel_tl_gset:Nx #1 #1
    \exp_args:Nc \__tl_build_end_loop:NN { \cs_to_str:N #1 ' } \tl_gclear:N
  }
\cs_new_protected:Npn \__tl_build_end_loop:NN #1#2
  {
    \if_meaning:w \c_empty_tl #1
      \exp_after:wN \use_none:nnnnnn
    \fi:
    #2 #1
    \exp_args:Nc \__tl_build_end_loop:NN { \cs_to_str:N #1 ' } #2
  }
\cs_new_protected:Npn \tl_build_get_intermediate:NN
  { \__tl_build_get:NNN \__kernel_tl_set:Nx }
\cs_new_protected:Npn \__tl_build_get:NNN #1#2#3
  { #1 #3 { \if_false: { \fi: \exp_after:wN \__tl_build_get:w #2 } } }
\cs_new:Npn \__tl_build_get:w #1 \__tl_build_last:NNn #2#3#4
  {
    \exp_not:n {#4}
    \if_meaning:w \c_empty_tl #3
      \exp_after:wN \__tl_build_get_end:w
    \fi:
    \exp_after:wN \__tl_build_get:w #3
  }
\cs_new:Npn \__tl_build_get_end:w #1#2#3
  { \__kernel_exp_not:w \exp_after:wN { \if_false: } \fi: }
%% File: l3str.dtx
```

```
\scan_new:N \s__str_mark
\scan_new:N \s__str_stop
\cs_new:Npn \__str_use_none_delimit_by_s_stop:w #1 \s__str_stop { }
\cs_new:Npn \__str_use_i_delimit_by_s_stop:nw #1 #2 \s__str_stop {#1}
\quark_new:N \q__str_recursion_tail
\quark_new:N \q__str_recursion_stop
\__kernel_quark_new_test:N \__str_if_recursion_tail_break:NN
\__kernel_quark_new_test:N \__str_if_recursion_tail_stop_do:Nn
\group_begin:
  \cs_set_protected:Npn \__str_tmp:n #1
    {
      \tl_if_blank:nF {#1}
        {
          \cs_new_eq:cc { str_ #1 :N } { tl_ #1 :N }
          \exp_args:Nc \cs_generate_variant:Nn { str_ #1 :N } { c }
          \__str_tmp:n
        }
    }
  \__str_tmp:n
    { new }
    { use }
    { clear }
    { gclear }
    { clear_new }
    { gclear_new }
    { }
\group_end:
\cs_new_eq:NN \str_set_eq:NN \tl_set_eq:NN
\cs_new_eq:NN \str_gset_eq:NN \tl_gset_eq:NN
\cs_generate_variant:Nn \str_set_eq:NN  { c , Nc , cc }
\cs_generate_variant:Nn \str_gset_eq:NN { c , Nc , cc }
\cs_new_eq:NN \str_concat:NNN \tl_concat:NNN
\cs_new_eq:NN \str_gconcat:NNN \tl_gconcat:NNN
\cs_generate_variant:Nn \str_concat:NNN  { ccc }
\cs_generate_variant:Nn \str_gconcat:NNN { ccc }
\cs_new_protected:Npn \str_set:Nn #1#2
  { \__kernel_tl_set:Nx #1 { \__kernel_tl_to_str:w {#2} } }
\cs_gset_protected:Npn \str_gset:Nn #1#2
  { \__kernel_tl_gset:Nx #1 { \__kernel_tl_to_str:w {#2} } }
\cs_new_protected:Npn \str_const:Nn #1#2
  {
    \__kernel_chk_if_free_cs:N #1
    \cs_gset_nopar:Npe #1 { \__kernel_tl_to_str:w {#2} }
  }
\cs_new_protected:Npn \str_put_left:Nn #1#2
  {
    \__kernel_tl_set:Nx #1
      { \__kernel_tl_to_str:w {#2} \__kernel_exp_not:w \exp_after:wN {#1} }
  }
\cs_new_protected:Npn \str_gput_left:Nn #1#2
  {
    \__kernel_tl_gset:Nx #1
      { \__kernel_tl_to_str:w {#2} \__kernel_exp_not:w \exp_after:wN {#1} }
  }
\cs_new_protected:Npn \str_put_right:Nn #1#2
  {
    \__kernel_tl_set:Nx #1
      { \__kernel_exp_not:w \exp_after:wN {#1} \__kernel_tl_to_str:w {#2} }
  }
\cs_new_protected:Npn \str_gput_right:Nn #1#2
  {
    \__kernel_tl_gset:Nx #1
      { \__kernel_exp_not:w \exp_after:wN {#1} \__kernel_tl_to_str:w {#2} }
  }
\cs_generate_variant:Nn \str_set:Nn        { NV , Ne , Nx , c , cV , ce , cx }
\cs_generate_variant:Nn \str_gset:Nn       { NV , Ne , Nx , c , cV , ce , cx }
\cs_generate_variant:Nn \str_const:Nn      { NV , Ne , Nx , c , cV , ce , cx }
\cs_generate_variant:Nn \str_put_left:Nn   { NV , Ne , Nx , c , cV , ce , cx }
\cs_generate_variant:Nn \str_gput_left:Nn  { NV , Ne , Nx , c , cV , ce , cx }
\cs_generate_variant:Nn \str_put_right:Nn  { NV , Ne , Nx , c , cV , ce , cx }
\cs_generate_variant:Nn \str_gput_right:Nn { NV , Ne , Nx , c , cV , ce , cx }
\cs_new_protected:Npn \str_replace_once:Nnn
  { \__str_replace:NNNnn \prg_do_nothing: \__kernel_tl_set:Nx  }
\cs_new_protected:Npn \str_greplace_once:Nnn
  { \__str_replace:NNNnn \prg_do_nothing: \__kernel_tl_gset:Nx }
\cs_new_protected:Npn \str_replace_all:Nnn
  { \__str_replace:NNNnn \__str_replace_next:w \__kernel_tl_set:Nx  }
\cs_new_protected:Npn \str_greplace_all:Nnn
  { \__str_replace:NNNnn \__str_replace_next:w \__kernel_tl_gset:Nx }
\cs_generate_variant:Nn \str_replace_once:Nnn  { c }
\cs_generate_variant:Nn \str_greplace_once:Nnn { c }
\cs_generate_variant:Nn \str_replace_all:Nnn   { c }
\cs_generate_variant:Nn \str_greplace_all:Nnn  { c }
\cs_new_protected:Npn \__str_replace:NNNnn #1#2#3#4#5
  {
    \tl_if_empty:nTF {#4}
      {
        \msg_error:nne { kernel } { empty-search-pattern } {#5}
      }
      {
        \use:e
          {
            \exp_not:n { \__str_replace_aux:NNNnnn #1 #2 #3 }
              { \tl_to_str:N #3 }
              { \tl_to_str:n {#4} } { \tl_to_str:n {#5} }
          }
      }
  }
\cs_new_protected:Npn \__str_replace_aux:NNNnnn #1#2#3#4#5#6
  {
    \cs_set:Npn \__str_replace_next:w ##1 #5 { ##1 #6 #1 }
    #2 #3
      {
        \__str_replace_next:w
        #4
        \__str_use_none_delimit_by_s_stop:w
        #5
        \s__str_stop
      }
  }
\cs_new_eq:NN \__str_replace_next:w ?
\cs_new_protected:Npn \str_remove_once:Nn #1#2
  { \str_replace_once:Nnn #1 {#2} { } }
\cs_new_protected:Npn \str_gremove_once:Nn #1#2
  { \str_greplace_once:Nnn #1 {#2} { } }
\cs_generate_variant:Nn \str_remove_once:Nn  { c }
\cs_generate_variant:Nn \str_gremove_once:Nn { c }
\cs_new_protected:Npn \str_remove_all:Nn #1#2
  { \str_replace_all:Nnn #1 {#2} { } }
\cs_new_protected:Npn \str_gremove_all:Nn #1#2
  { \str_greplace_all:Nnn #1 {#2} { } }
\cs_generate_variant:Nn \str_remove_all:Nn  { c }
\cs_generate_variant:Nn \str_gremove_all:Nn { c }
\prg_new_eq_conditional:NNn \str_if_exist:N \tl_if_exist:N
  { p , T , F , TF }
\prg_new_eq_conditional:NNn \str_if_exist:c \tl_if_exist:c
  { p , T , F , TF }
\prg_new_eq_conditional:NNn \str_if_empty:N \tl_if_empty:N
  { p , T , F , TF }
\prg_new_eq_conditional:NNn \str_if_empty:c \tl_if_empty:c
  { p , T , F , TF }
\prg_new_eq_conditional:NNn \str_if_empty:n \tl_if_empty:n
  { p , T , F , TF }
\cs_new_eq:NN \__str_if_eq:nn \tex_strcmp:D
\prg_new_conditional:Npnn \str_compare:nNn #1#2#3 { p , T , F , TF }
  {
    \if_int_compare:w
      \__str_if_eq:nn { \exp_not:n {#1} } { \exp_not:n {#3} }
      #2 \c_zero_int
      \prg_return_true: \else: \prg_return_false: \fi:
  }
\prg_new_conditional:Npnn \str_compare:eNe #1#2#3 { p , T , F , TF }
  {
    \if_int_compare:w \__str_if_eq:nn {#1} {#3} #2 \c_zero_int
      \prg_return_true: \else: \prg_return_false: \fi:
  }
\prg_new_conditional:Npnn \str_if_eq:nn #1#2 { p , T , F , TF }
  {
    \if:w 0 \__str_if_eq:nn { \exp_not:n {#1} } { \exp_not:n {#2} }
      \prg_return_true: \else: \prg_return_false: \fi:
  }
\prg_generate_conditional_variant:Nnn \str_if_eq:nn
  { V , v , o , nV , no , VV , nv } { p , T , F , TF }
\prg_new_conditional:Npnn \str_if_eq:ee #1#2 { p , T , F , TF }
  {
    \if:w 0 \__str_if_eq:nn {#1} {#2}
      \prg_return_true: \else: \prg_return_false: \fi:
  }
\prg_new_conditional:Npnn \str_if_eq:NN #1#2 { p , TF , T , F }
  {
    \if:w 0 \__str_if_eq:nn { \tl_to_str:N #1 } { \tl_to_str:N #2 }
      \prg_return_true: \else: \prg_return_false: \fi:
  }
\prg_generate_conditional_variant:Nnn \str_if_eq:NN
  { c , Nc , cc } { T , F , TF , p }
\prg_new_protected_conditional:Npnn \str_if_in:Nn #1#2 { T , F , TF }
  {
    \use:e
      { \tl_if_in:nnTF { \tl_to_str:N #1 } { \tl_to_str:n {#2} } }
      { \prg_return_true: } { \prg_return_false: }
  }
\prg_generate_conditional_variant:Nnn \str_if_in:Nn
  { c } { T , F , TF }
\prg_new_protected_conditional:Npnn \str_if_in:nn #1#2 { T , F , TF }
  {
    \use:e
      { \tl_if_in:nnTF { \tl_to_str:n {#1} } { \tl_to_str:n {#2} } }
      { \prg_return_true: } { \prg_return_false: }
  }
\cs_new:Npn \str_case:nn #1#2
  {
    \exp:w
    \__str_case:nnTF {#1} {#2} { } { }
  }
```

```
\cs_new:Npn \str_case:nnT #1#2#3
  {
    \exp:w
    \__str_case:nnTF {#1} {#2} {#3} { }
  }
\cs_new:Npn \str_case:nnF #1#2
  {
    \exp:w
    \__str_case:nnTF {#1} {#2} { }
  }
\cs_new:Npn \str_case:nnTF #1#2
  {
    \exp:w
    \__str_case:nnTF {#1} {#2}
  }
\cs_new:Npn \__str_case:nnTF #1#2#3#4
  { \__str_case:nw {#1} #2 {#1} { } \s__str_mark {#3} \s__str_mark {#4} \s__str_stop }
\cs_generate_variant:Nn \str_case:nn   { V , o , e , nV , nv , ne }
\prg_generate_conditional_variant:Nnn \str_case:nn
  { V , o , e , nV , nv , ne } { T , F , TF }
\cs_new_eq:NN \str_case:Nn   \str_case:Vn
\cs_new_eq:NN \str_case:NnT  \str_case:VnT
\cs_new_eq:NN \str_case:NnF  \str_case:VnF
\cs_new_eq:NN \str_case:NnTF \str_case:VnTF
\cs_new:Npn \__str_case:nw #1#2#3
  {
    \str_if_eq:nnTF {#1} {#2}
      { \__str_case_end:nw {#3} }
      { \__str_case:nw {#1} }
  }
\cs_new:Npn \str_case_e:nn #1#2
  {
    \exp:w
    \__str_case_e:nnTF {#1} {#2} { } { }
  }
\cs_new:Npn \str_case_e:nnT #1#2#3
  {
    \exp:w
    \__str_case_e:nnTF {#1} {#2} {#3} { }
  }
\cs_new:Npn \str_case_e:nnF #1#2
  {
    \exp:w
    \__str_case_e:nnTF {#1} {#2} { }
  }
\cs_new:Npn \str_case_e:nnTF #1#2
```

```
  {
    \exp:w
    \__str_case_e:nnTF {#1} {#2}
  }
\cs_new:Npn \__str_case_e:nnTF #1#2#3#4
  { \__str_case_e:nw {#1} #2 {#1} { } \s__str_mark {#3} \s__str_mark {#4} \s__str_stop }
\cs_generate_variant:Nn \str_case_e:nn { e }
\prg_generate_conditional_variant:Nnn \str_case_e:nn { e } { T , F , TF }
\cs_new:Npn \__str_case_e:nw #1#2#3
  {
    \str_if_eq:eeTF {#1} {#2}
      { \__str_case_end:nw {#3} }
      { \__str_case_e:nw {#1} }
  }
\cs_new:Npn \__str_case_end:nw #1#2#3 \s__str_mark #4#5 \s__str_stop
  { \exp_end: #1 #4 }
\cs_new:Npn \str_map_function:nN #1#2
  {
    \exp_after:wN \__str_map_function:w
    \exp_after:wN \__str_map_function:nn \exp_after:wN #2
      \__kernel_tl_to_str:w {#1}
      \q__str_recursion_tail ? ~
    \prg_break_point:Nn \str_map_break: { }
  }
\cs_new:Npn \str_map_function:NN
  { \exp_args:No \str_map_function:nN }
\cs_new:Npn \__str_map_function:w #1 ~
  { #1 { ~ { ~ } \__str_map_function:w } }
\cs_new:Npn \__str_map_function:nn #1#2
  {
    \if_meaning:w \q__str_recursion_tail #2
      \exp_after:wN \str_map_break:
    \fi:
    #1 #2 \__str_map_function:nn {#1}
  }
\cs_generate_variant:Nn \str_map_function:NN { c }
\cs_new_protected:Npn \str_map_inline:nn #1#2
  {
    \int_gincr:N \g__kernel_prg_map_int
    \cs_gset_protected:cpn
      { __str_map_ \int_use:N \g__kernel_prg_map_int :w } ##1 {#2}
    \use:e
      {
        \exp_not:N \__str_map_inline:NN
        \exp_not:c { __str_map_ \int_use:N \g__kernel_prg_map_int :w }
        \__kernel_str_to_other_fast:n {#1}
      }
      \q__str_recursion_tail
    \prg_break_point:Nn \str_map_break:
      { \int_gdecr:N \g__kernel_prg_map_int }
  }
\cs_new_protected:Npn \str_map_inline:Nn
  { \exp_args:No \str_map_inline:nn }
\cs_generate_variant:Nn \str_map_inline:Nn { c }
\cs_new:Npn \__str_map_inline:NN #1#2
  {
    \__str_if_recursion_tail_break:NN #2 \str_map_break:
    \exp_args:No #1 { \token_to_str:N #2 }
    \__str_map_inline:NN #1
  }
\cs_new_protected:Npn \str_map_variable:nNn #1#2#3
  {
    \use:e
      {
        \exp_not:n { \__str_map_variable:NnN #2 {#3} }
        \__kernel_str_to_other_fast:n {#1}
      }
      \q__str_recursion_tail
    \prg_break_point:Nn \str_map_break: { }
  }
\cs_new_protected:Npn \str_map_variable:NNn
  { \exp_args:No \str_map_variable:nNn }
\cs_new_protected:Npn \__str_map_variable:NnN #1#2#3
  {
    \__str_if_recursion_tail_break:NN #3 \str_map_break:
    \str_set:Nn #1 {#3}
    \use:n {#2}
    \__str_map_variable:NnN #1 {#2}
  }
\cs_generate_variant:Nn \str_map_variable:NNn { c }
\cs_new:Npn \str_map_break:
  { \prg_map_break:Nn \str_map_break: { } }
\cs_new:Npn \str_map_break:n
  { \prg_map_break:Nn \str_map_break: }
\cs_new:Npn \str_map_tokens:nn #1#2
  {
    \exp_args:Nno \use:nn
      { \__str_map_function:w \__str_map_function:nn {#2} }
      { \__kernel_tl_to_str:w {#1} }
      \q__str_recursion_tail ? ~
    \prg_break_point:Nn \str_map_break: { }
  }
```

```
\cs_new:Npn \str_map_tokens:Nn { \exp_args:No \str_map_tokens:nn }
\cs_generate_variant:Nn \str_map_tokens:Nn { c }
\cs_new:Npn \__kernel_str_to_other:n #1
  {
    \exp_after:wN \__str_to_other_loop:w
      \tl_to_str:n {#1} ~ A ~ A ~ A ~ A ~ A ~ A ~ A ~ A ~ \s__str_mark \s__str_stop
  }
\group_begin:
\tex_lccode:D `\* = `\␣ %
\tex_lccode:D `\A = `\A %
\tex_lowercase:D
  {
    \group_end:
    \cs_new:Npn \__str_to_other_loop:w
      #1 ~ #2 ~ #3 ~ #4 ~ #5 ~ #6 ~ #7 ~ #8 ~ #9 \s__str_stop
      {
        \if_meaning:w A #8
          \__str_to_other_end:w
        \fi:
        \__str_to_other_loop:w
        #9 #1 * #2 * #3 * #4 * #5 * #6 * #7 * #8 * \s__str_stop
      }
    \cs_new:Npn \__str_to_other_end:w \fi: #1 \s__str_mark #2 * A #3 \s__str_stop
      { \fi: #2 }
  }
\cs_new:Npn \__kernel_str_to_other_fast:n #1
  {
    \exp_after:wN \__str_to_other_fast_loop:w \tl_to_str:n {#1} ~
      A ~ A ~ A ~ A ~ A ~ A ~ A ~ A ~ A ~ \s__str_stop
  }
\group_begin:
\tex_lccode:D `\* = `\␣ %
\tex_lccode:D `\A = `\A %
\tex_lowercase:D
  {
    \group_end:
    \cs_new:Npn \__str_to_other_fast_loop:w
      #1 ~ #2 ~ #3 ~ #4 ~ #5 ~ #6 ~ #7 ~ #8 ~ #9 ~
      {
        \if_meaning:w A #9
          \__str_to_other_fast_end:w
        \fi:
        #1 * #2 * #3 * #4 * #5 * #6 * #7 * #8 * #9
        \__str_to_other_fast_loop:w *
      }
    \cs_new:Npn \__str_to_other_fast_end:w #1 * A #2 \s__str_stop {#1}
  }
\cs_new_eq:NN \__str_sep: \__kernel_int_sep:
\cs_new:Npn \str_item:Nn { \exp_args:No \str_item:nn }
\cs_generate_variant:Nn \str_item:Nn { c }
\cs_new:Npn \str_item:nn #1#2
  {
    \exp_args:Nf \tl_to_str:n
      {
        \exp_args:Nf \__str_item:nn
          { \__kernel_str_to_other:n {#1} } {#2}
      }
  }
\cs_new:Npn \str_item_ignore_spaces:nn #1
  { \exp_args:No \__str_item:nn { \tl_to_str:n {#1} } }
\cs_new:Npn \__str_item:nn #1#2
  {
    \exp_after:wN \__str_item:w
    \int_value:w \int_eval:n {#2} \exp_after:wN \__str_sep:
    \int_value:w \__str_count:n {#1} \__str_sep:
    #1 \s__str_stop
  }
\cs_new:Npn \__str_item:w #1 \__str_sep: #2 \__str_sep:
  {
    \int_compare:nNnTF {#1} < 0
      {
        \int_compare:nNnTF {#1} < {-#2}
          { \__str_use_none_delimit_by_s_stop:w }
          {
            \exp_after:wN \__str_use_i_delimit_by_s_stop:nw
            \exp:w \exp_after:wN \__str_skip_exp_end:w
              \int_value:w \int_eval:n { #1 + #2 } \__str_sep:
          }
      }
      {
        \int_compare:nNnTF {#1} > {#2}
          { \__str_use_none_delimit_by_s_stop:w }
          {
            \exp_after:wN \__str_use_i_delimit_by_s_stop:nw
            \exp:w \__str_skip_exp_end:w #1 \__str_sep: { }
          }
      }
  }
\cs_new:Npn \__str_skip_exp_end:w #1 \__str_sep:
  {
    \if_int_compare:w #1 > 8 \exp_stop_f:
      \exp_after:wN \__str_skip_loop:wNNNNNNNN
    \else:
      \exp_after:wN \__str_skip_end:w
      \int_value:w \int_eval:w
    \fi:
    #1 \__str_sep:
  }
\cs_new:Npn \__str_skip_loop:wNNNNNNNN #1 \__str_sep: #2#3#4#5#6#7#8#9
  {
    \exp_after:wN \__str_skip_exp_end:w
      \int_value:w \int_eval:n { #1 - 8 } \__str_sep:
  }
\cs_new:Npn \__str_skip_end:w #1 \__str_sep:
  {
    \exp_after:wN \__str_skip_end:NNNNNNNN
    \if_case:w #1 \exp_stop_f: \or: \or: \or: \or: \or: \or: \or: \or:
  }
\cs_new:Npn \__str_skip_end:NNNNNNNN #1#2#3#4#5#6#7#8 { \fi: \exp_end: }
\cs_new:Npn \str_range:Nnn { \exp_args:No \str_range:nnn }
\cs_generate_variant:Nn \str_range:Nnn { c }
\cs_new:Npn \str_range:nnn #1#2#3
  {
    \exp_args:Nf \tl_to_str:n
      {
        \exp_args:Nf \__str_range:nnn
          { \__kernel_str_to_other:n {#1} } {#2} {#3}
      }
  }
\cs_new:Npn \str_range_ignore_spaces:nnn #1
  { \exp_args:No \__str_range:nnn { \tl_to_str:n {#1} } }
\cs_new:Npn \__str_range:nnn #1#2#3
  {
    \exp_after:wN \__str_range:w
    \int_value:w \__str_count:n {#1} \exp_after:wN \__str_sep:
    \int_value:w \int_eval:n { (#2) - 1 } \exp_after:wN \__str_sep:
    \int_value:w \int_eval:n {#3} \__str_sep:
    #1 \s__str_stop
  }
\cs_new:Npn \__str_range:w #1 \__str_sep: #2 \__str_sep: #3 \__str_sep:
  {
    \exp_args:Nf \__str_range:nnw
      { \__str_range_normalize:nn {#2} {#1} }
      { \__str_range_normalize:nn {#3} {#1} }
  }
\cs_new:Npn \__str_range:nnw #1#2
  {
    \exp_after:wN \__str_collect_delimit_by_q_stop:w
    \int_value:w \int_eval:n { #2 - #1 } \exp_after:wN \__str_sep:
    \exp:w \__str_skip_exp_end:w #1 \__str_sep:
  }
\cs_new:Npn \__str_range_normalize:nn #1#2
  {
    \int_eval:n
      {
        \if_int_compare:w #1 < \c_zero_int
          \if_int_compare:w #1 < -#2 \exp_stop_f:
            0
          \else:
            #1 + #2 + 1
          \fi:
        \else:
          \if_int_compare:w #1 < #2 \exp_stop_f:
            #1
          \else:
            #2
          \fi:
        \fi:
      }
  }
\cs_new:Npn \__str_collect_delimit_by_q_stop:w #1 \__str_sep:
  { \__str_collect_loop:wn #1 \__str_sep: { } }
\cs_new:Npn \__str_collect_loop:wn #1 \__str_sep:
  {
    \if_int_compare:w #1 > 7 \exp_stop_f:
      \exp_after:wN \__str_collect_loop:wnNNNNNNN
    \else:
      \exp_after:wN \__str_collect_end:wn
    \fi:
    #1 \__str_sep:
  }
\cs_new:Npn \__str_collect_loop:wnNNNNNNN #1 \__str_sep: #2 #3#4#5#6#7#8#9
  {
    \exp_after:wN \__str_collect_loop:wn
    \int_value:w \int_eval:n { #1 - 7 } \__str_sep:
    { #2 #3#4#5#6#7#8#9 }
  }
\cs_new:Npn \__str_collect_end:wn #1 \__str_sep:
  {
    \exp_after:wN \__str_collect_end:nnnnnnnnw
    \if_case:w \if_int_compare:w #1 > \c_zero_int
      #1 \else: 0 \fi: \exp_stop_f:
      \or: \or: \or: \or: \or: \or: \fi:
  }
```

```
\cs_new:Npn \__str_collect_end:nnnnnnnnw #1#2#3#4#5#6#7#8 #9 \s__str_stop
  { #1#2#3#4#5#6#7#8 }
\cs_new:Npn \str_count_spaces:N
  { \exp_args:No \str_count_spaces:n }
\cs_generate_variant:Nn \str_count_spaces:N { c }
\cs_new:Npn \str_count_spaces:n #1
  {
    \int_eval:n
      {
        \exp_after:wN \__str_count_spaces_loop:w
        \tl_to_str:n {#1} ~
        X 7 ~ X 6 ~ X 5 ~ X 4 ~ X 3 ~ X 2 ~ X 1 ~ X 0 ~ X -1 ~
        \s__str_stop
      }
  }
\cs_new:Npn \__str_count_spaces_loop:w #1~#2~#3~#4~#5~#6~#7~#8~#9~
  {
    \if_meaning:w X #9
      \__str_use_i_delimit_by_s_stop:nw
    \fi:
    9 + \__str_count_spaces_loop:w
  }
\cs_new:Npn \str_count:N { \exp_args:No \str_count:n }
\cs_generate_variant:Nn \str_count:N { c }
\cs_new:Npn \str_count:n #1
  {
    \__str_count_aux:n
      {
        \str_count_spaces:n {#1}
        + \exp_after:wN \__str_count_loop:NNNNNNNNN \tl_to_str:n {#1}
      }
  }
\cs_new:Npn \__str_count:n #1
  {
    \__str_count_aux:n
      { \__str_count_loop:NNNNNNNNN #1 }
  }
\cs_new:Npn \str_count_ignore_spaces:n #1
  {
    \__str_count_aux:n
      { \exp_after:wN \__str_count_loop:NNNNNNNNN \tl_to_str:n {#1} }
  }
\cs_new:Npn \__str_count_aux:n #1
  {
    \int_eval:n
      {
        #1
        { X 8 } { X 7 } { X 6 }
        { X 5 } { X 4 } { X 3 }
        { X 2 } { X 1 } { X 0 }
        \s__str_stop
      }
  }
\cs_new:Npn \__str_count_loop:NNNNNNNNN #1#2#3#4#5#6#7#8#9
  {
    \if_meaning:w X #9
      \exp_after:wN \__str_use_none_delimit_by_s_stop:w
    \fi:
    9 + \__str_count_loop:NNNNNNNNN
  }
\cs_new:Npn \str_head:N { \exp_args:No \str_head:n }
\cs_generate_variant:Nn \str_head:N { c }
\cs_new:Npn \str_head:n #1
  {
    \exp_after:wN \__str_head:w
    \tl_to_str:n {#1}
    { { } } ~ \s__str_stop
  }
\cs_new:Npn \__str_head:w #1 ~ %
  { \__str_use_i_delimit_by_s_stop:nw #1 { ~ } }
\cs_new:Npn \str_head_ignore_spaces:n #1
  {
    \exp_after:wN \__str_use_i_delimit_by_s_stop:nw
    \tl_to_str:n {#1} { } \s__str_stop
  }
\cs_new:Npn \str_tail:N { \exp_args:No \str_tail:n }
\cs_generate_variant:Nn \str_tail:N { c }
\cs_new:Npn \str_tail:n #1
  {
    \exp_after:wN \__str_tail_auxi:w
    \reverse_if:N \if_charcode:w
        \scan_stop: \tl_to_str:n {#1} X X \s__str_stop
  }
\cs_new:Npn \__str_tail_auxi:w #1 X #2 \s__str_stop { \fi: #1 }
\cs_new:Npn \str_tail_ignore_spaces:n #1
  {
    \exp_after:wN \__str_tail_auxii:w
    \tl_to_str:n {#1} \s__str_mark \s__str_mark \s__str_stop
  }
\cs_new:Npn \__str_tail_auxii:w #1 #2 \s__str_mark #3 \s__str_stop { #2 }
\cs_new:Npn \str_casefold:n  #1 { \__str_change_case:nn {#1} { casefold } }
\cs_new:Npn \str_lowercase:n #1 { \__str_change_case:nn {#1} { lowercase } }
```

```
\cs_new:Npn \str_uppercase:n #1 { \__str_change_case:nn {#1} { uppercase } }
\cs_generate_variant:Nn \str_casefold:n  { V }
\cs_generate_variant:Nn \str_lowercase:n { f }
\cs_generate_variant:Nn \str_uppercase:n { f }
\cs_new:Npn \__str_change_case:nn #1
  {
    \exp_after:wN \__str_change_case_aux:nn \exp_after:wN
      { \tl_to_str:n {#1} }
  }
\cs_new:Npn \__str_change_case_aux:nn #1#2
  {
    \__str_change_case_loop:nw {#2} #1 \q__str_recursion_tail \q__str_recursion_stop
      \__str_change_case_result:n { }
  }
\cs_new:Npn \__str_change_case_output:nw #1#2 \__str_change_case_result:n #3
  { #2 \__str_change_case_result:n { #3 #1 } }
\cs_generate_variant:Nn  \__str_change_case_output:nw { f }
\cs_new:Npn \__str_change_case_end:wn #1 \__str_change_case_result:n #2
  { \tl_to_str:n {#2} }
\cs_new:Npn \__str_change_case_loop:nw #1#2 \q__str_recursion_stop
  {
    \tl_if_head_is_space:nTF {#2}
      { \__str_change_case_space:n }
      { \__str_change_case_char:nN }
    {#1} #2 \q__str_recursion_stop
  }
\exp_last_unbraced:NNNNo
  \cs_new:Npn \__str_change_case_space:n #1 \c_space_tl
  {
    \__str_change_case_output:nw { ~ }
    \__str_change_case_loop:nw {#1}
  }
\cs_new:Npn \__str_change_case_char:nN #1#2
  {
    \__str_if_recursion_tail_stop_do:Nn #2
      { \__str_change_case_end:wn }
    \__str_change_case_codepoint:nN {#1} #2
  }
\if_int_compare:w 0
  \cs_if_exist:NT \tex_XeTeXversion:D { 1 }
  \cs_if_exist:NT \tex_luatexversion:D { 1 }
  > 0 \exp_stop_f:
  \cs_new:Npn \__str_change_case_codepoint:nN #1#2
    { \__str_change_case_char:fnn { \int_eval:n {`#2} } {#1} {#2} }
\else:
    \cs_new:Npe \__str_change_case_codepoint:nN #1#2
      {
        \exp_not:N \int_compare:nNnTF {`#2} > { "80 }
          {
            \cs_if_exist:NTF \tex_pdftexversion:D
              { \exp_not:N \__str_change_case_char_auxi:nN }
              {
                \exp_not:N \int_compare:nNnTF {`#2} > { "FF }
                  { \exp_not:N \__str_change_case_char_auxii:nN }
                  { \exp_not:N \__str_change_case_char_auxi:nN }
              }
          }
          { \exp_not:N \__str_change_case_char_auxii:nN }
            {#1} #2
      }
    \cs_new:Npn \__str_change_case_char_auxi:nN #1#2
      {
        \int_compare:nNnTF {`#2} < { "E0 }
          { \__str_change_case_codepoint:nNN }
          {
            \int_compare:nNnTF {`#2} < { "F0 }
              { \__str_change_case_codepoint:nNNN }
              { \__str_change_case_codepoint:nNNNN }
          }
            {#1} #2
      }
    \cs_new:Npn \__str_change_case_char_auxii:nN #1#2
      { \__str_change_case_char:fnn { \int_eval:n {`#2} } {#1} {#2} }
    \cs_new:Npn \__str_change_case_codepoint:nNN #1#2#3
      {
        \__str_change_case_char:fnn
          { \int_eval:n { (`#2 - "C0) * "40 + `#3 - "80 } }
          {#1} {#2#3}
      }
    \cs_new:Npn \__str_change_case_codepoint:nNNN #1#2#3#4
      {
        \__str_change_case_char:fnn
          {
            \int_eval:n
              { (`#2 - "E0) * "1000 + (`#3 - "80) * "40 + `#4 - "80 }
          }
          {#1} {#2#3#4}
      }
    \cs_new:Npn \__str_change_case_codepoint:nNNNN #1#2#3#4#5
      {
        \__str_change_case_char:fnn
          {
            \int_eval:n
              {
                  (`#2 - "F0) * "40000
                + (`#3 - "80) * "1000
                + (`#4 - "80) * "40
                + `#5 - "80
              }
          }
          {#1} {#2#3#4#5}
      }
\fi:
\cs_new:Npn \__str_change_case_char:nnn #1#2#3
  {
    \__str_change_case_output:fw
      {
        \exp_args:Ne \__str_change_case_char_aux:nnn
          { \__kernel_codepoint_case:nn {#2} {#1} } {#1} {#3}
      }
    \__str_change_case_loop:nw {#2}
  }
\cs_generate_variant:Nn \__str_change_case_char:nnn { f }
\cs_new:Npn \__str_change_case_char_aux:nnn #1#2#3
  {
    \use:e { \__str_change_case_char:nnnnn #1 {#2} {#3} }
  }
\cs_new:Npn \__str_change_case_char:nnnnn #1#2#3#4#5
  {
    \int_compare:nNnTF {#1} = {#4}
      { \tl_to_str:n {#5} }
      {
        \codepoint_str_generate:n {#1}
        \tl_if_blank:nF {#2}
          {
            \codepoint_str_generate:n {#2}
            \tl_if_blank:nF {#3}
              { \codepoint_str_generate:n {#3} }
          }
      }
  }
\cs_new:Npn \str_mdfive_hash:n #1 { \tex_mdfivesum:D { \tl_to_str:n {#1} } }
\cs_new:Npn \str_mdfive_hash:e #1 { \tex_mdfivesum:D {#1} }
\str_const:Ne \c_ampersand_str   { \cs_to_str:N \& }
\str_const:Ne \c_atsign_str      { \cs_to_str:N \@ }
\str_const:Ne \c_backslash_str   { \cs_to_str:N \\ }
\str_const:Ne \c_left_brace_str  { \cs_to_str:N \{ }
\str_const:Ne \c_right_brace_str { \cs_to_str:N \} }
```

```
\str_const:Ne \c_circumflex_str  { \cs_to_str:N \^ }
\str_const:Ne \c_colon_str       { \cs_to_str:N \: }
\str_const:Ne \c_dollar_str      { \cs_to_str:N \$ }
\str_const:Ne \c_hash_str        { \cs_to_str:N \# }
\str_const:Ne \c_percent_str     { \cs_to_str:N \% }
\str_const:Ne \c_tilde_str       { \cs_to_str:N \~ }
\str_const:Ne \c_underscore_str  { \cs_to_str:N \_ }
\str_const:Ne \c_zero_str        { 0 }
\cs_new_eq:NN \c_empty_str \c_empty_tl
\str_new:N \l_tmpa_str
\str_new:N \l_tmpb_str
\str_new:N \g_tmpa_str
\str_new:N \g_tmpb_str
\cs_new_eq:NN \str_show:n \tl_show:n
\cs_new_protected:Npn \str_show:N #1
  {
    \__kernel_chk_tl_type:NnnT #1 { str } { \tl_to_str:N #1 }
      { \tl_show:N #1 }
  }
\cs_generate_variant:Nn \str_show:N { c }
\cs_new_eq:NN \str_log:n \tl_log:n
\cs_new_protected:Npn \str_log:N #1
  {
    \__kernel_chk_tl_type:NnnT #1 { str } { \tl_to_str:N #1 }
      { \tl_log:N #1 }
  }
\cs_generate_variant:Nn \str_log:N { c }
%% File: l3seq.dtx
\scan_new:N \s__seq
\scan_new:N \s__seq_mark
\scan_new:N \s__seq_stop
\cs_new:Npn \__seq_item:n
  {
    \msg_expandable_error:nn { seq } { misused }
    \use_none:n
  }
\tl_new:N \l__seq_internal_a_tl
\tl_new:N \l__seq_internal_b_tl
\cs_new_eq:NN \__seq_tmp:w ?
\tl_const:Nn \c_empty_seq { \s__seq }
\cs_new_protected:Npn \seq_new:N #1
  {
    \__kernel_chk_if_free_cs:N #1
    \cs_gset_eq:NN #1 \c_empty_seq
  }
\cs_generate_variant:Nn \seq_new:N { c }
```

```
\cs_new_protected:Npn \seq_clear:N  #1
  { \seq_set_eq:NN #1 \c_empty_seq }
\cs_generate_variant:Nn \seq_clear:N  { c }
\cs_new_protected:Npn \seq_gclear:N #1
  { \seq_gset_eq:NN #1 \c_empty_seq }
\cs_generate_variant:Nn \seq_gclear:N { c }
\cs_new_protected:Npn \seq_clear_new:N  #1
  { \seq_if_exist:NTF #1 { \seq_clear:N #1 } { \seq_new:N #1 } }
\cs_generate_variant:Nn \seq_clear_new:N  { c }
\cs_new_protected:Npn \seq_gclear_new:N #1
  { \seq_if_exist:NTF #1 { \seq_gclear:N #1 } { \seq_new:N #1 } }
\cs_generate_variant:Nn \seq_gclear_new:N { c }
\cs_new_eq:NN \seq_set_eq:NN  \tl_set_eq:NN
\cs_new_eq:NN \seq_set_eq:Nc  \tl_set_eq:Nc
\cs_new_eq:NN \seq_set_eq:cN  \tl_set_eq:cN
\cs_new_eq:NN \seq_set_eq:cc  \tl_set_eq:cc
\cs_new_eq:NN \seq_gset_eq:NN \tl_gset_eq:NN
\cs_new_eq:NN \seq_gset_eq:Nc \tl_gset_eq:Nc
\cs_new_eq:NN \seq_gset_eq:cN \tl_gset_eq:cN
\cs_new_eq:NN \seq_gset_eq:cc \tl_gset_eq:cc
\cs_new_protected:Npn \seq_set_from_clist:NN #1#2
  {
    \__kernel_tl_set:Nx #1
      { \s__seq \clist_map_function:NN #2 \__seq_wrap_item:n }
  }
\cs_new_protected:Npn \seq_set_from_clist:Nn #1#2
  {
    \__kernel_tl_set:Nx #1
      { \s__seq \clist_map_function:nN {#2} \__seq_wrap_item:n }
  }
\cs_new_protected:Npn \seq_gset_from_clist:NN #1#2
  {
    \__kernel_tl_gset:Nx #1
      { \s__seq \clist_map_function:NN #2 \__seq_wrap_item:n }
  }
\cs_new_protected:Npn \seq_gset_from_clist:Nn #1#2
  {
    \__kernel_tl_gset:Nx #1
      { \s__seq \clist_map_function:nN {#2} \__seq_wrap_item:n }
  }
\cs_generate_variant:Nn \seq_set_from_clist:NN  {     Nc }
\cs_generate_variant:Nn \seq_set_from_clist:NN  { c , cc }
\cs_generate_variant:Nn \seq_set_from_clist:Nn  { c      }
\cs_generate_variant:Nn \seq_gset_from_clist:NN {     Nc }
\cs_generate_variant:Nn \seq_gset_from_clist:NN { c , cc }
\cs_generate_variant:Nn \seq_gset_from_clist:Nn { c      }
```

```
\cs_new_protected:Npn \seq_const_from_clist:Nn #1#2
  {
    \tl_const:Ne #1
      { \s__seq \clist_map_function:nN {#2} \__seq_wrap_item:n }
  }
\cs_generate_variant:Nn \seq_const_from_clist:Nn { c }
\cs_new_protected:Npn \seq_set_split:Nnn
  { \__seq_set_split:NNNnn \__kernel_tl_set:Nx \tl_trim_spaces:n }
\cs_new_protected:Npn \seq_gset_split:Nnn
  { \__seq_set_split:NNNnn \__kernel_tl_gset:Nx \tl_trim_spaces:n }
\cs_new_protected:Npn \seq_set_split_keep_spaces:Nnn
  { \__seq_set_split:NNNnn \__kernel_tl_set:Nx \exp_not:n }
\cs_new_protected:Npn \seq_gset_split_keep_spaces:Nnn
  { \__seq_set_split:NNNnn \__kernel_tl_gset:Nx \exp_not:n }
\cs_new_protected:Npn \__seq_set_split:NNNnn #1#2#3#4#5
  {
    \tl_if_empty:nTF {#4}
      {
        \tl_set:Nn \l__seq_internal_a_tl
          { \tl_map_function:nN {#5} \__seq_wrap_item:n }
      }
      {
        \tl_set:Nn \l__seq_internal_a_tl
          {
            \__seq_set_split:Nw #2 \prg_do_nothing:
            #5
            \__seq_set_split_end:
          }
        \tl_replace_all:Nnn \l__seq_internal_a_tl {#4}
          {
            \__seq_set_split_end:
            \__seq_set_split:Nw #2 \prg_do_nothing:
          }
        \__kernel_tl_set:Nx \l__seq_internal_a_tl { \l__seq_internal_a_tl }
      }
    #1 #3 { \s__seq \l__seq_internal_a_tl }
  }
\cs_new:Npn \__seq_set_split:Nw #1#2 \__seq_set_split_end:
  {
    \exp_not:N \__seq_set_split:w
    \exp_args:No #1 {#2}
    \exp_not:N \__seq_set_split_end:
  }
\cs_new:Npn \__seq_set_split:w #1 \__seq_set_split_end:
  { \__seq_wrap_item:n {#1} }
\cs_generate_variant:Nn \seq_set_split:Nnn  { NV , NnV , NVV , Nne , Nee }
```

```
\cs_generate_variant:Nn \seq_set_split:Nnn  { Nnx , Nxx }
\cs_generate_variant:Nn \seq_gset_split:Nnn { NV , NnV , NVV , Nne , Nee }
\cs_generate_variant:Nn \seq_gset_split:Nnn { Nnx , Nxx }
\cs_generate_variant:Nn \seq_set_split_keep_spaces:Nnn  { NnV }
\cs_generate_variant:Nn \seq_gset_split_keep_spaces:Nnn { NnV }
\cs_new_protected:Npn \seq_set_filter:NNn
  { \__seq_set_filter:NNNn \__kernel_tl_set:Nx }
\cs_new_protected:Npn \seq_gset_filter:NNn
  { \__seq_set_filter:NNNn \__kernel_tl_gset:Nx }
\cs_new_protected:Npn \__seq_set_filter:NNNn #1#2#3#4
  {
    \__seq_push_item_def:n { \bool_if:nT {#4} { \__seq_wrap_item:n {##1} } }
    #1 #2 { #3 }
    \__seq_pop_item_def:
  }
\cs_new_protected:Npn \seq_set_regex_extract_once:Nnn #1#2#3
  { \regex_extract_once:nnN {#2} {#3} #1 }
\cs_generate_variant:Nn \seq_set_regex_extract_once:Nnn { c }
\cs_new_protected:Npn \seq_set_regex_extract_once:NNn #1#2#3
  { \regex_extract_once:NnN #2 {#3} #1 }
\cs_generate_variant:Nn \seq_set_regex_extract_once:NNn { c }
\cs_new_protected:Npn \seq_set_regex_extract_all:Nnn #1#2#3
  { \regex_extract_all:nnN {#2} {#3} #1 }
\cs_generate_variant:Nn \seq_set_regex_extract_all:Nnn { c }
\cs_new_protected:Npn \seq_set_regex_extract_all:NNn #1#2#3
  { \regex_extract_all:NnN #2 {#3} #1 }
\cs_generate_variant:Nn \seq_set_regex_extract_all:NNn { c }
\cs_new_protected:Npn \seq_set_regex_split:Nnn #1#2#3
  { \regex_split:nnN {#2} {#3} #1 }
\cs_generate_variant:Nn \seq_set_regex_split:Nnn { c }
\cs_new_protected:Npn \seq_set_regex_split:NNn #1#2#3
  { \regex_split:NnN #2 {#3} #1 }
\cs_generate_variant:Nn \seq_set_regex_split:NNn { c }
\group_begin:
  \cs_set_protected:Npn \__seq_tmp:w #1#2#3
    {
      \cs_new_protected:cpe { seq_gset_regex_ #1 :N #2 n } ##1##2##3
        {
          \group_begin:
            \seq_set_eq:NN \exp_not:N \l__seq_tmp_seq ##1
            \exp_not:c { regex_ #1 :Nn #2 }
              #3 {##2} {##3} \exp_not:N \l__seq_tmp_seq
            \seq_gset_eq:NN ##1 \exp_not:N \l__seq_tmp_seq
          \group_end:
        }
      \cs_generate_variant:cn { seq_gset_regex_ #1 :N #2 n } { c }
    }
  \__seq_tmp:w { extract_once } n { }
  \__seq_tmp:w { extract_once } N \use:n
  \__seq_tmp:w { extract_all } n { }
  \__seq_tmp:w { extract_all } N \use:n
  \__seq_tmp:w { split } n { }
  \__seq_tmp:w { split } N \use:n
\group_end:
\cs_new_protected:Npn \seq_concat:NNN #1#2#3
  { \tl_set:Nf #1 { \exp_after:wN \use_i:nn \exp_after:wN #2 #3 } }
\cs_new_protected:Npn \seq_gconcat:NNN #1#2#3
  { \tl_gset:Nf #1 { \exp_after:wN \use_i:nn \exp_after:wN #2 #3 } }
\cs_generate_variant:Nn \seq_concat:NNN  { ccc }
\cs_generate_variant:Nn \seq_gconcat:NNN { ccc }
\prg_new_eq_conditional:NNn \seq_if_exist:N \cs_if_exist:N
  { TF , T , F , p }
\prg_new_eq_conditional:NNn \seq_if_exist:c \cs_if_exist:c
  { TF , T , F , p }
\cs_new_protected:Npn \seq_put_left:Nn #1#2
  {
    \__kernel_tl_set:Nx #1
      {
        \exp_not:n { \s__seq \__seq_item:n {#2} }
        \exp_not:f { \exp_after:wN \__seq_put_left_aux:w #1 }
      }
  }
\cs_new_protected:Npn \seq_gput_left:Nn #1#2
  {
    \__kernel_tl_gset:Nx #1
      {
        \exp_not:n { \s__seq \__seq_item:n {#2} }
        \exp_not:f { \exp_after:wN \__seq_put_left_aux:w #1 }
      }
  }
\cs_new:Npn \__seq_put_left_aux:w \s__seq { \exp_stop_f: }
\cs_generate_variant:Nn \seq_put_left:Nn  {     NV , Nv , Ne , No , Nx }
\cs_generate_variant:Nn \seq_put_left:Nn  { c , cV , cv , ce , co ,cx }
\cs_generate_variant:Nn \seq_gput_left:Nn {     NV , Nv , Ne , No , Nx }
\cs_generate_variant:Nn \seq_gput_left:Nn { c , cV , cv , ce , co , cx }
\cs_new_protected:Npn \seq_put_right:Nn #1#2
  { \tl_put_right:Nn #1 { \__seq_item:n {#2} } }
\cs_new_protected:Npn \seq_gput_right:Nn #1#2
  { \tl_gput_right:Nn #1 { \__seq_item:n {#2} } }
\cs_generate_variant:Nn \seq_put_right:Nn  {     NV , Nv , Ne , No , Nx }
\cs_generate_variant:Nn \seq_put_right:Nn  { c , cV , cv , ce , co , cx }
\cs_generate_variant:Nn \seq_gput_right:Nn {     NV , Nv , Ne , No , Nx }
```

```
\cs_generate_variant:Nn \seq_gput_right:Nn { c , cV , cv , ce , co , cx }
\cs_new:Npn \__seq_wrap_item:n #1 { \exp_not:n { \__seq_item:n {#1} } }
\seq_new:N \l__seq_tmp_seq
\cs_new_protected:Npn \seq_remove_duplicates:N
  { \__seq_remove_duplicates:NN \seq_set_eq:NN }
\cs_new_protected:Npn \seq_gremove_duplicates:N
  { \__seq_remove_duplicates:NN \seq_gset_eq:NN }
\cs_new_protected:Npn \__seq_remove_duplicates:NN #1#2
  {
    \seq_clear:N \l__seq_tmp_seq
    \seq_map_inline:Nn #2
      {
        \seq_if_in:NnF \l__seq_tmp_seq {##1}
          { \seq_put_right:Nn \l__seq_tmp_seq {##1} }
      }
    #1 #2 \l__seq_tmp_seq
  }
\cs_generate_variant:Nn \seq_remove_duplicates:N  { c }
\cs_generate_variant:Nn \seq_gremove_duplicates:N { c }
\cs_new_protected:Npn \seq_remove_all:Nn
  { \__seq_remove_all_aux:NNn \__kernel_tl_set:Nx }
\cs_new_protected:Npn \seq_gremove_all:Nn
  { \__seq_remove_all_aux:NNn \__kernel_tl_gset:Nx }
\cs_new_protected:Npn \__seq_remove_all_aux:NNn #1#2#3
  {
    \__seq_push_item_def:n
      {
        \str_if_eq:nnT {##1} {#3}
          {
            \if_false: { \fi: }
            \tl_set:Nn \l__seq_internal_b_tl {##1}
            #1 #2
              { \if_false: } \fi:
                \exp_not:o {#2}
                \tl_if_eq:NNT \l__seq_internal_a_tl \l__seq_internal_b_tl
                  { \use_none:nn }
          }
        \__seq_wrap_item:n {##1}
      }
    \tl_set:Nn \l__seq_internal_a_tl {#3}
    #1 #2 {#2}
    \__seq_pop_item_def:
  }
\cs_generate_variant:Nn \seq_remove_all:Nn  { NV , Ne , c , cV , ce }
\cs_generate_variant:Nn \seq_remove_all:Nn  { Nx , cx }
\cs_generate_variant:Nn \seq_gremove_all:Nn { NV , Ne , c , cV , ce }
```

```
\cs_generate_variant:Nn \seq_gremove_all:Nn { Nx , cx }
\cs_new_eq:NN \__seq_int_eval:w \tex_numexpr:D
\cs_new_protected:Npn \seq_set_item:Nnn #1#2#3
  { \__seq_set_item:NnnNN #1 {#2} {#3} \__kernel_tl_set:Nx \use_i:nn }
\cs_new_protected:Npn \seq_gset_item:Nnn #1#2#3
  { \__seq_set_item:NnnNN #1 {#2} {#3} \__kernel_tl_gset:Nx \use_i:nn }
\cs_generate_variant:Nn \seq_set_item:Nnn { c }
\cs_generate_variant:Nn \seq_gset_item:Nnn { c }
\prg_new_protected_conditional:Npnn \seq_set_item:Nnn #1#2#3 { TF , T , F }
  { \__seq_set_item:NnnNN #1 {#2} {#3} \__kernel_tl_set:Nx \use_ii:nn }
\prg_new_protected_conditional:Npnn \seq_gset_item:Nnn #1#2#3 { TF , T , F }
  { \__seq_set_item:NnnNN #1 {#2} {#3} \__kernel_tl_gset:Nx \use_ii:nn }
\prg_generate_conditional_variant:Nnn \seq_set_item:Nnn { c } { TF , T , F }
\prg_generate_conditional_variant:Nnn \seq_gset_item:Nnn { c } { TF , T , F }
\cs_new_protected:Npn \__seq_set_item:NnnNN #1#2#3
  {
    \tl_set:Nn \l__seq_internal_a_tl { \__seq_item:n {#3} }
    \exp_args:Nff \__seq_set_item:nnNNNN
      { \int_eval:n {#2} } { \seq_count:N #1 } #1 \use_none:nn
  }
\cs_new_protected:Npn \__seq_set_item:nnNNNN #1#2
  {
    \int_compare:nNnTF {#1} > 0
      { \int_compare:nNnF {#1} > {#2} { \__seq_set_item:nNnnNNNN { #1 - 1 } } }
      {
        \int_compare:nNnF {#1} < {-#2}
          {
            \int_compare:nNnF {#1} = 0
              { \__seq_set_item:nNnnNNNN { #2 + #1 } }
          }
      }
    \__seq_set_item_false:nnNNNN {#1} {#2}
  }
\cs_new_protected:Npn \__seq_set_item_false:nnNNNN #1#2#3#4#5#6
  {
    #6
      {
        \msg_error:nneee { seq } { item-too-large }
          { \token_to_str:N #3 } {#2} {#1}
      }
      { \prg_return_false: }
  }
\cs_new_protected:Npn \__seq_set_item:nNnnNNNN #1#2#3#4#5#6#7#8
  {
    #7 #5
      {
        \s__seq
        \exp_after:wN \__seq_set_item:wn
        \int_value:w \__seq_int_eval:w #1
        #5 \s__seq_stop #6
      }
    #8 { } { \prg_return_true: }
  }
\cs_new:Npn \__seq_set_item:wn #1 \__seq_item:n #2
  {
    \if_meaning:w 0 #1 \__seq_set_item_end:w \fi:
    \exp_not:n { \__seq_item:n {#2} }
    \exp_after:wN \__seq_set_item:wn
    \int_value:w \__seq_int_eval:w #1 - 1 \s__seq
  }
\cs_new:Npn \__seq_set_item_end:w #1 \exp_not:n #2 #3 \s__seq #4 \s__seq_stop #5
  {
    #1
    \exp_not:o \l__seq_internal_a_tl
    \exp_not:n {#4}
    #5 #2
  }
\cs_new_protected:Npn \seq_reverse:N
  { \__seq_reverse:NN \__kernel_tl_set:Nx }
\cs_new_protected:Npn \seq_greverse:N
  { \__seq_reverse:NN \__kernel_tl_gset:Nx }
\cs_new_protected:Npn \__seq_reverse:NN #1 #2
  {
    \cs_set_eq:NN \__seq_tmp:w \__seq_item:n
    \cs_set_eq:NN \__seq_item:n \__seq_reverse_item:nwn
    #1 #2 { #2 \exp_not:n { } }
    \cs_set_eq:NN \__seq_item:n \__seq_tmp:w
  }
\cs_new:Npn \__seq_reverse_item:nwn #1 #2 \exp_not:n #3
  {
    #2
    \exp_not:n { \__seq_item:n {#1} #3 }
  }
\cs_generate_variant:Nn \seq_reverse:N  { c }
\cs_generate_variant:Nn \seq_greverse:N { c }
\prg_new_conditional:Npnn \seq_if_empty:N #1 { p , T , F , TF }
  {
    \if_meaning:w #1 \c_empty_seq
      \prg_return_true:
    \else:
      \prg_return_false:
    \fi:
  }
\prg_generate_conditional_variant:Nnn \seq_if_empty:N
  { c } { p , T , F , TF }
\seq_new:N \g__seq_internal_seq
\cs_new_protected:Npn \seq_shuffle:N { \__seq_shuffle:NN \seq_set_eq:NN }
\cs_new_protected:Npn \seq_gshuffle:N { \__seq_shuffle:NN \seq_gset_eq:NN }
\cs_new_protected:Npn \__seq_shuffle:NN #1#2
  {
    \int_compare:nNnTF { \seq_count:N #2 } > \c_max_register_int
      {
        \msg_error:nne { seq } { shuffle-too-large }
          { \token_to_str:N #2 }
      }
      {
        \group_begin:
          \int_zero:N \l__seq_internal_a_int
          \__seq_push_item_def:
          \cs_gset_eq:NN \__seq_item:n \__seq_shuffle_item:n
          #2
          \__seq_pop_item_def:
          \seq_gclear:N \g__seq_internal_seq
          \int_step_inline:nn \l__seq_internal_a_int
            {
              \seq_gput_right:Ne \g__seq_internal_seq
                { \tex_the:D \tex_toks:D ##1 }
            }
        \group_end:
        #1 #2 \g__seq_internal_seq
        \seq_gclear:N \g__seq_internal_seq
      }
  }
\cs_new_protected:Npn \__seq_shuffle_item:n
  {
    \int_incr:N \l__seq_internal_a_int
    \int_set:Nn \l__seq_internal_b_int
      { 1 + \tex_uniformdeviate:D \l__seq_internal_a_int }
    \tex_toks:D \l__seq_internal_a_int
      = \tex_toks:D \l__seq_internal_b_int
    \tex_toks:D \l__seq_internal_b_int
  }
\cs_generate_variant:Nn \seq_shuffle:N { c }
\cs_generate_variant:Nn \seq_gshuffle:N { c }
\prg_new_protected_conditional:Npnn \seq_if_in:Nn #1#2
  { T , F , TF }
  {
    \group_begin:
      \tl_set:Nn \l__seq_internal_a_tl {#2}
      \cs_set_protected:Npn \__seq_item:n ##1
        {
          \tl_set:Nn \l__seq_internal_b_tl {##1}
          \if_meaning:w \l__seq_internal_a_tl \l__seq_internal_b_tl
            \exp_after:wN \__seq_if_in:
          \fi:
        }
      #1
    \group_end:
    \prg_return_false:
    \prg_break_point:
  }
\cs_new:Npn \__seq_if_in:
  { \prg_break:n { \group_end: \prg_return_true: } }
\prg_generate_conditional_variant:Nnn \seq_if_in:Nn
  { NV , Nv , Ne , No , Nx , c , cV , cv , ce , co , cx } { T , F , TF }
\cs_new_protected:Npn \__seq_pop:NNNN #1#2#3#4
  {
    \if_meaning:w #3 \c_empty_seq
      \tl_set:Nn #4 { \q_no_value }
    \else:
      #1#2#3#4
    \fi:
  }
\cs_new_protected:Npn \__seq_pop_TF:NNNN #1#2#3#4
  {
    \if_meaning:w #3 \c_empty_seq
      % \tl_set:Nn #4 { \q_no_value }
      \prg_return_false:
    \else:
      #1#2#3#4
      \prg_return_true:
    \fi:
  }
\cs_new_protected:Npn \seq_get_left:NN #1#2
  {
    \__kernel_tl_set:Nx #2
      {
        \exp_after:wN \__seq_get_left:wnw
        #1 \__seq_item:n { \q_no_value } \s__seq_stop
      }
  }
\cs_new:Npn \__seq_get_left:wnw #1 \__seq_item:n #2#3 \s__seq_stop
  { \exp_not:n {#2} }
\cs_generate_variant:Nn \seq_get_left:NN { c }
```

```
\cs_new_protected:Npn \seq_pop_left:NN
  { \__seq_pop:NNNN \__seq_pop_left:NNN \tl_set:Nn }
\cs_new_protected:Npn \seq_gpop_left:NN
  { \__seq_pop:NNNN \__seq_pop_left:NNN \tl_gset:Nn }
\cs_new_protected:Npn \__seq_pop_left:NNN #1#2#3
  { \exp_after:wN \__seq_pop_left:wnwNNN #2 \s__seq_stop #1#2#3 }
\cs_new_protected:Npn \__seq_pop_left:wnwNNN
    #1 \__seq_item:n #2#3 \s__seq_stop #4#5#6
  {
    #4 #5 { #1 #3 }
    \tl_set:Nn #6 {#2}
  }
\cs_generate_variant:Nn \seq_pop_left:NN  { c }
\cs_generate_variant:Nn \seq_gpop_left:NN { c }
\cs_new_protected:Npn \seq_get_right:NN #1#2
  {
    \__kernel_tl_set:Nx #2
      {
        \exp_after:wN \use_i_ii:nnn
        \exp_after:wN \__seq_get_right_loop:nw
        \exp_after:wN \q_no_value
        #1
        \__seq_get_right_end:NnN \__seq_item:n
      }
  }
\cs_new:Npn \__seq_get_right_loop:nw #1#2 \__seq_item:n
  {
    #2 \use_none:n {#1}
    \__seq_get_right_loop:nw
  }
\cs_new:Npn \__seq_get_right_end:NnN #1#2#3 { \exp_not:n {#2} }
\cs_generate_variant:Nn \seq_get_right:NN { c }
\cs_new_protected:Npn \seq_pop_right:NN
  { \__seq_pop:NNNN \__seq_pop_right:NNN \__kernel_tl_set:Nx }
\cs_new_protected:Npn \seq_gpop_right:NN
  { \__seq_pop:NNNN \__seq_pop_right:NNN \__kernel_tl_gset:Nx }
\cs_new_protected:Npn \__seq_pop_right:NNN #1#2#3
  {
    \cs_set_eq:NN \__seq_tmp:w \__seq_item:n
    \cs_set_eq:NN \__seq_item:n \scan_stop:
    #1 #2
      { \if_false: } \fi: \s__seq
        \exp_after:wN \use_i:nnn
        \exp_after:wN \__seq_pop_right_loop:nn
        #2
        {
          \if_false: { \fi: }
          \__kernel_tl_set:Nx #3
        }
        { } \use_none:nn
    \cs_set_eq:NN \__seq_item:n \__seq_tmp:w
  }
\cs_new:Npn \__seq_pop_right_loop:nn #1#2
  {
    #2 { \exp_not:n {#1} }
    \__seq_pop_right_loop:nn
  }
\cs_generate_variant:Nn \seq_pop_right:NN  { c }
\cs_generate_variant:Nn \seq_gpop_right:NN { c }
\prg_new_protected_conditional:Npnn \seq_get_left:NN #1#2 { T , F , TF }
  { \__seq_pop_TF:NNNN \prg_do_nothing: \seq_get_left:NN #1#2 }
\prg_new_protected_conditional:Npnn \seq_get_right:NN #1#2 { T , F , TF }
  { \__seq_pop_TF:NNNN \prg_do_nothing: \seq_get_right:NN #1#2 }
\prg_generate_conditional_variant:Nnn \seq_get_left:NN
  { c } { T , F , TF }
\prg_generate_conditional_variant:Nnn \seq_get_right:NN
  { c } { T , F , TF }
\prg_new_protected_conditional:Npnn \seq_pop_left:NN #1#2
  { T , F , TF }
  { \__seq_pop_TF:NNNN \__seq_pop_left:NNN \tl_set:Nn #1 #2 }
\prg_new_protected_conditional:Npnn \seq_gpop_left:NN #1#2
  { T , F , TF }
  { \__seq_pop_TF:NNNN \__seq_pop_left:NNN \tl_gset:Nn #1 #2 }
\prg_new_protected_conditional:Npnn \seq_pop_right:NN #1#2
  { T , F , TF }
  { \__seq_pop_TF:NNNN \__seq_pop_right:NNN \__kernel_tl_set:Nx #1 #2 }
\prg_new_protected_conditional:Npnn \seq_gpop_right:NN #1#2
  { T , F , TF }
  { \__seq_pop_TF:NNNN \__seq_pop_right:NNN \__kernel_tl_gset:Nx #1 #2 }
\prg_generate_conditional_variant:Nnn \seq_pop_left:NN { c }
  { T , F , TF }
\prg_generate_conditional_variant:Nnn \seq_gpop_left:NN { c }
  { T , F , TF }
\prg_generate_conditional_variant:Nnn \seq_pop_right:NN { c }
  { T , F , TF }
\prg_generate_conditional_variant:Nnn \seq_gpop_right:NN { c }
  { T , F , TF }
\cs_new:Npn \seq_item:Nn #1
  { \exp_after:wN \__seq_item:wNn #1 \s__seq_stop #1 }
\cs_new:Npn \__seq_item:wNn \s__seq #1 \s__seq_stop #2#3
  {
    \exp_args:Nf \__seq_item:nwn
```

```
      { \exp_args:Nf \__seq_item:nN { \int_eval:n {#3} } #2 }
    #1
    \prg_break: \__seq_item:n { }
    \prg_break_point:
  }
\cs_new:Npn \__seq_item:nN #1#2
  {
    \int_compare:nNnTF {#1} < 0
      { \int_eval:n { \seq_count:N #2 + 1 + #1 } }
      {#1}
  }
\cs_new:Npn \__seq_item:nwn #1#2 \__seq_item:n #3
  {
    #2
    \int_compare:nNnTF {#1} = 1
      { \prg_break:n { \exp_not:n {#3} } }
      { \exp_args:Nf \__seq_item:nwn { \int_eval:n { #1 - 1 } } }
  }
\cs_generate_variant:Nn \seq_item:Nn { NV , Ne , c , cV , ce }
\cs_new:Npn \seq_rand_item:N #1
  {
    \seq_if_empty:NF #1
      { \seq_item:Nn #1 { \int_rand:nn { 1 } { \seq_count:N #1 } } }
  }
\cs_generate_variant:Nn \seq_rand_item:N { c }
\cs_new:Npn \seq_map_break:
  { \prg_map_break:Nn \seq_map_break: { } }
\cs_new:Npn \seq_map_break:n
  { \prg_map_break:Nn \seq_map_break: }
\cs_new:Npn \seq_map_function:NN #1#2
  {
    \exp_after:wN \use_i_ii:nnn
    \exp_after:wN \__seq_map_function:Nw
    \exp_after:wN #2
    #1
    \prg_break:
    \__seq_item:n { } \__seq_item:n { } \__seq_item:n { } \__seq_item:n { }
    \prg_break_point:
    \prg_break_point:Nn \seq_map_break: { }
  }
\cs_new:Npn \__seq_map_function:Nw #1
    #2 \__seq_item:n #3
    #4 \__seq_item:n #5
    #6 \__seq_item:n #7
    #8 \__seq_item:n #9
  {
    #2 #1 {#3}
    #4 #1 {#5}
    #6 #1 {#7}
    #8 #1 {#9}
    \__seq_map_function:Nw #1
  }
\cs_generate_variant:Nn \seq_map_function:NN { c }
\cs_new_protected:Npn \__seq_push_item_def:n
  {
    \__seq_push_item_def:
    \cs_gset:Npn \__seq_item:n ##1
  }
\cs_new_protected:Npn \__seq_push_item_def:e
  {
    \__seq_push_item_def:
    \cs_gset:Npe \__seq_item:n ##1
  }
\cs_new_protected:Npn \__seq_push_item_def:
  {
    \int_gincr:N \g__kernel_prg_map_int
    \cs_gset_eq:cN { __seq_map_ \int_use:N \g__kernel_prg_map_int :w }
      \__seq_item:n
  }
\cs_new_protected:Npn \__seq_pop_item_def:
  {
    \cs_gset_eq:Nc \__seq_item:n
      { __seq_map_ \int_use:N \g__kernel_prg_map_int :w }
    \int_gdecr:N \g__kernel_prg_map_int
  }
\cs_new_protected:Npn \seq_map_inline:Nn #1#2
  {
    \__seq_push_item_def:n {#2}
    #1
    \prg_break_point:Nn \seq_map_break: { \__seq_pop_item_def: }
  }
\cs_generate_variant:Nn \seq_map_inline:Nn { c }
\cs_new:Npn \seq_map_tokens:Nn #1#2
  {
    \exp_last_unbraced:Nno
      \use_i:nn { \__seq_map_tokens:nw {#2} } #1
    \prg_break:
    \__seq_item:n { } \__seq_item:n { } \__seq_item:n { } \__seq_item:n { }
    \prg_break_point:
    \prg_break_point:Nn \seq_map_break: { }
  }
\cs_generate_variant:Nn \seq_map_tokens:Nn { c }
```

```
\cs_new:Npn \__seq_map_tokens:nw #1
    #2 \__seq_item:n #3
    #4 \__seq_item:n #5
    #6 \__seq_item:n #7
    #8 \__seq_item:n #9
  {
    #2 \use:n {#1} {#3}
    #4 \use:n {#1} {#5}
    #6 \use:n {#1} {#7}
    #8 \use:n {#1} {#9}
    \__seq_map_tokens:nw {#1}
  }
\cs_new_protected:Npn \seq_map_variable:NNn #1#2#3
  {
    \__seq_push_item_def:e
      {
        \tl_set:Nn \exp_not:N #2 {##1}
        \exp_not:n {#3}
      }
    #1
    \prg_break_point:Nn \seq_map_break: { \__seq_pop_item_def: }
  }
\cs_generate_variant:Nn \seq_map_variable:NNn {     Nc }
\cs_generate_variant:Nn \seq_map_variable:NNn { c , cc }
\cs_new_eq:NN \__seq_sep: \__kernel_int_sep:
\cs_new:Npn \seq_map_indexed_function:NN #1#2
  {
    \__seq_map_indexed:NN #1#2
    \prg_break_point:Nn \seq_map_break: { }
  }
\cs_new_protected:Npn \seq_map_indexed_inline:Nn #1#2
  {
    \int_gincr:N \g__kernel_prg_map_int
    \cs_gset_protected:cpn
      { __seq_map_ \int_use:N \g__kernel_prg_map_int :w } ##1##2 {#2}
    \exp_args:NNc \__seq_map_indexed:NN #1
      { __seq_map_ \int_use:N \g__kernel_prg_map_int :w }
    \prg_break_point:Nn \seq_map_break:
      { \int_gdecr:N \g__kernel_prg_map_int }
  }
\cs_new:Npn \__seq_map_indexed:NN #1#2
  {
    \exp_after:wN \__seq_map_indexed:Nw
    \exp_after:wN #2
    \int_value:w 1
    \exp_after:wN \use_i:nn
    \exp_after:wN \__seq_sep:
    #1
    \prg_break: \__seq_item:n { } \prg_break_point:
  }
\cs_new:Npn \__seq_map_indexed:Nw #1#2 \__seq_sep: #3 \__seq_item:n #4
  {
    #3
    #1 {#2} {#4}
    \exp_after:wN \__seq_map_indexed:Nw
    \exp_after:wN #1
    \int_value:w \int_eval:w 1 + #2 \__seq_sep:
  }
\cs_new:Npn \seq_map_pairwise_function:NNN #1#2#3
  { \exp_after:wN \__seq_map_pairwise_function:wNN #2 \s__seq_stop #1 #3 }
\cs_new:Npn \__seq_map_pairwise_function:wNN \s__seq #1 \s__seq_stop #2#3
  {
    \exp_after:wN \__seq_map_pairwise_function:wNw #2 \s__seq_stop #3
      #1 { ? \prg_break: } { }
    \prg_break_point:
    \prg_break_point:Nn \seq_map_break: { }
  }
\cs_new:Npn \__seq_map_pairwise_function:wNw \s__seq #1 \s__seq_stop #2
  {
    \__seq_map_pairwise_function:Nnnwnn #2
      #1 { ? \prg_break: } { }
    \s__seq_stop
  }
\cs_new:Npn \__seq_map_pairwise_function:Nnnwnn #1#2#3#4 \s__seq_stop #5#6
  {
    \use_none:n #2
    \use_none:n #5
    #1 {#3} {#6}
    \__seq_map_pairwise_function:Nnnwnn #1 #4 \s__seq_stop
  }
\cs_generate_variant:Nn \seq_map_pairwise_function:NNN { Nc , c , cc }
\cs_new_protected:Npn \seq_set_map_e:NNn
  { \__seq_set_map_e:NNNn \__kernel_tl_set:Nx }
\cs_new_protected:Npn \seq_gset_map_e:NNn
  { \__seq_set_map_e:NNNn \__kernel_tl_gset:Nx }
\cs_new_protected:Npn \__seq_set_map_e:NNNn #1#2#3#4
  {
    \__seq_push_item_def:n { \exp_not:N \__seq_item:n {#4} }
    #1 #2 { #3 }
    \__seq_pop_item_def:
  }
\cs_new_protected:Npn \seq_set_map:NNn
```

```
  { \__seq_set_map:NNNn \__kernel_tl_set:Nx }
\cs_new_protected:Npn \seq_gset_map:NNn
  { \__seq_set_map:NNNn \__kernel_tl_gset:Nx }
\cs_new_protected:Npn \__seq_set_map:NNNn #1#2#3#4
  {
    \__seq_push_item_def:n { \exp_not:n { \__seq_item:n {#4} } }
    #1 #2 { #3 }
    \__seq_pop_item_def:
  }
\cs_new:Npn \seq_count:N #1
  {
    \int_eval:n
      {
        \exp_after:wN \use_i:nn
        \exp_after:wN \__seq_count:w
        #1
        \__seq_count_end:w \__seq_item:n 7
        \__seq_count_end:w \__seq_item:n 6
        \__seq_count_end:w \__seq_item:n 5
        \__seq_count_end:w \__seq_item:n 4
        \__seq_count_end:w \__seq_item:n 3
        \__seq_count_end:w \__seq_item:n 2
        \__seq_count_end:w \__seq_item:n 1
        \__seq_count_end:w \__seq_item:n 0
        \prg_break_point:
      }
  }
\cs_new:Npn \__seq_count:w
    #1 \__seq_item:n #2 \__seq_item:n #3 \__seq_item:n #4 \__seq_item:n
    #5 \__seq_item:n #6 \__seq_item:n #7 \__seq_item:n #8 #9 \__seq_item:n
  { #9 8 + \__seq_count:w }
\cs_new:Npn \__seq_count_end:w 8 + \__seq_count:w #1#2 \prg_break_point: {#1}
\cs_generate_variant:Nn \seq_count:N { c }
\cs_new:Npn \seq_use:Nnnn #1#2#3#4
  {
    \seq_if_exist:NTF #1
      {
        \int_case:nnF { \seq_count:N #1 }
          {
            { 0 } { }
            { 1 } { \exp_after:wN \__seq_use:NNnNnn #1 ? { } { } }
            { 2 } { \exp_after:wN \__seq_use:NNnNnn #1 {#2} }
          }
          {
            \exp_after:wN \__seq_use_setup:w #1 \__seq_item:n
            \s__seq_mark { \__seq_use:nwwwwnwn {#3} }
            \s__seq_mark { \__seq_use:nwwn {#4} }
            \s__seq_stop { }
          }
      }
      {
        \msg_expandable_error:nnn
          { kernel } { bad-variable } {#1}
      }
  }
\cs_generate_variant:Nn \seq_use:Nnnn { c }
\cs_new:Npn \__seq_use:NNnNnn #1#2#3#4#5#6 { \exp_not:n { #3 #6 #5 } }
\cs_new:Npn \__seq_use_setup:w \s__seq { \__seq_use:nwwwwnwn { } }
\cs_new:Npn \__seq_use:nwwwwnwn
    #1 \__seq_item:n #2 \__seq_item:n #3 \__seq_item:n #4#5
    \s__seq_mark #6#7 \s__seq_stop #8
  {
    #6 \__seq_item:n {#3} \__seq_item:n {#4} #5
    \s__seq_mark {#6} #7 \s__seq_stop { #8 #1 #2 }
  }
\cs_new:Npn \__seq_use:nwwn #1 \__seq_item:n #2 #3 \s__seq_stop #4
  { \exp_not:n { #4 #1 #2 } }
\cs_new:Npn \seq_use:Nn #1#2
  { \seq_use:Nnnn #1 {#2} {#2} {#2} }
\cs_generate_variant:Nn \seq_use:Nn { c }
\cs_new_eq:NN \seq_push:Nn \seq_put_left:Nn
\cs_generate_variant:Nn \seq_push:Nn { NV , Nv , Ne , c , cV , cv , ce }
\cs_generate_variant:Nn \seq_push:Nn { No , Nx , co , cx }
\cs_new_eq:NN \seq_gpush:Nn \seq_gput_left:Nn
\cs_generate_variant:Nn \seq_gpush:Nn { NV , Nv , Ne , c , cV , cv , ce }
\cs_generate_variant:Nn \seq_gpush:Nn { No , Nx , co , cx }
\cs_new_eq:NN \seq_get:NN \seq_get_left:NN
\cs_new_eq:NN \seq_get:cN \seq_get_left:cN
\cs_new_eq:NN \seq_pop:NN \seq_pop_left:NN
\cs_new_eq:NN \seq_pop:cN \seq_pop_left:cN
\cs_new_eq:NN \seq_gpop:NN \seq_gpop_left:NN
\cs_new_eq:NN \seq_gpop:cN \seq_gpop_left:cN
\prg_new_eq_conditional:NNn \seq_get:NN  \seq_get_left:NN  { T , F , TF }
\prg_new_eq_conditional:NNn \seq_get:cN  \seq_get_left:cN  { T , F , TF }
\prg_new_eq_conditional:NNn \seq_pop:NN  \seq_pop_left:NN  { T , F , TF }
\prg_new_eq_conditional:NNn \seq_pop:cN  \seq_pop_left:cN  { T , F , TF }
\prg_new_eq_conditional:NNn \seq_gpop:NN \seq_gpop_left:NN { T , F , TF }
\prg_new_eq_conditional:NNn \seq_gpop:cN \seq_gpop_left:cN { T , F , TF }
\cs_new_protected:Npn \seq_show:N { \__seq_show:NN \msg_show:nneeee }
\cs_generate_variant:Nn \seq_show:N { c }
\cs_new_protected:Npn \seq_log:N { \__seq_show:NN \msg_log:nneeee }
\cs_generate_variant:Nn \seq_log:N { c }
```

```
\cs_new_protected:Npn \__seq_show:NN #1#2
  {
    \__kernel_chk_tl_type:NnnT #2 { seq }
      {
        \s__seq
        \exp_after:wN \use_i:nn \exp_after:wN \__seq_show_validate:nn #2
        \q_recursion_tail \q_recursion_tail \q_recursion_stop
      }
      {
        #1 { seq } { show }
          { \token_to_str:N #2 }
          { \seq_map_function:NN #2 \msg_show_item:n }
          { } { }
      }
  }
\cs_new:Npn \__seq_show_validate:nn #1#2
  {
    \quark_if_recursion_tail_stop:n {#2}
    \__seq_wrap_item:n {#2}
    \__seq_show_validate:nn
  }
\seq_new:N \l_tmpa_seq
\seq_new:N \l_tmpb_seq
\seq_new:N \g_tmpa_seq
\seq_new:N \g_tmpb_seq
%% File: l3int.dtx
\cs_new_eq:NN \int_value:w      \tex_number:D
\cs_new_eq:NN \__int_eval:w       \tex_numexpr:D
\cs_new_eq:NN \__int_eval_end:    \tex_relax:D
\cs_new_eq:NN \if_int_odd:w     \tex_ifodd:D
\cs_new_eq:NN \if_case:w        \tex_ifcase:D
\scan_new:N \s__int_mark
\scan_new:N \s__int_stop
\cs_new:Npn \__int_use_none_delimit_by_s_stop:w #1 \s__int_stop { }
\quark_new:N \q__int_recursion_tail
\quark_new:N \q__int_recursion_stop
\__kernel_quark_new_test:N \__int_if_recursion_tail_stop_do:Nn
\__kernel_quark_new_test:N \__int_if_recursion_tail_stop:N
\cs_new:Npn \int_eval:n #1
  { \tex_the:D \__int_eval:w #1 \__int_eval_end: }
\cs_new:Npn \int_eval:w { \tex_the:D \__int_eval:w }
\cs_new_eq:NN \__int_sep: \__kernel_int_sep:
\cs_new:Npn \int_sign:n #1
  {
    \int_value:w \exp_after:wN \__int_sign:Nw
      \int_value:w \__int_eval:w #1 \__int_eval_end: \__int_sep:
```

```
    \exp_stop_f:
  }
\cs_new:Npn \__int_sign:Nw #1#2 \__int_sep:
  {
    \if_meaning:w 0 #1
      0
    \else:
      \if_meaning:w - #1 - \fi: 1
    \fi:
  }
\cs_new:Npn \int_abs:n #1
  {
    \int_value:w \exp_after:wN \__int_abs:N
      \int_value:w \__int_eval:w #1 \__int_eval_end:
    \exp_stop_f:
  }
\cs_new:Npn \__int_abs:N #1
  { \if_meaning:w - #1 \else: \exp_after:wN #1 \fi: }
\cs_new:Npn \int_max:nn #1#2
  {
    \int_value:w \exp_after:wN \__int_maxmin:wwN
      \int_value:w \__int_eval:w #1 \exp_after:wN \__int_sep:
      \int_value:w \__int_eval:w #2 \__int_sep:
      >
    \exp_stop_f:
  }
\cs_new:Npn \int_min:nn #1#2
  {
    \int_value:w \exp_after:wN \__int_maxmin:wwN
      \int_value:w \__int_eval:w #1 \exp_after:wN \__int_sep:
      \int_value:w \__int_eval:w #2 \__int_sep:
      <
    \exp_stop_f:
  }
\cs_new:Npn \__int_maxmin:wwN #1 \__int_sep: #2 \__int_sep: #3
  {
    \if_int_compare:w #1 #3 #2 ~
      #1
    \else:
      #2
    \fi:
  }
\cs_new:Npn \int_div_truncate:nn #1#2
  {
    \int_value:w \__int_eval:w
      \exp_after:wN \__int_div_truncate:NwNw
      \int_value:w \__int_eval:w #1 \exp_after:wN \__int_sep:
      \int_value:w \__int_eval:w #2 \__int_sep:
    \__int_eval_end:
  }
\cs_new:Npn \__int_div_truncate:NwNw #1#2 \__int_sep: #3#4 \__int_sep:
  {
    \if_meaning:w 0 #1
      0
    \else:
      (
        #1#2
        \if_meaning:w - #1 + \else: - \fi:
        ( \if_meaning:w - #3 - \fi: #3#4 - 1 ) / 2
      )
    \fi:
    / #3#4
  }
\cs_new:Npn \int_div_round:nn #1#2
  { \int_value:w \__int_eval:w ( #1 ) / ( #2 ) \__int_eval_end: }
\cs_new:Npn \int_mod:nn #1#2
  {
    \int_value:w \__int_eval:w \exp_after:wN \__int_mod:ww
      \int_value:w \__int_eval:w #1 \exp_after:wN \__int_sep:
      \int_value:w \__int_eval:w #2 \__int_sep:
    \__int_eval_end:
  }
\cs_new:Npn \__int_mod:ww #1 \__int_sep: #2 \__int_sep:
  { #1 - ( \__int_div_truncate:NwNw #1 \__int_sep: #2 \__int_sep: ) * #2 }
\cs_new:Npn \__kernel_int_add:nnn #1#2#3
  {
    \int_value:w \__int_eval:w #1
      \if_int_compare:w #2 < \c_zero_int \exp_after:wN \reverse_if:N \fi:
      \if_int_compare:w #1 < \c_zero_int + #2 + #3 \else: + #3 + #2 \fi:
    \__int_eval_end:
  }
\cs_new_protected:Npn \int_new:N #1
  {
    \__kernel_chk_if_free_cs:N #1
    \cs:w newcount \cs_end: #1
  }
\cs_generate_variant:Nn \int_new:N { c }
\cs_new_protected:Npn \int_const:Nn #1#2
  { \__int_const:eN { \int_eval:n {#2} } #1 }
\cs_generate_variant:Nn \int_const:Nn { c }
\cs_new_protected:Npn \__int_const:nN #1#2
  {
    \int_compare:nNnTF {#1} < \c_zero_int
      {
        \int_new:N #2
        \tex_global:D
      }
      {
        \int_compare:nNnTF {#1} > \c__int_max_constdef_int
          {
            \int_new:N #2
            \tex_global:D
          }
          {
            \__kernel_chk_if_free_cs:N #2
            \tex_global:D \__int_constdef:Nw
          }
      }
    #2 = \__int_eval:w #1 \__int_eval_end:
  }
\cs_generate_variant:Nn \__int_const:nN { e }
\if_int_odd:w 0
  \cs_if_exist:NT \tex_luatexversion:D { 1 }
  \cs_if_exist:NT \tex_omathchardef:D  { 1 }
  \cs_if_exist:NT \tex_XeTeXversion:D  { 1 } ~
    \cs_if_exist:NTF \tex_omathchardef:D
      { \cs_new_eq:NN \__int_constdef:Nw \tex_omathchardef:D }
      { \cs_new_eq:NN \__int_constdef:Nw \tex_chardef:D }
    \tex_global:D \__int_constdef:Nw \c__int_max_constdef_int 1114111 ~
\else:
  \cs_new_eq:NN \__int_constdef:Nw \tex_mathchardef:D
  \tex_global:D \__int_constdef:Nw \c__int_max_constdef_int 32767 ~
\fi:
\cs_new_protected:Npn \int_zero:N  #1 { #1 = \c_zero_int }
\cs_new_protected:Npn \int_gzero:N #1 { \tex_global:D #1 = \c_zero_int }
\cs_generate_variant:Nn \int_zero:N  { c }
\cs_generate_variant:Nn \int_gzero:N { c }
\cs_new_protected:Npn \int_zero_new:N  #1
  { \int_if_exist:NTF #1 { \int_zero:N #1 } { \int_new:N #1 } }
\cs_new_protected:Npn \int_gzero_new:N #1
  { \int_if_exist:NTF #1 { \int_gzero:N #1 } { \int_new:N #1 } }
\cs_generate_variant:Nn \int_zero_new:N  { c }
\cs_generate_variant:Nn \int_gzero_new:N { c }
\cs_new_protected:Npn \int_set_eq:NN #1#2 { #1 = #2 }
\cs_generate_variant:Nn \int_set_eq:NN { c , Nc , cc }
\cs_new_protected:Npn \int_gset_eq:NN #1#2 { \tex_global:D #1 = #2 }
\cs_generate_variant:Nn \int_gset_eq:NN { c , Nc , cc }
\prg_new_eq_conditional:NNn \int_if_exist:N \cs_if_exist:N
```

```
  { TF , T , F , p }
\prg_new_eq_conditional:NNn \int_if_exist:c \cs_if_exist:c
  { TF , T , F , p }
\cs_new_protected:Npn \int_add:Nn #1#2
  { \tex_advance:D #1 \__int_eval:w #2 \__int_eval_end: }
\cs_new_protected:Npn \int_sub:Nn #1#2
  { \tex_advance:D #1 - \__int_eval:w #2 \__int_eval_end: }
\cs_new_protected:Npn \int_gadd:Nn #1#2
  { \tex_global:D \tex_advance:D #1 \__int_eval:w #2 \__int_eval_end: }
\cs_new_protected:Npn \int_gsub:Nn #1#2
  { \tex_global:D \tex_advance:D #1 - \__int_eval:w #2 \__int_eval_end: }
\cs_generate_variant:Nn \int_add:Nn  { c }
\cs_generate_variant:Nn \int_gadd:Nn { c }
\cs_generate_variant:Nn \int_sub:Nn  { c }
\cs_generate_variant:Nn \int_gsub:Nn { c }
\cs_new_protected:Npn \int_incr:N #1
  { \tex_advance:D #1 \c_one_int }
\cs_new_protected:Npn \int_decr:N #1
  { \tex_advance:D #1 - \c_one_int }
\cs_new_protected:Npn \int_gincr:N #1
  { \tex_global:D \tex_advance:D #1 \c_one_int }
\cs_new_protected:Npn \int_gdecr:N #1
  { \tex_global:D \tex_advance:D #1 - \c_one_int }
\cs_generate_variant:Nn \int_incr:N  { c }
\cs_generate_variant:Nn \int_decr:N  { c }
\cs_generate_variant:Nn \int_gincr:N { c }
\cs_generate_variant:Nn \int_gdecr:N { c }
\cs_new_protected:Npn \int_set:Nn #1#2
  { #1 = \__int_eval:w #2 \__int_eval_end: }
\cs_new_protected:Npn \int_gset:Nn #1#2
  { \tex_global:D #1 = \__int_eval:w #2 \__int_eval_end: }
\cs_generate_variant:Nn \int_set:Nn  { NV , c , cV }
\cs_generate_variant:Nn \int_gset:Nn { NV , c , cV }
\cs_new_protected:Npn \int_set_regex_count:Nnn #1#2#3
  { \regex_count:nnN {#2} {#3} #1 }
\cs_generate_variant:Nn \int_set_regex_count:Nnn { c }
\cs_new_protected:Npn \int_gset_regex_count:Nnn #1#2#3
  {
    \group_begin:
      \int_set_eq:NN \l__int_internal_a_int #1
      \regex_count:nnN {#2} {#3} \l__int_internal_a_int
      \int_gset_eq:NN #1 \l__int_internal_a_int
    \group_end:
  }
\cs_generate_variant:Nn \int_gset_regex_count:Nnn { c }
\cs_new_protected:Npn \int_set_regex_count:NNn #1#2#3
```

```
  { \regex_count:NnN #2 {#3} #1 }
\cs_generate_variant:Nn \int_set_regex_count:Nnn { c }
\cs_new_protected:Npn \int_gset_regex_count:NNn #1#2#3
  {
    \group_begin:
      \int_set_eq:NN \l__int_internal_a_int #1
      \regex_count:NnN #2 {#3} \l__int_internal_a_int
      \int_gset_eq:NN #1 \l__int_internal_a_int
    \group_end:
  }
\cs_generate_variant:Nn \int_gset_regex_count:NNn { c }
\cs_new_eq:NN \int_use:N \tex_the:D
\cs_new:Npn \int_use:c #1 { \tex_the:D \cs:w #1 \cs_end: }
\cs_new_protected:Npn \__int_compare_error:
  {
    \if_int_compare:w \c_zero_int \c_zero_int \fi:
    =
    \__int_compare_error:
  }
\cs_new:Npn \__int_compare_error:Nw
    #1#2 \s__int_stop
  {
    { }
    \c_zero_int \fi:
    \msg_expandable_error:nnn
      { kernel } { unknown-comparison } {#1}
    \prg_return_false:
  }
\prg_new_conditional:Npnn \int_compare:n #1 { p , T , F , TF }
  {
    \exp_after:wN \__int_compare:w
    \int_value:w \__int_eval:w #1 \__int_compare_error:
  }
\cs_new:Npn \__int_compare:w #1 \__int_compare_error:
  {
    \exp_after:wN \if_false: \int_value:w
      \__int_compare:Nw #1 e { = nd_ } \s__int_stop
  }
\cs_new:Npn \__int_compare:Nw #1#2 \s__int_stop
  {
    \exp_after:wN \__int_compare:NNw
      \__int_to_roman:w - 0 #2 \s__int_mark
    #1#2 \s__int_stop
  }
\cs_new:Npn \__int_compare:NNw #1#2#3 \s__int_mark
  {
    \__kernel_exp_not:w
    \use:c
      {
        __int_compare_ \token_to_str:N #1
        \if_meaning:w = #2 =  \fi:
        :NNw
      }
      \__int_compare_error:Nw #1
  }
\cs_new:cpn { __int_compare_end_=:NNw } #1#2#3 e #4 \s__int_stop
  {
    {#3} \exp_stop_f:
    \prg_return_false: \else: \prg_return_true: \fi:
  }
\cs_new:Npn \__int_compare:nnN #1#2#3
  {
        {#2} \exp_stop_f:
      \prg_return_false: \exp_after:wN \__int_use_none_delimit_by_s_stop:w
    \fi:
    #1 #2 #3 \exp_after:wN \__int_compare:Nw \int_value:w \__int_eval:w
  }
\cs_new:cpn { __int_compare_=:NNw } #1#2#3 =
  { \__int_compare:nnN { \reverse_if:N \if_int_compare:w } {#3} = }
\cs_new:cpn { __int_compare_<:NNw } #1#2#3 <
  { \__int_compare:nnN { \reverse_if:N \if_int_compare:w } {#3} < }
\cs_new:cpn { __int_compare_>:NNw } #1#2#3 >
  { \__int_compare:nnN { \reverse_if:N \if_int_compare:w } {#3} > }
\cs_new:cpn { __int_compare_==:NNw } #1#2#3 ==
  { \__int_compare:nnN { \reverse_if:N \if_int_compare:w } {#3} = }
\cs_new:cpn { __int_compare_!=:NNw } #1#2#3 !=
  { \__int_compare:nnN { \if_int_compare:w } {#3} = }
\cs_new:cpn { __int_compare_<=:NNw } #1#2#3 <=
  { \__int_compare:nnN { \if_int_compare:w } {#3} > }
\cs_new:cpn { __int_compare_>=:NNw } #1#2#3 >=
  { \__int_compare:nnN { \if_int_compare:w } {#3} < }
\prg_new_conditional:Npnn \int_compare:nNn #1#2#3 { p , T , F , TF }
  {
    \if_int_compare:w \__int_eval:w #1 #2 \__int_eval:w #3 \__int_eval_end:
      \prg_return_true:
    \else:
      \prg_return_false:
    \fi:
  }
\prg_new_conditional:Npnn \int_if_zero:n #1 { p , T , F , TF }
  {
    \if_int_compare:w \__int_eval:w #1 = \c_zero_int
```

```
      \prg_return_true:
    \else:
      \prg_return_false:
    \fi:
  }
\cs_new:Npn \int_case:nnTF #1
  {
    \exp:w
    \exp_args:Nf \__int_case:nnTF { \int_eval:n {#1} }
  }
\cs_new:Npn \int_case:nnT #1#2#3
  {
    \exp:w
    \exp_args:Nf \__int_case:nnTF { \int_eval:n {#1} } {#2} {#3} { }
  }
\cs_new:Npn \int_case:nnF #1#2
  {
    \exp:w
    \exp_args:Nf \__int_case:nnTF { \int_eval:n {#1} } {#2} { }
  }
\cs_new:Npn \int_case:nn #1#2
  {
    \exp:w
    \exp_args:Nf \__int_case:nnTF { \int_eval:n {#1} } {#2} { } { }
  }
\cs_new:Npn \__int_case:nnTF #1#2#3#4
  { \__int_case:nw {#1} #2 {#1} { } \s__int_mark {#3} \s__int_mark {#4} \s__int_stop }
\cs_new:Npn \__int_case:nw #1#2#3
  {
    \int_compare:nNnTF {#1} = {#2}
      { \__int_case_end:nw {#3} }
      { \__int_case:nw {#1} }
  }
\cs_new:Npn \__int_case_end:nw #1#2#3 \s__int_mark #4#5 \s__int_stop
  { \exp_end: #1 #4 }
\prg_new_conditional:Npnn \int_if_odd:n #1 { p , T , F , TF}
  {
    \if_int_odd:w \__int_eval:w #1 \__int_eval_end:
      \prg_return_true:
    \else:
      \prg_return_false:
    \fi:
  }
\prg_new_conditional:Npnn \int_if_even:n #1 { p , T , F , TF}
  {
    \reverse_if:N \if_int_odd:w \__int_eval:w #1 \__int_eval_end:
      \prg_return_true:
    \else:
      \prg_return_false:
    \fi:
  }
\cs_new:Npn \int_while_do:nn #1#2
  {
    \int_compare:nT {#1}
      {
        #2
        \int_while_do:nn {#1} {#2}
      }
  }
\cs_new:Npn \int_until_do:nn #1#2
  {
    \int_compare:nF {#1}
      {
        #2
        \int_until_do:nn {#1} {#2}
      }
  }
\cs_new:Npn \int_do_while:nn #1#2
  {
    #2
    \int_compare:nT {#1}
      { \int_do_while:nn {#1} {#2} }
  }
\cs_new:Npn \int_do_until:nn #1#2
  {
    #2
    \int_compare:nF {#1}
      { \int_do_until:nn {#1} {#2} }
  }
\cs_new:Npn \int_while_do:nNnn #1#2#3#4
  {
    \int_compare:nNnT {#1} #2 {#3}
      {
        #4
        \int_while_do:nNnn {#1} #2 {#3} {#4}
      }
  }
\cs_new:Npn \int_until_do:nNnn #1#2#3#4
  {
    \int_compare:nNnF {#1} #2 {#3}
      {
        #4
        \int_until_do:nNnn {#1} #2 {#3} {#4}
      }
  }
\cs_new:Npn \int_do_while:nNnn #1#2#3#4
  {
    #4
    \int_compare:nNnT {#1} #2 {#3}
      { \int_do_while:nNnn {#1} #2 {#3} {#4} }
  }
\cs_new:Npn \int_do_until:nNnn #1#2#3#4
  {
    #4
    \int_compare:nNnF {#1} #2 {#3}
      { \int_do_until:nNnn {#1} #2 {#3} {#4} }
  }
\cs_new:Npn \int_step_function:nnnN #1#2#3
  {
    \exp_after:wN \__int_step:w
    \int_value:w \__int_eval:w #1 \exp_after:wN \__int_sep:
    \int_value:w \__int_eval:w #2 \exp_after:wN \__int_sep:
    \int_value:w \__int_eval:w #3 \__int_sep:
  }
\cs_new:Npn \__int_step:w #1 \__int_sep: #2 \__int_sep: #3 \__int_sep: #4
  {
    \int_compare:nNnTF {#2} > \c_zero_int
      { \__int_step:Nw > }
      {
        \if_meaning:w 0 #2
          \exp_after:wN \use_ii:nn
        \fi:
        \use_none:n
          {
            \msg_expandable_error:nnn
              { kernel } { zero-step } {#4}
            \prg_break:
          }
        \__int_step:Nw <
      }
      #1 \__int_sep: {#2} {#3} {#4}
    \prg_break_point:
  }
\cs_new:Npn \__int_step:Nw #1#2 \__int_sep: #3#4#5
  {
    \if_int_compare:w #2 #1 #4 \exp_stop_f:
      \prg_break:n
    \fi:
```

```
    #5 {#2}
    \exp_after:wN \__int_step:Nw
    \exp_after:wN #1
    \int_value:w \__int_eval:w #2 + #3 \__int_sep: {#3} {#4} {#5}
  }
\cs_new:Npn \int_step_function:nN
  { \int_step_function:nnnN \c_one_int \c_one_int }
\cs_new:Npn \int_step_function:nnN #1
  { \int_step_function:nnnN {#1} \c_one_int }
\cs_new_eq:NN \int_step_tokens:nn   \int_step_function:nN
\cs_new_eq:NN \int_step_tokens:nnn  \int_step_function:nnN
\cs_new_eq:NN \int_step_tokens:nnnn \int_step_function:nnnN
\cs_new_protected:Npn \int_step_inline:nn
  { \int_step_inline:nnnn { 1 } { 1 } }
\cs_new_protected:Npn \int_step_inline:nnn #1
  { \int_step_inline:nnnn {#1} { 1 } }
\cs_new_protected:Npn \int_step_inline:nnnn
  {
    \int_gincr:N \g__kernel_prg_map_int
    \exp_args:NNc \__int_step:NNnnnn
      \cs_gset_protected:Npn
      { __int_map_ \int_use:N \g__kernel_prg_map_int :w }
  }
\cs_new_protected:Npn \int_step_variable:nNn
  { \int_step_variable:nnnNn { 1 } { 1 } }
\cs_new_protected:Npn \int_step_variable:nnNn #1
  { \int_step_variable:nnnNn {#1} { 1 } }
\cs_new_protected:Npn \int_step_variable:nnnNn #1#2#3#4#5
  {
    \int_gincr:N \g__kernel_prg_map_int
    \exp_args:NNc \__int_step:NNnnnn
      \cs_gset_protected:Npe
      { __int_map_ \int_use:N \g__kernel_prg_map_int :w }
      {#1}{#2}{#3}
      {
        \tl_set:Nn \exp_not:N #4 {##1}
        \exp_not:n {#5}
      }
  }
\cs_new_protected:Npn \__int_step:NNnnnn #1#2#3#4#5#6
  {
    #1 #2 ##1 {#6}
    \int_step_function:nnnN {#3} {#4} {#5} #2
    \prg_break_point:Nn \scan_stop: { \int_gdecr:N \g__kernel_prg_map_int }
  }
\cs_new_eq:NN \int_to_arabic:n \int_eval:n
```

```
\cs_generate_variant:Nn \int_to_arabic:n { v }
\cs_new:Npn \int_to_symbols:nnn #1#2#3
  {
    \int_compare:nNnTF {#1} > {#2}
      {
        \__int_to_symbols:ennn
          {
            \int_case:nn
              { 1 + \int_mod:nn { #1 - 1 } {#2} }
              {#3}
          }
          {#1} {#2} {#3}
      }
      { \int_case:nn {#1} {#3} }
  }
\cs_new:Npn \__int_to_symbols:nnnn #1#2#3#4
  {
    \exp_args:Nf \int_to_symbols:nnn
      { \int_div_truncate:nn { #2 - 1 } {#3} } {#3} {#4}
    #1
  }
\cs_generate_variant:Nn \__int_to_symbols:nnnn { e }
\cs_new:Npn \int_to_alph:n #1
  {
    \int_to_symbols:nnn {#1} { 26 }
      {
        {  1 } { a }
        {  2 } { b }
        {  3 } { c }
        {  4 } { d }
        {  5 } { e }
        {  6 } { f }
        {  7 } { g }
        {  8 } { h }
        {  9 } { i }
        { 10 } { j }
        { 11 } { k }
        { 12 } { l }
        { 13 } { m }
        { 14 } { n }
        { 15 } { o }
        { 16 } { p }
        { 17 } { q }
        { 18 } { r }
        { 19 } { s }
        { 20 } { t }
        { 21 } { u }
        { 22 } { v }
        { 23 } { w }
        { 24 } { x }
        { 25 } { y }
        { 26 } { z }
      }
  }
\cs_new:Npn \int_to_Alph:n #1
  {
    \int_to_symbols:nnn {#1} { 26 }
      {
        {  1 } { A }
        {  2 } { B }
        {  3 } { C }
        {  4 } { D }
        {  5 } { E }
        {  6 } { F }
        {  7 } { G }
        {  8 } { H }
        {  9 } { I }
        { 10 } { J }
        { 11 } { K }
        { 12 } { L }
        { 13 } { M }
        { 14 } { N }
        { 15 } { O }
        { 16 } { P }
        { 17 } { Q }
        { 18 } { R }
        { 19 } { S }
        { 20 } { T }
        { 21 } { U }
        { 22 } { V }
        { 23 } { W }
        { 24 } { X }
        { 25 } { Y }
        { 26 } { Z }
      }
  }
\cs_new:Npn \int_to_base:nn #1
  { \exp_args:Nf \__int_to_base:nn { \int_eval:n {#1} } }
\cs_new:Npn \int_to_Base:nn #1
  { \exp_args:Nf \__int_to_Base:nn { \int_eval:n {#1} } }
\cs_new:Npn \__int_to_base:nn #1#2
  {
    \int_compare:nNnTF {#1} < 0
      { \exp_args:No \__int_to_base:nnN { \use_none:n #1 } {#2} - }
      { \__int_to_base:nnN {#1} {#2} \c_empty_tl }
  }
\cs_new:Npn \__int_to_Base:nn #1#2
  {
    \int_compare:nNnTF {#1} < 0
      { \exp_args:No \__int_to_Base:nnN { \use_none:n #1 } {#2} - }
      { \__int_to_Base:nnN {#1} {#2} \c_empty_tl }
  }
\cs_new:Npn \__int_to_base:nnN #1#2#3
  {
    \int_compare:nNnTF {#1} < {#2}
      { \exp_last_unbraced:Nf #3 { \__int_to_letter:n {#1} } }
      {
        \exp_args:Nf \__int_to_base:nnnN
          { \__int_to_letter:n { \int_mod:nn {#1} {#2} } }
          {#1}
          {#2}
          #3
      }
  }
\cs_new:Npn \__int_to_base:nnnN #1#2#3#4
  {
    \exp_args:Nf \__int_to_base:nnN
      { \int_div_truncate:nn {#2} {#3} }
      {#3}
      #4
    #1
  }
\cs_new:Npn \__int_to_Base:nnN #1#2#3
  {
    \int_compare:nNnTF {#1} < {#2}
      { \exp_last_unbraced:Nf #3 { \__int_to_Letter:n {#1} } }
      {
        \exp_args:Nf \__int_to_Base:nnnN
          { \__int_to_Letter:n { \int_mod:nn {#1} {#2} } }
          {#1}
          {#2}
          #3
      }
  }
\cs_new:Npn \__int_to_Base:nnnN #1#2#3#4
  {
    \exp_args:Nf \__int_to_Base:nnN
      { \int_div_truncate:nn {#2} {#3} }
      {#3}
      #4
    #1
  }
\cs_new:Npn \__int_to_letter:n #1
  {
    \exp_after:wN \exp_after:wN
    \if_case:w \__int_eval:w #1 - 10 \__int_eval_end:
         a
    \or: b
    \or: c
    \or: d
    \or: e
    \or: f
    \or: g
    \or: h
    \or: i
    \or: j
    \or: k
    \or: l
    \or: m
    \or: n
    \or: o
    \or: p
    \or: q
    \or: r
    \or: s
    \or: t
    \or: u
    \or: v
    \or: w
    \or: x
    \or: y
    \or: z
    \else: \int_value:w \__int_eval:w #1 \exp_after:wN \__int_eval_end:
    \fi:
  }
\cs_new:Npn \__int_to_Letter:n #1
  {
    \exp_after:wN \exp_after:wN
    \if_case:w \__int_eval:w #1 - 10 \__int_eval_end:
         A
    \or: B
    \or: C
    \or: D
    \or: E
    \or: F
    \or: G
    \or: H
    \or: I
    \or: J
    \or: K
    \or: L
    \or: M
    \or: N
    \or: O
    \or: P
    \or: Q
    \or: R
    \or: S
    \or: T
    \or: U
    \or: V
    \or: W
    \or: X
    \or: Y
    \or: Z
    \else: \int_value:w \__int_eval:w #1 \exp_after:wN \__int_eval_end:
    \fi:
  }
\cs_new:Npn \int_to_bin:n #1
  { \int_to_base:nn {#1} { 2 } }
\cs_new:Npn \int_to_hex:n #1
  { \int_to_base:nn {#1} { 16 } }
\cs_new:Npn \int_to_Hex:n #1
  { \int_to_Base:nn {#1} { 16 } }
\cs_new:Npn \int_to_oct:n #1
  { \int_to_base:nn {#1} { 8 } }
\cs_new:Npn \int_to_roman:n #1
  {
    \exp_after:wN \__int_to_roman:N
      \__int_to_roman:w \int_eval:n {#1} Q
  }
\cs_new:Npn \__int_to_roman:N #1
  {
    \use:c { __int_to_roman_ #1 :w }
    \__int_to_roman:N
  }
\cs_new:Npn \int_to_Roman:n #1
  {
    \exp_after:wN \__int_to_Roman_aux:N
      \__int_to_roman:w \int_eval:n {#1} Q
  }
\cs_new:Npn \__int_to_Roman_aux:N #1
  {
    \use:c { __int_to_Roman_ #1 :w }
    \__int_to_Roman_aux:N
  }
\cs_new:Npn \__int_to_roman_i:w { i }
\cs_new:Npn \__int_to_roman_v:w { v }
\cs_new:Npn \__int_to_roman_x:w { x }
\cs_new:Npn \__int_to_roman_l:w { l }
\cs_new:Npn \__int_to_roman_c:w { c }
\cs_new:Npn \__int_to_roman_d:w { d }
\cs_new:Npn \__int_to_roman_m:w { m }
\cs_new:Npn \__int_to_roman_Q:w #1 { }
\cs_new:Npn \__int_to_Roman_i:w { I }
\cs_new:Npn \__int_to_Roman_v:w { V }
\cs_new:Npn \__int_to_Roman_x:w { X }
\cs_new:Npn \__int_to_Roman_l:w { L }
\cs_new:Npn \__int_to_Roman_c:w { C }
\cs_new:Npn \__int_to_Roman_d:w { D }
\cs_new:Npn \__int_to_Roman_m:w { M }
\cs_new:Npn \__int_to_Roman_Q:w #1 { }
\cs_new:Npn \__int_pass_signs:wn #1
  {
    \if:w + \if:w - \exp_not:N #1 + \fi: \exp_not:N #1
      \exp_after:wN \__int_pass_signs:wn
    \else:
      \exp_after:wN \__int_pass_signs_end:wn
      \exp_after:wN #1
    \fi:
  }
\cs_new:Npn \__int_pass_signs_end:wn #1 \s__int_stop #2 { #2 #1 }
\cs_new:Npn \int_from_alph:n #1
  {
    \int_eval:n
      {
        \exp_after:wN \__int_pass_signs:wn \tl_to_str:n {#1}
          \s__int_stop { \__int_from_alph:nN { 0 } }
        \q__int_recursion_tail \q__int_recursion_stop
      }
  }
\cs_new:Npn \__int_from_alph:nN #1#2
  {
    \__int_if_recursion_tail_stop_do:Nn #2 {#1}
    \exp_args:Nf \__int_from_alph:nN
      { \int_eval:n { #1 * 26 + \__int_from_alph:N #2 } }
  }
\cs_new:Npn \__int_from_alph:N #1
  { `#1 - \int_compare:nNnTF { `#1 } < { 91 } { 64 } { 96 } }
\cs_new:Npn \int_from_base:nn #1#2
  {
    \int_eval:n
      {
        \exp_after:wN \__int_pass_signs:wn \tl_to_str:n {#1}
          \s__int_stop { \__int_from_base:nnN { 0 } {#2} }
        \q__int_recursion_tail \q__int_recursion_stop
      }
  }
\cs_new:Npn \__int_from_base:nnN #1#2#3
  {
    \__int_if_recursion_tail_stop_do:Nn #3 {#1}
    \exp_args:Nf \__int_from_base:nnN
      { \int_eval:n { #1 * #2 + \__int_from_base:N #3 } }
      {#2}
  }
\cs_new:Npn \__int_from_base:N #1
  {
    \int_compare:nNnTF { `#1 } < { 58 }
      {#1}
      { `#1 - \int_compare:nNnTF { `#1 } < { 91 } { 55 } { 87 } }
  }
\cs_new:Npn \int_from_bin:n #1
  { \int_from_base:nn {#1} { 2 } }
\cs_new:Npn \int_from_hex:n #1
  { \int_from_base:nn {#1} { 16 } }
\cs_new:Npn \int_from_oct:n #1
  { \int_from_base:nn {#1} { 8 } }
\int_const:cn { c__int_from_roman_i_int } { 1 }
\int_const:cn { c__int_from_roman_v_int } { 5 }
\int_const:cn { c__int_from_roman_x_int } { 10 }
\int_const:cn { c__int_from_roman_l_int } { 50 }
\int_const:cn { c__int_from_roman_c_int } { 100 }
\int_const:cn { c__int_from_roman_d_int } { 500 }
\int_const:cn { c__int_from_roman_m_int } { 1000 }
\int_const:cn { c__int_from_roman_I_int } { 1 }
\int_const:cn { c__int_from_roman_V_int } { 5 }
\int_const:cn { c__int_from_roman_X_int } { 10 }
\int_const:cn { c__int_from_roman_L_int } { 50 }
\int_const:cn { c__int_from_roman_C_int } { 100 }
\int_const:cn { c__int_from_roman_D_int } { 500 }
\int_const:cn { c__int_from_roman_M_int } { 1000 }
\cs_new:Npn \int_from_roman:n #1
```

```
  {
    \int_eval:n
      {
        (
          0
          \exp_after:wN \__int_from_roman:NN \tl_to_str:n {#1}
          \q__int_recursion_tail \q__int_recursion_tail \q__int_recursion_stop
        )
      }
  }
\cs_new:Npn \__int_from_roman:NN #1#2
  {
    \__int_if_recursion_tail_stop:N #1
    \int_if_exist:cF { c__int_from_roman_ #1 _int }
      { \__int_from_roman_error:w }
    \__int_if_recursion_tail_stop_do:Nn #2
      { + \use:c { c__int_from_roman_ #1 _int } }
    \int_if_exist:cF { c__int_from_roman_ #2 _int }
      { \__int_from_roman_error:w }
    \int_compare:nNnTF
      { \use:c { c__int_from_roman_ #1 _int } }
      <
      { \use:c { c__int_from_roman_ #2 _int } }
      {
        + \use:c { c__int_from_roman_ #2 _int }
        - \use:c { c__int_from_roman_ #1 _int }
        \__int_from_roman:NN
      }
      {
        + \use:c { c__int_from_roman_ #1 _int }
        \__int_from_roman:NN #2
      }
  }
\cs_new:Npn \__int_from_roman_error:w #1 \q__int_recursion_stop #2
  { #2 * 0 - 1 }
\cs_new_eq:NN \int_show:N \__kernel_register_show:N
\cs_generate_variant:Nn \int_show:N { c }
\cs_new_protected:Npn \int_show:n
  { \__kernel_msg_show_eval:Nn \int_eval:n }
\cs_new_eq:NN \int_log:N \__kernel_register_log:N
\cs_generate_variant:Nn \int_log:N { c }
\cs_new_protected:Npn \int_log:n
  { \__kernel_msg_log_eval:Nn \int_eval:n }
\int_const:Nn \c_one_int { 1 }
\int_const:Nn \c_max_int { 2 147 483 647 }
\int_const:Nn \c_max_char_int
```

```
  {
    \if_int_odd:w 0
      \cs_if_exist:NT \tex_luatexversion:D  { 1 }
      \cs_if_exist:NT \tex_XeTeXversion:D    { 1 } ~
      "10FFFF
    \else:
      "FF
    \fi:
  }
\int_new:N \l_tmpa_int
\int_new:N \l_tmpb_int
\int_new:N \g_tmpa_int
\int_new:N \g_tmpb_int
\int_new:N \l__seq_internal_a_int
\int_new:N \l__seq_internal_b_int
%% File: l3flag.dtx
\cs_new_eq:NN \__flag_sep: \__kernel_int_sep:
\cs_new_protected:Npn \flag_new:N #1
  { \cs_new_protected:Npe #1 { \cs_to_str:N #1 } }
\cs_generate_variant:Nn \flag_new:N { c }
\flag_new:N \l_tmpa_flag
\flag_new:N \l_tmpb_flag
\cs_new_protected:Npn \flag_clear:N #1
  {
    \__flag_clear:wN 0 \__flag_sep: #1
    \prg_break_point:
  }
\cs_generate_variant:Nn \flag_clear:N { c }
\cs_new_protected:Npn \__flag_clear:wN #1 \__flag_sep: #2
  {
    \if_cs_exist:w #2 #1 \cs_end: \else:
      \prg_break:n
    \fi:
    \cs_set_eq:cN { #2 #1 } \tex_undefined:D
    \exp_after:wN \__flag_clear:wN
    \int_value:w \int_eval:w \c_one_int + #1 \__flag_sep: #2
  }
\cs_new_protected:Npn \flag_clear_new:N #1
  { \flag_if_exist:NTF #1 { \flag_clear:N } { \flag_new:N } #1 }
\cs_generate_variant:Nn \flag_clear_new:N { c }
\cs_new_protected:Npn \flag_show:N { \__flag_show:NN \tl_show:n }
\cs_generate_variant:Nn \flag_show:N { c }
\cs_new_protected:Npn \flag_log:N { \__flag_show:NN \tl_log:n }
\cs_generate_variant:Nn \flag_log:N { c }
\cs_new_protected:Npn \__flag_show:NN #1#2
  {
    \__kernel_chk_defined:NT #2
      { \exp_args:Ne #1 { \tl_to_str:n { #2 height } = \flag_height:N #2 } }
  }
\prg_new_eq_conditional:NNn \flag_if_exist:N \cs_if_exist:N
  { TF , T , F , p }
\prg_new_eq_conditional:NNn \flag_if_exist:c \cs_if_exist:c
  { TF , T , F , p }
\prg_new_conditional:Npnn \flag_if_raised:N #1 { p , T , F , TF }
  {
    \if_cs_exist:w #1 0 \cs_end:
      \prg_return_true:
    \else:
      \prg_return_false:
    \fi:
  }
\prg_generate_conditional_variant:Nnn \flag_if_raised:N
  { c } { p , T , F , TF }
\cs_new:Npn \flag_height:N #1 { \__flag_height_loop:wN 0 \__flag_sep: #1 }
\cs_new:Npn \__flag_height_loop:wN #1 \__flag_sep: #2
  {
    \if_cs_exist:w #2 #1 \cs_end: \else:
      \exp_after:wN \__flag_height_end:wN
    \fi:
    \exp_after:wN \__flag_height_loop:wN
    \int_value:w \int_eval:w \c_one_int + #1 \__flag_sep: #2
  }
\cs_new:Npn \__flag_height_end:wN #1 + #2 \__flag_sep: #3 {#2}
\cs_generate_variant:Nn \flag_height:N { c }
\cs_new:Npn \flag_raise:N #1
  { \exp_after:wN \use_none:n \cs:w #1 \flag_height:N #1 \cs_end: }
\cs_generate_variant:Nn \flag_raise:N { c }
\cs_new:Npn \flag_ensure_raised:N #1
  { \exp_after:wN \use_none:n \cs:w #1 0 \cs_end: }
\cs_generate_variant:Nn \flag_ensure_raised:N { c }
\cs_new_protected:Npn \flag_new:n #1 { \flag_new:c { l_#1_flag } }
\cs_new_protected:Npn \flag_clear:n #1 { \flag_clear:c { l_#1_flag } }
\cs_new_protected:Npn \flag_clear_new:n #1 { \flag_clear_new:c { l_#1_flag } }
\cs_new:Npn \flag_if_exist_p:n #1 { \flag_if_exist_p:c { l_#1_flag } }
\cs_new:Npn \flag_if_exist:nT #1 { \flag_if_exist:cT { l_#1_flag } }
\cs_new:Npn \flag_if_exist:nF #1 { \flag_if_exist:cF { l_#1_flag } }
\cs_new:Npn \flag_if_exist:nTF #1 { \flag_if_exist:cTF { l_#1_flag } }
\cs_new:Npn \flag_if_raised_p:n #1 { \flag_if_raised_p:c { l_#1_flag } }
\cs_new:Npn \flag_if_raised:nT #1 { \flag_if_raised:cT { l_#1_flag } }
\cs_new:Npn \flag_if_raised:nF #1 { \flag_if_raised:cF { l_#1_flag } }
\cs_new:Npn \flag_if_raised:nTF #1 { \flag_if_raised:cTF { l_#1_flag } }
\cs_new:Npn \flag_height:n #1 { \flag_height:c { l_#1_flag } }
```

```
\cs_new:Npn \flag_raise:n #1 { \flag_raise:c { l_#1_flag } }
\cs_new:Npn \flag_ensure_raised:n #1 { \flag_ensure_raised:c { l_#1_flag } }
\cs_new_protected:Npn \flag_show:n { \__flag_show:Nn \tl_show:n }
\cs_new_protected:Npn \flag_log:n { \__flag_show:Nn \tl_log:n }
\cs_new_protected:Npn \__flag_show:Nn #1#2
  {
    \exp_args:Nc \__kernel_chk_defined:NT { l_#2_flag }
      {
        \exp_args:Ne #1
          { \tl_to_str:n { flag~#2~height } = \flag_height:n {#2} }
      }
  }
%% File: l3prg.dtx
\cs_new_eq:NN \if_predicate:w \tex_ifodd:D
\cs_new_protected:Npn \bool_new:N #1 { \cs_new_eq:NN #1 \c_false_bool }
\cs_generate_variant:Nn \bool_new:N { c }
\cs_new_protected:Npn \bool_const:Nn #1#2
  {
    \__kernel_chk_if_free_cs:N #1
    \tex_global:D \tex_chardef:D #1 = \bool_if_p:n {#2}
  }
\cs_generate_variant:Nn \bool_const:Nn { c }
\cs_new_protected:Npn \bool_set_true:N #1
  { \cs_set_eq:NN #1 \c_true_bool }
\cs_new_protected:Npn \bool_set_false:N #1
  { \cs_set_eq:NN #1 \c_false_bool }
\cs_new_protected:Npn \bool_gset_true:N #1
  { \cs_gset_eq:NN #1 \c_true_bool }
\cs_new_protected:Npn \bool_gset_false:N #1
  { \cs_gset_eq:NN #1 \c_false_bool }
\cs_generate_variant:Nn \bool_set_true:N   { c }
\cs_generate_variant:Nn \bool_set_false:N  { c }
\cs_generate_variant:Nn \bool_gset_true:N  { c }
\cs_generate_variant:Nn \bool_gset_false:N { c }
\cs_new_eq:NN \bool_set_eq:NN  \tl_set_eq:NN
\cs_new_eq:NN \bool_gset_eq:NN \tl_gset_eq:NN
\cs_generate_variant:Nn \bool_set_eq:NN { Nc, cN, cc }
\cs_generate_variant:Nn \bool_gset_eq:NN { Nc, cN, cc }
\cs_new_protected:Npn \bool_set:Nn #1#2
  {
    \exp_last_unbraced:NNNf
      \tex_chardef:D #1 = { \bool_if_p:n {#2} }
  }
\cs_new_protected:Npn \bool_gset:Nn #1#2
  {
    \exp_last_unbraced:NNNNf
```

```
      \tex_global:D \tex_chardef:D #1 = { \bool_if_p:n {#2} }
  }
\cs_generate_variant:Nn \bool_set:Nn  { c }
\cs_generate_variant:Nn \bool_gset:Nn { c }
\cs_new_protected:Npn \bool_set_inverse:N #1
  { \bool_if:NTF #1 { \bool_set_false:N } { \bool_set_true:N } #1 }
\cs_generate_variant:Nn \bool_set_inverse:N { c }
\cs_new_protected:Npn \bool_gset_inverse:N #1
  { \bool_if:NTF #1 { \bool_gset_false:N } { \bool_gset_true:N } #1 }
\cs_generate_variant:Nn \bool_gset_inverse:N { c }
\quark_new:N \q__bool_recursion_tail
\quark_new:N \q__bool_recursion_stop
\cs_new:Npn \__bool_use_i_delimit_by_q_recursion_stop:nw
  #1 #2 \q__bool_recursion_stop {#1}
\__kernel_quark_new_test:N \__bool_if_recursion_tail_stop_do:nn
\prg_new_conditional:Npnn \bool_if:N #1 { p , T , F , TF }
  {
    \if_bool:N #1
      \prg_return_true:
    \else:
      \prg_return_false:
    \fi:
  }
\prg_generate_conditional_variant:Nnn \bool_if:N { c } { p , T , F , TF }
\cs_new:Npe \bool_to_str:N #1
  {
    \exp_not:N \bool_if:NTF #1
      { \tl_to_str:n { true } } { \tl_to_str:n { false } }
  }
\cs_generate_variant:Nn \bool_to_str:N { c }
\cs_new:Npe \bool_to_str:n #1
  {
    \exp_not:N \bool_if:nTF {#1}
      { \tl_to_str:n { true } } { \tl_to_str:n { false } }
  }
\cs_new_protected:Npn \bool_show:n
  { \__kernel_msg_show_eval:Nn \bool_to_str:n }
\cs_new_protected:Npn \bool_log:n
  { \__kernel_msg_log_eval:Nn \bool_to_str:n }
\cs_new_protected:Npn \bool_show:N { \__bool_show:NN \tl_show:n }
\cs_generate_variant:Nn \bool_show:N { c }
\cs_new_protected:Npn \bool_log:N { \__bool_show:NN \tl_log:n }
\cs_generate_variant:Nn \bool_log:N { c }
\cs_new_protected:Npn \__bool_show:NN #1#2
  {
    \__kernel_chk_defined:NT #2
```

```
      {
        \token_case_meaning:NnF #2
          {
            \c_true_bool { \exp_args:Ne #1 { \token_to_str:N #2 = true } }
            \c_false_bool { \exp_args:Ne #1 { \token_to_str:N #2 = false } }
          }
          {
            \msg_error:nneee { kernel } { bad-type }
              { \token_to_str:N #2 } { \token_to_meaning:N #2 } { bool }
          }
      }
  }
\bool_new:N \l_tmpa_bool
\bool_new:N \l_tmpb_bool
\bool_new:N \g_tmpa_bool
\bool_new:N \g_tmpb_bool
\prg_new_eq_conditional:NNn \bool_if_exist:N \cs_if_exist:N
  { TF , T , F , p }
\prg_new_eq_conditional:NNn \bool_if_exist:c \cs_if_exist:c
  { TF , T , F , p }
\prg_new_conditional:Npnn \bool_if:n #1 { T , F , TF }
  {
    \if_predicate:w \bool_if_p:n {#1}
      \prg_return_true:
    \else:
      \prg_return_false:
    \fi:
  }
\cs_new:Npn \bool_if_p:n { \exp_args:Nf \__bool_if_p:n }
\cs_new:Npn \__bool_if_p:n #1
  {
    \tl_if_empty:oT { \use_none:nn #1 . } { \__bool_if_p_aux:w }
    \group_align_safe_begin:
    \exp_after:wN
    \group_align_safe_end:
    \exp:w \exp_end_continue_f:w % (
    \__bool_get_next:NN \use_i:nnnn #1 )
  }
\cs_new:Npn \__bool_if_p_aux:w #1 \use_i:nnnn #2#3
  { \bool_if:NTF #2 \c_true_bool \c_false_bool }
\cs_new:Npn \__bool_get_next:NN #1#2
  {
    \use:c
      {
        __bool_
        \if_meaning:w !#2 ! \else: \if_meaning:w (#2 ( \else: p \fi: \fi:
```

```
        :Nw
      }
      #1 #2
  }
\cs_new:cpn { __bool_!:Nw } #1#2
  {
    \exp_after:wN \__bool_get_next:NN
    #1 \use_ii:nnnn \use_i:nnnn \use_iii:nnnn \use_iv:nnnn
  }
\cs_new:cpn { __bool_(:Nw } #1#2
  {
    \exp_after:wN \__bool_choose:NNN \exp_after:wN #1
    \int_value:w \__bool_get_next:NN \use_i:nnnn
  }
\cs_new:cpn { __bool_p:Nw } #1
  { \exp_after:wN \__bool_choose:NNN \exp_after:wN #1 \int_value:w }
\cs_new:Npn \__bool_choose:NNN #1#2#3
  {
    \use:c
      {
        __bool_ \token_to_str:N #3 _
        #1 #2 { \if_meaning:w 0 #2 1 \else: 0 \fi: } 2 0 :
      }
  }
\cs_new:cpn { __bool_)_0: } { \c_false_bool }
\cs_new:cpn { __bool_)_1: } { \c_true_bool }
\cs_new:cpn { __bool_)_2: } { \c_true_bool }
\cs_new:cpn { __bool_&_0: } & { \__bool_get_next:NN \use_iv:nnnn }
\cs_new:cpn { __bool_&_1: } & { \__bool_get_next:NN \use_i:nnnn }
\cs_new:cpn { __bool_&_2: } & { \__bool_get_next:NN \use_iii:nnnn }
\cs_new:cpn { __bool_|_0: } | { \__bool_get_next:NN \use_i:nnnn }
\cs_new:cpn { __bool_|_1: } | { \__bool_get_next:NN \use_iii:nnnn }
\cs_new:cpn { __bool_|_2: } | { \__bool_get_next:NN \use_iii:nnnn }
\cs_new:Npn \bool_lazy_all_p:n #1
  { \__bool_lazy_all:n #1 \q__bool_recursion_tail \q__bool_recursion_stop }
\prg_new_conditional:Npnn \bool_lazy_all:n #1 { T , F , TF }
  {
    \if_predicate:w \bool_lazy_all_p:n {#1}
      \prg_return_true:
    \else:
      \prg_return_false:
    \fi:
  }
\cs_new:Npn \__bool_lazy_all:n #1
  {
    \__bool_if_recursion_tail_stop_do:nn {#1} { \c_true_bool }
    \bool_if:nF {#1}
      { \__bool_use_i_delimit_by_q_recursion_stop:nw { \c_false_bool } }
    \__bool_lazy_all:n
  }
\prg_new_conditional:Npnn \bool_lazy_and:nn #1#2 { p , T , F , TF }
  {
    \if_predicate:w
        \bool_if:nTF {#1} { \bool_if_p:n {#2} } { \c_false_bool }
      \prg_return_true:
    \else:
      \prg_return_false:
    \fi:
  }
\cs_new:Npn \bool_lazy_any_p:n #1
  { \__bool_lazy_any:n #1 \q__bool_recursion_tail \q__bool_recursion_stop }
\prg_new_conditional:Npnn \bool_lazy_any:n #1 { T , F , TF }
  {
    \if_predicate:w \bool_lazy_any_p:n {#1}
      \prg_return_true:
    \else:
      \prg_return_false:
    \fi:
  }
\cs_new:Npn \__bool_lazy_any:n #1
  {
    \__bool_if_recursion_tail_stop_do:nn {#1} { \c_false_bool }
    \bool_if:nT {#1}
      { \__bool_use_i_delimit_by_q_recursion_stop:nw { \c_true_bool } }
    \__bool_lazy_any:n
  }
\prg_new_conditional:Npnn \bool_lazy_or:nn #1#2 { p , T , F , TF }
  {
    \if_predicate:w
        \bool_if:nTF {#1} { \c_true_bool } { \bool_if_p:n {#2} }
      \prg_return_true:
    \else:
      \prg_return_false:
    \fi:
  }
\cs_new:Npn \bool_not_p:n #1 { \bool_if_p:n { ! ( #1 ) } }
\prg_new_conditional:Npnn \bool_xor:nn #1#2 { p , T , F , TF }
  {
    \bool_if:nT {#1} \reverse_if:N
    \if_predicate:w \bool_if_p:n {#2}
      \prg_return_true:
    \else:
      \prg_return_false:
    \fi:
  }
\cs_new:Npn \bool_while_do:Nn #1#2
  { \bool_if:NT #1 { #2 \bool_while_do:Nn #1 {#2} } }
\cs_new:Npn \bool_until_do:Nn #1#2
  { \bool_if:NF #1 { #2 \bool_until_do:Nn #1 {#2} } }
\cs_generate_variant:Nn \bool_while_do:Nn { c }
\cs_generate_variant:Nn \bool_until_do:Nn { c }
\cs_new:Npn \bool_do_while:Nn #1#2
  { #2 \bool_if:NT #1 { \bool_do_while:Nn #1 {#2} } }
\cs_new:Npn \bool_do_until:Nn #1#2
  { #2 \bool_if:NF #1 { \bool_do_until:Nn #1 {#2} } }
\cs_generate_variant:Nn \bool_do_while:Nn { c }
\cs_generate_variant:Nn \bool_do_until:Nn { c }
\cs_new:Npn \bool_while_do:nn #1#2
  {
    \bool_if:nT {#1}
      {
        #2
        \bool_while_do:nn {#1} {#2}
      }
  }
\cs_new:Npn \bool_do_while:nn #1#2
  {
    #2
    \bool_if:nT {#1} { \bool_do_while:nn {#1} {#2} }
  }
\cs_new:Npn \bool_until_do:nn #1#2
  {
    \bool_if:nF {#1}
      {
        #2
        \bool_until_do:nn {#1} {#2}
      }
  }
\cs_new:Npn \bool_do_until:nn #1#2
  {
    #2
    \bool_if:nF {#1} { \bool_do_until:nn {#1} {#2}  }
  }
\scan_new:N \s__bool_mark
\scan_new:N \s__bool_stop
\cs_new:Npn \bool_case:nTF
  { \exp:w \__bool_case:nTF }
\cs_new:Npn \bool_case:nT #1#2
```

```
  { \exp:w \__bool_case:nTF {#1} {#2} { } }
\cs_new:Npn \bool_case:nF #1
  { \exp:w \__bool_case:nTF {#1} { } }
\cs_new:Npn \bool_case:n #1
  { \exp:w \__bool_case:nTF {#1} { } { } }
\cs_new:Npn \__bool_case:nTF #1#2#3
  {
    \__bool_case:w
    #1 \c_true_bool { } \s__bool_mark {#2} \s__bool_mark {#3} \s__bool_stop
  }
\cs_new:Npn \__bool_case:w #1#2
  {
    \bool_if:nTF {#1}
      { \__bool_case_end:nw {#2} }
      { \__bool_case:w }
  }
\cs_new:Npn \__bool_case_end:nw #1#2#3 \s__bool_mark #4#5 \s__bool_stop
  { \exp_end: #1 #4 }
\cs_new:Npn \prg_replicate:nn #1
  {
    \exp:w
      \exp_after:wN \__prg_replicate_first:N
      \int_value:w \int_eval:n {#1}
      \cs_end:
  }
\cs_new:Npn \__prg_replicate:N #1
  { \cs:w __prg_replicate_#1 :n \__prg_replicate:N }
\cs_new:Npn \__prg_replicate_first:N #1
  { \cs:w __prg_replicate_first_ #1 :n \__prg_replicate:N }
\cs_new:Npn \__prg_replicate_ :n #1 { \cs_end: }
\cs_new:cpn { __prg_replicate_0:n } #1
  { \cs_end: {#1#1#1#1#1#1#1#1#1#1} }
\cs_new:cpn { __prg_replicate_1:n } #1
  { \cs_end: {#1#1#1#1#1#1#1#1#1#1} #1 }
\cs_new:cpn { __prg_replicate_2:n } #1
  { \cs_end: {#1#1#1#1#1#1#1#1#1#1} #1#1 }
\cs_new:cpn { __prg_replicate_3:n } #1
  { \cs_end: {#1#1#1#1#1#1#1#1#1#1} #1#1#1 }
\cs_new:cpn { __prg_replicate_4:n } #1
  { \cs_end: {#1#1#1#1#1#1#1#1#1#1} #1#1#1#1 }
\cs_new:cpn { __prg_replicate_5:n } #1
  { \cs_end: {#1#1#1#1#1#1#1#1#1#1} #1#1#1#1#1 }
\cs_new:cpn { __prg_replicate_6:n } #1
  { \cs_end: {#1#1#1#1#1#1#1#1#1#1} #1#1#1#1#1#1 }
\cs_new:cpn { __prg_replicate_7:n } #1
  { \cs_end: {#1#1#1#1#1#1#1#1#1#1} #1#1#1#1#1#1#1 }
```

```
\cs_new:cpn { __prg_replicate_8:n } #1
  { \cs_end: {#1#1#1#1#1#1#1#1#1#1} #1#1#1#1#1#1#1#1 }
\cs_new:cpn { __prg_replicate_9:n } #1
  { \cs_end: {#1#1#1#1#1#1#1#1#1#1} #1#1#1#1#1#1#1#1#1 }
\cs_new:cpn { __prg_replicate_first_-:n } #1
  {
    \exp_end:
    \msg_expandable_error:nn { prg } { negative-replication }
  }
\cs_new:cpn { __prg_replicate_first_0:n } #1 { \exp_end: }
\cs_new:cpn { __prg_replicate_first_1:n } #1 { \exp_end: #1 }
\cs_new:cpn { __prg_replicate_first_2:n } #1 { \exp_end: #1#1 }
\cs_new:cpn { __prg_replicate_first_3:n } #1 { \exp_end: #1#1#1 }
\cs_new:cpn { __prg_replicate_first_4:n } #1 { \exp_end: #1#1#1#1 }
\cs_new:cpn { __prg_replicate_first_5:n } #1 { \exp_end: #1#1#1#1#1 }
\cs_new:cpn { __prg_replicate_first_6:n } #1 { \exp_end: #1#1#1#1#1#1 }
\cs_new:cpn { __prg_replicate_first_7:n } #1 { \exp_end: #1#1#1#1#1#1#1 }
\cs_new:cpn { __prg_replicate_first_8:n } #1 { \exp_end: #1#1#1#1#1#1#1#1 }
\cs_new:cpn { __prg_replicate_first_9:n } #1
  { \exp_end: #1#1#1#1#1#1#1#1#1 }
\prg_new_conditional:Npnn \mode_if_vertical: { p , T , F , TF }
  { \if_mode_vertical: \prg_return_true: \else: \prg_return_false: \fi: }
\prg_new_conditional:Npnn \mode_if_horizontal: { p , T , F , TF }
  { \if_mode_horizontal: \prg_return_true: \else: \prg_return_false: \fi: }
\prg_new_conditional:Npnn \mode_if_inner: { p , T , F , TF }
  { \if_mode_inner: \prg_return_true: \else: \prg_return_false: \fi: }
\prg_new_conditional:Npnn \mode_if_math: { p , T , F , TF }
  { \if_mode_math: \prg_return_true: \else: \prg_return_false: \fi: }
\group_begin:
\tex_catcode:D `\^^@ = 2 \exp_stop_f:
\cs_new:Npn \group_align_safe_begin:
  { \exp:w \if_false: { \fi: `^^@ \exp_stop_f: }
\tex_catcode:D `\^^@ = 1 \exp_stop_f:
\cs_new:Npn \group_align_safe_end:
  { \exp:w `^^@ \if_false: } \fi: \exp_stop_f: }
\group_end:
\int_new:N \g__kernel_prg_map_int
%% File: l3sys.dtx
\tl_new:N \l__sys_tmp_tl
\cs_new_protected:Npn \__sys_const:nn #1#2
  {
    \bool_if:nTF {#2}
      {
        \cs_new_eq:cN { #1 :T }  \use:n
        \cs_new_eq:cN { #1 :F }  \use_none:n
        \cs_new_eq:cN { #1 :TF } \use_i:nn
        \cs_new_eq:cN { #1 _p: } \c_true_bool
      }
      {
        \cs_new_eq:cN { #1 :T }  \use_none:n
        \cs_new_eq:cN { #1 :F }  \use:n
        \cs_new_eq:cN { #1 :TF } \use_ii:nn
        \cs_new_eq:cN { #1 _p: } \c_false_bool
      }
  }
\str_const:Ne \c_sys_engine_str
  {
    \cs_if_exist:NT \tex_luatexversion:D { luatex }
    \cs_if_exist:NT \tex_pdftexversion:D { pdftex }
    \cs_if_exist:NT \tex_kanjiskip:D
      {
        \cs_if_exist:NTF \tex_enablecjktoken:D
          { uptex }
          { ptex }
      }
    \cs_if_exist:NT \tex_XeTeXversion:D { xetex }
  }
\tl_map_inline:nn { { luatex } { pdftex } { ptex } { uptex } { xetex } }
  {
    \__sys_const:nn { sys_if_engine_ #1 }
      { \str_if_eq_p:Vn \c_sys_engine_str {#1} }
  }
\__sys_const:nn
  { sys_if_engine_opentype }
  { \cs_if_exist_p:N \tex_Umathcode:D }
\group_begin:
  \cs_set_eq:NN \lua_now:e    \tex_directlua:D
  \str_const:Ne \c_sys_engine_exec_str
    {
      \sys_if_engine_pdftex:T { pdf }
      \sys_if_engine_xetex:T  { xe  }
      \sys_if_engine_ptex:T   { ep  }
      \sys_if_engine_uptex:T  { eup }
      \sys_if_engine_luatex:T
        {
          lua \lua_now:e
            {
              if (pcall(require, 'luaharfbuzz')) then ~
                tex.print("hb") ~
              end
            }
        }
      tex
    }
\group_end:
\str_const:Ne \c_sys_engine_format_str
  {
    \cs_if_exist:NTF \fmtname
      {
        \bool_lazy_or:nnTF
          { \str_if_eq_p:Vn \fmtname { plain } }
          { \str_if_eq_p:Vn \fmtname { LaTeX2e } }
          {
            \sys_if_engine_pdftex:T
              { \int_compare:nNnT { \tex_pdfoutput:D } = { 1 } { pdf } }
            \sys_if_engine_xetex:T  { xe }
            \sys_if_engine_ptex:T   { p  }
            \sys_if_engine_uptex:T  { up }
            \sys_if_engine_luatex:T
              {
                \int_compare:nNnT { \tex_pdfoutput:D } = { 0 } { dvi }
                lua
              }
            \str_if_eq:VnTF \fmtname { LaTeX2e }
              { latex }
              {
                \bool_lazy_and:nnT
                  { \sys_if_engine_pdftex_p: }
                  { \int_compare_p:nNn { \tex_pdfoutput:D } = { 0 } }
                    { e }
                tex
              }
          }
          { \fmtname }
      }
      { unknown }
  }
\str_const:Ne \c_sys_engine_version_str
  {
    \str_case:on \c_sys_engine_str
      {
        { pdftex }
          {
            \int_div_truncate:nn { \tex_pdftexversion:D } { 100 }
            .
            \int_mod:nn { \tex_pdftexversion:D } { 100 }
            .
            \tex_pdftexrevision:D
        }
      { ptex }
        {
          \cs_if_exist:NT \tex_ptexversion:D
            {
              p
              \int_use:N  \tex_ptexversion:D
              .
              \int_use:N \tex_ptexminorversion:D
              \tex_ptexrevision:D
              -
              \int_use:N \tex_epTeXversion:D
            }
        }
      { luatex }
        {
          \int_div_truncate:nn { \tex_luatexversion:D } { 100 }
          .
          \int_mod:nn { \tex_luatexversion:D } { 100 }
          .
          \tex_luatexrevision:D
        }
      { uptex }
        {
          \cs_if_exist:NT \tex_ptexversion:D
            {
              p
              \int_use:N  \tex_ptexversion:D
              .
              \int_use:N \tex_ptexminorversion:D
              \tex_ptexrevision:D
              -
              u
              \int_use:N  \tex_uptexversion:D
              \tex_uptexrevision:D
              -
              \int_use:N \tex_epTeXversion:D
            }
        }
      { xetex }
        {
          \int_use:N \tex_XeTeXversion:D
          \tex_XeTeXrevision:D
        }
    }
}
```

```
\cs_new_protected:Npn \sys_load_backend:n #1
  {
    \sys_finalise:
    \str_if_exist:NTF \c_sys_backend_str
      {
        \str_if_eq:VnF \c_sys_backend_str {#1}
          { \msg_error:nn { sys } { backend-set } }
      }
      {
        \tl_if_blank:nF {#1}
          { \tl_gset:Nn \g__sys_backend_tl {#1} }
        \__sys_load_backend_check:N \g__sys_backend_tl
        \str_const:Ne \c_sys_backend_str { \g__sys_backend_tl }
        \__kernel_sys_configuration_load:n
          { l3backend- \c_sys_backend_str }
      }
  }
\cs_new_protected:Npn \__sys_load_backend_check:N #1
  {
    \sys_if_engine_xetex:TF
      {
        \str_case:VnF #1
          {
            { dvisvgm }   { }
            { xdvipdfmx } { \tl_gset:Nn #1 { xetex } }
            { xetex }     { }
          }
          {
            \msg_error:nnee { sys } { wrong-backend }
              #1 { xetex }
            \tl_gset:Nn #1 { xetex }
          }
      }
      {
        \sys_if_output_pdf:TF
          {
            \str_if_eq:VnTF #1 { pdfmode }
              {
                \sys_if_engine_luatex:TF
                  { \tl_gset:Nn #1 { luatex } }
                  { \tl_gset:Nn #1 { pdftex } }
              }
              {
                \bool_lazy_or:nnF
                  { \str_if_eq_p:Vn #1 { luatex } }
                  { \str_if_eq_p:Vn #1 { pdftex } }
                  {
                    \msg_error:nnee { sys } { wrong-backend }
                      #1 { \sys_if_engine_luatex:TF { luatex } { pdftex } }
                    \sys_if_engine_luatex:TF
                      { \tl_gset:Nn #1 { luatex } }
                      { \tl_gset:Nn #1 { pdftex } }
                  }
              }
          }
          {
            \str_case:VnF #1
              {
                { dvipdfmx } { }
                { dvips }    { }
                { dvisvgm }  { }
              }
              {
                \msg_error:nnee { sys } { wrong-backend }
                  #1 { dvips }
                \tl_gset:Nn #1 { dvips }
              }
          }
      }
  }
\cs_new_protected:Npn \sys_ensure_backend:
  {
    \str_if_exist:NF \c_sys_backend_str
      { \sys_load_backend:n { } }
  }
\bool_new:N \g__sys_debug_bool
\cs_new_protected:Npn \sys_load_debug:
  {
    \bool_if:NF \g__sys_debug_bool
      { \__kernel_sys_configuration_load:n { l3debug } }
    \bool_gset_true:N \g__sys_debug_bool
  }
\cs_if_exist:NT \@expl@finalise@setup@@
  {
    \tl_gput_right:Nn \@expl@finalise@setup@@
      {
        \tl_gput_right:Nn \@kernel@after@begindocument
          {
            \cs_gset_protected:Npn \sys_load_debug:
              { \msg_error:nn { sys } { load-debug-in-preamble } }
          }
      }
  }
\tl_new:N \l__sys_internal_tl
\tl_const:Ne \c__sys_marker_tl { : \token_to_str:N : }
\cs_new_protected:Npn \sys_get_shell:nnN #1#2#3
  {
    \sys_get_shell:nnNF {#1} {#2} #3
      { \tl_set:Nn #3 { \q_no_value } }
  }
\prg_new_protected_conditional:Npnn \sys_get_shell:nnN #1#2#3 { T , F , TF }
  {
    \sys_if_shell:TF
      { \exp_args:No \__sys_get:nnN { \tl_to_str:n {#1} } {#2} #3 }
      { \prg_return_false: }
  }
\cs_new_protected:Npn \__sys_get:nnN #1#2#3
  {
    \tl_if_in:nnTF {#1} { " }
      {
        \msg_error:nne
          { kernel } { quote-in-shell } {#1}
        \prg_return_false:
      }
      {
        \group_begin:
          \if_false: { \fi:
          \int_set_eq:NN \tex_tracingnesting:D \c_zero_int
          \exp_args:No \tex_everyeof:D { \c__sys_marker_tl }
          #2 \scan_stop:
          \exp_after:wN \__sys_get_do:Nw
          \exp_after:wN #3
          \exp_after:wN \prg_do_nothing:
            \tex_input:D | "#1" \scan_stop:
        \if_false: } \fi:
        \prg_return_true:
      }
  }
\exp_args:Nno \use:nn
  { \cs_new_protected:Npn \__sys_get_do:Nw #1#2 }
  { \c__sys_marker_tl }
  {
    \group_end:
    \tl_set:No #1 {#2}
  }
\sys_if_engine_luatex:F
  { \int_const:Nn \c__sys_shell_stream_int { 18 } }
\sys_if_engine_luatex:TF
```

```
  {
    \cs_new_protected:Npn \sys_shell_now:n #1
      { \__sys_shell_now:e { \exp_not:n {#1} } }
  }
  {
    \cs_new_protected:Npn \sys_shell_now:n #1
      { \iow_now:Nn \c__sys_shell_stream_int {#1} }
  }
\cs_generate_variant:Nn \sys_shell_now:n { e, x }
\sys_if_engine_luatex:TF
  {
    \cs_new_protected:Npn \sys_shell_shipout:n #1
    { \__sys_shell_shipout:e { \exp_not:n {#1} } }
  }
  {
    \cs_new_protected:Npn \sys_shell_shipout:n #1
      { \iow_shipout:Nn \c__sys_shell_stream_int {#1} }
  }
\cs_generate_variant:Nn \sys_shell_shipout:n { e , x }
\cs_new_protected:Npn \__kernel_sys_everyjob:
  {
    \tl_use:N \g__sys_everyjob_tl
    \tl_gclear:N \g__sys_everyjob_tl
  }
\cs_new_protected:Npn \__sys_everyjob:n #1
  { \tl_gput_right:Nn \g__sys_everyjob_tl {#1} }
\tl_new:N \g__sys_everyjob_tl
\__sys_everyjob:n
  { \cs_new_eq:NN \c_sys_jobname_str \tex_jobname:D }
\__sys_everyjob:n
  {
    \group_begin:
      \cs_set:Npn \__sys_tmp:w #1
        {
          \str_if_eq:eeTF { \cs_meaning:N #1 } { \token_to_str:N #1 }
            { #1 }
            {
              \cs_if_exist:NTF \tex_primitive:D
                {
                  \bool_lazy_and:nnTF
                    { \sys_if_engine_xetex_p: }
                    {
                      \int_compare_p:nNn
                        { \exp_after:wN \use_none:n \tex_XeTeXrevision:D }
                          < { 99999 }
                    }
                    { 0 }
                    { \tex_primitive:D #1 }
                }
                { 0 }
            }
        }
      \int_const:Nn \c_sys_minute_int
        { \int_mod:nn { \__sys_tmp:w \time } { 60 } }
      \int_const:Nn \c_sys_hour_int
        { \int_div_truncate:nn { \__sys_tmp:w \time } { 60 } }
      \int_const:Nn \c_sys_day_int   { \__sys_tmp:w \day }
      \int_const:Nn \c_sys_month_int { \__sys_tmp:w \month }
      \int_const:Nn \c_sys_year_int  { \__sys_tmp:w \year }
    \group_end:
  }
\__sys_everyjob:n
  {
    \str_const:Ne \c_sys_timestamp_str
      {
        \cs_if_exist:NTF \tex_directlua:D
          { \tex_directlua:D { tex.print(pdf.getcreationdate()) } }
          { \tex_creationdate:D }
      }
  }
\__sys_everyjob:n
  {
    \cs_new:Npn \sys_rand_seed: { \tex_the:D \tex_randomseed:D }
  }
\__sys_everyjob:n
  {
    \cs_new_protected:Npn \sys_gset_rand_seed:n #1
      { \tex_setrandomseed:D \int_eval:n {#1} \exp_stop_f: }
  }
\cs_new:Npe \sys_timer:
  {
    \sys_if_engine_luatex:TF
      { \exp_not:N \__sys_elapsedtime: }
      { \exp_not:N \int_value:w \exp_not:N \tex_elapsedtime:D }
  }
\__sys_everyjob:n
  {
    \int_const:Nn \c_sys_shell_escape_int
      {
        \sys_if_engine_luatex:TF
          {
            \tex_directlua:D
```

```
              { tex.sprint(status.shell_escape~or~os.execute()) }
          }
          { \tex_shellescape:D }
      }
  }
\__sys_everyjob:n
  {
    \__sys_const:nn { sys_if_shell }
      { \int_compare_p:nNn \c_sys_shell_escape_int > 0 }
    \__sys_const:nn { sys_if_shell_unrestricted }
      { \int_compare_p:nNn \c_sys_shell_escape_int = 1 }
    \__sys_const:nn { sys_if_shell_restricted }
      { \int_compare_p:nNn \c_sys_shell_escape_int = 2 }
  }
\cs_new_protected:Npn \sys_get_query:nN #1#2
  { \sys_get_query:nnnN {#1} { } { } #2 }
\cs_new_protected:Npn \sys_get_query:nnN #1#2#3
  { \sys_get_query:nnnN {#1} { } {#2} #3 }
\cs_new_protected:Npn \sys_get_query:nnnN #1#2#3#4
  {
    \tl_clear:N #4
    \__sys_get_query_auxi:neeN {#1} {#2} {#3} #4
  }
\cs_new:Npn \__sys_get_query_auxi:nnnN #1#2#3#4
  {
    \__sys_get_query_auxii:neeN {#1}
      { \tl_if_blank:nF {#2} { \tl_to_str:n { ~ #2 } } }
      {
        \tl_if_blank:nF {#3}
          {
            \c_space_tl
            \sys_if_shell_restricted:F '
            \tl_to_str:n {#3}
            \sys_if_shell_restricted:F '
          }
      }
      #4
  }
\cs_generate_variant:Nn \__sys_get_query_auxi:nnnN { nee }
\cs_new_protected:Npn \__sys_get_query_auxii:nnnN #1#2#3#4
  {
    \sys_if_shell:T
      {
        \sys_get_shell:nnN
          { l3sys-query~#1 #2 #3 }
          {
```

```
      \int_step_inline:nnn { 0 } { `A - 1 }
        { \char_set_catcode_other:n {##1} }
      \int_step_inline:nnn { `Z + 1 } { `a - 1 }
        { \char_set_catcode_other:n {##1} }
      \int_step_inline:nnn { `z + 1 } { 127 }
        { \char_set_catcode_other:n {##1} }
      \char_set_catcode_active:n { `\_ }
      \tex_endlinechar:D 13 \scan_stop:
     }
    \l__sys_tmp_tl
    \tl_if_empty:NF \l__sys_tmp_tl
     {
      \exp_after:wN \__sys_get_query:Nw \exp_after:wN #4
        \l__sys_tmp_tl \q_stop
     }
   }
 }
\cs_generate_variant:Nn \__sys_get_query_auxii:nnnN { nee }
\group_begin:
 \tex_lccode:D `\* = 13 \scan_stop:
 \tex_lowercase:D
  {
   \group_end:
   \cs_new_protected:Npn \__sys_get_query:Nw #1#2 * \q_stop
  }
  { \tl_set:Nn #1 {#2} }
\cs_new_protected:Npn \sys_split_query:nN #1#2
 { \sys_split_query:nnnN {#1} { } { } #2 }
\cs_new_protected:Npn \sys_split_query:nnN #1#2#3
 { \sys_split_query:nnnN {#1} { } {#2} #3 }
\group_begin:
 \tex_lccode:D `\* = 13 \scan_stop:
 \tex_lowercase:D
  {
   \group_end:
   \cs_new_protected:Npn \sys_split_query:nnnN #1#2#3#4
    {
     \seq_clear:N #4
     \sys_get_query:nnnN {#1} {#2} {#3} \l__sys_tmp_tl
     \tl_if_empty:NF \l__sys_tmp_tl
      { \seq_set_split:NnV #4 * \l__sys_tmp_tl }
    }
  }
\__sys_everyjob:n
 { \cs_gset_eq:NN \g_file_curr_name_str \tex_jobname:D }
\cs_new_protected:Npn \sys_finalise:
  {
    \__kernel_sys_everyjob:
    \tl_use:N \g__sys_finalise_tl
    \tl_gclear:N \g__sys_finalise_tl
  }
\cs_new_protected:Npn \__sys_finalise:n #1
  { \tl_gput_right:Nn \g__sys_finalise_tl {#1} }
\tl_new:N \g__sys_finalise_tl
\__sys_finalise:n
  {
    \str_const:Ne \c_sys_output_str
      {
        \int_compare:nNnTF
          { \cs_if_exist_use:NF \tex_pdfoutput:D { 0 } } > { 0 }
          { pdf }
          { dvi }
      }
    \__sys_const:nn { sys_if_output_dvi }
      { \str_if_eq_p:Vn \c_sys_output_str { dvi } }
    \__sys_const:nn { sys_if_output_pdf }
      { \str_if_eq_p:Vn \c_sys_output_str { pdf } }
  }
\tl_new:N \g__sys_backend_tl
\__sys_finalise:n
  {
    \__kernel_tl_gset:Nx \g__sys_backend_tl
      {
        \sys_if_engine_xetex:TF
          { xetex }
          {
            \sys_if_output_pdf:TF
              {
                \sys_if_engine_pdftex:TF
                  { pdftex }
                  { luatex }
              }
              { dvips }
          }
      }
  }
\__sys_finalise:n
  {
    \cs_if_exist:NT \@classoptionslist
      {
        \cs_if_eq:NNF \@classoptionslist \scan_stop:
          {
            \clist_map_inline:Nn \@classoptionslist
              {
                \str_case:nnT {#1}
                  {
                    { dvipdfmx }
                      { \tl_gset:Nn \g__sys_backend_tl { dvipdfmx } }
                    { dvips }
                      { \tl_gset:Nn \g__sys_backend_tl { dvips } }
                    { dvisvgm }
                      { \tl_gset:Nn \g__sys_backend_tl { dvisvgm } }
                    { pdftex }
                      { \tl_gset:Nn \g__sys_backend_tl { pdfmode } }
                    { xetex }
                      { \tl_gset:Nn \g__sys_backend_tl { xdvipdfmx } }
                  }
                  { \clist_remove_all:Nn \@unusedoptionlist {#1} }
              }
          }
      }
  }
%% File: l3clist.dtx
\cs_new_eq:NN \c_empty_clist \c_empty_tl
\tl_new:N \l__clist_internal_clist
\scan_new:N \s__clist_mark
\scan_new:N \s__clist_stop
\cs_new:Npn \__clist_use_none_delimit_by_s_mark:w #1 \s__clist_mark { }
\cs_new:Npn \__clist_use_none_delimit_by_s_stop:w #1 \s__clist_stop { }
\cs_new:Npn \__clist_use_i_delimit_by_s_stop:nw #1 #2 \s__clist_stop {#1}
\cs_new_protected:Npn \__clist_tmp:w { }
\cs_new:Npn \__clist_trim_next:w #1 ,
  {
    \tl_if_empty:oTF { \use_none:nn #1 ? }
      { \__clist_trim_next:w \prg_do_nothing: }
      { \tl_trim_spaces_apply:oN {#1} \exp_end: }
  }
\cs_new:Npn \__clist_sanitize:n #1
  {
    \exp_after:wN \__clist_sanitize:Nn \exp_after:wN \c_empty_tl
    \exp:w \__clist_trim_next:w \prg_do_nothing:
    #1 , \s__clist_stop \prg_break: , \prg_break_point:
  }
\cs_new:Npn \__clist_sanitize:Nn #1#2
  {
    \__clist_use_none_delimit_by_s_stop:w #2 \s__clist_stop
    #1 \__clist_wrap_item:w #2 ,
    \exp_after:wN \__clist_sanitize:Nn \exp_after:wN ,
    \exp:w \__clist_trim_next:w \prg_do_nothing:
  }
\prg_new_conditional:Npnn \__clist_if_wrap:n #1 { TF }
  {
    \tl_if_empty:oTF
      {
        \__clist_if_wrap:w
          \s__clist_mark ? #1 ~ \s__clist_mark ? ~ #1
          \s__clist_mark , ~ \s__clist_mark #1 ,
      }
      {
        \tl_if_head_is_group:nTF { #1 { } }
          {
            \tl_if_empty:nTF {#1}
              { \prg_return_true: }
              {
                \tl_if_empty:oTF { \use_none:n #1}
                  { \prg_return_true: }
                  { \prg_return_false: }
              }
          }
          { \prg_return_false: }
      }
      { \prg_return_true: }
  }
\cs_new:Npn \__clist_if_wrap:w #1 \s__clist_mark ? ~ #2 ~ \s__clist_mark #3 , { }
\cs_new:Npn \__clist_wrap_item:w #1 ,
  { \__clist_if_wrap:nTF {#1} { \exp_not:n { {#1} } } { \exp_not:n {#1} } }
\cs_new_eq:NN \clist_new:N \tl_new:N
\cs_new_eq:NN \clist_new:c \tl_new:c
\cs_new_protected:Npn \clist_const:Nn #1#2
  { \tl_const:Ne #1 { \__clist_sanitize:n {#2} } }
\cs_generate_variant:Nn \clist_const:Nn { Ne , c , ce }
\cs_generate_variant:Nn \clist_const:Nn { Nx , cx }
\cs_new_eq:NN \clist_clear:N  \tl_clear:N
\cs_new_eq:NN \clist_clear:c  \tl_clear:c
\cs_new_eq:NN \clist_gclear:N \tl_gclear:N
\cs_new_eq:NN \clist_gclear:c \tl_gclear:c
\cs_new_eq:NN \clist_clear_new:N  \tl_clear_new:N
\cs_new_eq:NN \clist_clear_new:c  \tl_clear_new:c
\cs_new_eq:NN \clist_gclear_new:N \tl_gclear_new:N
\cs_new_eq:NN \clist_gclear_new:c \tl_gclear_new:c
\cs_new_eq:NN \clist_set_eq:NN  \tl_set_eq:NN
\cs_new_eq:NN \clist_set_eq:Nc  \tl_set_eq:Nc
\cs_new_eq:NN \clist_set_eq:cN  \tl_set_eq:cN
\cs_new_eq:NN \clist_set_eq:cc  \tl_set_eq:cc
```

```
\cs_new_eq:NN \clist_gset_eq:NN \tl_gset_eq:NN
\cs_new_eq:NN \clist_gset_eq:Nc \tl_gset_eq:Nc
\cs_new_eq:NN \clist_gset_eq:cN \tl_gset_eq:cN
\cs_new_eq:NN \clist_gset_eq:cc \tl_gset_eq:cc
\cs_new_protected:Npn \clist_set_from_seq:NN
  { \__clist_set_from_seq:NNNN \clist_clear:N  \__kernel_tl_set:Nx  }
\cs_new_protected:Npn \clist_gset_from_seq:NN
  { \__clist_set_from_seq:NNNN \clist_gclear:N \__kernel_tl_gset:Nx }
\cs_new_protected:Npn \__clist_set_from_seq:NNNN #1#2#3#4
  {
    \seq_if_empty:NTF #4
      { #1 #3 }
      {
        #2 #3
          {
            \exp_after:wN \use_none:n \exp:w \exp_end_continue_f:w
            \seq_map_function:NN #4 \__clist_set_from_seq:n
          }
      }
  }
\cs_new:Npn \__clist_set_from_seq:n #1
  {
    ,
    \__clist_if_wrap:nTF {#1}
      { \exp_not:n { {#1} } }
      { \exp_not:n {#1} }
  }
\cs_generate_variant:Nn \clist_set_from_seq:NN  {     Nc }
\cs_generate_variant:Nn \clist_set_from_seq:NN  { c , cc }
\cs_generate_variant:Nn \clist_gset_from_seq:NN {     Nc }
\cs_generate_variant:Nn \clist_gset_from_seq:NN { c , cc }
\cs_new_protected:Npn \clist_concat:NNN
  { \__clist_concat:NNNN \__kernel_tl_set:Nx }
\cs_new_protected:Npn \clist_gconcat:NNN
  { \__clist_concat:NNNN \__kernel_tl_gset:Nx }
\cs_new_protected:Npn \__clist_concat:NNNN #1#2#3#4
  {
    #1 #2
      {
        \exp_not:o #3
        \clist_if_empty:NF #3 { \clist_if_empty:NF #4 { , } }
        \exp_not:o #4
      }
  }
\cs_generate_variant:Nn \clist_concat:NNN  { ccc }
\cs_generate_variant:Nn \clist_gconcat:NNN { ccc }
```

```
\prg_new_eq_conditional:NNn \clist_if_exist:N \cs_if_exist:N
  { TF , T , F , p }
\prg_new_eq_conditional:NNn \clist_if_exist:c \cs_if_exist:c
  { TF , T , F , p }
\cs_new_protected:Npn \clist_set:Nn #1#2
  { \__kernel_tl_set:Nx #1 { \__clist_sanitize:n {#2} } }
\cs_new_protected:Npn \clist_gset:Nn #1#2
  { \__kernel_tl_gset:Nx #1 { \__clist_sanitize:n {#2} } }
\cs_generate_variant:Nn \clist_set:Nn  { NV , Ne , c , cV , ce }
\cs_generate_variant:Nn \clist_set:Nn  { No , Nx , co , cx }
\cs_generate_variant:Nn \clist_gset:Nn { NV , Ne , c , cV , ce }
\cs_generate_variant:Nn \clist_gset:Nn { No , Nx , co , cx }
\cs_new_protected:Npn \clist_put_left:Nn
  { \__clist_put_left:NNNn \clist_concat:NNN \clist_set:Nn }
\cs_new_protected:Npn \clist_gput_left:Nn
  { \__clist_put_left:NNNn \clist_gconcat:NNN \clist_set:Nn }
\cs_new_protected:Npn \__clist_put_left:NNNn #1#2#3#4
  {
    #2 \l__clist_internal_clist {#4}
    #1 #3 \l__clist_internal_clist #3
  }
\cs_generate_variant:Nn \clist_put_left:Nn  { NV , Nv , Ne , c , cV , cv , ce }
\cs_generate_variant:Nn \clist_put_left:Nn  { No , Nx , co , cx }
\cs_generate_variant:Nn \clist_gput_left:Nn { NV , Nv , Ne , c , cV , cv , ce }
\cs_generate_variant:Nn \clist_gput_left:Nn { No , Nx , co , cx }
\cs_new_protected:Npn \clist_put_right:Nn
  { \__clist_put_right:NNNn \clist_concat:NNN \clist_set:Nn }
\cs_new_protected:Npn \clist_gput_right:Nn
  { \__clist_put_right:NNNn \clist_gconcat:NNN \clist_set:Nn }
\cs_new_protected:Npn \__clist_put_right:NNNn #1#2#3#4
  {
    #2 \l__clist_internal_clist {#4}
    #1 #3 #3 \l__clist_internal_clist
  }
\cs_generate_variant:Nn \clist_put_right:Nn
  { NV , Nv , Ne , c , cV , cv , ce }
\cs_generate_variant:Nn \clist_put_right:Nn
  { No , Nx , co , cx }
\cs_generate_variant:Nn \clist_gput_right:Nn
  { NV , Nv , Ne , c , cV , cv , ce }
\cs_generate_variant:Nn \clist_gput_right:Nn
  { No , Nx , co , cx }
\cs_new_protected:Npn \clist_get:NN #1#2
  {
    \if_meaning:w #1 \c_empty_clist
      \tl_set:Nn #2 { \q_no_value }
    \else:
      \exp_after:wN \__clist_get:wN #1 , \s__clist_stop #2
    \fi:
  }
\cs_new_protected:Npn \__clist_get:wN #1 , #2 \s__clist_stop #3
  { \tl_set:Nn #3 {#1} }
\cs_generate_variant:Nn \clist_get:NN { c }
\cs_new_protected:Npn \clist_pop:NN
  { \__clist_pop:NNN \__kernel_tl_set:Nx }
\cs_new_protected:Npn \clist_gpop:NN
  { \__clist_pop:NNN \__kernel_tl_gset:Nx }
\cs_new_protected:Npn \__clist_pop:NNN #1#2#3
  {
    \if_meaning:w #2 \c_empty_clist
      \tl_set:Nn #3 { \q_no_value }
    \else:
      \exp_after:wN \__clist_pop:wwNNN #2 , \s__clist_mark \s__clist_stop #1#2#3
    \fi:
  }
\cs_new_protected:Npn \__clist_pop:wwNNN #1 , #2 \s__clist_stop #3#4#5
  {
    \tl_set:Nn #5 {#1}
    #3 #4
      {
        \__clist_pop:wN \prg_do_nothing:
          #2 \exp_not:o
          , \s__clist_mark \use_none:n
        \s__clist_stop
      }
  }
\cs_new:Npn \__clist_pop:wN #1 , \s__clist_mark #2 #3 \s__clist_stop { #2 {#1} }
\cs_generate_variant:Nn \clist_pop:NN  { c }
\cs_generate_variant:Nn \clist_gpop:NN { c }
\prg_new_protected_conditional:Npnn \clist_get:NN #1#2 { T , F , TF }
  {
    \if_meaning:w #1 \c_empty_clist
      \prg_return_false:
    \else:
      \exp_after:wN \__clist_get:wN #1 , \s__clist_stop #2
      \prg_return_true:
    \fi:
  }
\prg_generate_conditional_variant:Nnn \clist_get:NN { c } { T , F , TF }
\prg_new_protected_conditional:Npnn \clist_pop:NN #1#2 { T , F , TF }
  { \__clist_pop_TF:NNN \__kernel_tl_set:Nx #1 #2 }
\prg_new_protected_conditional:Npnn \clist_gpop:NN #1#2 { T , F , TF }
```

```
  { \__clist_pop_TF:NNN \__kernel_tl_gset:Nx #1 #2 }
\cs_new_protected:Npn \__clist_pop_TF:NNN #1#2#3
  {
    \if_meaning:w #2 \c_empty_clist
      \prg_return_false:
    \else:
      \exp_after:wN \__clist_pop:wwNNN #2 , \s__clist_mark \s__clist_stop #1#2#3
      \prg_return_true:
    \fi:
  }
\prg_generate_conditional_variant:Nnn \clist_pop:NN { c } { T , F , TF }
\prg_generate_conditional_variant:Nnn \clist_gpop:NN { c } { T , F , TF }
\cs_new_eq:NN \clist_push:Nn  \clist_put_left:Nn
\cs_generate_variant:Nn \clist_push:Nn { NV , No , Nx , c , cV , co , cx }
\cs_new_eq:NN \clist_gpush:Nn \clist_gput_left:Nn
\cs_generate_variant:Nn \clist_gpush:Nn { NV , No , Nx , c , cV , co , cx }
\clist_new:N \l__clist_internal_remove_clist
\seq_new:N \l__clist_internal_remove_seq
\cs_new_protected:Npn \clist_remove_duplicates:N
  { \__clist_remove_duplicates:NN \clist_set_eq:NN }
\cs_new_protected:Npn \clist_gremove_duplicates:N
  { \__clist_remove_duplicates:NN \clist_gset_eq:NN }
\cs_new_protected:Npn \__clist_remove_duplicates:NN #1#2
  {
    \clist_clear:N \l__clist_internal_remove_clist
    \clist_map_inline:Nn #2
      {
        \clist_if_in:NnF \l__clist_internal_remove_clist {##1}
          {
            \tl_put_right:Ne \l__clist_internal_remove_clist
              {
                \clist_if_empty:NF \l__clist_internal_remove_clist { , }
                \__clist_if_wrap:nTF {##1} { \exp_not:n { {##1} } } { \exp_not:n {##1} }
              }
          }
      }
    #1 #2 \l__clist_internal_remove_clist
  }
\cs_generate_variant:Nn \clist_remove_duplicates:N  { c }
\cs_generate_variant:Nn \clist_gremove_duplicates:N { c }
\cs_new_protected:Npn \clist_remove_all:Nn
  { \__clist_remove_all:NNNn \clist_set_from_seq:NN \__kernel_tl_set:Nx }
\cs_new_protected:Npn \clist_gremove_all:Nn
  { \__clist_remove_all:NNNn \clist_gset_from_seq:NN \__kernel_tl_gset:Nx }
\cs_new_protected:Npn \__clist_remove_all:NNNn #1#2#3#4
  {
    \__clist_if_wrap:nTF {#4}
      {
        \seq_set_from_clist:NN \l__clist_internal_remove_seq #3
        \seq_remove_all:Nn \l__clist_internal_remove_seq {#4}
        #1 #3 \l__clist_internal_remove_seq
      }
      {
        \cs_set:Npn \__clist_tmp:w ##1 , #4 ,
          {
            ##1
            , \s__clist_mark , \__clist_use_none_delimit_by_s_stop:w ,
            \__clist_remove_all:
          }
        #2 #3
          {
            \exp_after:wN \__clist_remove_all:
            #3 , \s__clist_mark , #4 , \s__clist_stop
          }
        \clist_if_empty:NF #3
          {
            #2 #3
              {
                \exp_args:No \exp_not:o
                  { \exp_after:wN \use_none:n #3 }
              }
          }
      }
  }
\cs_new:Npn \__clist_remove_all:
  { \exp_after:wN \__clist_remove_all:w \__clist_tmp:w , }
\cs_new:Npn \__clist_remove_all:w #1 , \s__clist_mark , #2 , { \exp_not:n {#1} }
\cs_generate_variant:Nn \clist_remove_all:Nn  { c , NV , cV , Ne , ce }
\cs_generate_variant:Nn \clist_gremove_all:Nn { c , NV , cV , Ne , ce }
\cs_new_protected:Npn \clist_reverse:N #1
  { \__kernel_tl_set:Nx #1 { \exp_args:No \clist_reverse:n {#1} } }
\cs_new_protected:Npn \clist_greverse:N #1
  { \__kernel_tl_gset:Nx #1 { \exp_args:No \clist_reverse:n {#1} } }
\cs_generate_variant:Nn \clist_reverse:N { c }
\cs_generate_variant:Nn \clist_greverse:N { c }
\cs_new:Npn \clist_reverse:n #1
  {
    \__clist_reverse:wwNww ? #1 ,
      \s__clist_mark \__clist_reverse:wwNww ! ,
      \s__clist_mark \__clist_reverse_end:ww
      \s__clist_stop ? \s__clist_mark
  }
```

```
\cs_new:Npn \__clist_reverse:wwNww
    #1 , #2 \s__clist_mark #3 #4 \s__clist_stop ? #5 \s__clist_mark
  { #3 ? #2 \s__clist_mark #3 #4 \s__clist_stop #1 , #5 \s__clist_mark }
\cs_new:Npn \__clist_reverse_end:ww #1 ! #2 , \s__clist_mark
  { \exp_not:o { \use_none:n #2 } }
\prg_new_eq_conditional:NNn \clist_if_empty:N \tl_if_empty:N
  { p , T , F , TF }
\prg_new_eq_conditional:NNn \clist_if_empty:c \tl_if_empty:c
  { p , T , F , TF }
\prg_new_conditional:Npnn \clist_if_empty:n #1 { p , T , F , TF }
  {
    \__clist_if_empty_n:w ? #1
    , \s__clist_mark \prg_return_false:
    , \s__clist_mark \prg_return_true:
    \s__clist_stop
  }
\cs_new:Npn \__clist_if_empty_n:w #1 ,
  {
    \tl_if_empty:oTF { \use_none:nn #1 ? }
      { \__clist_if_empty_n:w ? }
      { \__clist_if_empty_n:wNw }
  }
\cs_new:Npn \__clist_if_empty_n:wNw #1 \s__clist_mark #2#3 \s__clist_stop {#2}
\prg_new_protected_conditional:Npnn \clist_if_in:Nn #1#2 { T  , F , TF }
  {
    \exp_args:No \__clist_if_in_return:nnN #1 {#2} #1
  }
\prg_new_protected_conditional:Npnn \clist_if_in:nn #1#2 { T  , F , TF }
  {
    \clist_set:Nn \l__clist_internal_clist {#1}
    \exp_args:No \__clist_if_in_return:nnN \l__clist_internal_clist {#2}
      \l__clist_internal_clist
  }
\cs_new_protected:Npn \__clist_if_in_return:nnN #1#2#3
  {
    \__clist_if_wrap:nTF {#2}
      {
        \cs_set:Npe \__clist_tmp:w ##1
          {
            \exp_not:N \tl_if_eq:nnT {##1}
            \exp_not:n
              {
                {#2}
                { \clist_map_break:n { \prg_return_true: \use_none:n } }
              }
          }
        \clist_map_function:NN #3 \__clist_tmp:w
        \prg_return_false:
      }
      {
        \cs_set:Npn \__clist_tmp:w ##1 ,#2, { }
        \tl_if_empty:oTF
          { \__clist_tmp:w ,#1, {} {} ,#2, }
          { \prg_return_false: } { \prg_return_true: }
      }
  }
\prg_generate_conditional_variant:Nnn \clist_if_in:Nn
  { NV , No , c , cV , co } { T , F , TF }
\prg_generate_conditional_variant:Nnn \clist_if_in:nn
  { nV , no } { T , F , TF }
\cs_new:Npn \clist_map_function:NN #1#2
  {
    \clist_if_empty:NF #1
      {
        \exp_after:wN \__clist_map_function:Nw \exp_after:wN #2 #1 ,
          \s__clist_stop , \s__clist_stop , \s__clist_stop , \s__clist_stop ,
          \s__clist_stop , \s__clist_stop , \s__clist_stop , \s__clist_stop ,
        \prg_break_point:Nn \clist_map_break: { }
      }
  }
\cs_new:Npn \__clist_map_function:Nw #1 #2, #3, #4, #5, #6, #7, #8, #9,
  {
    \__clist_use_none_delimit_by_s_stop:w
      #9 \__clist_map_function_end:w \s__clist_stop
    #1 {#2} #1 {#3} #1 {#4} #1 {#5} #1 {#6} #1 {#7} #1 {#8} #1 {#9}
    \__clist_map_function:Nw #1
  }
\cs_new:Npn \__clist_map_function_end:w \s__clist_stop #1#2
  {
    \__clist_use_none_delimit_by_s_stop:w #2 \clist_map_break: \s__clist_stop
    #1 {#2}
    \__clist_map_function_end:w \s__clist_stop
  }
\cs_generate_variant:Nn \clist_map_function:NN { c }
\cs_new:Npn \clist_map_function:nN #1#2
  {
    \exp_after:wN \__clist_map_function_n:Nn \exp_after:wN #2
    \exp:w \__clist_trim_next:w \prg_do_nothing: #1 ,
      \s__clist_stop \clist_map_break: ,
    \prg_break_point:Nn \clist_map_break: { }
  }
\cs_generate_variant:Nn \clist_map_function:nN { e }
```

```
\cs_new:Npn \__clist_map_function_n:Nn #1 #2
  {
    \__clist_use_none_delimit_by_s_stop:w #2 \s__clist_stop
    \__clist_map_unbrace:wn #2 , #1
    \exp_after:wN \__clist_map_function_n:Nn \exp_after:wN #1
    \exp:w \__clist_trim_next:w \prg_do_nothing:
  }
\cs_new:Npn \__clist_map_unbrace:wn #1, #2 { #2 {#1} }
\cs_new_protected:Npn \clist_map_inline:Nn #1#2
  {
    \clist_if_empty:NF #1
      {
        \int_gincr:N \g__kernel_prg_map_int
        \cs_gset_protected:cpn
          { __clist_map_ \int_use:N \g__kernel_prg_map_int :w } ##1 {#2}
        \exp_last_unbraced:Nco \__clist_map_function:Nw
          { __clist_map_ \int_use:N \g__kernel_prg_map_int :w }
          #1 ,
          \s__clist_stop , \s__clist_stop , \s__clist_stop , \s__clist_stop ,
          \s__clist_stop , \s__clist_stop , \s__clist_stop , \s__clist_stop ,
        \prg_break_point:Nn \clist_map_break:
          { \int_gdecr:N \g__kernel_prg_map_int }
      }
  }
\cs_new_protected:Npn \clist_map_inline:nn #1
  {
    \clist_set:Nn \l__clist_internal_clist {#1}
    \clist_map_inline:Nn \l__clist_internal_clist
  }
\cs_generate_variant:Nn \clist_map_inline:Nn { c }
\cs_new_protected:Npn \clist_map_variable:NNn #1#2#3
  { \clist_map_tokens:Nn #1 { \__clist_map_variable:Nnn #2 {#3} } }
\cs_generate_variant:Nn \clist_map_variable:NNn { c }
\cs_new_protected:Npn \__clist_map_variable:Nnn #1#2#3
  { \tl_set:Nn #1 {#3} #2 }
\cs_new_protected:Npn \clist_map_variable:nNn #1
  {
    \clist_set:Nn \l__clist_internal_clist {#1}
    \clist_map_variable:NNn \l__clist_internal_clist
  }
\cs_new:Npn \clist_map_tokens:Nn #1#2
  {
    \clist_if_empty:NF #1
      {
        \exp_last_unbraced:Nno \__clist_map_tokens:nw {#2} #1 ,
          \s__clist_stop , \s__clist_stop , \s__clist_stop , \s__clist_stop ,
          \s__clist_stop , \s__clist_stop , \s__clist_stop , \s__clist_stop ,
        \prg_break_point:Nn \clist_map_break: { }
      }
  }
\cs_new:Npn \__clist_map_tokens:nw #1 #2, #3, #4, #5, #6, #7, #8, #9,
  {
    \__clist_use_none_delimit_by_s_stop:w
      #9 \__clist_map_tokens_end:w \s__clist_stop
    \use:n {#1} {#2} \use:n {#1} {#3} \use:n {#1} {#4} \use:n {#1} {#5}
    \use:n {#1} {#6} \use:n {#1} {#7} \use:n {#1} {#8} \use:n {#1} {#9}
    \__clist_map_tokens:nw {#1}
  }
\cs_new:Npn \__clist_map_tokens_end:w \s__clist_stop \use:n #1#2
  {
    \__clist_use_none_delimit_by_s_stop:w #2 \clist_map_break: \s__clist_stop
    #1 {#2}
    \__clist_map_tokens_end:w \s__clist_stop
  }
\cs_generate_variant:Nn \clist_map_tokens:Nn { c }
\cs_new:Npn \clist_map_tokens:nn #1#2
  {
    \__clist_map_tokens_n:nw {#2}
    \prg_do_nothing: #1 , \s__clist_stop \clist_map_break: ,
    \prg_break_point:Nn \clist_map_break: { }
  }
\cs_new:Npn \__clist_map_tokens_n:nw #1#2 ,
  {
    \tl_if_empty:oF { \use_none:nn #2 ? }
      {
        \__clist_use_none_delimit_by_s_stop:w #2 \s__clist_stop
        \tl_trim_spaces_apply:oN {#2} \use_ii_i:nn
        \__clist_map_unbrace:wn , {#1}
      }
    \__clist_map_tokens_n:nw {#1} \prg_do_nothing:
  }
\cs_new:Npn \clist_map_break:
  { \prg_map_break:Nn \clist_map_break: { } }
\cs_new:Npn \clist_map_break:n
  { \prg_map_break:Nn \clist_map_break: }
\cs_new:Npn \clist_count:N #1
  {
    \int_eval:n
      {
        0
        \clist_map_function:NN #1 \__clist_count:n
      }
  }
\cs_generate_variant:Nn \clist_count:N { c }
\cs_new:Npn \__clist_count:n #1 { + 1 }
\cs_set_protected:Npn \__clist_tmp:w #1
  {
    \cs_new:Npn \clist_count:n ##1
      {
        \int_eval:n
          {
            0
            \__clist_count:w #1
            ##1 , \s__clist_stop \prg_break: , \prg_break_point:
          }
      }
    \cs_new:Npn \__clist_count:w ##1 ,
      {
        \__clist_use_none_delimit_by_s_stop:w ##1 \s__clist_stop
        \tl_if_blank:nF {##1} { + 1 }
        \__clist_count:w #1
      }
  }
\exp_args:No \__clist_tmp:w \c_space_tl
\cs_generate_variant:Nn \clist_count:n { e }
\cs_new:Npn \clist_use:Nnnn #1#2#3#4
  {
    \clist_if_exist:NTF #1
      {
        \int_case:nnF { \clist_count:N #1 }
          {
            { 0 } { }
            { 1 } { \exp_after:wN \__clist_use:wwn #1 , , { } }
            { 2 } { \exp_after:wN \__clist_use:wwn #1 , {#2} }
          }
          {
            \exp_after:wN \__clist_use:nwwwwnwn
            \exp_after:wN { \exp_after:wN } #1 ,
            \s__clist_mark , { \__clist_use:nwwwwnwn {#3} }
            \s__clist_mark , { \__clist_use:nwwn {#4} }
            \s__clist_stop { }
          }
      }
      {
        \msg_expandable_error:nnn
          { kernel } { bad-variable } {#1}
      }
  }
```

```
\cs_generate_variant:Nn \clist_use:Nnnn { c }
\cs_new:Npn \__clist_use:wwn #1 , #2 , #3 { \exp_not:n { #1 #3 #2 } }
\cs_new:Npn \__clist_use:nwwwwnwn
    #1#2 , #3 , #4 , #5 \s__clist_mark , #6#7 \s__clist_stop #8
  { #6 {#3} , {#4} , #5 \s__clist_mark , {#6} #7 \s__clist_stop { #8 #1 #2 } }
\cs_new:Npn \__clist_use:nwwn #1#2 , #3 \s__clist_stop #4
  { \exp_not:n { #4 #1 #2 } }
\cs_new:Npn \clist_use:Nn #1#2
  { \clist_use:Nnnn #1 {#2} {#2} {#2} }
\cs_generate_variant:Nn \clist_use:Nn { c }
\cs_new_eq:NN \clist_use:N \tl_use:N
\cs_generate_variant:Nn \clist_use:N { c }
\cs_new:Npn \clist_use:nnnn #1#2#3#4
  {
    \__clist_use:Nw \__clist_use_none_delimit_by_s_stop:w
    \__clist_use:Nw \__clist_use_one:w
    \__clist_use:Nw \__clist_use_end:w
    \__clist_use_more:w ;
      {#2} {#3} {#4} ;
    \prg_do_nothing: #1 , \s__clist_mark ,
    \s__clist_stop
  }
\cs_new:Npn \__clist_use:Nw #1#2 ; #3 ; #4 ,
  {
    \tl_if_empty:oTF { \use_none:nn #4 ? }
      { \__clist_use:Nw #1#2 ; }
      {
        \__clist_use_none_delimit_by_s_mark:w #4 #1 \s__clist_mark
        \tl_trim_spaces_apply:oN {#4} \use_ii_i:nn
        \__clist_map_unbrace:wn , { #2 ; }
      }
    #3 ; \prg_do_nothing:
  }
\cs_new:Npn \__clist_use_one:w \s__clist_mark #1 , #2#3#4 \s__clist_stop
  { \exp_not:n {#3} }
\cs_new:Npn \__clist_use_end:w
    \s__clist_mark #1 , #2#3#4#5#6 \s__clist_stop
  { \exp_not:n { #4 #5 #3 } }
\cs_new:Npn \__clist_use_more:w ; #1#2#3#4#5#6 ;
  {
    \exp_not:n { #3 #5 }
    \__clist_use:Nw \__clist_use_end:w \__clist_use_more:w ;
    {#1} {#2} {#6} {#5} {#6} ;
  }
\cs_new:Npn \clist_use:nn #1#2 { \clist_use:nnnn {#1} {#2} {#2} {#2} }
\cs_new:Npn \clist_item:Nn #1#2
```

```
  {
    \__clist_item:ffoN
      { \clist_count:N #1 }
      { \int_eval:n {#2} }
      #1
      \__clist_item_N_loop:nw
  }
\cs_new:Npn \__clist_item:nnnN #1#2#3#4
  {
    \int_compare:nNnTF {#2} < 0
      {
        \int_compare:nNnTF {#2} < { - #1 }
          { \__clist_use_none_delimit_by_s_stop:w }
          { \exp_args:Nf #4 { \int_eval:n { #2 + 1 + #1 } } }
      }
      {
        \int_compare:nNnTF {#2} > {#1}
          { \__clist_use_none_delimit_by_s_stop:w }
          { #4 {#2} }
      }
    { } , #3 , \s__clist_stop
  }
\cs_generate_variant:Nn \__clist_item:nnnN { ffo, ff }
\cs_new:Npn \__clist_item_N_loop:nw #1 #2,
  {
    \int_compare:nNnTF {#1} = 0
      { \__clist_use_i_delimit_by_s_stop:nw { \exp_not:n {#2} } }
      { \exp_args:Nf \__clist_item_N_loop:nw { \int_eval:n { #1 - 1 } } }
  }
\cs_generate_variant:Nn \clist_item:Nn { c }
\cs_new:Npn \clist_item:nn #1#2
  {
    \__clist_item:ffnN
      { \clist_count:n {#1} }
      { \int_eval:n {#2} }
      {#1}
      \__clist_item_n:nw
  }
\cs_generate_variant:Nn \clist_item:nn { e }
\cs_new:Npn \__clist_item_n:nw #1
  { \__clist_item_n_loop:nw {#1} \prg_do_nothing: }
\cs_new:Npn \__clist_item_n_loop:nw #1 #2,
  {
    \exp_args:No \tl_if_blank:nTF {#2}
      { \__clist_item_n_loop:nw {#1} \prg_do_nothing: }
      {
        \int_compare:nNnTF {#1} = 0
          { \exp_args:No \__clist_item_n_end:n {#2} }
          {
            \exp_args:Nf \__clist_item_n_loop:nw
              { \int_eval:n { #1 - 1 } }
              \prg_do_nothing:
          }
      }
  }
\cs_new:Npn \__clist_item_n_end:n #1 #2 \s__clist_stop
  { \tl_trim_spaces_apply:nN {#1} \__clist_item_n_strip:n }
\cs_new:Npn \__clist_item_n_strip:n #1 { \__clist_item_n_strip:w #1 , }
\cs_new:Npn \__clist_item_n_strip:w #1 , { \exp_not:n {#1} }
\cs_new:Npn \clist_rand_item:n #1
  { \exp_args:Nf \__clist_rand_item:nn { \clist_count:n {#1} } {#1} }
\cs_new:Npn \__clist_rand_item:nn #1#2
  {
    \int_compare:nNnF {#1} = 0
      { \clist_item:nn {#2} { \int_rand:nn { 1 } {#1} } }
  }
\cs_new:Npn \clist_rand_item:N #1
  {
    \clist_if_empty:NF #1
      { \clist_item:Nn #1 { \int_rand:nn { 1 } { \clist_count:N #1 } } }
  }
\cs_generate_variant:Nn \clist_rand_item:N { c }
\cs_new_protected:Npn \clist_show:N { \__clist_show:NN \msg_show:nneeee }
\cs_generate_variant:Nn \clist_show:N { c }
\cs_new_protected:Npn \clist_log:N { \__clist_show:NN \msg_log:nneeee }
\cs_generate_variant:Nn \clist_log:N { c }
\cs_new_protected:Npn \__clist_show:NN #1#2
  {
    \__kernel_chk_tl_type:NnnT #2 { clist } { \exp_not:o #2 }
      {
        \int_compare:nNnTF { \clist_count:N #2 }
          = { \exp_args:No \clist_count:n #2 }
          {
            #1 { clist } { show }
              { \token_to_str:N #2 }
              { \clist_map_function:NN #2 \msg_show_item:n }
              { } { }
          }
          {
            \msg_error:nnee { clist } { non-clist }
              { \token_to_str:N #2 } { \tl_to_str:N #2 }
          }
      }
  }
\cs_new_protected:Npn \clist_show:n { \__clist_show:Nn \msg_show:nneeee }
\cs_new_protected:Npn \clist_log:n { \__clist_show:Nn \msg_log:nneeee }
\cs_new_protected:Npn \__clist_show:Nn #1#2
  {
    #1 { clist } { show }
      { } { \clist_map_function:nN {#2} \msg_show_item:n } { } { }
  }
\clist_new:N \l_tmpa_clist
\clist_new:N \l_tmpb_clist
\clist_new:N \g_tmpa_clist
\clist_new:N \g_tmpb_clist
%% File: l3token.dtx
\scan_new:N \s__char_stop
\quark_new:N \q__char_no_value
\__kernel_quark_new_conditional:Nn \__char_quark_if_no_value:N { TF }
\cs_new_protected:Npn \char_set_catcode:nn #1#2
  { \tex_catcode:D \int_eval:n {#1} = \int_eval:n {#2} \exp_stop_f: }
\cs_new:Npn \char_value_catcode:n #1
  { \tex_the:D \tex_catcode:D \int_eval:n {#1} \exp_stop_f: }
\cs_new_protected:Npn \char_show_value_catcode:n #1
  { \exp_args:Nf \tl_show:n { \char_value_catcode:n {#1} } }
\cs_new_protected:Npn \char_set_catcode_escape:N #1
  { \char_set_catcode:nn { `#1 } { 0 } }
\cs_new_protected:Npn \char_set_catcode_group_begin:N #1
  { \char_set_catcode:nn { `#1 } { 1 } }
\cs_new_protected:Npn \char_set_catcode_group_end:N #1
  { \char_set_catcode:nn { `#1 } { 2 } }
\cs_new_protected:Npn \char_set_catcode_math_toggle:N #1
  { \char_set_catcode:nn { `#1 } { 3 } }
\cs_new_protected:Npn \char_set_catcode_alignment:N #1
  { \char_set_catcode:nn { `#1 } { 4 } }
\cs_new_protected:Npn \char_set_catcode_end_line:N #1
  { \char_set_catcode:nn { `#1 } { 5 } }
\cs_new_protected:Npn \char_set_catcode_parameter:N #1
  { \char_set_catcode:nn { `#1 } { 6 } }
\cs_new_protected:Npn \char_set_catcode_math_superscript:N #1
  { \char_set_catcode:nn { `#1 } { 7 } }
\cs_new_protected:Npn \char_set_catcode_math_subscript:N #1
  { \char_set_catcode:nn { `#1 } { 8 } }
\cs_new_protected:Npn \char_set_catcode_ignore:N #1
  { \char_set_catcode:nn { `#1 } { 9 } }
\cs_new_protected:Npn \char_set_catcode_space:N #1
  { \char_set_catcode:nn { `#1 } { 10 } }
\cs_new_protected:Npn \char_set_catcode_letter:N #1
```

```
  { \char_set_catcode:nn { `#1 } { 11 } }
\cs_new_protected:Npn \char_set_catcode_other:N #1
  { \char_set_catcode:nn { `#1 } { 12 } }
\cs_new_protected:Npn \char_set_catcode_active:N #1
  { \char_set_catcode:nn { `#1 } { 13 } }
\cs_new_protected:Npn \char_set_catcode_comment:N #1
  { \char_set_catcode:nn { `#1 } { 14 } }
\cs_new_protected:Npn \char_set_catcode_invalid:N #1
  { \char_set_catcode:nn { `#1 } { 15 } }
\cs_new_protected:Npn \char_set_catcode_escape:n #1
  { \char_set_catcode:nn {#1} { 0 } }
\cs_new_protected:Npn \char_set_catcode_group_begin:n #1
  { \char_set_catcode:nn {#1} { 1 } }
\cs_new_protected:Npn \char_set_catcode_group_end:n #1
  { \char_set_catcode:nn {#1} { 2 } }
\cs_new_protected:Npn \char_set_catcode_math_toggle:n #1
  { \char_set_catcode:nn {#1} { 3 } }
\cs_new_protected:Npn \char_set_catcode_alignment:n #1
  { \char_set_catcode:nn {#1} { 4 } }
\cs_new_protected:Npn \char_set_catcode_end_line:n #1
  { \char_set_catcode:nn {#1} { 5 } }
\cs_new_protected:Npn \char_set_catcode_parameter:n #1
  { \char_set_catcode:nn {#1} { 6 } }
\cs_new_protected:Npn \char_set_catcode_math_superscript:n #1
  { \char_set_catcode:nn {#1} { 7 } }
\cs_new_protected:Npn \char_set_catcode_math_subscript:n #1
  { \char_set_catcode:nn {#1} { 8 } }
\cs_new_protected:Npn \char_set_catcode_ignore:n #1
  { \char_set_catcode:nn {#1} { 9 } }
\cs_new_protected:Npn \char_set_catcode_space:n #1
  { \char_set_catcode:nn {#1} { 10 } }
\cs_new_protected:Npn \char_set_catcode_letter:n #1
  { \char_set_catcode:nn {#1} { 11 } }
\cs_new_protected:Npn \char_set_catcode_other:n #1
  { \char_set_catcode:nn {#1} { 12 } }
\cs_new_protected:Npn \char_set_catcode_active:n #1
  { \char_set_catcode:nn {#1} { 13 } }
\cs_new_protected:Npn \char_set_catcode_comment:n #1
  { \char_set_catcode:nn {#1} { 14 } }
\cs_new_protected:Npn \char_set_catcode_invalid:n #1
  { \char_set_catcode:nn {#1} { 15 } }
\cs_new_protected:Npn \char_set_mathcode:nn #1#2
  { \tex_mathcode:D \int_eval:n {#1} = \int_eval:n {#2} \exp_stop_f: }
\cs_new:Npn \char_value_mathcode:n #1
  { \tex_the:D \tex_mathcode:D \int_eval:n {#1} \exp_stop_f: }
\cs_new_protected:Npn \char_show_value_mathcode:n #1
```

```
  { \exp_args:Nf \tl_show:n { \char_value_mathcode:n {#1} } }
\cs_new_protected:Npn \char_set_lccode:nn #1#2
  { \tex_lccode:D \int_eval:n {#1} = \int_eval:n {#2} \exp_stop_f: }
\cs_new:Npn \char_value_lccode:n #1
  { \tex_the:D \tex_lccode:D \int_eval:n {#1} \exp_stop_f: }
\cs_new_protected:Npn \char_show_value_lccode:n #1
  { \exp_args:Nf \tl_show:n { \char_value_lccode:n {#1} } }
\cs_new_protected:Npn \char_set_uccode:nn #1#2
  { \tex_uccode:D \int_eval:n {#1} = \int_eval:n {#2} \exp_stop_f: }
\cs_new:Npn \char_value_uccode:n #1
  { \tex_the:D \tex_uccode:D \int_eval:n {#1} \exp_stop_f: }
\cs_new_protected:Npn \char_show_value_uccode:n #1
  { \exp_args:Nf \tl_show:n { \char_value_uccode:n {#1} } }
\cs_new_protected:Npn \char_set_sfcode:nn #1#2
  { \tex_sfcode:D \int_eval:n {#1} = \int_eval:n {#2} \exp_stop_f: }
\cs_new:Npn \char_value_sfcode:n #1
  { \tex_the:D \tex_sfcode:D \int_eval:n {#1} \exp_stop_f: }
\cs_new_protected:Npn \char_show_value_sfcode:n #1
  { \exp_args:Nf \tl_show:n { \char_value_sfcode:n {#1} } }
\seq_new:N \l_char_special_seq
\seq_set_split:Nnn \l_char_special_seq { }
  { \␣ \" \# \$ \% \& \\ \^ \_ \{ \} \~ }
\seq_new:N \l_char_active_seq
\seq_set_split:Nnn \l_char_active_seq { }
  { \" \$ \& \^ \_ \~ }
\group_begin:
  \char_set_catcode_active:N \^^@
  \cs_set_protected:Npn \__char_tmp:nN #1#2
    {
      \cs_new_protected:cpn { #1 :nN } ##1
        {
          \group_begin:
            \char_set_lccode:nn { `\^^@ } { ##1 }
          \tex_lowercase:D { \group_end: #2 ^^@ }
        }
      \cs_new_protected:cpe { #1 :NN } ##1
        { \exp_not:c { #1 : nN } { `##1 } }
    }
  \__char_tmp:nN { char_set_active_eq }  \cs_set_eq:NN
  \__char_tmp:nN { char_gset_active_eq } \cs_gset_eq:NN
\group_end:
\cs_generate_variant:Nn \char_set_active_eq:NN  { Nc }
\cs_generate_variant:Nn \char_gset_active_eq:NN { Nc }
\cs_generate_variant:Nn \char_set_active_eq:nN  { nc }
\cs_generate_variant:Nn \char_gset_active_eq:nN { nc }
\cs_new_eq:NN \__char_int_to_roman:w \tex_romannumeral:D
```

```
\cs_new_eq:NN \__char_sep: \__kernel_int_sep:
\cs_new:Npn \char_generate:nn #1#2
  {
    \exp:w \exp_after:wN \__char_generate_aux:w
      \int_value:w \int_eval:n {#1} \exp_after:wN \__char_sep:
      \int_value:w \int_eval:n {#2} \__char_sep:
  }
\cs_new:Npn \__char_generate_aux:w #1 \__char_sep: #2 \__char_sep:
  {
    \if_int_odd:w 0
        \if_int_compare:w #2 < 1  \exp_stop_f: 1 \fi:
        \if_int_compare:w #2 = 5  \exp_stop_f: 1 \fi:
        \if_int_compare:w #2 = 9  \exp_stop_f: 1 \fi:
        \if_int_compare:w #2 > 13 \exp_stop_f: 1 \fi: \exp_stop_f:
      \msg_expandable_error:nn { char }
        { invalid-catcode }
    \else:
      \if_int_odd:w 0
        \if_int_compare:w #1 < \c_zero_int 1 \fi:
        \if_int_compare:w #1 > \c_max_char_int 1 \fi: \exp_stop_f:
        \msg_expandable_error:nn { char }
          { out-of-range }
      \else:
        \if_int_compare:w #2#1 = 100 \exp_stop_f:
          \msg_expandable_error:nn { char } { null-space }
        \else:
          \__char_generate_aux:nnw {#1} {#2}
        \fi:
      \fi:
    \fi:
    \exp_end:
  }
\tl_new:N \l__char_tmp_tl
\group_begin:
  \char_set_catcode_active:N \^^L
  \cs_set:Npn ^^L { }
  \if_cs_exist:N \tex_Ucharcat:D
      \cs_new:Npn \__char_generate_aux:nnw #1#2#3 \exp_end:
        {
          #3
          \exp_after:wN \exp_end:
          \tex_Ucharcat:D #1 \exp_stop_f: #2 \exp_stop_f:
        }
  \else:
    \char_set_catcode_active:n { 0 }
    \tl_set:Nn \l__char_tmp_tl { \exp_not:N ^^@ \exp_not:N \or: }
    \char_set_catcode_other:n { 0 }
    \tl_put_right:Nn \l__char_tmp_tl { ^^@ \exp_not:N \or: }
    \char_set_catcode_letter:n { 0 }
    \tl_put_right:Nn \l__char_tmp_tl { ^^@ \exp_not:N \or: }
    \tl_put_right:Nn \l__char_tmp_tl { \use:n { ~ } \exp_not:N \or: }
    \tl_put_right:Nn \l__char_tmp_tl { \exp_not:N \or: }
    \char_set_catcode_math_subscript:n { 0 }
    \tl_put_right:Nn \l__char_tmp_tl { ^^@ \exp_not:N \or: }
    \char_set_catcode_math_superscript:n { 0 }
    \tl_put_right:Nn \l__char_tmp_tl { ^^@ \exp_not:N \or: }
    \char_set_catcode_parameter:n { 0 }
    \tl_put_right:Nn \l__char_tmp_tl { ^^@ \exp_not:N \or: }
    \tl_put_right:Nn \l__char_tmp_tl { { \if_false: } \fi: \exp_not:N \or: }
    \char_set_catcode_alignment:n { 0 }
    \tl_put_right:Nn \l__char_tmp_tl { ^^@ \exp_not:N \or: }
    \char_set_catcode_math_toggle:n { 0 }
    \tl_put_right:Nn \l__char_tmp_tl { ^^@ \exp_not:N \or: }
    \char_set_catcode_group_end:n { 0 }
    \tl_put_right:Nn \l__char_tmp_tl { \if_false: { \fi: ^^@ \exp_not:N \or: } % }
    \char_set_catcode_group_begin:n { 0 } % {
    \tl_put_right:Nn \l__char_tmp_tl { ^^@ \exp_not:N \or: } }
     \cs_set_protected:Npn \__char_tmp:n #1
      {
        \char_set_lccode:nn { 0 } {#1}
        \char_set_lccode:nn { 32 } {#1}
        \exp_args:Ne \tex_lowercase:D
          {
            \tl_const:Ne
              \exp_not:c { c__char_ \__char_int_to_roman:w #1 _tl }
              { \exp_not:o \l__char_tmp_tl }
          }
      }
     \int_step_function:nnN { 0 } { 255 }  \__char_tmp:n
     \cs_new:Npn \__char_generate_aux:nnw #1#2#3 \exp_end:
      {
        #3
        \if_false: { \fi:
        \exp_after:wN \exp_after:wN \exp_after:wN \exp_end:
        \exp_after:wN \exp_after:wN
        \if_case:w \tex_numexpr:D 13 - #2
          \exp_after:wN \exp_after:wN \exp_after:wN \exp_after:wN
          \exp_after:wN \exp_after:wN \exp_after:wN \scan_stop:
          \exp_after:wN \exp_after:wN \exp_after:wN \exp_not:N
            \cs:w c__char_ \__char_int_to_roman:w #1 _tl \cs_end:
        }
        \fi:
        }
  \fi:
\group_end:
\group_begin:
  \char_set_catcode_active:N *
  \char_set_lccode:nn { `* } { `\␣}
  \tex_lowercase:D { \tl_const:Nn \c_catcode_active_space_tl { * } }
\group_end:
\tl_const:Ne \c_catcode_other_space_tl { \char_generate:nn { `\␣ } { 12 } }
\scan_new:N \s__token_mark
\scan_new:N \s__token_stop
\cs_new:Npn \token_to_catcode:N
  { \int_value:w \group_align_safe_begin: \__token_to_catcode:N }
\cs_new:Npn \__token_to_catcode:N #1
  {
    \if_catcode:w \exp_not:N #1 \c_catcode_letter_token
      11
    \else:
      \if_catcode:w \exp_not:N #1 \c_catcode_other_token
        12
      \else:
        \if_catcode:w \exp_not:N #1 \c_math_toggle_token
          3
        \else:
          \if_catcode:w \exp_not:N #1 \c_alignment_token
            4
          \else:
            \if_catcode:w \exp_not:N #1 ##
              6
            \else:
              \if_catcode:w \exp_not:N #1 \c_math_superscript_token
                7
              \else:
                \if_catcode:w \exp_not:N #1 \c_math_subscript_token
                  8
                \else:
                  \if_catcode:w \exp_not:N #1 \c_group_begin_token
                    1
                  \else:
                    \if_catcode:w \exp_not:N #1 \c_group_end_token
                      2
                    \else:
                      \if_catcode:w \exp_not:N #1 \c_space_token
                        10
                      \else:
                        \token_if_cs:NTF #1 { 16 } { 13 }
                      \fi:
                    \fi:
                  \fi:
                \fi:
              \fi:
            \fi:
          \fi:
        \fi:
      \fi:
    \fi:
    \group_align_safe_end:
    \exp_stop_f:
  }
\group_begin:
  \__kernel_chk_if_free_cs:N \c_group_begin_token
  \tex_global:D \tex_let:D \c_group_begin_token {
  \__kernel_chk_if_free_cs:N \c_group_end_token
  \tex_global:D \tex_let:D \c_group_end_token }
  \char_set_catcode_math_toggle:N \*
  \cs_new_eq:NN \c_math_toggle_token *
  \char_set_catcode_alignment:N \*
  \cs_new_eq:NN \c_alignment_token *
  \cs_new_eq:NN \c_parameter_token #
  \cs_new_eq:NN \c_math_superscript_token ^
  \char_set_catcode_math_subscript:N \*
  \cs_new_eq:NN \c_math_subscript_token *
  \__kernel_chk_if_free_cs:N \c_space_token
  \use:n { \tex_global:D \tex_let:D \c_space_token = ~ } ~
  \cs_new_eq:NN \c_catcode_letter_token a
  \cs_new_eq:NN \c_catcode_other_token 1
\group_end:
\group_begin:
  \char_set_catcode_active:N \*
  \tl_const:Nn \c__token_active_tl { \exp_not:N * }
\group_end:
\prg_new_conditional:Npnn \token_if_group_begin:N #1 { p , T ,  F , TF }
  {
    \if_catcode:w \exp_not:N #1 \c_group_begin_token
      \prg_return_true: \else: \prg_return_false: \fi:
  }
\prg_new_conditional:Npnn \token_if_group_end:N #1 { p , T ,  F , TF }
  {
    \if_catcode:w \exp_not:N #1 \c_group_end_token
      \prg_return_true: \else: \prg_return_false: \fi:
  }
\prg_new_conditional:Npnn \token_if_math_toggle:N #1 { p , T ,  F , TF }
```

```
  {
    \if_catcode:w \exp_not:N #1 \c_math_toggle_token
      \prg_return_true: \else: \prg_return_false: \fi:
  }
\prg_new_conditional:Npnn \token_if_alignment:N #1 { p , T ,  F , TF }
  {
    \if_catcode:w \exp_not:N #1 \c_alignment_token
      \prg_return_true: \else: \prg_return_false: \fi:
  }
\group_begin:
\cs_set_eq:NN \c_parameter_token \scan_stop:
\prg_new_conditional:Npnn \token_if_parameter:N #1 { p , T ,  F , TF }
  {
    \if_catcode:w \exp_not:N #1 \c_parameter_token
      \prg_return_true: \else: \prg_return_false: \fi:
  }
\group_end:
\prg_new_conditional:Npnn \token_if_math_superscript:N #1
  { p , T ,  F , TF }
  {
    \if_catcode:w \exp_not:N #1 \c_math_superscript_token
      \prg_return_true: \else: \prg_return_false: \fi:
  }
\prg_new_conditional:Npnn \token_if_math_subscript:N #1 { p , T ,  F , TF }
  {
    \if_catcode:w \exp_not:N #1 \c_math_subscript_token
      \prg_return_true: \else: \prg_return_false: \fi:
  }
\prg_new_conditional:Npnn \token_if_space:N #1 { p , T ,  F , TF }
  {
    \if_catcode:w \exp_not:N #1 \c_space_token
      \prg_return_true: \else: \prg_return_false: \fi:
  }
\prg_new_conditional:Npnn \token_if_letter:N #1 { p , T ,  F , TF }
  {
    \if_catcode:w \exp_not:N #1 \c_catcode_letter_token
      \prg_return_true: \else: \prg_return_false: \fi:
  }
\prg_new_conditional:Npnn \token_if_other:N #1 { p , T ,  F , TF }
  {
    \if_catcode:w \exp_not:N #1 \c_catcode_other_token
      \prg_return_true: \else: \prg_return_false: \fi:
  }
\prg_new_conditional:Npnn \token_if_active:N #1 { p , T ,  F , TF }
  {
    \if_catcode:w \exp_not:N #1 \c__token_active_tl
```

```
      \prg_return_true: \else: \prg_return_false: \fi:
  }
\prg_new_eq_conditional:NNn \token_if_eq_meaning:NN \cs_if_eq:NN
  { p , T , F , TF }
\prg_new_conditional:Npnn \token_if_eq_catcode:NN #1#2 { p , T ,  F , TF }
  {
    \if_catcode:w \exp_not:N #1 \exp_not:N #2
      \prg_return_true: \else: \prg_return_false: \fi:
  }
\prg_new_conditional:Npnn \token_if_eq_charcode:NN #1#2 { p , T ,  F , TF }
  {
    \if_charcode:w \exp_not:N #1 \exp_not:N #2
      \prg_return_true: \else: \prg_return_false: \fi:
  }
\use:e
  {
    \prg_new_conditional:Npnn \exp_not:N \token_if_macro:N #1
      { p , T ,  F , TF }
      {
        \exp_not:N \exp_after:wN \exp_not:N \__token_if_macro_p:w
        \exp_not:N \token_to_meaning:N #1 \tl_to_str:n { ma : }
          \s__token_stop
      }
    \cs_new:Npn \exp_not:N  \__token_if_macro_p:w
      #1 \tl_to_str:n { ma } #2 \c_colon_str #3 \s__token_stop
  }
      {
        \str_if_eq:nnTF { #2 } { cro }
          { \prg_return_true: }
          { \prg_return_false: }
      }
\prg_new_conditional:Npnn \token_if_cs:N #1 { p , T ,  F , TF }
  {
    \if_catcode:w \exp_not:N #1 \scan_stop:
      \prg_return_true: \else: \prg_return_false: \fi:
  }
\prg_new_conditional:Npnn \token_if_expandable:N #1 { p , T ,  F , TF }
  {
    \exp_after:wN \if_meaning:w \exp_not:N #1 #1
      \prg_return_false:
    \else:
      \if_cs_exist:N #1
        \prg_return_true:
      \else:
        \prg_return_false:
      \fi:
```

```
    \fi:
  }
\group_begin:
\cs_set_protected:Npn \__token_tmp:w #1
  {
    \use:e
      {
        \cs_new:Npn \exp_not:c { __token_delimit_by_ #1 :w }
            ##1 \tl_to_str:n {#1} ##2 \s__token_stop
          { ##1 \tl_to_str:n {#1} }
      }
  }
\__token_tmp:w { char" }
\__token_tmp:w { count }
\__token_tmp:w { dimen }
\__token_tmp:w { ~ font }
\__token_tmp:w { macro }
\__token_tmp:w { muskip }
\__token_tmp:w { skip }
\__token_tmp:w { toks }
\group_end:
\group_begin:
\cs_set_protected:Npn \__token_tmp:w #1#2#3
  {
    \use:e
      {
        \prg_new_conditional:Npnn \exp_not:c { token_if_ #1 :N } ##1
          { p , T ,  F , TF }
          {
            \cs_if_exist:cT { tex_ #2 :D }
              {
                \exp_not:N \if_meaning:w ##1 \exp_not:c { tex_ #2 :D }
                \exp_not:N \prg_return_false:
                \exp_not:N \else:
                \exp_not:N \if_meaning:w ##1 \exp_not:c { tex_ #2 def:D }
                \exp_not:N \prg_return_false:
                \exp_not:N \else:
              }
            \exp_not:N \str_if_eq:eeTF
              {
                \exp_not:N \exp_after:wN
                \exp_not:c { __token_delimit_by_ #2 :w }
                \exp_not:N \token_to_meaning:N ##1
                ? \tl_to_str:n {#2} \s__token_stop
              }
              { \exp_not:n {#3} }
              { \exp_not:N \prg_return_true: }
              { \exp_not:N \prg_return_false: }
            \cs_if_exist:cT { tex_ #2 :D }
              {
                \exp_not:N \fi:
                \exp_not:N \fi:
              }
          }
      }
  }
\__token_tmp:w { chardef } { char" } { \token_to_str:N \char" }
\__token_tmp:w { mathchardef } { char" } { \token_to_str:N \mathchar" }
\__token_tmp:w { long_macro } { macro } { \tl_to_str:n { \long } macro }
\__token_tmp:w { protected_macro } { macro }
  { \tl_to_str:n { \protected } macro }
\__token_tmp:w { protected_long_macro } { macro }
  { \token_to_str:N \protected \tl_to_str:n { \long } macro }
\__token_tmp:w { font_selection } { ~ font } { select ~ font }
\__token_tmp:w { dim_register } { dimen } { \token_to_str:N \dimen }
\__token_tmp:w { int_register } { count } { \token_to_str:N \count }
\__token_tmp:w { muskip_register } { muskip } { \token_to_str:N \muskip }
\__token_tmp:w { skip_register } { skip } { \token_to_str:N \skip }
\__token_tmp:w { toks_register } { toks } { \token_to_str:N \toks }
\group_end:
\sys_if_engine_luatex:TF
  {
    \prg_new_conditional:Npnn \token_if_primitive:N #1 { p , T , F , TF }
      {
        \__token_if_primitive_lua:N #1
      }
  }
  {
    \tex_global:D \tex_chardef:D \c__token_A_int = `A ~ %
    \use:e
      {
        \prg_new_conditional:Npnn \exp_not:N \token_if_primitive:N #1
          { p , T , F , TF }
          {
            \exp_not:N \token_if_macro:NTF #1
              \exp_not:N \prg_return_false:
              {
                \exp_not:N \exp_after:wN \exp_not:N \__token_if_primitive:NNw
                \exp_not:N \token_to_meaning:N #1
                  \tl_to_str:n { : : : } \s__token_stop #1
              }
          }
        \cs_new:Npn \exp_not:N \__token_if_primitive:NNw
          #1#2 #3 \c_colon_str #4 \s__token_stop
          {
            \exp_not:N \tl_if_empty:oTF
              { \exp_not:N \__token_if_primitive_space:w #3 ~ }
              {
                \exp_not:N \__token_if_primitive_loop:N #3
                  \c_colon_str \s__token_stop
              }
              { \exp_not:N \__token_if_primitive_nullfont:N }
          }
      }
    \cs_new:Npn \__token_if_primitive_space:w #1 ~ { }
    \cs_new:Npn \__token_if_primitive_nullfont:N #1
      {
        \if_meaning:w \tex_nullfont:D #1
          \prg_return_true:
        \else:
          \prg_return_false:
        \fi:
      }
    \cs_new:Npn \__token_if_primitive_loop:N #1
      {
        \if_int_compare:w `#1 < \c__token_A_int %
          \exp_after:wN \__token_if_primitive:Nw
          \exp_after:wN #1
        \else:
          \exp_after:wN \__token_if_primitive_loop:N
        \fi:
      }
    \cs_new:Npn \__token_if_primitive:Nw #1 #2 \s__token_stop
      {
        \if:w : #1
          \exp_after:wN \__token_if_primitive_undefined:N
        \else:
          \prg_return_false:
          \exp_after:wN \use_none:n
        \fi:
      }
    \cs_new:Npn \__token_if_primitive_undefined:N #1
      {
        \if_cs_exist:N #1
          \prg_return_true:
        \else:
          \prg_return_false:
        \fi:
      }
  }
\cs_new:Npn \token_case_catcode:Nn #1#2
  { \exp:w \__token_case:NNnTF \token_if_eq_catcode:NNTF #1 {#2} { } { } }
\cs_new:Npn \token_case_catcode:NnT #1#2#3
  { \exp:w \__token_case:NNnTF \token_if_eq_catcode:NNTF #1 {#2} {#3} { } }
\cs_new:Npn \token_case_catcode:NnF #1#2
  { \exp:w \__token_case:NNnTF \token_if_eq_catcode:NNTF #1 {#2} { } }
\cs_new:Npn \token_case_catcode:NnTF
  { \exp:w \__token_case:NNnTF \token_if_eq_catcode:NNTF }
\cs_new:Npn \token_case_charcode:Nn #1#2
  { \exp:w \__token_case:NNnTF \token_if_eq_charcode:NNTF #1 {#2} { } { } }
\cs_new:Npn \token_case_charcode:NnT #1#2#3
  { \exp:w \__token_case:NNnTF \token_if_eq_charcode:NNTF #1 {#2} {#3} { } }
\cs_new:Npn \token_case_charcode:NnF #1#2
  { \exp:w \__token_case:NNnTF \token_if_eq_charcode:NNTF #1 {#2} { } }
\cs_new:Npn \token_case_charcode:NnTF
  { \exp:w \__token_case:NNnTF \token_if_eq_charcode:NNTF }
\cs_new:Npn \token_case_meaning:Nn #1#2
  { \exp:w \__token_case:NNnTF \token_if_eq_meaning:NNTF #1 {#2} { } { } }
\cs_new:Npn \token_case_meaning:NnT #1#2#3
  { \exp:w \__token_case:NNnTF \token_if_eq_meaning:NNTF #1 {#2} {#3} { } }
\cs_new:Npn \token_case_meaning:NnF #1#2
  { \exp:w \__token_case:NNnTF \token_if_eq_meaning:NNTF #1 {#2} { } }
\cs_new:Npn \token_case_meaning:NnTF
  { \exp:w \__token_case:NNnTF \token_if_eq_meaning:NNTF }
\cs_new:Npn \__token_case:NNnTF #1#2#3#4#5
  {
    \__token_case:NNw #1 #2 #3 #2 { }
    \s__token_mark {#4}
    \s__token_mark {#5}
    \s__token_stop
  }
\cs_new:Npn \__token_case:NNw #1#2#3#4
  {
    #1 #2 #3
      { \__token_case_end:nw {#4} }
      { \__token_case:NNw #1 #2 }
  }
\cs_new:Npn \__token_case_end:nw #1#2#3 \s__token_mark #4#5 \s__token_stop
  { \exp_end: #1 #4 }
\cs_new_eq:NN \l_peek_token ?
\cs_new_eq:NN \g_peek_token ?
\cs_new_eq:NN \l__peek_search_token ?
\tl_new:N \l__peek_search_tl
\cs_new:Npn \__peek_true:w  { }
```

```
\cs_new:Npn \__peek_true_aux:w  { }
\cs_new:Npn \__peek_false:w { }
\cs_new:Npn \__peek_tmp:w { }
\scan_new:N \s__peek_mark
\scan_new:N \s__peek_stop
\cs_new:Npn \__peek_use_none_delimit_by_s_stop:w #1 \s__peek_stop { }
\cs_new_protected:Npn \peek_after:Nw
  { \tex_futurelet:D \l_peek_token }
\cs_new_protected:Npn \peek_gafter:Nw
  { \tex_global:D \tex_futurelet:D \g_peek_token }
\cs_new_protected:Npn \__peek_true_remove:w
  {
    \tex_afterassignment:D \__peek_true_aux:w
    \cs_set_eq:NN \__peek_tmp:w
  }
\cs_new_protected:Npn \peek_remove_spaces:n #1
  {
    \cs_set:Npe \__peek_false:w { \exp_not:n {#1} }
    \group_align_safe_begin:
    \cs_set:Npn \__peek_true_aux:w { \peek_after:Nw \__peek_remove_spaces: }
    \__peek_true_aux:w
  }
\cs_new_protected:Npn \__peek_remove_spaces:
  {
    \if_meaning:w \l_peek_token \c_space_token
      \exp_after:wN \__peek_true_remove:w
    \else:
      \group_align_safe_end:
      \exp_after:wN \__peek_false:w
    \fi:
  }
\cs_new_protected:Npn \peek_remove_filler:n #1
  {
    \cs_set:Npn \__peek_true_aux:w { \__peek_remove_filler:w }
    \cs_set:Npe \__peek_false:w
      {
        \exp_not:N \group_align_safe_end:
        \exp_not:n {#1}
      }
    \group_align_safe_begin:
    \__peek_remove_filler:w
  }
\cs_new_protected:Npn \__peek_remove_filler:w
  {
    \exp_after:wN \peek_after:Nw \exp_after:wN \__peek_remove_filler:
    \exp:w \exp_end_continue_f:w
  }
\cs_new_protected:Npn \__peek_remove_filler:
  {
    \if_catcode:w \exp_not:N \l_peek_token \c_space_token
      \exp_after:wN \__peek_true_remove:w
    \else:
      \if_meaning:w \l_peek_token \scan_stop:
        \exp_after:wN \exp_after:wN \exp_after:wN
          \__peek_true_remove:w
      \else:
        \exp_after:wN \exp_after:wN \exp_after:wN
          \__peek_remove_filler_expand:w
      \fi:
    \fi:
  }
\cs_new_protected:Npn \__peek_remove_filler_expand:w
  {
    \exp_after:wN \if_meaning:w \exp_not:N \l_peek_token \l_peek_token
      \exp_after:wN \__peek_false:w
    \else:
      \exp_after:wN \__peek_remove_filler:w
    \fi:
  }
\cs_new_protected:Npn \__peek_token_generic_aux:NNNTF #1#2#3#4#5
  {
    \group_align_safe_begin:
    \cs_set_eq:NN \l__peek_search_token #3
    \tl_set:Nn \l__peek_search_tl {#3}
    \cs_set:Npe \__peek_true_aux:w
      {
        \exp_not:N \group_align_safe_end:
        \exp_not:n {#4}
      }
    \cs_set_eq:NN \__peek_true:w #1
    \cs_set:Npe \__peek_false:w
      {
        \exp_not:N \group_align_safe_end:
        \exp_not:n {#5}
      }
    \peek_after:Nw #2
  }
\cs_new_protected:Npn \__peek_token_generic:NNTF
  { \__peek_token_generic_aux:NNNTF \__peek_true_aux:w }
\cs_new_protected:Npn \__peek_token_generic:NNT #1#2#3
  { \__peek_token_generic:NNTF #1 #2 {#3} { } }
\cs_new_protected:Npn \__peek_token_generic:NNF #1#2#3
```

```
  { \__peek_token_generic:NNTF #1 #2 { } {#3} }
\cs_new_protected:Npn \__peek_token_remove_generic:NNTF
  { \__peek_token_generic_aux:NNNTF \__peek_true_remove:w }
\cs_new_protected:Npn \__peek_token_remove_generic:NNT #1#2#3
  { \__peek_token_remove_generic:NNTF #1 #2 {#3} { } }
\cs_new_protected:Npn \__peek_token_remove_generic:NNF #1#2#3
  { \__peek_token_remove_generic:NNTF #1 #2 { } {#3} }
\cs_new:Npn \__peek_execute_branches_meaning:
  {
    \if_meaning:w \l_peek_token \l__peek_search_token
      \exp_after:wN \__peek_true:w
    \else:
      \exp_after:wN \__peek_false:w
    \fi:
  }
\cs_new:Npn \__peek_execute_branches_catcode:
  { \if_catcode:w \__peek_execute_branches_catcode_aux: }
\cs_new:Npn \__peek_execute_branches_charcode:
  { \if_charcode:w \__peek_execute_branches_catcode_aux: }
\cs_new:Npn \__peek_execute_branches_catcode_aux:
  {
        \if_catcode:w \exp_not:N \l_peek_token \scan_stop:
          \exp_after:wN \exp_after:wN
          \exp_after:wN \__peek_execute_branches_catcode_auxii:N
          \exp_after:wN \exp_not:N
        \else:
          \exp_after:wN \__peek_execute_branches_catcode_auxiii:
        \fi:
  }
\cs_new:Npn \__peek_execute_branches_catcode_auxii:N #1
  {
        \exp_not:N #1
        \exp_after:wN \exp_not:N \l__peek_search_tl
      \exp_after:wN \__peek_true:w
    \else:
      \exp_after:wN \__peek_false:w
    \fi:
    #1
  }
\cs_new:Npn \__peek_execute_branches_catcode_auxiii:
  {
        \exp_not:N \l_peek_token
        \exp_after:wN \exp_not:N \l__peek_search_tl
      \exp_after:wN \__peek_true:w
    \else:
      \exp_after:wN \__peek_false:w
    \fi:
  }
\tl_map_inline:nn { { catcode } { charcode } { meaning } }
  {
    \tl_map_inline:nn { { } { _remove } }
      {
        \tl_map_inline:nn { { TF } { T } { F } }
          {
            \cs_new_protected:cpe { peek_ #1 ##1 :N ####1 }
              {
                \exp_not:c { __peek_token ##1 _generic:NN ####1 }
                \exp_not:c { __peek_execute_branches_ #1 : }
              }
          }
      }
  }
\group_begin:
  \cs_set_protected:Npn \__peek_tmp:w #1 \s__peek_stop
    {
      \cs_new_protected:Npn \__peek_execute_branches_N_type:
        {
          \if_int_odd:w
              \if_catcode:w \exp_not:N \l_peek_token {   \c_zero_int \fi:
              \if_catcode:w \exp_not:N \l_peek_token }   \c_zero_int \fi:
              \if_meaning:w \l_peek_token \c_space_token \c_zero_int \fi:
              \c_one_int
            \exp_after:wN \__peek_N_type:w
              \token_to_meaning:N \l_peek_token
              \s__peek_mark \__peek_N_type_aux:nnw
              #1 \s__peek_mark \__peek_use_none_delimit_by_s_stop:w
              \s__peek_stop
            \exp_after:wN \__peek_true:w
          \else:
            \exp_after:wN \__peek_false:w
          \fi:
        }
      \cs_new_protected:Npn \__peek_N_type:w ##1 #1 ##2 \s__peek_mark ##3
        { ##3 {##1} {##2} }
    }
  \exp_after:wN \__peek_tmp:w \tl_to_str:n { outer } \s__peek_stop
\group_end:
\cs_new_protected:Npn \__peek_N_type_aux:nnw #1 #2 #3 \fi:
  {
    \fi:
    \tl_if_in:noTF {#1} { \tl_to_str:n {ma} }
      { \__peek_true:w }
      { \tl_if_empty:nTF {#2} { \__peek_true:w } { \__peek_false:w } }
  }
\cs_new_protected:Npn \peek_N_type:TF
  {
    \__peek_token_generic:NNTF
      \__peek_execute_branches_N_type: \scan_stop:
  }
\cs_new_protected:Npn \peek_N_type:T
  { \__peek_token_generic:NNT \__peek_execute_branches_N_type: \scan_stop: }
\cs_new_protected:Npn \peek_N_type:F
  { \__peek_token_generic:NNF \__peek_execute_branches_N_type: \scan_stop: }
%% File: l3prop.dtx
\cs_new_eq:NN \__prop_tmp:w ?
\tl_new:N \l__prop_internal_tl
\scan_new:N \s__prop_mark
\scan_new:N \s__prop_stop
\quark_new:N \q__prop_recursion_tail
\quark_new:N \q__prop_recursion_stop
\__kernel_quark_new_test:N \__prop_if_recursion_tail_stop:n
\cs_generate_variant:Nn \__prop_if_recursion_tail_stop:n { o }
\scan_new:N \s__prop
\cs_new_protected:Npn \__prop_chk:w { \__prop_chk_loop:nw { } }
\cs_new_protected:Npn \__prop_chk_loop:nw #1
  {
    \peek_meaning:NTF \__prop_pair:wn
      { \__prop_chk_get:nw {#1} }
      { \msg_error:nne { prop } { misused } {#1} }
  }
\cs_new_protected:Npn \__prop_chk_get:nw #1 \__prop_pair:wn #2 \s__prop #3
  { \__prop_chk_loop:nw { #1 , ~ {#2} = { \tl_to_str:n {#3} } } }
\cs_new:Npn \__prop_pair:wn #1 \s__prop #2 { }
\cs_new_protected:Npn \__prop_flatten:w #1 \s__prop #2#3
  { \use:e { \__prop_flatten_aux:N #3 } }
\cs_new:Npn \__prop_flatten:N #1
  { \exp_after:wN \__prop_flatten_aux:w #1 }
\cs_new:Npn \__prop_flatten_aux:w #1 \s__prop #2 { \__prop_flatten_aux:N }
\cs_new:Npn \__prop_flatten_aux:N #1
  {
    \s__prop \__prop_chk:w
    \exp_after:wN \__prop_flatten_loop:w #1
    \use_none:nnnn \__prop_pair:wn \s__prop { }
  }
\cs_new:Npn \__prop_flatten_loop:w #1#2#3 \__prop_pair:wn #4 \s__prop #5
  {
    #3
    \exp_not:n { \__prop_pair:wn #4 \s__prop {#5} }
    \exp_after:wN \__prop_flatten_loop:w
  }
\int_new:N \g__prop_prefix_int
\int_const:Nn \c__prop_basis_int { \c_max_char_int - `\! }
\cs_new_protected:Npn \__prop_next_prefix:
  {
    \tl_set:Ne \l__prop_internal_tl
      { \__prop_to_prefix:n { \g__prop_prefix_int } }
    \int_gincr:N \g__prop_prefix_int
  }
\cs_new:Npn \__prop_to_prefix:n #1
  {
    \int_compare:nNnTF {#1} > \c__prop_basis_int
      {
        \exp_args:Nf \__prop_to_prefix:n
          { \int_div_truncate:nn {#1} \c__prop_basis_int }
        \exp_args:Nf \__prop_to_prefix:n
          { \int_mod:nn {#1} \c__prop_basis_int }
      }
      { \char_generate:nn { `\! + #1 } { 12 } }
  }
\cs_new:Npn \__prop_if_flat:NTF #1
  {
    \prop_if_exist:NT #1
    \exp_after:wN \__prop_if_flat_aux:w #1
    \s__prop_mark \use_ii:nn
    \__prop_flatten:w \s__prop_mark \use_i:nn \s__prop_stop
  }
\cs_new:Npn \__prop_if_flat_aux:w
    #1 \__prop_flatten:w #2 \s__prop_mark #3 #4 \s__prop_stop {#3}
\tl_const:Nn \c_empty_prop { \s__prop \__prop_chk:w }
\cs_new_protected:Npn \prop_new:N #1
  {
    \__kernel_chk_if_free_cs:N #1
    \cs_gset_eq:NN #1 \c_empty_prop
  }
\cs_generate_variant:Nn \prop_new:N { c }
\cs_new_protected:Npn \prop_new_linked:N #1
  {
    \__kernel_chk_if_free_cs:N #1
    \__prop_new_linked:N #1
  }
\cs_new_protected:Npn \__prop_new_linked:N #1
  {
    \__prop_next_prefix:
    \cs_gset_nopar:Npe #1
```

```
      {
        \__prop_flatten:w
        \exp_not:c { __prop ~ \l__prop_internal_tl }
        \s__prop { \l__prop_internal_tl }
        \exp_not:c { __prop ~ \l__prop_internal_tl }
      }
    \cs_gset_nopar:cpe { __prop ~ \l__prop_internal_tl }
      {
        \exp_not:N \use_none:n
        \exp_not:N #1
      }
  }
\cs_generate_variant:Nn \prop_new_linked:N { c }
\cs_new_protected:Npn \prop_clear:N
  { \__prop_clear:NNN \cs_set_eq:NN \cs_set_nopar:Npe }
\cs_generate_variant:Nn \prop_clear:N  { c }
\cs_new_protected:Npn \prop_gclear:N
  { \__prop_clear:NNN \cs_gset_eq:NN \cs_gset_nopar:Npe }
\cs_generate_variant:Nn \prop_gclear:N { c }
\cs_new_protected:Npn \__prop_clear:NNN #1#2#3
  {
    \__prop_if_flat:NTF #3
      { #1 #3 \c_empty_prop }
      { \exp_after:wN \__prop_clear:wNNN #3 #1 #2 #3 }
  }
\cs_new_protected:Npn \__prop_clear:wNNN
    \__prop_flatten:w #1 \s__prop #2#3#4#5#6
  {
    \__prop_clear_entries:NN #4 #3
    #5 #6 { \exp_not:n { \__prop_flatten:w #1 \s__prop {#2} #1 } }
    #5 #1 { \exp_not:n { \use_none:n #6 } }
  }
\cs_new_protected:Npn \__prop_clear_entries:NN #1#2
  {
    \exp_after:wN \__prop_clear_loop:Nw \exp_after:wN #1 #2
    \use_none:nnnn \__prop_pair:wn \s__prop { }
  }
\cs_new_protected:Npn \__prop_clear_loop:Nw
    #1#2#3#4 \__prop_pair:wn #5 \s__prop #6
  {
    #1 #3 \tex_undefined:D
    #4
    \exp_after:wN \__prop_clear_loop:Nw
    \exp_after:wN #1
  }
\cs_new_protected:Npn \prop_clear_new:N  #1
```

```
  { \prop_if_exist:NTF #1 { \prop_clear:N #1 } { \prop_new:N #1 } }
\cs_generate_variant:Nn \prop_clear_new:N  { c }
\cs_new_protected:Npn \prop_gclear_new:N #1
  { \prop_if_exist:NTF #1 { \prop_gclear:N #1 } { \prop_new:N #1 } }
\cs_generate_variant:Nn \prop_gclear_new:N { c }
\cs_new_protected:Npn \prop_clear_new_linked:N  #1
  { \prop_if_exist:NTF #1 { \prop_clear:N #1 } { \prop_new_linked:N #1 } }
\cs_generate_variant:Nn \prop_clear_new_linked:N  { c }
\cs_new_protected:Npn \prop_gclear_new_linked:N #1
  { \prop_if_exist:NTF #1 { \prop_gclear:N #1 } { \prop_new_linked:N #1 } }
\cs_generate_variant:Nn \prop_gclear_new_linked:N { c }
\cs_new_protected:Npn \prop_set_eq:NN
  { \__prop_set_eq:NNNN \cs_set_eq:NN \cs_set_nopar:Npe }
\cs_generate_variant:Nn \prop_set_eq:NN { Nc , cN , cc }
\cs_new_protected:Npn \prop_gset_eq:NN
  { \__prop_set_eq:NNNN \cs_gset_eq:NN \cs_gset_nopar:Npe }
\cs_generate_variant:Nn \prop_gset_eq:NN { Nc , cN , cc }
\cs_new_protected:Npn \__prop_set_eq:NNNN #1#2#3#4
  {
    \cs_if_eq:NNF #3#4
      {
        \__prop_if_flat:NTF #3
          {
            \__prop_if_flat:NTF #4
              { #1 #3 #4 }
              { #2 #3 { \__prop_flatten:N #4 } }
          }
          { \exp_after:wN \__prop_set_eq:wNNNN #3 #1#2#3#4 }
      }
  }
\cs_new_protected:Npn \__prop_set_eq:wNNNN
    \__prop_flatten:w #1 \s__prop #2#3#4#5#6#7
  {
    \__prop_clear_entries:NN #4 #3
    \exp_args:Nf \__prop_set_eq:nNnNN {#7} #1 {#2} #5 #6
  }
\cs_new_protected:Npn \__prop_set_eq:nNnNN #1#2#3#4#5
  {
    \use_i:nnn
      {
        \__prop_set_eq_loop:NNnw #5 #4 {#3}
        \__prop_flatten:w #2 \s__prop {#3}
      }
      #1
    \use_none:n \__prop_pair:wn ? \s__prop
  }
```

```
\cs_new_protected:Npn \__prop_set_eq_loop:NNnw
    #1#2#3#4 \s__prop #5#6 \__prop_pair:wn #7 \s__prop
  {
    \tl_set:Ne \l__prop_internal_tl { \exp_not:c { __prop ~ #3 #6 ~ #7 } }
    #2 #1 { \exp_not:n { #4 \s__prop {#5} } \exp_not:o \l__prop_internal_tl }
    \use_none:n #6 \__prop_set_eq_end:w
    \exp_after:wN \__prop_set_eq_loop:NNnw \l__prop_internal_tl #2 {#3}
    \use_none:n #1 \__prop_pair:wn #7 \s__prop
  }
\cs_new_protected:Npn \__prop_set_eq_end:w
    \exp_after:wN \__prop_set_eq_loop:NNnw #1#2#3
    \use_none:n #4#5 \s__prop
  {
    \exp_after:wN #2 \l__prop_internal_tl { \exp_not:n { \use_none:n #4 } }
  }
\cs_new_protected:Npn \prop_make_flat:N #1
  {
    \int_compare:nNnTF { \tex_currentgrouplevel:D } = 0
      {
        \__prop_if_flat:NTF #1 { }
          { \exp_args:NNf \__prop_make_flat:Nn #1 {#1} }
      }
      {
        \msg_error:nnee { prop } { inner-make }
          { \token_to_str:N \prop_make_flat:N } { \token_to_str:N #1 }
      }
  }
\cs_generate_variant:Nn \prop_make_flat:N { c }
\cs_new_protected:Npn \__prop_make_flat:Nn #1#2
  {
    \exp_after:wN \__prop_clear:wNNN #1 \cs_set_eq:NN \cs_set_nopar:Npe #1
    \cs_set_nopar:Npe #1 { \exp_not:n {#2} }
  }
\cs_new_protected:Npn \prop_make_linked:N #1
  {
    \int_compare:nNnTF { \tex_currentgrouplevel:D } = 0
      {
        \__prop_if_flat:NTF #1
          { \exp_args:NNo \__prop_make_linked:Nn #1 {#1} } { }
      }
      {
        \msg_error:nnee { prop } { inner-make }
          { \token_to_str:N \prop_make_linked:N } { \token_to_str:N #1 }
      }
  }
\cs_generate_variant:Nn \prop_make_linked:N { c }
```

```
\cs_new_protected:Npn \__prop_make_linked:Nn #1#2
  {
    \__prop_new_linked:N #1
    \tl_set:Nn \l__prop_internal_tl {#2}
    \exp_after:wN \__prop_set_eq:wNNNN #1
      \cs_set_eq:NN \cs_set_nopar:Npe #1 \l__prop_internal_tl
  }
\prop_new:N \l_tmpa_prop
\prop_new:N \l_tmpb_prop
\prop_new:N \g_tmpa_prop
\prop_new:N \g_tmpb_prop
\cs_new_protected:Npn \prop_concat:NNN
  { \__prop_concat:NNNNN \cs_set_eq:NN \cs_set_nopar:Npe }
\cs_generate_variant:Nn \prop_concat:NNN { ccc }
\cs_new_protected:Npn \prop_gconcat:NNN
  { \__prop_concat:NNNNN \cs_gset_eq:NN \cs_gset_nopar:Npe }
\cs_generate_variant:Nn \prop_gconcat:NNN { ccc }
\cs_new_protected:Npn \__prop_concat:NNNNN #1#2#3#4#5
  {
    \cs_if_eq:NNTF #3 #5
      { \__prop_concat:nNNN \use_none:nnn #2 #3 #4 }
      {
        \__prop_set_eq:NNNN #1 #2 #3 #4
        \__prop_concat:nNNN { } #2 #3 #5
      }
  }
\cs_new_protected:Npn \__prop_concat:nNNN #1#2#3#4
  {
    \cs_gset_eq:NN \__prop_tmp:w \__prop_pair:wn
    \cs_gset_protected:Npn \__prop_pair:wn ##1 \s__prop
      { \__prop_put:nNNnn {#1} #2 #3 {##1} }
    \exp_last_unbraced:Nf \use_none:nn #4
    \cs_gset_eq:NN \__prop_pair:wn \__prop_tmp:w
  }
\cs_new_protected:Npn \prop_put_from_keyval:Nn #1
  { \__prop_from_keyval:nn { \__prop_put:nNNnn { } \cs_set_nopar:Npe #1 } }
\cs_generate_variant:Nn \prop_put_from_keyval:Nn { c }
\cs_new_protected:Npn \prop_gput_from_keyval:Nn #1
  { \__prop_from_keyval:nn { \__prop_put:nNNnn { } \cs_gset_nopar:Npe #1 } }
\cs_generate_variant:Nn \prop_gput_from_keyval:Nn { c }
\cs_new_protected:Npn \__prop_from_keyval:nn
  {
    \bool_if:NTF \l__kernel_keyval_allow_blank_keys_bool
      { \__prop_from_keyval:Nnn \c_true_bool }
      { \__prop_from_keyval:Nnn \c_false_bool }
  }
```

```
\cs_new_protected:Npn \__prop_from_keyval:Nnn #1#2#3
  {
    \bool_set_eq:NN \l__kernel_keyval_allow_blank_keys_bool \c_true_bool
    \keyval_parse:nnn \__prop_missing_eq:n {#2} {#3}
    \bool_set_eq:NN \l__kernel_keyval_allow_blank_keys_bool #1
  }
\cs_new_protected:Npn \__prop_missing_eq:n
  { \msg_error:nnn { prop } { prop-keyval } }
\cs_new_protected:Npn \prop_set_from_keyval:Nn #1
  {
    \__prop_clear:NNN \cs_set_eq:NN \cs_set_nopar:Npe #1
    \prop_put_from_keyval:Nn #1
  }
\cs_generate_variant:Nn \prop_set_from_keyval:Nn { c }
\cs_new_protected:Npn \prop_gset_from_keyval:Nn #1
  {
    \__prop_clear:NNN \cs_gset_eq:NN \cs_gset_nopar:Npe #1
    \prop_gput_from_keyval:Nn #1
  }
\cs_generate_variant:Nn \prop_gset_from_keyval:Nn { c }
\cs_new_protected:Npn \prop_const_from_keyval:Nn #1
  {
    \prop_new:N #1
    \__prop_from_keyval:nn { \__prop_put:nNNnn { } \cs_gset_nopar:Npe #1 }
  }
\cs_generate_variant:Nn \prop_const_from_keyval:Nn { c }
\cs_new_protected:Npn \prop_const_linked_from_keyval:Nn #1
  {
    \prop_new_linked:N #1
    \__prop_from_keyval:nn { \__prop_put:nNNnn { } \cs_gset_nopar:Npe #1 }
  }
\cs_generate_variant:Nn \prop_const_linked_from_keyval:Nn { c }
\cs_new_protected:Npn \__prop_split:NnTFn #1#2
  {
    \exp_after:wN \__prop_split_aux:nNTFn
    \exp_after:wN { \tl_to_str:n {#2} } #1
  }
\cs_new_protected:Npn \__prop_split_aux:nNTFn #1#2#3
  {
    \cs_set:Npn \__prop_split_flat:w \__prop_split_linked:w ##1
      \__prop_pair:wn #1 \s__prop ##2 ##3 \s__prop_mark ##4 ##5 \s__prop_stop
      { ##4 {#3} }
    \exp_after:wN \__prop_split_test:wn #2 \s__prop_mark \use_i:nnn
      \__prop_pair:wn #1 \s__prop { ? \fi: \__prop_split_wrong:Nw #2 }
      \s__prop_mark \use_ii:nnn
      \s__prop_stop
  }
\cs_new:Npn \__prop_split_flat:w { }
\cs_new_protected:Npn \__prop_split_test:wn #1 \s__prop #2
  {
    \if_meaning:w \__prop_chk:w #2 \exp_after:wN \__prop_split_flat:w \fi:
    \__prop_split_linked:w
  }
\cs_new_protected:Npn \__prop_split_linked:w #1 \s__prop_stop #2#3 {#3}
\cs_new_protected:Npn \__prop_split_wrong:Nw #1#2 \s__prop_stop #3#4
  {
    \prop_show:N #1
    \cs_gset_eq:NN #1 \c_empty_prop
    #3
  }
\cs_new_protected:Npn \prop_get:NnN #1#2#3
  {
    \__prop_get:NnnTF #1 {#2}
      { \tl_set:Nn #3 } { } { \tl_set:Nn #3 { \q_no_value } }
  }
\cs_generate_variant:Nn \prop_get:NnN { NV , Nv , Ne , c , cV , cv , ce }
\cs_generate_variant:Nn \prop_get:NnN { No , Nx , co , cx }
\cs_generate_variant:Nn \prop_get:NnN { cnc }
\prg_new_protected_conditional:Npnn \prop_get:NnN #1#2#3 { T , F , TF }
  {
    \__prop_get:NnnTF #1 {#2}
      { \tl_set:Nn #3 } \prg_return_true: \prg_return_false:
  }
\prg_generate_conditional_variant:Nnn \prop_get:NnN
  { NV , Nv , Ne , c , cV , cv , ce } { T , F , TF }
\prg_generate_conditional_variant:Nnn \prop_get:NnN
  { No , Nx , co , cx } { T , F , TF }
\prg_generate_conditional_variant:Nnn \prop_get:NnN
  { cnc } { T , F , TF }
\cs_new_protected:Npn \__prop_get:NnnTF #1#2#3#4#5
  {
    \__prop_split:NnTFn #1 {#2}
      { #3 {##2} #4 }
      {#5}
      { \exp_after:wN \__prop_get_linked:w #1 {#2} {#3} {#4} {#5} }
  }
\cs_new:Npn \__prop_get_linked:w
    \__prop_flatten:w #1 \s__prop #2#3#4#5#6#7
  {
    \if_cs_exist:w __prop ~ #2 ~ \tl_to_str:n {#4} \cs_end:
      \exp_after:wN \exp_after:wN \exp_after:wN \__prop_get_linked_aux:w
      \cs:w __prop ~ #2 ~ \tl_to_str:n {#4} \exp_after:wN \cs_end:
```

```
    \else:
      \exp_after:wN \__prop_get_linked_aux:w
    \fi:
    \s__prop_mark {#5} {#6}
    \s__prop { } \s__prop_mark \use_none:n {#7}
    \s__prop_stop
  }
\cs_new:Npn \__prop_get_linked_aux:w
    #1 \s__prop #2 #3 \s__prop_mark #4 #5 #6 \s__prop_stop { #4 {#2} #5 }
\cs_new:Npn \prop_item:Nn #1#2
  {
    \__prop_if_flat:NTF #1
      {
        \exp_args:NNo \prop_map_tokens:Nn #1
          {
            \exp_after:wN \__prop_item:nnn
            \exp_after:wN { \tl_to_str:n {#2} }
          }
      }
      { \exp_after:wN \__prop_get_linked:w #1 {#2} \exp_not:n { } { } }
  }
\cs_new:Npn \__prop_item:nnn #1#2#3
  {
    \str_if_eq:eeT {#1} {#2}
      { \prop_map_break:n { \exp_not:n {#3} } }
  }
\cs_generate_variant:Nn \prop_item:Nn { NV , No , Ne , c , cV , co , ce }
\cs_new_protected:Npn \__prop_pop:NnNNnTF #1#2#3#4#5#6#7
  {
    \__prop_split:NnTFn #1 {#2}
      {
        #4 #1 { \exp_not:n { \s__prop \__prop_chk:w ##1 ##3 } }
        #5 {##2}
        #6
      }
      {#7}
      {
        \exp_after:wN \__prop_pop_linked:wnNNnTF #1 {#2}
          #3 #4 {#5} {#6} {#7}
      }
  }
\cs_new_protected:Npn \__prop_pop_linked:wnNNnTF
    \__prop_flatten:w #1 \s__prop #2#3#4#5#6#7
  {
    \if_cs_exist:w __prop ~ #2 ~ \tl_to_str:n {#4} \cs_end:
      \exp_after:wN \__prop_pop_linked:NNNn
      \cs:w __prop ~ #2 ~ \tl_to_str:n {#4} \cs_end:
      #5 #6 {#7}
    \else:
      \exp_after:wN \use_iii:nnn
    \fi:
    \use_i:nn
  }
\cs_new_protected:Npn \__prop_pop_linked:NNNn #1#2#3#4
  {
    \if_meaning:w \scan_stop: #1
      \exp_after:wN \exp_after:wN \exp_after:wN \use_iii:nnn
    \else:
      \exp_after:wN \__prop_pop_linked:w #1 #1 #2 #3 {#4}
    \fi:
  }
\cs_new_protected:Npn \__prop_pop_linked:w
    \use_none:n #1#2 \s__prop #3#4#5#6#7#8
  {
    #6 #5 \tex_undefined:D
    #7 #1
      {
        \exp_after:wN \__prop_pop_linked_prev:w #1
        \exp_not:N #4
      }
    #7 #4
      {
        \exp_not:n { \use_none:n #1 }
        \exp_not:f { \exp_after:wN \__prop_pop_linked_next:w #4 }
      }
    #8 {#3}
  }
\cs_new:Npn \__prop_pop_linked_prev:w #1 \s__prop #2#3
  { \exp_not:n { #1 \s__prop {#2} } }
\cs_new:Npn \__prop_pop_linked_next:w \use_none:n #1 { \exp_stop_f: }
\cs_new_protected:Npn \prop_remove:Nn #1#2
  {
    \__prop_pop:NnNNnTF #1 {#2}
      \cs_set_eq:NN \cs_set_nopar:Npe
      \use_none:n { } { }
  }
\cs_new_protected:Npn \prop_gremove:Nn #1#2
  {
    \__prop_pop:NnNNnTF #1 {#2}
      \cs_gset_eq:NN \cs_gset_nopar:Npe
      \use_none:n { } { }
  }
```

```
\cs_generate_variant:Nn \prop_remove:Nn  { NV , Ne , c , cV , ce }
\cs_generate_variant:Nn \prop_gremove:Nn { NV , Ne , c , cV , ce }
\cs_new_protected:Npn \prop_pop:NnN #1#2#3
  {
    \__prop_pop:NnNNnTF #1 {#2}
      \cs_set_eq:NN \cs_set_nopar:Npe
      { \tl_set:Nn #3 } { } { \tl_set:Nn #3 { \q_no_value } }
  }
\cs_new_protected:Npn \prop_gpop:NnN #1#2#3
  {
    \__prop_pop:NnNNnTF #1 {#2}
      \cs_gset_eq:NN \cs_gset_nopar:Npe
      { \tl_set:Nn #3 } { } { \tl_set:Nn #3 { \q_no_value } }
  }
\cs_generate_variant:Nn \prop_pop:NnN  {     NV , No }
\cs_generate_variant:Nn \prop_pop:NnN  { c , cV , co }
\cs_generate_variant:Nn \prop_gpop:NnN {     NV , No }
\cs_generate_variant:Nn \prop_gpop:NnN { c , cV , co }
\prg_new_protected_conditional:Npnn \prop_pop:NnN #1#2#3 { T , F , TF }
  {
    \__prop_pop:NnNNnTF #1 {#2}
      \cs_set_eq:NN \cs_set_nopar:Npe
      { \tl_set:Nn #3 } \prg_return_true: \prg_return_false:
  }
\prg_new_protected_conditional:Npnn \prop_gpop:NnN #1#2#3 { T , F , TF }
  {
    \__prop_pop:NnNNnTF #1 {#2}
      \cs_gset_eq:NN \cs_gset_nopar:Npe
      { \tl_set:Nn #3 } \prg_return_true: \prg_return_false:
  }
\prg_generate_conditional_variant:Nnn \prop_pop:NnN
  { NV , No , c , cV , co } { T , F , TF }
\prg_generate_conditional_variant:Nnn \prop_gpop:NnN
  { NV , No , c , cV , co } { T , F , TF }
\cs_new_protected:Npn \prop_put:Nnn
  { \__prop_put:nNNnn { } \cs_set_nopar:Npe }
\cs_new_protected:Npn \prop_gput:Nnn
  { \__prop_put:nNNnn { } \cs_gset_nopar:Npe }
\cs_new_protected:Npn \prop_put_if_not_in:Nnn
  { \__prop_put:nNNnn \use_none:nnn \cs_set_nopar:Npe }
\cs_new_protected:Npn \prop_gput_if_not_in:Nnn
  { \__prop_put:nNNnn \use_none:nnn \cs_gset_nopar:Npe }
\cs_generate_variant:Nn \prop_put:Nnn
  {
         NnV , Nnv , Nne , NV , NVV , NVv , NVe ,
    Nv , NvV , Nvv , Nve , Ne , NeV , Nev , Nee
  }
\cs_generate_variant:Nn \prop_put:Nnn
  { Nno , No , Noo , Nnx , NVx , NxV , Nxx }
\cs_generate_variant:Nn \prop_put:Nnn
  {
    c  , cnV , cnv , cne , cV , cVV , cVv , cVe ,
    cv , cvV , cvv , cve , ce , ceV , cev , cee
  }
\cs_generate_variant:Nn \prop_put:Nnn
  { cno , co , coo , cnx , cVx , cxV , cxx }
\cs_generate_variant:Nn \prop_gput:Nnn
  {
         NnV , Nnv , Nne , NV , NVV , NVv , NVe ,
    Nv , NvV , Nvv , Nve , Ne , NeV , Nev , Nee
  }
\cs_generate_variant:Nn \prop_gput:Nnn
  { Nno , No , Noo , Nnx , NVx , NxV , Nxx }
\cs_generate_variant:Nn \prop_gput:Nnn
  {
    c  , cnV , cnv , cne , cV , cVV , cVv , cVe ,
    cv , cvV , cvv , cve , ce , ceV , cev , cee
  }
\cs_generate_variant:Nn \prop_gput:Nnn
  { cno , co , coo , cnx , cVx , cxV , cxx }
\cs_generate_variant:Nn \prop_put_if_not_in:Nnn
  {
         NnV , Nnv , Nne , NV , NVV , NVv , NVe ,
    Nv , NvV , Nvv , Nve , Ne , NeV , Nev , Nee ,
    c  , cnV , cnv , cne , cV , cVV , cVv , cVe ,
    cv , cvV , cvv , cve , ce , ceV , cev , cee
  }
\cs_generate_variant:Nn \prop_gput_if_not_in:Nnn
  {
         NnV , Nnv , Nne , NV , NVV , NVv , NVe ,
    Nv , NvV , Nvv , Nve , Ne , NeV , Nev , Nee ,
    c  , cnV , cnv , cne , cV , cVV , cVv , cVe ,
    cv , cvV , cvv , cve , ce , ceV , cev , cee
  }
\cs_new_protected:Npn \__prop_put:nNNnn #1#2#3#4#5
  {
    \tl_set:Nn \l__prop_internal_tl
      {
        \exp_not:N \__prop_pair:wn \tl_to_str:n {#4}
        \s__prop { \exp_not:n {#5} }
      }
    \__prop_split:NnTFn #3 {#4}
```

```
      {
        #1 #2 #3
          {
            \s__prop \__prop_chk:w \exp_not:n {##1}
            \l__prop_internal_tl \exp_not:n {##3}
          }
      }
      { #2 #3 { \exp_not:o {#3} \l__prop_internal_tl } }
      { \exp_after:wN \__prop_put_linked:wnnN #3 {#4} {#1} #2 }
  }
\cs_new_protected:Npn \__prop_put_linked:wnnN
    \__prop_flatten:w #1 \s__prop #2#3#4
  {
    \exp_after:wN \__prop_put_linked:NNnN
    \cs:w __prop ~ #2 ~ \tl_to_str:n {#4} \cs_end:
    #1
  }
\cs_new_protected:Npn \__prop_put_linked:NNnN #1#2#3#4
  {
    \if_meaning:w \scan_stop: #1
      \exp_after:wN \__prop_put_linked_new:w #2 #1 #2 #4
    \else:
      \exp_after:wN \__prop_put_linked_old:w #1 { #3 #4 #1 }
    \fi:
  }
\cs_new_protected:Npn \__prop_put_linked_new:w
    \use_none:n #1#2#3#4
  {
    #4 #1
      {
        \exp_after:wN \__prop_pop_linked_prev:w #1
        \exp_not:N #2
      }
    #4 #2
      {
        \exp_not:n { \use_none:n #1 }
        \l__prop_internal_tl
        \exp_not:N #3
      }
    #4 #3 { \exp_not:n { \use_none:n #2 } }
  }
\cs_new_protected:Npn \__prop_put_linked_old:w
    \use_none:n #1#2 \s__prop #3#4#5
  {
    #5
      {
        \exp_not:n { \use_none:n #1 }
        \l__prop_internal_tl
        \exp_not:N #4
      }
  }
\prg_new_eq_conditional:NNn \prop_if_exist:N \cs_if_exist:N
  { TF , T , F , p }
\prg_new_eq_conditional:NNn \prop_if_exist:c \cs_if_exist:c
  { TF , T , F , p }
\prg_new_conditional:Npnn \prop_if_empty:N #1 { p , T , F , TF }
  {
    \if_meaning:w #1 \c_empty_prop
      \prg_return_true:
    \else:
      \exp_after:wN \__prop_if_empty_return:w #1
        \__prop_flatten:w 2 \s__prop 34 \s__prop_stop
    \fi:
  }
\cs_new:Npn \__prop_if_empty_return:w
    #1 \__prop_flatten:w #2 \s__prop #3#4#5 \s__prop_stop
  {
    \if_meaning:w #2 #4
      \prg_return_true:
    \else:
      \prg_return_false:
    \fi:
  }
\prg_generate_conditional_variant:Nnn \prop_if_empty:N
  { c } { p , T , F , TF }
\prg_new_conditional:Npnn \prop_if_in:Nn #1#2 { p , T , F , TF }
  {
    \__prop_if_flat:NTF #1
      {
        \exp_after:wN \prop_map_tokens:Nn \exp_after:wN #1
          {
            \exp_after:wN \__prop_if_in_flat:nnn
            \exp_after:wN { \tl_to_str:n {#2} }
          }
        \prg_return_false:
      }
      {
        \exp_after:wN \__prop_get_linked:w #1 {#2}
        \use_none:n \prg_return_true: \prg_return_false:
      }
  }
\cs_new:Npn \__prop_if_in_flat:nnn #1#2#3
```

```
  {
    \str_if_eq:eeT {#1} {#2}
      { \prop_map_break:n { \use_i:nn \prg_return_true: } }
  }
\prg_generate_conditional_variant:Nnn \prop_if_in:Nn
  { NV , Ne , No , c , cV , ce , co } { p , T , F , TF }
\cs_new:Npn \prop_map_function:NN #1#2
  {
    \exp_last_unbraced:Nnf
      \use_i:nnn { \__prop_map_function:Nw #2 } #1
    \__prop_pair:wn \fi: \prop_map_break: \s__prop { }
    \__prop_pair:wn \fi: \prop_map_break: \s__prop { }
    \__prop_pair:wn \fi: \prop_map_break: \s__prop { }
    \__prop_pair:wn \fi: \prop_map_break: \s__prop { }
    \prg_break_point:Nn \prop_map_break: { }
  }
\cs_new:Npn \__prop_map_function:Nw #1
    \__prop_pair:wn #2 \s__prop #3
    \__prop_pair:wn #4 \s__prop #5
    \__prop_pair:wn #6 \s__prop #7
    \__prop_pair:wn #8 \s__prop #9
  {
    \if_false: #2 \fi: #1 {#2} {#3}
    \if_false: #4 \fi: #1 {#4} {#5}
    \if_false: #6 \fi: #1 {#6} {#7}
    \if_false: #8 \fi: #1 {#8} {#9}
    \__prop_map_function:Nw #1
  }
\cs_generate_variant:Nn \prop_map_function:NN { Nc , c , cc }
\cs_new_protected:Npn \prop_map_inline:Nn #1#2
  {
    \cs_gset_eq:cN
      { __prop_map_ \int_use:N \g__kernel_prg_map_int :wn } \__prop_pair:wn
    \int_gincr:N \g__kernel_prg_map_int
    \cs_gset_protected:Npn \__prop_pair:wn ##1 \s__prop ##2 {#2}
    \exp_last_unbraced:Nf \use_none:nn #1
    \prg_break_point:Nn \prop_map_break:
      {
        \int_gdecr:N \g__kernel_prg_map_int
        \cs_gset_eq:Nc \__prop_pair:wn
          { __prop_map_ \int_use:N \g__kernel_prg_map_int :wn }
      }
  }
\cs_generate_variant:Nn \prop_map_inline:Nn { c }
\cs_new:Npn \prop_map_tokens:Nn #1#2
  {
    \exp_last_unbraced:Nnf
      \use_i:nnn { \__prop_map_tokens:nw {#2} } #1
    \__prop_pair:wn \fi: \prop_map_break: \s__prop { }
    \__prop_pair:wn \fi: \prop_map_break: \s__prop { }
    \__prop_pair:wn \fi: \prop_map_break: \s__prop { }
    \__prop_pair:wn \fi: \prop_map_break: \s__prop { }
    \prg_break_point:Nn \prop_map_break: { }
  }
\cs_new:Npn \__prop_map_tokens:nw #1
    \__prop_pair:wn #2 \s__prop #3
    \__prop_pair:wn #4 \s__prop #5
    \__prop_pair:wn #6 \s__prop #7
    \__prop_pair:wn #8 \s__prop #9
  {
    \if_false: #2 \fi: \use:n {#1} {#2} {#3}
    \if_false: #4 \fi: \use:n {#1} {#4} {#5}
    \if_false: #6 \fi: \use:n {#1} {#6} {#7}
    \if_false: #8 \fi: \use:n {#1} {#8} {#9}
    \__prop_map_tokens:nw {#1}
  }
\cs_generate_variant:Nn \prop_map_tokens:Nn { c }
\cs_new:Npn \prop_map_break:
  { \prg_map_break:Nn \prop_map_break: { } }
\cs_new:Npn \prop_map_break:n
  { \prg_map_break:Nn \prop_map_break: }
\cs_new:Npn \prop_count:N #1
  {
    \int_eval:n
      {
        0
        \prop_map_function:NN #1 \__prop_count:nn
      }
  }
\cs_new:Npn \__prop_count:nn #1#2 { + 1 }
\cs_generate_variant:Nn \prop_count:N { c }
\cs_new:Npn \prop_to_keyval:N #1
  {
    \__kernel_exp_not:w
      \prop_if_empty:NTF #1
        { {} }
        {
          \exp_after:wN \exp_after:wN \exp_after:wN
          {
            \tex_expanded:D
              {
                \exp_not:N \use_none:n
                \prop_map_function:NN #1 \__prop_to_keyval:nn
              }
          }
        }
  }
\cs_new:Npn \__prop_to_keyval:nn #1#2
  { , ~ {#1} =~ { \__kernel_exp_not:w {#2} } }
\cs_new_protected:Npn \prop_show:N { \__prop_show:NN \msg_show:nneeee }
\cs_generate_variant:Nn \prop_show:N { c }
\cs_new_protected:Npn \prop_log:N { \__prop_show:NN \msg_log:nneeee }
\cs_generate_variant:Nn \prop_log:N { c }
\cs_new_protected:Npn \__prop_show:NN #1#2
  {
    \__kernel_chk_tl_type:NnnT #2 { prop }
      {
        \__prop_if_flat:NTF #2
          {
            \s__prop \__prop_chk:w
            \exp_after:wN \__prop_show_flat:w #2
            \s__prop { }
            \__prop_pair:wn \q__prop_recursion_tail \s__prop { }
            \q__prop_recursion_stop
          }
          { \exp_after:wN \__prop_show_linked:w #2 \s__prop ! ? \s__prop_stop }
      }
      {
        \__prop_if_flat:NTF #2
          { \__prop_show_finally:NNn #1 #2 { flat } }
          {
            \tl_set:Nn \l__prop_internal_tl { #1 #2 }
            \exp_after:wN \__prop_show_prepare:w #2 #2
          }
      }
  }
\cs_new:Npn \__prop_show_flat:w #1 \__prop_pair:wn #2 \s__prop #3
  {
    \__prop_if_recursion_tail_stop:n {#2}
    \exp_not:N \__prop_pair:wn \tl_to_str:n {#2} \s__prop \exp_not:n { {#3} }
    \__prop_show_flat:w
  }
\cs_new:Npn \__prop_show_linked:w #1 \s__prop #2#3#4 \s__prop_stop
  {
    \exp_not:N \__prop_flatten:w
    \exp_not:c { __prop ~ \tl_to_str:n {#2} }
    \s__prop { \tl_to_str:n {#2} }
    \exp_not:n {#3}
  }
\cs_new_protected:Npn \__prop_show_finally:NNn #1#2#3
  {
    #1 { prop } { show }
      { \token_to_str:N #2 }
      { \prop_map_function:NN #2 \msg_show_item:nn }
      {#3} { }
  }
\cs_new_protected:Npn \__prop_show_prepare:w
    \__prop_flatten:w #1 \s__prop #2#3#4
  {
    \use:e
      {
        \cs_set_nopar:Npn \exp_not:N \__prop_tmp:w
            ##1 \token_to_str:N #1 ##2 \s__prop_mark ##3 \s__prop_stop
          {
            \exp_not:N \tl_if_empty:nTF {##1}
              {
                \exp_not:N \tl_if_head_is_space:nTF {##2}
                  { \exp_not:N \exp_args:Nf \__prop_show_loop_key:wNNN }
                  { \exp_not:N \tl_if_empty:nTF }
                {##2}
              }
              { \exp_not:N \use_ii:nn }
            \__prop_show_end:NNN
            \__prop_show_bad_name:NNN
          }
      }
    \exp_last_unbraced:NNNo \__prop_show_loop:NNw #1 #4 #4
  }
\cs_new_protected:Npn \__prop_show_loop:NNw #1#2 #3 \s__prop #4#5
  {
    \exp_last_two_unbraced:Noo \__prop_tmp:w
      { \token_to_str:N #5 \s__prop_mark }
      { \token_to_str:N #1 \s__prop_mark \s__prop_stop }
      #1 #2 #5
  }
\cs_new_protected:Npn \__prop_show_bad_name:NNN #1#2#3
  {
    \msg_error:nneeee { prop } { bad-link }
      { \tl_tail:N \l__prop_internal_tl }
      { \token_to_str:N #2 }
      { \token_to_str:N #3 }
      { \token_to_str:N #1 }
  }
\cs_new_protected:Npn \__prop_show_end:NNN #1#2#3
```

```
  {
    \__kernel_chk_tl_type:NnnT #3
      { \tl_tail:N \l__prop_internal_tl prop~entry }
      { \exp_not:n { \use_none:n #2 } }
      {
        \exp_after:wN \__prop_show_finally:NNn
        \l__prop_internal_tl { linked }
      }
  }
\cs_new_protected:Npn \__prop_show_loop_key:wNNN #1#2#3#4#5#6
  {
    \__kernel_chk_tl_type:NnnT #6
      { \tl_tail:N \l__prop_internal_tl prop~entry }
      {
        \exp_not:n { \use_none:n #5 }
        \exp_after:wN \__prop_show_flat:w #6 \s__prop { }
        \__prop_pair:wn \q__prop_recursion_tail \s__prop { }
        \q__prop_recursion_stop
        \tl_item:Nn #6 { -1 }
      }
      { \exp_last_unbraced:NNNo \__prop_show_loop:NNw #4 #6 #6 }
  }
%% File: l3msg.dtx
\tl_new:N \l__msg_internal_tl
\str_new:N \l__msg_name_str
\str_new:N \l__msg_text_str
\scan_new:N \s__msg_mark
\scan_new:N \s__msg_stop
\cs_new:Npn \__msg_use_none_delimit_by_s_stop:w #1 \s__msg_stop { }
\tl_const:Nn \c__msg_text_prefix_tl      { msg~text~>~ }
\tl_const:Nn \c__msg_more_text_prefix_tl { msg~extra~text~>~ }
\prg_new_conditional:Npnn \msg_if_exist:nn #1#2 { p , T , F , TF }
  {
    \cs_if_exist:cTF { \c__msg_text_prefix_tl #1 / #2 }
      { \prg_return_true: } { \prg_return_false: }
  }
\cs_new_protected:Npn \__msg_chk_free:nn #1#2
  {
    \msg_if_exist:nnT {#1} {#2}
      {
        \msg_error:nnnn { msg } { already-defined }
          {#1} {#2}
      }
  }
\cs_new_protected:Npn \msg_new:nnnn #1#2#3#4
  {
    \__msg_chk_free:nn {#1} {#2}
    \cs_gset:cpn { \c__msg_text_prefix_tl #1 / #2 }
      ##1##2##3##4 {#3}
    \cs_gset:cpn { \c__msg_more_text_prefix_tl #1 / #2 }
      ##1##2##3##4 {#4}
  }
\cs_generate_variant:Nn \msg_new:nnnn { nnee , nnxx }
\cs_new_protected:Npn \msg_new:nnn #1#2#3
  { \msg_new:nnnn {#1} {#2} {#3} { } }
\cs_generate_variant:Nn \msg_new:nnn { nne , nnx }
\cs_new_protected:Npn \msg_set:nnnn #1#2#3#4
  {
    \cs_set:cpn { \c__msg_text_prefix_tl #1 / #2 }
      ##1##2##3##4 {#3}
    \cs_set:cpn { \c__msg_more_text_prefix_tl #1 / #2 }
      ##1##2##3##4 {#4}
  }
\cs_new_protected:Npn \msg_set:nnn #1#2#3
  { \msg_set:nnnn {#1} {#2} {#3} { } }
\tl_const:Nn \c__msg_coding_error_text_tl
  {
    This~is~a~coding~error.
    \\ \\
  }
\tl_const:Nn \c__msg_continue_text_tl
  { Type~<return>~to~continue }
\tl_const:Nn \c__msg_critical_text_tl
  { Reading~the~current~file~'\g_file_curr_name_str'~will~stop. }
\tl_const:Nn \c__msg_fatal_text_tl
  { This~is~a~fatal~error:~LaTeX~will~abort. }
\tl_const:Nn \c__msg_help_text_tl
  { For~immediate~help~type~H~<return> }
\tl_const:Nn \c__msg_no_info_text_tl
  {
    LaTeX~does~not~know~anything~more~about~this~error,~sorry.
    \c__msg_return_text_tl
  }
\tl_const:Nn \c__msg_on_line_text_tl { on~line }
\tl_const:Nn \c__msg_return_text_tl
  {
    \\ \\
    Try~typing~<return>~to~proceed.
    \\
    If~that~doesn't~work,~type~X~<return>~to~quit.
  }
\tl_const:Nn \c__msg_trouble_text_tl
```

```
  {
    \\ \\
    More~errors~will~almost~certainly~follow: \\
    the~LaTeX~run~should~be~aborted.
  }
\cs_new:Npn \msg_line_number: { \int_use:N \tex_inputlineno:D }
\cs_gset:Npn \msg_line_context:
  {
    \c__msg_on_line_text_tl
    \c_space_tl
    \msg_line_number:
  }
\cs_new_protected:Npn \__msg_interrupt:NnnnN #1#2#3#4#5
  {
    \str_set:Ne \l__msg_text_str { #1 {#2} }
    \str_set:Ne \l__msg_name_str { \msg_module_name:n {#2} }
    \cs_if_eq:cNTF
      { \c__msg_more_text_prefix_tl #2 / #3 }
      \__msg_no_more_text:nnnn
      {
        \__msg_interrupt_wrap:nnn
          { \use:c { \c__msg_text_prefix_tl #2 / #3 } #4 }
          { \c__msg_continue_text_tl }
          {
            \c__msg_no_info_text_tl
            \tl_if_empty:NF #5
              { \\ \\ #5 }
          }
      }
      {
        \__msg_interrupt_wrap:nnn
          { \use:c { \c__msg_text_prefix_tl #2 / #3 } #4 }
          { \c__msg_help_text_tl }
          {
            \use:c { \c__msg_more_text_prefix_tl #2 / #3 } #4
            \tl_if_empty:NF #5
              { \\ \\ #5 }
          }
      }
  }
\cs_new:Npn \__msg_no_more_text:nnnn #1#2#3#4 { }
\cs_new_protected:Npn \__msg_interrupt_wrap:nnn #1#2#3
  {
    \iow_wrap:nnnN { \\ #3 } { } { } \__msg_interrupt_more_text:n
    \group_begin:
      \int_sub:Nn \l_iow_line_count_int { 2 }
      \iow_wrap:nenN { \l__msg_text_str : ~ #1 }
        {
          ( \l__msg_name_str )
          \prg_replicate:nn
            {
                \str_count:N \l__msg_text_str
              - \str_count:N \l__msg_name_str
              + 2
            }
            { ~ }
        }
        { } \__msg_interrupt_text:n
    \iow_wrap:nnnN { \l__msg_internal_tl \\ \\ #2 } { } { }
      \__msg_interrupt:n
  }
\cs_new_protected:Npn \__msg_interrupt_text:n #1
  {
    \group_end:
    \tl_set:Nn \l__msg_internal_tl {#1}
  }
\cs_new_protected:Npn \__msg_interrupt_more_text:n #1
  { \exp_args:Ne \tex_errhelp:D { #1 \iow_newline: } }
\group_begin:
  \char_set_lccode:nn { 38 } { 32 } % &
  \char_set_lccode:nn { 46 } { 32 } % .
  \char_set_lccode:nn { 123 } { 32 } % {
  \char_set_lccode:nn { 125 } { 32 } % }
  \char_set_catcode_active:N \&
\tex_lowercase:D
  {
    \group_end:
    \cs_new_protected:Npn \__msg_interrupt:n #1
      {
        \iow_term:n { }
        \__kernel_iow_with:Nnn \tex_newlinechar:D { `\^^J }
          {
            \__kernel_iow_with:Nnn \tex_errorcontextlines:D { -1 }
              {
                \group_begin:
                  \cs_set_protected:Npn &
                    {
                      \tex_errmessage:D
                        {
                          #1
                          \use_none:n
                            { ............................................ }
                        }
                    }
                  \exp_after:wN
                \group_end:
                &
              }
          }
      }
  }
\int_gset:Nn \tex_errorcontextlines:D { -1 }
\cs_new:Npn \msg_fatal_text:n #1
  {
    Fatal ~
    \msg_error_text:n {#1}
  }
\cs_new:Npn \msg_critical_text:n #1
  {
    Critical ~
    \msg_error_text:n {#1}
  }
\cs_new:Npn \msg_error_text:n #1
  { \__msg_text:nn {#1} { Error } }
\cs_new:Npn \msg_warning_text:n #1
  { \__msg_text:nn {#1} { Warning } }
\cs_new:Npn \msg_info_text:n #1
  { \__msg_text:nn {#1} { Info } }
\cs_new:Npn \__msg_text:nn #1#2
  {
    \exp_args:Nf \__msg_text:n { \msg_module_type:n {#1} }
    \exp_args:Nf \__msg_text:n { \msg_module_name:n {#1} }
    #2
  }
\cs_new:Npn \__msg_text:n #1
  {
    \tl_if_blank:nF {#1}
      { #1 ~ }
  }
\prop_new:N \g_msg_module_name_prop
\prop_new:N \g_msg_module_type_prop
\prop_gput:Nnn \g_msg_module_type_prop { LaTeX } { }
\cs_new:Npn \msg_module_type:n #1
  {
    \prop_if_in:NnTF \g_msg_module_type_prop {#1}
      { \prop_item:Nn \g_msg_module_type_prop {#1} }
      { Package }
  }
```

```
\cs_new:Npn \msg_module_name:n #1
  {
    \prop_if_in:NnTF \g_msg_module_name_prop {#1}
      { \prop_item:Nn \g_msg_module_name_prop {#1} }
      {#1}
  }
\cs_new:Npn \msg_see_documentation_text:n #1
  {
    See~the~ \msg_module_name:n {#1} ~
    documentation~for~further~information.
  }
\group_begin:
  \cs_set_protected:Npn \__msg_class_new:nn #1#2
    {
      \prop_new:c { l__msg_redirect_ #1 _prop }
      \cs_new_protected:cpn { __msg_ #1 _code:nnnnnn }
          ##1##2##3##4##5##6 {#2}
      \cs_new_protected:cpn { msg_ #1 :nnnnnn } ##1##2##3##4##5##6
        {
          \use:e
            {
              \exp_not:n { \__msg_use:nnnnnnn {#1} {##1} {##2} }
                { \tl_to_str:n {##3} } { \tl_to_str:n {##4} }
                { \tl_to_str:n {##5} } { \tl_to_str:n {##6} }
            }
        }
      \cs_new_protected:cpe { msg_ #1 :nnnnn } ##1##2##3##4##5
        { \exp_not:c { msg_ #1 :nnnnnn } {##1} {##2} {##3} {##4} {##5} { } }
      \cs_new_protected:cpe { msg_ #1 :nnnn } ##1##2##3##4
        { \exp_not:c { msg_ #1 :nnnnnn } {##1} {##2} {##3} {##4} { } { } }
      \cs_new_protected:cpe { msg_ #1 :nnn } ##1##2##3
        { \exp_not:c { msg_ #1 :nnnnnn } {##1} {##2} {##3} { } { } { } }
      \cs_new_protected:cpe { msg_ #1 :nn } ##1##2
        { \exp_not:c { msg_ #1 :nnnnnn } {##1} {##2} { } { } { } { } }
      \cs_generate_variant:cn { msg_ #1 :nnn }
        { nnV , nne , nnx }
      \cs_generate_variant:cn { msg_ #1 :nnnn }
        { nnVV , nnVn , nnnV , nnne , nnnx , nnee , nnxx }
      \cs_generate_variant:cn { msg_ #1 :nnnnn }
        { nnnee , nnnxx , nneee , nnxxx }
      \cs_generate_variant:cn { msg_ #1 :nnnnnn } { nneeee , nnxxxx }
    }
  \__msg_class_new:nn { fatal }
    {
      \__msg_interrupt:NnnnN
        \msg_fatal_text:n {#1} {#2}
        { {#3} {#4} {#5} {#6} }
        \c__msg_fatal_text_tl
      \__msg_fatal_exit:
    }
  \cs_new_protected:Npn \__msg_fatal_exit:
    {
      \tex_batchmode:D
      \tex_read:D -1 to \l__msg_internal_tl
    }
  \__msg_class_new:nn { critical }
    {
      \__msg_interrupt:NnnnN
        \msg_critical_text:n {#1} {#2}
        { {#3} {#4} {#5} {#6} }
        \c__msg_critical_text_tl
      \tex_endinput:D
    }
  \cs_undefine:N \msg_error:nnee
  \cs_undefine:N \msg_error:nne
  \cs_undefine:N \msg_error:nn
  \__msg_class_new:nn { error }
    {
      \__msg_interrupt:NnnnN
        \msg_error_text:n {#1} {#2}
        { {#3} {#4} {#5} {#6} }
        \c_empty_tl
    }
  \cs_new_protected:Npn \__msg_info_aux:NNnnnnnn #1#2#3#4#5#6#7#8
    {
      \str_set:Ne \l__msg_text_str { #2 {#3} }
      \str_set:Ne \l__msg_name_str { \msg_module_name:n {#3} }
      #1 { }
      \iow_wrap:nenN
        {
          \l__msg_text_str : ~
          \use:c { \c__msg_text_prefix_tl #3 / #4 } {#5} {#6} {#7} {#8}
        }
        {
          ( \l__msg_name_str )
          \prg_replicate:nn
            {
                \str_count:N \l__msg_text_str
              - \str_count:N \l__msg_name_str
            }
            { ~ }
        }
      { } #1
    #1 { }
  }
\__msg_class_new:nn { warning }
  {
    \__msg_info_aux:NNnnnnnn \iow_term:n \msg_warning_text:n
      {#1} {#2} {#3} {#4} {#5} {#6}
  }
\__msg_class_new:nn { note }
  {
    \__msg_info_aux:NNnnnnnn \iow_term:n \msg_info_text:n
      {#1} {#2} {#3} {#4} {#5} {#6}
  }
\__msg_class_new:nn { info }
  {
    \__msg_info_aux:NNnnnnnn \iow_log:n \msg_info_text:n
      {#1} {#2} {#3} {#4} {#5} {#6}
  }
\__msg_class_new:nn { log }
  {
    \iow_wrap:nnnN
      { \use:c { \c__msg_text_prefix_tl #1 / #2 } {#3} {#4} {#5} {#6} }
      { } { } \iow_log:n
  }
\__msg_class_new:nn { term }
  {
    \iow_wrap:nnnN
      { \use:c { \c__msg_text_prefix_tl #1 / #2 } {#3} {#4} {#5} {#6} }
      { } { } \iow_term:n
  }
\__msg_class_new:nn { none } { }
\__msg_class_new:nn { show }
  {
    \iow_wrap:nnnN
      { \use:c { \c__msg_text_prefix_tl #1 / #2 } {#3} {#4} {#5} {#6} }
      { } { } \__msg_show:n
  }
\cs_new_protected:Npn \__msg_show:n #1
  {
    \tl_if_in:nnTF { ^^J #1 } { ^^J > ~ }
      {
        \tl_if_in:nnTF { #1 \s__msg_mark } { . \s__msg_mark }
          { \__msg_show_dot:w } { \__msg_show:w }
        ^^J #1 \s__msg_stop
      }
      { \__msg_show:nn { ? #1 } { } }
    }
  \cs_new:Npn \__msg_show_dot:w #1 ^^J > ~ #2 . \s__msg_stop
    { \__msg_show:nn {#1} {#2} }
  \cs_new:Npn \__msg_show:w #1 ^^J > ~ #2 \s__msg_stop
    { \__msg_show:nn {#1} {#2} }
  \cs_new_protected:Npn \__msg_show:nn #1#2
    {
      \tl_if_empty:nF {#1}
        { \exp_args:No \iow_term:n { \use_none:n #1 } }
      \tl_set:Nn \l__msg_internal_tl {#2}
      \__kernel_iow_with:Nnn \tex_newlinechar:D { 10 }
        {
          \__kernel_iow_with:Nnn \tex_errorcontextlines:D { -1 }
            {
              \tex_showtokens:D \exp_after:wN \exp_after:wN \exp_after:wN
                { \exp_after:wN \l__msg_internal_tl }
            }
        }
    }
\group_end:
\cs_new:Npe \msg_show_item:n #1
  { ^^J > ~ \c_space_tl \exp_not:N \tl_to_str:n { {#1} } }
\cs_new:Npe \msg_show_item_unbraced:n #1
  { ^^J > ~ \c_space_tl \exp_not:N \tl_to_str:n {#1} }
\cs_new:Npe \msg_show_item:nn #1#2
  {
    ^^J > \use:nn { ~ } { ~ }
    \exp_not:N \tl_to_str:n { {#1} }
    \use:nn { ~ } { ~ } => \use:nn { ~ } { ~ }
    \exp_not:N \tl_to_str:n { {#2} }
  }
\cs_new:Npe \msg_show_item_unbraced:nn #1#2
  {
    ^^J > \use:nn { ~ } { ~ }
    \exp_not:N \tl_to_str:n {#1}
    \use:nn { ~ } { ~ } => \use:nn { ~ } { ~ }
    \exp_not:N \tl_to_str:n {#2}
  }
\cs_new:Npn \__msg_class_chk_exist:nT #1
  {
    \cs_if_free:cTF { __msg_ #1 _code:nnnnnn }
      { \msg_error:nnn { msg } { class-unknown } {#1} }
  }
\tl_new:N \l__msg_class_tl
\tl_new:N \l__msg_current_class_tl
\prop_new:N \l__msg_redirect_prop
```

```
\seq_new:N \l__msg_hierarchy_seq
\seq_new:N \l__msg_class_loop_seq
\cs_new_protected:Npn \__msg_use:nnnnnnn #1#2#3#4#5#6#7
  {
    \cs_if_exist_use:N \conditionally@traceoff
    \msg_if_exist:nnTF {#2} {#3}
      {
        \__msg_class_chk_exist:nT {#1}
          {
            \tl_set:Nn \l__msg_current_class_tl {#1}
            \cs_set_protected:Npe \__msg_use_code:
              {
                \exp_not:n
                  {
                    \use:c { __msg_ \l__msg_class_tl _code:nnnnnn }
                      {#2} {#3} {#4} {#5} {#6} {#7}
                  }
              }
            \__msg_use_redirect_name:n { #2 / #3 }
          }
      }
      { \msg_error:nnnn { msg } { unknown } {#2} {#3} }
    \cs_if_exist_use:N \conditionally@traceon
  }
\cs_new_protected:Npn \__msg_use_code: { }
\cs_new_protected:Npn \__msg_use_redirect_name:n #1
  {
    \prop_get:NnNTF \l__msg_redirect_prop { / #1 } \l__msg_class_tl
      { \__msg_use_code: }
      {
        \seq_clear:N \l__msg_hierarchy_seq
        \__msg_use_hierarchy:nwwN { }
          #1 \s__msg_mark \__msg_use_hierarchy:nwwN
          /  \s__msg_mark \__msg_use_none_delimit_by_s_stop:w
          \s__msg_stop
        \__msg_use_redirect_module:n { }
      }
  }
\cs_new_protected:Npn \__msg_use_hierarchy:nwwN #1#2 / #3 \s__msg_mark #4
  {
    \seq_put_left:Nn \l__msg_hierarchy_seq {#1}
    #4 { #1 / #2 } #3 \s__msg_mark #4
  }
\cs_new_protected:Npn \__msg_use_redirect_module:n #1
  {
    \seq_map_inline:Nn \l__msg_hierarchy_seq
```

```
      {
        \prop_get:cnNTF { l__msg_redirect_ \l__msg_current_class_tl _prop }
          {##1} \l__msg_class_tl
          {
            \seq_map_break:n
              {
                \tl_if_eq:NNTF \l__msg_current_class_tl \l__msg_class_tl
                  { \__msg_use_code: }
                  {
                    \tl_set_eq:NN \l__msg_current_class_tl \l__msg_class_tl
                    \__msg_use_redirect_module:n {##1}
                  }
              }
          }
          {
            \str_if_eq:nnT {##1} {#1}
              {
                \tl_set_eq:NN \l__msg_class_tl \l__msg_current_class_tl
                \seq_map_break:n { \__msg_use_code: }
              }
          }
      }
  }
\cs_new_protected:Npn \msg_redirect_name:nnn #1#2#3
  {
    \tl_if_empty:nTF {#3}
      { \prop_remove:Nn \l__msg_redirect_prop { / #1 / #2 } }
      {
        \__msg_class_chk_exist:nT {#3}
          { \prop_put:Nnn \l__msg_redirect_prop { / #1 / #2 } {#3} }
      }
  }
\cs_new_protected:Npn \msg_redirect_class:nn
  { \__msg_redirect:nnn { } }
\cs_new_protected:Npn \msg_redirect_module:nnn #1
  { \__msg_redirect:nnn { / #1 } }
\cs_new_protected:Npn \__msg_redirect:nnn #1#2#3
  {
    \__msg_class_chk_exist:nT {#2}
      {
        \tl_if_empty:nTF {#3}
          { \prop_remove:cn { l__msg_redirect_ #2 _prop } {#1} }
          {
            \__msg_class_chk_exist:nT {#3}
              {
                \prop_put:cnn { l__msg_redirect_ #2 _prop } {#1} {#3}
                \tl_set:Nn \l__msg_current_class_tl {#2}
                \seq_clear:N \l__msg_class_loop_seq
                \__msg_redirect_loop_chk:nnn {#2} {#3} {#1}
              }
          }
      }
  }
\cs_new_protected:Npn \__msg_redirect_loop_chk:nnn #1#2#3
  {
    \seq_put_right:Nn \l__msg_class_loop_seq {#1}
    \prop_get:cnNT { l__msg_redirect_ #1 _prop } {#3} \l__msg_class_tl
      {
        \str_if_eq:VnF \l__msg_class_tl {#1}
          {
            \tl_if_eq:NNTF \l__msg_class_tl \l__msg_current_class_tl
              {
                \prop_put:cnn { l__msg_redirect_ #2 _prop } {#3} {#2}
                \msg_warning:nneeee
                  { msg } { redirect-loop }
                  { \seq_item:Nn \l__msg_class_loop_seq { 1 } }
                  { \seq_item:Nn \l__msg_class_loop_seq { 2 } }
                  {#3}
                  {
                    \seq_map_function:NN \l__msg_class_loop_seq
                      \__msg_redirect_loop_list:n
                    { \seq_item:Nn \l__msg_class_loop_seq { 1 } }
                  }
              }
              { \__msg_redirect_loop_chk:onn \l__msg_class_tl {#2} {#3} }
          }
      }
  }
\cs_generate_variant:Nn \__msg_redirect_loop_chk:nnn { o }
\cs_new:Npn \__msg_redirect_loop_list:n #1 { {#1} ~ => ~ }
\cs_new_protected:Npn \__kernel_msg_show_eval:Nn #1#2
  { \exp_args:Nf \__msg_show_eval:nnN { #1 {#2} } {#2} \tl_show:n }
\cs_new_protected:Npn \__kernel_msg_log_eval:Nn #1#2
  { \exp_args:Nf \__msg_show_eval:nnN { #1 {#2} } {#2} \tl_log:n }
\cs_new_protected:Npn \__msg_show_eval:nnN #1#2#3 { #3 { #2 = #1 } }
\cs_new_protected:Npn \__kernel_msg_new:nnnn #1
  { \msg_new:nnnn { LaTeX / #1 } }
\cs_new_protected:Npn \__kernel_msg_new:nnn #1
  { \msg_new:nnn { LaTeX / #1 } }
\cs_new_protected:Npn \__kernel_msg_info:nnxx #1
  { \msg_info:nnee { LaTeX / #1 } }
\cs_new_protected:Npn \__kernel_msg_warning:nnx #1
```

```
  { \msg_warning:nne { LaTeX / #1 } }
\cs_new_protected:Npn \__kernel_msg_warning:nnxx #1
  { \msg_warning:nnee { LaTeX / #1 } }
\cs_new_protected:Npn \__kernel_msg_error:nnx #1
  { \msg_error:nne { LaTeX / #1 } }
\cs_new_protected:Npn \__kernel_msg_error:nnxx #1
  { \msg_error:nnee { LaTeX / #1 } }
\cs_new_protected:Npn \__kernel_msg_error:nnxxx #1
  { \msg_error:nneee { LaTeX / #1 } }
\cs_new:Npn \__kernel_msg_expandable_error:nnn #1
  { \msg_expandable_error:nnn { LaTeX / #1 } }
\cs_new:Npn \__kernel_msg_expandable_error:nnf #1
  { \msg_expandable_error:nnf { LaTeX / #1 } }
\cs_new:Npn \__kernel_msg_expandable_error:nnff #1
  { \msg_expandable_error:nnff { LaTeX / #1 } }
\msg_new:nnnn { msg } { already-defined }
  { Message~'#2'~for~module~'#1'~already~defined. }
  {
    \c__msg_coding_error_text_tl
    LaTeX~was~asked~to~define~a~new~message~called~'#2'\\
    by~the~module~'#1':~this~message~already~exists.
    \c__msg_return_text_tl
  }
\msg_new:nnnn { msg } { unknown }
  { Unknown~message~'#2'~for~module~'#1'. }
  {
    \c__msg_coding_error_text_tl
    LaTeX~was~asked~to~display~a~message~called~'#2'\\
    by~the~module~'#1':~this~message~does~not~exist.
    \c__msg_return_text_tl
  }
\msg_new:nnnn { msg } { class-unknown }
  { Unknown~message~class~'#1'. }
  {
    LaTeX~has~been~asked~to~redirect~messages~to~a~class~'#1':\\
    this~was~never~defined.
    \c__msg_return_text_tl
  }
\msg_new:nnnn { msg } { redirect-loop }
  {
    Message~redirection~loop~caused~by~ {#1} ~=>~ {#2}
    \tl_if_empty:nF {#3} { ~for~module~' \use_none:n #3 ' } .
  }
  {
    Adding~the~message~redirection~ {#1} ~=>~ {#2}
    \tl_if_empty:nF {#3} { ~for~the~module~' \use_none:n #3 ' } ~
    created~an~infinite~loop\\\\
    \iow_indent:n { #4 \\\\ }
  }
\msg_new:nnnn { kernel } { bad-number-of-arguments }
  { Function~'#1'~cannot~be~defined~with~#2~arguments. }
  {
    \c__msg_coding_error_text_tl
    LaTeX~has~been~asked~to~define~a~function~'#1'~with~
    #2~arguments.~
    TeX~allows~between~0~and~9~arguments~for~a~single~function.
  }
\msg_new:nnnn { kernel } { command-already-defined }
  { Control~sequence~#1~already~defined. }
  {
    \c__msg_coding_error_text_tl
    LaTeX~has~been~asked~to~create~a~new~control~sequence~'#1'~
    but~this~name~has~already~been~used~elsewhere. \\ \\
    The~current~meaning~is:\\
    \␣\␣#2
  }
\msg_new:nnnn { kernel } { command-not-defined }
  { Control~sequence~#1~undefined. }
  {
    \c__msg_coding_error_text_tl
    LaTeX~has~been~asked~to~use~a~control~sequence~'#1':\\
    this~has~not~been~defined~yet.
  }
\msg_new:nnnn { kernel } { empty-search-pattern }
  { Empty~search~pattern. }
  {
    \c__msg_coding_error_text_tl
    LaTeX~has~been~asked~to~replace~an~empty~pattern~by~'#1':~that~
    would~lead~to~an~infinite~loop!
  }
\msg_new:nnnn { kernel } { non-base-function }
  { Function~'#1'~is~not~a~base~function }
  {
    \c__msg_coding_error_text_tl
    Functions~defined~through~\iow_char:N\\cs_new:Nn~must~have~
    a~signature~consisting~of~only~normal~arguments~'N'~and~'n'.~
    The~signature~'#2'~of~'#1'~contains~other~arguments~'#3'.~
    To~define~variants~use~\iow_char:N\\cs_generate_variant:Nn~
    and~to~define~other~functions~use~\iow_char:N\\cs_new:Npn.
  }
\msg_new:nnnn { kernel } { missing-colon }
  { Function~'#1'~contains~no~':'. }
  {
    \c__msg_coding_error_text_tl
    Code-level~functions~must~contain~':'~to~separate~the~
    argument~specification~from~the~function~name.~This~is~
    needed~when~defining~conditionals~or~variants,~or~when~building~a~
    parameter~text~from~the~number~of~arguments~of~the~function.
  }
\msg_new:nnnn { kernel } { overflow }
  { Integers~larger~than~2^{30}-1~cannot~be~stored~in~arrays. }
  {
    An~attempt~was~made~to~store~#3~
    \tl_if_empty:nF {#2} { at~position~#2~ } in~the~array~'#1'.~
    The~largest~allowed~value~#4~will~be~used~instead.
  }
\msg_new:nnnn { kernel } { out-of-bounds }
  { Access~to~an~entry~beyond~an~array's~bounds. }
  {
    An~attempt~was~made~to~access~or~store~data~at~position~#2~of~the~
    array~'#1',~but~this~array~has~entries~at~positions~from~1~to~#3.
  }
\msg_new:nnnn { kernel } { protected-predicate }
  { Predicate~'#1'~must~be~expandable. }
  {
    \c__msg_coding_error_text_tl
    LaTeX~has~been~asked~to~define~'#1'~as~a~protected~predicate.~
    Only~expandable~tests~can~have~a~predicate~version.
  }
\msg_new:nnn { kernel } { randint-backward-range }
  { Wrong~order~of~bounds~in~\iow_char:N\\int_rand:nn{#1}{#2}. }
\msg_new:nnnn { kernel } { conditional-base-undefined }
  { Undefined~conditional~base~function~'#1'. }
  {
    \c__msg_coding_error_text_tl
    LaTeX~has~been~asked~to~define~a~variant~of~the~conditional~'#1',~
    but~the~latter~is~not~defined.
  }
\msg_new:nnnn { kernel } { conditional-form-unknown }
  { Conditional~form~'#1'~for~function~'#2'~unknown. }
  {
    \c__msg_coding_error_text_tl
    LaTeX~has~been~asked~to~define~the~conditional~form~'#1'~of~
    the~function~'#2',~but~only~'TF',~'T',~'F',~and~'p'~forms~exist.
  }
\msg_new:nnnn { kernel } { variant-too-long }
  { Variant~form~'#1'~longer~than~base~signature~of~'#2'. }
  {
    \c__msg_coding_error_text_tl
    LaTeX~has~been~asked~to~create~a~variant~of~the~function~'#2'~
    with~a~signature~starting~with~'#1',~but~that~is~longer~than~
    the~signature~(part~after~the~colon)~of~'#2'.
  }
\msg_new:nnnn { kernel } { invalid-variant }
  { Variant~form~'#1'~invalid~for~base~form~'#2'. }
  {
    \c__msg_coding_error_text_tl
    LaTeX~has~been~asked~to~create~a~variant~of~the~function~'#2'~
    with~a~signature~starting~with~'#1',~but~cannot~change~an~argument~
    from~type~'#3'~to~type~'#4'.
  }
\msg_new:nnnn { kernel } { invalid-exp-args }
  { Invalid~variant~specifier~'#1'~in~'#2'. }
  {
    \c__msg_coding_error_text_tl
    LaTeX~has~been~asked~to~create~an~\iow_char:N\\exp_args:N...~
    function~with~signature~'N#2'~but~'#1'~is~not~a~valid~argument~
    specifier.
  }
\msg_new:nnn { kernel } { deprecated-variant }
  {
    Variant~form~'#1'~deprecated~for~base~form~'#2'.~
    One~should~not~change~an~argument~from~type~'#3'~to~type~'#4'
    \str_case:nnF {#3}
      {
        { n } { :~use~a~'\token_if_eq_charcode:NNTF #4 c v V'~variant? }
        { N } { :~base~form~only~accepts~a~single~token~argument. }
        {#4} { :~base~form~is~already~a~variant. }
      } { . }
  }
\msg_new:nnn { char } { active }
  { Cannot~generate~active~chars. }
\msg_new:nnn { char } { invalid-catcode }
  { Invalid~catcode~for~char~generation. }
\msg_new:nnn { char } { null-space }
  { Cannot~generate~null~char~as~a~space. }
\msg_new:nnn { char } { out-of-range }
  { Charcode~requested~out~of~engine~range. }
\msg_new:nnn { dim } { zero-unit }
  { Zero~unit~in~conversion. }
\msg_new:nnnn { kernel } { quote-in-shell }
  { Quotes~in~shell~command~'#1'. }
  { Shell~commands~cannot~contain~quotes~("). }
\msg_new:nnnn { keys } { no-property }
```

```
  { No~property~given~in~definition~of~key~'#1'. }
  {
    \c__msg_coding_error_text_tl
    Inside~\keys_define:nn  each~key~name~
    needs~a~property:  \\ \\
    \iow_indent:n { #1 .<property> } \\ \\
    LaTeX~did~not~find~a~'.'~to~indicate~the~start~of~a~property.
  }
\msg_new:nnnn { keys } { property-boolean-values-only }
  { The~property~'#1'~accepts~boolean~values~only. }
  {
    \c__msg_coding_error_text_tl
    The~property~'#1'~only~accepts~the~values~'true'~and~'false'.
  }
\msg_new:nnnn { keys } { property-requires-value }
  { The~property~'#1'~requires~a~value. }
  {
    \c__msg_coding_error_text_tl
    LaTeX~was~asked~to~set~property~'#1'~for~key~'#2'.\\
    No~value~was~given~for~the~property,~and~one~is~required.
  }
\msg_new:nnnn { keys } { property-unknown }
  { The~key~property~'#1'~is~unknown. }
  {
    \c__msg_coding_error_text_tl
    LaTeX~has~been~asked~to~set~the~property~'#1'~for~key~'#2':~
    this~property~is~not~defined.
  }
\msg_new:nnnn { quark } { invalid-function }
  { Quark~test~function~'#1'~is~invalid. }
  {
    \c__msg_coding_error_text_tl
    LaTeX~has~been~asked~to~create~quark~test~function~'#1'~
    \tl_if_empty:nTF {#2}
      { but~that~name~ }
      { with~signature~'#2',~but~that~signature~ }
    is~not~valid.
  }
\__kernel_msg_new:nnn { quark } { invalid }
  { Invalid~quark~variable~'#1'. }
\msg_new:nnnn { scanmark } { already-defined }
  { Scan~mark~#1~already~defined. }
  {
    \c__msg_coding_error_text_tl
    LaTeX~has~been~asked~to~create~a~new~scan~mark~'#1'~
    but~this~name~has~already~been~used~for~a~scan~mark.
  }
\msg_new:nnnn { seq } { item-too-large }
  { Sequence~'#1'~does~not~have~an~item~#3 }
  {
    An~attempt~was~made~to~push~or~pop~the~item~at~position~#3~
    of~'#1',~but~this~
    \int_compare:nTF { #3 = 0 }
      { position~does~not~exist. }
      { sequence~only~has~#2~item \int_compare:nF { #2 = 1 } {s}. }
  }
\msg_new:nnnn { seq } { shuffle-too-large }
  { The~sequence~#1~is~too~long~to~be~shuffled~by~TeX. }
  {
    TeX~has~ \int_eval:n { \c_max_register_int + 1 } ~
    toks~registers:~this~only~allows~to~shuffle~up~to~
    \int_use:N \c_max_register_int \ items.~
    The~list~will~not~be~shuffled.
  }
\msg_new:nnnn { kernel } { variable-not-defined }
  { Variable~#1~undefined. }
  {
    \c__msg_coding_error_text_tl
    LaTeX~has~been~asked~to~show~a~variable~#1,~but~this~has~not~
    been~defined~yet.
  }
\msg_new:nnnn { kernel } { bad-type }
  { Variable~'#1'~is~not~a~valid~#3. }
  {
    \c__msg_coding_error_text_tl
    The~variable~'#1'~with~\tl_if_empty:nTF {#4} {meaning} {value}\\\\
    \iow_indent:n {#2}\\\\
    should~be~a~#3~variable,~but~
    \tl_if_empty:nTF {#4}
      { it~is~not \str_if_eq:nnF {#3} { bool } { ~a~short~macro } . }
      {
        it~does~not~have~the~correct~
        \str_if_eq:nnTF {#2} {#4}
          { category~codes. }
          { internal~structure:\\\\\iow_indent:n {#4} }
      }
  }
\msg_new:nnnn { prop } { bad-link }
  { Variable~'#1'~is~not~a~valid~(linked)~prop. }
  {
    \c__msg_coding_error_text_tl
    The~variable~'#1'~has~an~incorrect~internal~structure.~
    Its~internal~entry~'#2'~points~to~'#3',~whose~name~is~not~of~the~
    form~'#4~<key>'.
  }
\msg_new:nnnn { clist } { non-clist }
  { Variable~'#1'~is~not~a~valid~clist. }
  {
    \c__msg_coding_error_text_tl
    The~variable~'#1'~with~value\\\\
    \iow_indent:n {#2}\\\\
    should~be~a~clist~variable,~but~it~includes~empty~or~blank~items~
    without~braces.
  }
\msg_new:nnnn { prop } { misused }
  { A~property~list~was~misused. }
  {
    \c__msg_coding_error_text_tl
    A~property~list~variable~was~used~without~an~accessor~function.~
    It~
    \tl_if_empty:nTF {#1}
      { is~empty. }
      { contains~the~key-value~pairs \use_none:n #1 . }
  }
\msg_new:nnnn { prop } { inner-make }
  { '#1'~ cannot~ be~ used~ in~ a~ group. }
  {
    \c__msg_coding_error_text_tl
    The~ command~ '#1'~ was~ applied~ to~ the~ property~ list~
    variable~ '#2', but~ the~ storage~ type~ can~ only~ be~ changed~
    at~ the~ outermost~ group~ level.
  }
\msg_new:nnn { kernel } { bad-exp-end-f }
  { Misused~\exp_end_continue_f:w or~:nw }
\msg_new:nnn { kernel } { bad-variable }
  { Erroneous~variable~#1 used! }
\msg_new:nnn { seq } { misused }
  { A~sequence~was~misused. }
\msg_new:nnn { prg } { negative-replication }
  { Negative~argument~for~\iow_char:N\\prg_replicate:nn. }
\msg_new:nnn { prop } { prop-keyval }
  { Missing~'='~in~'#1'~(in~'..._keyval:Nn') }
\msg_new:nnn { kernel } { unknown-comparison }
  { Relation~'#1'~not~among~=,<,>,==,!=,<=,>=. }
\msg_new:nnn { kernel } { zero-step }
  { Zero~step~size~for~function~#1. }
\msg_new:nnn { clist } { show }
  {
    The~comma~list~ \tl_if_empty:nF {#1} { #1 ~ }
    \tl_if_empty:nTF {#2}
      { is~empty \\>~ . }
      { contains~the~items~(without~outer~braces): #2 . }
  }
\msg_new:nnn { intarray } { show }
  { The~integer~array~#1~contains~#2~items: \\ #3 . }
\msg_new:nnn { prop } { show }
  {
    The~ \str_if_eq:nnF {#3} { flat } { #3~ }
    property~list~#1~
    \tl_if_empty:nTF {#2}
      { is~empty \\>~ . }
      { contains~the~pairs~(without~outer~braces): #2 . }
  }
\msg_new:nnn { seq } { show }
  {
    The~sequence~#1~
    \tl_if_empty:nTF {#2}
      { is~empty \\>~ . }
      { contains~the~items~(without~outer~braces): #2 . }
  }
\msg_new:nnn { kernel } { show-streams }
  {
    \tl_if_empty:nTF {#2} { No~ } { The~following~ }
    \str_case:nn {#1}
      {
        { ior } { input ~ }
        { iow } { output ~ }
      }
    streams~are~
    \tl_if_empty:nTF {#2} { open } { in~use: #2 . }
  }
\msg_new:nnnn { sys } { backend-set }
  { Backend~configuration~already~set. }
  {
    Run-time~backend~selection~may~only~be~carried~out~once~during~a~run.~
    This~second~attempt~to~set~them~will~be~ignored.
  }
\msg_new:nnnn { sys } { load-debug-in-preamble }
  { Load~debug~support~in~the~preamble. }
  {
    Debugging~requires~support~loaded~in~the~preamble: \\
    Use~\sys_load_debug:~before~\begin{document}.
  }
\msg_new:nnnn { sys } { wrong-backend }
```

```
  { Backend~request~inconsistent~with~engine:~using~'#2'~backend. }
  {
    You~have~requested~backend~'#1',~but~this~is~not~suitable~for~use~with~the~
    active~engine.~LaTeX~will~use~the~'#2'~backend~instead.
  }
\cs_set_protected:Npn \__msg_tmp:w #1
  {
    \cs_new:Npn #1 ? { }
    \cs_new:Npn \__msg_expandable_error:nn ##1##2
      {
        \exp_after:wN \exp_after:wN
        \exp_after:wN \__msg_use_none_delimit_by_s_stop:w
        \use:n { #1 ~ ! ~ ##2 : ~ ##1 } \s__msg_stop
      }
  }
\exp_args:Nc \__msg_tmp:w { ??? }
\exp_args_generate:n { oooo }
\cs_new:Npn \msg_expandable_error:nnnnnn #1#2#3#4#5#6
  {
    \exp_args:Nee \__msg_expandable_error:nn
      {
        \exp_args:Nc \exp_args:Noooo
          { \c__msg_text_prefix_tl #1 / #2 }
          { \tl_to_str:n {#3} }
          { \tl_to_str:n {#4} }
          { \tl_to_str:n {#5} }
          { \tl_to_str:n {#6} }
      }
      { \msg_error_text:n {#1} }
  }
\cs_new:Npn \msg_expandable_error:nnnnn #1#2#3#4#5
  { \msg_expandable_error:nnnnnn {#1} {#2} {#3} {#4} {#5} { } }
\cs_new:Npn \msg_expandable_error:nnnn #1#2#3#4
  { \msg_expandable_error:nnnnnn {#1} {#2} {#3} {#4} { } { } }
\cs_new:Npn \msg_expandable_error:nnn #1#2#3
  { \msg_expandable_error:nnnnnn {#1} {#2} {#3} { } { } { } }
\cs_new:Npn \msg_expandable_error:nn #1#2
  { \msg_expandable_error:nnnnnn {#1} {#2} { } { } { } { } }
\cs_generate_variant:Nn \msg_expandable_error:nnnnnn { nnffff }
\cs_generate_variant:Nn \msg_expandable_error:nnnnn  { nnfff }
\cs_generate_variant:Nn \msg_expandable_error:nnnn   { nnff }
\cs_generate_variant:Nn \msg_expandable_error:nnn    { nnf }
\prop_gput:Nnn \g_msg_module_name_prop { kernel } { LaTeX }
\prop_gput:Nnn \g_msg_module_type_prop { kernel } { }
\clist_map_inline:nn
  {
    char , clist , coffin , debug , deprecation , dim, msg ,
    quark , prg , prop , scanmark , seq , sys
  }
  {
    \prop_gput:Nnn \g_msg_module_name_prop {#1} { LaTeX }
    \prop_gput:Nnn \g_msg_module_type_prop {#1} { }
  }
\prop_gput:Nnn \g_msg_module_name_prop { LaTeX / cmd } { LaTeX }
\prop_gput:Nnn \g_msg_module_type_prop { LaTeX / cmd } { }
\prop_gput:Nnn \g_msg_module_name_prop { LaTeX / ltcmd } { LaTeX }
\prop_gput:Nnn \g_msg_module_type_prop { LaTeX / ltcmd } { }
%% File: l3file.dtx
\tl_new:N  \l__ior_internal_tl
\int_const:Nn \c__ior_term_ior { 16 }
\seq_new:N \g__ior_streams_seq
\tl_new:N \l__ior_stream_tl
\prop_new:N \g__ior_streams_prop
\int_step_inline:nnn
  { 0 }
  {
    \cs_if_exist:NTF \contextversion
      { \tex_count:D 38 ~ }
      {
        \tex_count:D 16 ~ %
        \cs_if_exist:NT \loccount { - 1 }
      }
  }
  {
    \prop_gput:Nnn \g__ior_streams_prop {#1} { Reserved~by~format }
  }
\cs_new_protected:Npn \ior_new:N #1 { \cs_new_eq:NN #1 \c__ior_term_ior }
\cs_generate_variant:Nn \ior_new:N { c }
\ior_new:N \g_tmpa_ior
\ior_new:N \g_tmpb_ior
\cs_new_protected:Npn \ior_open:Nn #1#2
  { \ior_open:NnF #1 {#2} { \__kernel_file_missing:n {#2} } }
\cs_generate_variant:Nn \ior_open:Nn { c }
\tl_new:N \l__ior_file_name_tl
\prg_new_protected_conditional:Npnn \ior_open:Nn #1#2 { T , F , TF }
  {
    \file_get_full_name:nNTF {#2} \l__ior_file_name_tl
      {
        \__kernel_ior_open:No #1 \l__ior_file_name_tl
        \prg_return_true:
      }
      { \prg_return_false: }
  }
\prg_generate_conditional_variant:Nnn \ior_open:Nn { c } { T , F , TF }
\exp_args:NNf \cs_new_protected:Npn \__ior_new:N
  { \exp_args:NNc \exp_after:wN \exp_stop_f: { newread } }
\cs_if_exist:NT \contextversion
  {
    \cs_new_eq:NN \__ior_new_aux:N \__ior_new:N
    \cs_gset_protected:Npn \__ior_new:N #1
      {
        \cs_undefine:N #1
        \__ior_new_aux:N #1
      }
  }
\cs_new_protected:Npn \__kernel_ior_open:Nn #1#2
  {
    \ior_close:N #1
    \seq_gpop:NNTF \g__ior_streams_seq \l__ior_stream_tl
      { \__ior_open_stream:Nn #1 {#2} }
      {
        \__ior_new:N #1
        \__kernel_tl_set:Nx \l__ior_stream_tl { \int_eval:n {#1} }
        \__ior_open_stream:Nn #1 {#2}
      }
  }
\cs_generate_variant:Nn \__kernel_ior_open:Nn { No }
\cs_new_protected:Npe \__ior_open_stream:Nn #1#2
  {
    \tex_global:D \tex_chardef:D #1 = \exp_not:N \l__ior_stream_tl \scan_stop:
    \prop_gput:NVn \exp_not:N \g__ior_streams_prop #1 {#2}
    \tex_openin:D #1
      \sys_if_engine_luatex:TF
        { {#2} }
        {  \exp_not:N \__kernel_file_name_quote:n {#2} \scan_stop: }
  }
\cs_new_protected:Npn \ior_shell_open:Nn #1#2
  {
    \sys_if_shell:TF
      { \__ior_shell_open:oN { \tl_to_str:n {#2} } #1 }
      { \msg_error:nn { kernel } { pipe-failed } }
  }
\cs_new_protected:Npn \__ior_shell_open:nN #1#2
  {
    \tl_if_in:nnTF {#1} { " }
      {
        \msg_error:nne
          { kernel } { quote-in-shell } {#1}
      }
      { \__kernel_ior_open:Nn #2 { |#1 } }
  }
\cs_generate_variant:Nn \__ior_shell_open:nN { o }
\msg_new:nnnn { kernel } { pipe-failed }
  { Cannot~run~piped~system~commands. }
  {
    LaTeX~tried~to~call~a~system~process~but~this~was~not~possible.\\
    Try~the~"--shell-escape"~(or~"--enable-pipes")~option.
  }
\cs_new_protected:Npn \ior_close:N #1
  {
    \int_compare:nT { -1 < #1 < \c__ior_term_ior }
      {
        \tex_closein:D #1
        \prop_gremove:NV \g__ior_streams_prop #1
        \seq_if_in:NVF \g__ior_streams_seq #1
          { \seq_gpush:NV \g__ior_streams_seq #1 }
        \cs_gset_eq:NN #1 \c__ior_term_ior
      }
  }
\cs_generate_variant:Nn \ior_close:N { c }
\cs_new_protected:Npn \ior_show:N { \__ior_show:NN \tl_show:n }
\cs_generate_variant:Nn \ior_show:N { c }
\cs_new_protected:Npn \ior_log:N { \__ior_show:NN \tl_log:n }
\cs_generate_variant:Nn \ior_log:N { c }
\cs_new_protected:Npn \__ior_show:NN #1#2
  {
    \__kernel_chk_defined:NT #2
      {
        \prop_get:NVNTF \g__ior_streams_prop #2 \l__ior_internal_tl
          {
            \exp_args:Ne #1
              { \token_to_str:N #2 ~ open: ~ \l__ior_internal_tl }
          }
          { \exp_args:Ne #1 { \token_to_str:N #2 ~ closed } }
      }
  }
\cs_new_protected:Npn \ior_show_list: { \__ior_list:N \msg_show:nneeee }
\cs_new_protected:Npn \ior_log_list: { \__ior_list:N \msg_log:nneeee }
\cs_new_protected:Npn \__ior_list:N #1
  {
    #1 { kernel } { show-streams }
      { ior }
      {
        \prop_map_function:NN \g__ior_streams_prop
```

```
          \msg_show_item_unbraced:nn
      }
      { } { }
  }
\cs_new_eq:NN \if_eof:w \tex_ifeof:D
\prg_new_conditional:Npnn \ior_if_eof:N #1 { p , T , F , TF }
  {
    \if_int_compare:w -1 < #1
      \if_int_compare:w #1 < \c__ior_term_ior
        \if_eof:w #1
          \prg_return_true:
        \else:
          \prg_return_false:
        \fi:
      \else:
        \prg_return_true:
      \fi:
    \else:
      \prg_return_true:
    \fi:
  }
\cs_new_protected:Npn \ior_get:NN #1#2
  { \ior_get:NNF #1 #2 { \tl_set:Nn #2 { \q_no_value } } }
\cs_new_protected:Npn \__ior_get:NN #1#2
  { \tex_read:D #1 to #2 }
\prg_new_protected_conditional:Npnn \ior_get:NN #1#2 { T , F , TF }
  {
    \ior_if_eof:NTF #1
      { \prg_return_false: }
      {
        \__ior_get:NN #1 #2
        \prg_return_true:
      }
  }
\cs_new_protected:Npn \ior_str_get:NN #1#2
  { \ior_str_get:NNF #1 #2 { \tl_set:Nn #2 { \q_no_value } } }
\cs_new_protected:Npn \__ior_str_get:NN #1#2
  {
    \exp_args:Nno \use:n
      {
        \int_set:Nn \tex_endlinechar:D { -1 }
        \tex_readline:D #1 to #2
        \int_set:Nn \tex_endlinechar:D
      }   { \int_use:N \tex_endlinechar:D }
  }
\prg_new_protected_conditional:Npnn \ior_str_get:NN #1#2 { T , F , TF }
```

```
  {
    \ior_if_eof:NTF #1
      { \prg_return_false: }
      {
        \__ior_str_get:NN #1 #2
        \prg_return_true:
      }
  }
\int_const:Nn \c__ior_term_noprompt_ior { -1 }
\cs_new_protected:Npn \ior_get_term:nN #1#2
  { \__ior_get_term:NnN \__ior_get:NN {#1} #2 }
\cs_new_protected:Npn \ior_str_get_term:nN #1#2
  { \__ior_get_term:NnN \__ior_str_get:NN {#1} #2 }
\cs_new_protected:Npn \__ior_get_term:NnN #1#2#3
  {
    \group_begin:
      \tex_escapechar:D = -1 \scan_stop:
      \tl_if_blank:nTF {#2}
        { \exp_args:NNc #1 \c__ior_term_noprompt_ior }
        { \exp_args:NNc #1 \c__ior_term_ior }
          {#2}
    \exp_args:NNNv \group_end:
    \tl_set:Nn #3 {#2}
  }
\cs_new:Npn \ior_map_break:
  { \prg_map_break:Nn \ior_map_break: { } }
\cs_new:Npn \ior_map_break:n
  { \prg_map_break:Nn \ior_map_break: }
\cs_new_protected:Npn \ior_map_inline:Nn
  { \__ior_map_inline:NNn \__ior_get:NN }
\cs_new_protected:Npn \ior_str_map_inline:Nn
  { \__ior_map_inline:NNn \__ior_str_get:NN }
\cs_new_protected:Npn \__ior_map_inline:NNn
  {
    \int_gincr:N \g__kernel_prg_map_int
    \exp_args:Nc \__ior_map_inline:NNNn
      { __ior_map_ \int_use:N \g__kernel_prg_map_int :n }
  }
\cs_new_protected:Npn \__ior_map_inline:NNNn #1#2#3#4
  {
    \cs_gset_protected:Npn #1 ##1 {#4}
    \ior_if_eof:NF #3 { \__ior_map_inline_loop:NNN #1#2#3 }
    \prg_break_point:Nn \ior_map_break:
      { \int_gdecr:N \g__kernel_prg_map_int }
  }
\cs_new_protected:Npn \__ior_map_inline_loop:NNN #1#2#3
```

```
  {
    #2 #3 \l__ior_internal_tl
    \if_eof:w #3
      \exp_after:wN \ior_map_break:
    \fi:
    \exp_args:No #1 \l__ior_internal_tl
    \__ior_map_inline_loop:NNN #1#2#3
  }
\cs_new_protected:Npn \ior_map_variable:NNn
  { \__ior_map_variable:NNNn \ior_get:NN }
\cs_new_protected:Npn \ior_str_map_variable:NNn
  { \__ior_map_variable:NNNn \ior_str_get:NN }
\cs_new_protected:Npn \__ior_map_variable:NNNn #1#2#3#4
  {
    \ior_if_eof:NF #2 { \__ior_map_variable_loop:NNNn #1#2#3 {#4} }
    \prg_break_point:Nn \ior_map_break: { }
  }
\cs_new_protected:Npn \__ior_map_variable_loop:NNNn #1#2#3#4
  {
    #1 #2 #3
    \if_eof:w #2
      \exp_after:wN \ior_map_break:
    \fi:
    #4
    \__ior_map_variable_loop:NNNn #1#2#3 {#4}
  }
\tl_new:N  \l__iow_internal_tl
\int_const:Nn \c_log_iow  { -1 }
\int_const:Nn \c_term_iow
  {
    \bool_lazy_and:nnTF
      { \sys_if_engine_luatex_p: }
      { \int_compare_p:nNn \tex_luatexversion:D > { 80 } }
      { 128 }
      { 16 }
  }
\seq_new:N \g__iow_streams_seq
\tl_new:N \l__iow_stream_tl
\prop_new:N \g__iow_streams_prop
\int_step_inline:nnn
  { 0 }
  {
    \cs_if_exist:NTF \contextversion
      { \tex_count:D 39 ~ }
      {
        \tex_count:D 17 ~
        \cs_if_exist:NT \loccount { - 1 }
      }
  }
  {
    \prop_gput:Nnn \g__iow_streams_prop {#1} { Reserved~by~format }
  }
\scan_new:N \s__iow_mark
\scan_new:N \s__iow_stop
\cs_new:Npn \__iow_use_i_delimit_by_s_stop:nw #1 #2 \s__iow_stop {#1}
\quark_new:N \q__iow_nil
\cs_new_protected:Npn \iow_new:N #1 { \cs_new_eq:NN #1 \c_term_iow }
\cs_generate_variant:Nn \iow_new:N { c }
\iow_new:N \g_tmpa_iow
\iow_new:N \g_tmpb_iow
\exp_args:NNf \cs_new_protected:Npn \__iow_new:N
  { \exp_args:NNc \exp_after:wN \exp_stop_f: { newwrite } }
\cs_if_exist:NT \contextversion
  {
    \cs_new_eq:NN \__iow_new_aux:N \__iow_new:N
    \cs_gset_protected:Npn \__iow_new:N #1
      {
        \cs_undefine:N #1
        \__iow_new_aux:N #1
      }
  }
\tl_new:N \l__iow_file_name_tl
\cs_new_protected:Npn \iow_open:Nn #1#2
  {
    \__kernel_tl_set:Nx \l__iow_file_name_tl
      { \__kernel_file_name_sanitize:n {#2} }
    \__kernel_iow_open:No #1 \l__iow_file_name_tl
  }
\cs_generate_variant:Nn \iow_open:Nn { NV , c , cV }
\cs_new_protected:Npn \__kernel_iow_open:Nn #1#2
  {
    \iow_close:N #1
    \seq_gpop:NNTF \g__iow_streams_seq \l__iow_stream_tl
      { \__iow_open_stream:Nn #1 {#2} }
      {
        \__iow_new:N #1
        \__kernel_tl_set:Nx \l__iow_stream_tl { \int_eval:n {#1} }
        \__iow_open_stream:Nn #1 {#2}
      }
  }
\cs_generate_variant:Nn \__kernel_iow_open:Nn { No }
\cs_new_protected:Npn \__iow_open_stream:Nn #1#2
```

```
  {
    \tex_global:D \tex_chardef:D #1 = \l__iow_stream_tl \scan_stop:
    \prop_gput:NVn \g__iow_streams_prop #1 {#2}
    \tex_immediate:D \tex_openout:D
        #1 \__kernel_file_name_quote:n {#2} \scan_stop:
  }
\cs_generate_variant:Nn \__iow_open_stream:Nn { NV }
\cs_new_protected:Npn \iow_shell_open:Nn #1#2
  {
    \sys_if_shell:TF
      { \__iow_shell_open:oN { \tl_to_str:n {#2} } #1 }
      { \msg_error:nn { kernel } { pipe-failed } }
  }
\cs_new_protected:Npn \__iow_shell_open:nN #1#2
  {
    \tl_if_in:nnTF {#1} { " }
      {
        \msg_error:nne
          { kernel } { quote-in-shell } {#1}
      }
      { \__kernel_iow_open:Nn #2 { |#1 } }
  }
\cs_generate_variant:Nn \__iow_shell_open:nN { o }
\cs_new_protected:Npn \iow_close:N #1
  {
    \int_compare:nT { \c_log_iow < #1 < \c_term_iow }
      {
        \tex_immediate:D \tex_closeout:D #1
        \prop_gremove:NV \g__iow_streams_prop #1
        \seq_if_in:NVF \g__iow_streams_seq #1
          { \seq_gpush:NV \g__iow_streams_seq #1 }
        \cs_gset_eq:NN #1 \c_term_iow
      }
  }
\cs_generate_variant:Nn \iow_close:N { c }
\cs_new_protected:Npn \iow_show:N { \__iow_show:NN \tl_show:n }
\cs_generate_variant:Nn \iow_show:N { c }
\cs_new_protected:Npn \iow_log:N { \__iow_show:NN \tl_log:n }
\cs_generate_variant:Nn \iow_log:N { c }
\cs_new_protected:Npn \__iow_show:NN #1#2
  {
    \__kernel_chk_defined:NT #2
      {
        \prop_get:NVNTF \g__iow_streams_prop #2 \l__iow_internal_tl
          {
            \exp_args:Ne #1
```

```
              { \token_to_str:N #2 ~ open: ~ \l__iow_internal_tl }
          }
          { \exp_args:Ne #1 { \token_to_str:N #2 ~ closed } }
      }
  }
\cs_new_protected:Npn \iow_show_list: { \__iow_list:N \msg_show:nneeee }
\cs_new_protected:Npn \iow_log_list: { \__iow_list:N \msg_log:nneeee }
\cs_new_protected:Npn \__iow_list:N #1
  {
    #1 { kernel } { show-streams }
      { iow }
      {
        \prop_map_function:NN \g__iow_streams_prop
          \msg_show_item_unbraced:nn
      }
      { } { }
  }
\cs_new_protected:Npn \iow_shipout_e:Nn #1#2
  { \tex_write:D #1 {#2} }
\cs_generate_variant:Nn \iow_shipout_e:Nn { Ne , c, ce }
\cs_new_protected:Npn \iow_shipout:Nn #1#2
  { \tex_write:D #1 { \exp_not:n {#2} } }
\cs_generate_variant:Nn \iow_shipout:Nn { Ne , c, ce }
\cs_generate_variant:Nn \iow_shipout:Nn { Nx , cx }
\cs_new_protected:Npn \__kernel_iow_with:Nnn #1#2
  {
    \int_compare:nNnTF {#1} = {#2}
      { \use:n }
      { \__iow_with:oNnn { \int_use:N #1 } #1 {#2} }
  }
\cs_new_protected:Npn \__iow_with:nNnn #1#2#3#4
  {
    \int_set:Nn #2 {#3}
    #4
    \int_set:Nn #2 {#1}
  }
\cs_generate_variant:Nn \__iow_with:nNnn { o }
\cs_new_protected:Npn \iow_now:Nn #1#2
  {
    \__kernel_iow_with:Nnn \tex_newlinechar:D { `\^^J }
      { \tex_immediate:D \tex_write:D #1 { \exp_not:n {#2} } }
  }
\cs_generate_variant:Nn \iow_now:Nn { NV , Ne , c , cV , ce }
\cs_generate_variant:Nn \iow_now:Nn { Nx , cx }
\cs_new_protected:Npn \iow_log:n  { \iow_now:Nn \c_log_iow  }
\cs_set_protected:Npn \iow_log:e  { \iow_now:Ne \c_log_iow  }
```

```
\cs_generate_variant:Nn \iow_log:n { x }
\cs_new_protected:Npn \iow_term:n { \iow_now:Nn \c_term_iow }
\cs_set_protected:Npn \iow_term:e { \iow_now:Ne \c_term_iow }
\cs_generate_variant:Nn \iow_term:n { x }
\cs_new:Npn \iow_newline: { ^^J }
\cs_new_eq:NN \iow_char:N \cs_to_str:N
\int_new:N  \l_iow_line_count_int
\int_set:Nn \l_iow_line_count_int { 78 }
\tl_new:N \l__iow_newline_tl
\int_new:N \l__iow_line_target_int
\tl_new:N \l__iow_one_indent_tl
\int_new:N \l__iow_one_indent_int
\cs_new:Npn \__iow_unindent:w { }
\cs_new_protected:Npn \__iow_set_indent:n #1
  {
    \__kernel_tl_set:Nx \l__iow_one_indent_tl
      { \exp_args:No \__kernel_str_to_other_fast:n { \tl_to_str:n {#1} } }
    \int_set:Nn \l__iow_one_indent_int
      { \str_count:N \l__iow_one_indent_tl }
    \exp_last_unbraced:NNo
      \cs_set:Npn \__iow_unindent:w \l__iow_one_indent_tl { }
  }
\exp_args:Ne \__iow_set_indent:n { \prg_replicate:nn { 4 } { ~ } }
\tl_new:N \l__iow_indent_tl
\int_new:N \l__iow_indent_int
\tl_new:N \l__iow_line_tl
\tl_new:N \l__iow_line_part_tl
\bool_new:N \l__iow_line_break_bool
\tl_new:N \l__iow_wrap_tl
\group_begin:
  \int_set:Nn \tex_escapechar:D { -1 }
  \tl_const:Ne \c__iow_wrap_marker_tl
    { \tl_to_str:n { \^^I \^^O \^^W \^^_ \^^W \^^R \^^A \^^P } }
\group_end:
\tl_map_inline:nn
  { { end } { newline } { allow_break } { indent } { unindent } }
  {
    \tl_const:ce { c__iow_wrap_ #1 _marker_tl }
      {
        \c__iow_wrap_marker_tl
        #1
        \c_catcode_other_space_tl
      }
  }
\cs_new_protected:Npn \iow_wrap_allow_break:
  {
    \msg_error:nnnn { kernel } { iow-indent }
      { \iow_wrap:nnnN } { \iow_wrap_allow_break: }
  }
\cs_new:Npe \__iow_wrap_allow_break: { \c__iow_wrap_allow_break_marker_tl }
\cs_new:Npn \__iow_wrap_allow_break_error:
  {
    \msg_expandable_error:nnnn { kernel } { iow-indent }
      { \iow_wrap:nnnN } { \iow_wrap_allow_break: }
  }
\cs_new_protected:Npn \iow_indent:n #1
  {
    \msg_error:nnnnn { kernel } { iow-indent }
      { \iow_wrap:nnnN } { \iow_indent:n } {#1}
    #1
  }
\cs_new:Npe \__iow_indent:n #1
  {
    \c__iow_wrap_indent_marker_tl
    #1
    \c__iow_wrap_unindent_marker_tl
  }
\cs_new:Npn \__iow_indent_error:n #1
  {
    \msg_expandable_error:nnnnn { kernel } { iow-indent }
      { \iow_wrap:nnnN } { \iow_indent:n } {#1}
    #1
  }
\cs_new_protected:Npn \iow_wrap:nnnN #1#2#3#4
  {
    \group_begin:
      \cs_if_exist_use:N \conditionally@traceoff
      \int_set:Nn \tex_escapechar:D { -1 }
      \cs_set:Npe \{ { \token_to_str:N \{ }
      \cs_set:Npe \# { \token_to_str:N \# }
      \cs_set:Npe \} { \token_to_str:N \} }
      \cs_set:Npe \% { \token_to_str:N \% }
      \cs_set:Npe \~ { \token_to_str:N \~ }
      \int_set:Nn \tex_escapechar:D { 92 }
      \cs_set_eq:NN \\ \iow_newline:
      \cs_set_eq:NN \  \c_catcode_other_space_tl
      \cs_set_eq:NN \iow_wrap_allow_break: \__iow_wrap_allow_break:
      \cs_set_eq:NN \iow_indent:n \__iow_indent:n
      #3
      \cs_set_eq:NN \protect \token_to_str:N
      \__kernel_tl_set:Nx \l__iow_wrap_tl {#1}
      \cs_set_eq:NN \iow_wrap_allow_break: \__iow_wrap_allow_break_error:
```

```
      \cs_set_eq:NN \iow_indent:n \__iow_indent_error:n
      \__kernel_tl_set:Nx \l__iow_newline_tl { \iow_newline: #2 }
      \__kernel_tl_set:Nx \l__iow_newline_tl { \tl_to_str:N \l__iow_newline_tl }
      \int_set:Nn \l__iow_line_target_int
        { \l_iow_line_count_int - \str_count:N \l__iow_newline_tl + 1 }
      \int_compare:nNnT { \l__iow_line_target_int } < 0
        {
          \tl_set:Nn \l__iow_newline_tl { \iow_newline: }
          \int_set:Nn \l__iow_line_target_int
            { \l_iow_line_count_int + 1 }
        }
      \__iow_wrap_do:
    \exp_args:NNf \group_end:
    #4 { \tl_to_str:N \l__iow_wrap_tl }
  }
\cs_generate_variant:Nn \iow_wrap:nnnN { ne }
\cs_new_protected:Npn \__iow_wrap_do:
  {
    \__kernel_tl_set:Nx \l__iow_wrap_tl
      {
        \exp_args:No \__kernel_str_to_other_fast:n \l__iow_wrap_tl
        \c__iow_wrap_end_marker_tl
      }
    \__kernel_tl_set:Nx \l__iow_wrap_tl
      {
        \exp_after:wN \__iow_wrap_fix_newline:w \l__iow_wrap_tl
          ^^J \q__iow_nil ^^J \s__iow_stop
      }
    \exp_after:wN \__iow_wrap_start:w \l__iow_wrap_tl
  }
\cs_new:Npn \__iow_wrap_fix_newline:w #1 ^^J #2 ^^J
  {
    #1
    \if_meaning:w \q__iow_nil #2
      \__iow_use_i_delimit_by_s_stop:nw
    \fi:
    \c__iow_wrap_newline_marker_tl
    \__iow_wrap_fix_newline:w #2 ^^J
  }
\cs_new_protected:Npn \__iow_wrap_start:w
  {
    \bool_set_false:N \l__iow_line_break_bool
    \tl_clear:N \l__iow_line_tl
    \tl_clear:N \l__iow_line_part_tl
    \tl_set:Nn \l__iow_wrap_tl { ~ \use_none:n }
    \int_zero:N \l__iow_indent_int
    \tl_clear:N \l__iow_indent_tl
    \__iow_wrap_chunk:nw { \l_iow_line_count_int }
  }
\cs_new_eq:NN \__iow_sep: \__kernel_int_sep:
\cs_set_protected:Npn \__iow_tmp:w #1#2
  {
    \cs_new_protected:Npn \__iow_wrap_chunk:nw ##1##2 #2
      {
        \tl_if_empty:nTF {##2}
          {
            \tl_clear:N \l__iow_line_part_tl
            \__iow_wrap_next:nw {##1}
          }
          {
            \tl_if_empty:NTF \l__iow_line_tl
              {
                \__iow_wrap_line:nw
                  { \l__iow_indent_tl }
                  ##1 - \l__iow_indent_int \__iow_sep:
              }
              { \__iow_wrap_line:nw { } ##1 \__iow_sep: }
            ##2 #1
            \__iow_wrap_end_chunk:w 7 6 5 4 3 2 1 0 \s__iow_stop
          }
      }
    \cs_new_protected:Npn \__iow_wrap_next:nw ##1##2 #1
      { \use:c { __iow_wrap_##2:n } {##1} }
  }
\exp_args:NVV \__iow_tmp:w \c_catcode_other_space_tl \c__iow_wrap_marker_tl
\cs_new_protected:Npn \__iow_wrap_line:nw #1
  {
    \tex_edef:D \l__iow_line_part_tl { \if_false: } \fi:
    #1
    \exp_after:wN \__iow_wrap_line_loop:w
    \int_value:w \int_eval:w
  }
\cs_new:Npn \__iow_wrap_line_loop:w #1 \__iow_sep: #2#3#4#5#6#7#8#9
  {
    \if_int_compare:w #1 < 8 \exp_stop_f:
      \__iow_wrap_line_aux:Nw #1
    \fi:
    #2 #3 #4 #5 #6 #7 #8 #9
    \exp_after:wN \__iow_wrap_line_loop:w
    \int_value:w \int_eval:w #1 - 8 \__iow_sep:
  }
\cs_new:Npn \__iow_wrap_line_aux:Nw #1#2#3 \exp_after:wN #4 \__iow_sep:
  {
    #2
    \exp_after:wN \__iow_wrap_line_end:NnnnnnnnN
    \exp_after:wN #1
    \exp:w \exp_end_continue_f:w
    \exp_after:wN \exp_after:wN
    \if_case:w #1 \exp_stop_f:
         \prg_do_nothing:
    \or: \use_none:n
    \or: \use_none:nn
    \or: \use_none:nnn
    \or: \use_none:nnnn
    \or: \use_none:nnnnn
    \or: \use_none:nnnnnn
    \or: \__iow_wrap_line_seven:nnnnnnn
    \fi:
    { } { } { } { } { } { } { } #3
  }
\cs_new:Npn \__iow_wrap_line_seven:nnnnnnn #1#2#3#4#5#6#7 { \exp_stop_f: }
\cs_new:Npn \__iow_wrap_line_end:NnnnnnnnN #1#2#3#4#5#6#7#8#9
  {
    #2 #3 #4 #5 #6 #7 #8
    \use_none:nnnnn \int_eval:w 8 - \__iow_sep: #9
    \token_if_eq_charcode:NNTF \c_space_token #9
      { \__iow_wrap_line_end:nw { } }
      { \if_false: { \fi: } \__iow_wrap_break:w #9 }
  }
\cs_new:Npn \__iow_wrap_line_end:nw #1
  {
    \if_false: { \fi: }
    \__iow_wrap_store_do:n {#1}
    \__iow_wrap_next_line:w
  }
\cs_new:Npn \__iow_wrap_end_chunk:w
    #1 \int_eval:w #2 - #3 \__iow_sep: #4#5 \s__iow_stop
  {
    \if_false: { \fi: }
    \exp_args:Nf \__iow_wrap_next:nw { \int_eval:n { #2 - #4 } }
  }
\cs_set_protected:Npn \__iow_tmp:w #1
  {
    \cs_new:Npn \__iow_wrap_break:w
      {
        \tex_edef:D \l__iow_line_part_tl
          { \if_false: } \fi:
            \exp_after:wN \__iow_wrap_break_first:w
            \l__iow_line_part_tl
            #1
            { ? \__iow_wrap_break_end:w }
            \s__iow_mark
      }
    \cs_new:Npn \__iow_wrap_break_first:w ##1 #1 ##2
      {
        \use_none:nn ##2 \__iow_wrap_break_none:w
        \__iow_wrap_break_loop:w ##1 #1 ##2
      }
    \cs_new:Npn \__iow_wrap_break_none:w ##1##2 #1 ##3 \s__iow_mark ##4 #1
      {
        \tl_if_empty:NTF \l__iow_line_tl
          { ##2 ##4 \__iow_wrap_line_end:nw { } }
          { \__iow_wrap_line_end:nw { \__iow_wrap_trim:N } ##2 ##4 #1 }
      }
    \cs_new:Npn \__iow_wrap_break_loop:w ##1 #1 ##2 #1 ##3
      {
        \use_none:n ##3
        ##1 #1
        \__iow_wrap_break_loop:w ##2 #1 ##3
      }
    \cs_new:Npn \__iow_wrap_break_end:w ##1 #1 ##2 ##3 #1 ##4 \s__iow_mark
      { ##1 \__iow_wrap_line_end:nw { } ##3 }
  }
\exp_args:NV \__iow_tmp:w \c_catcode_other_space_tl
\cs_new_protected:Npn \__iow_wrap_next_line:w #1#2 \s__iow_stop
  {
    \tl_clear:N \l__iow_line_tl
    \token_if_eq_meaning:NNTF #1 \__iow_wrap_end_chunk:w
      {
        \tl_clear:N \l__iow_line_part_tl
        \bool_set_true:N \l__iow_line_break_bool
        \__iow_wrap_next:nw { \l__iow_line_target_int }
      }
      {
        \__iow_wrap_line:nw
          { \l__iow_indent_tl }
          \l__iow_line_target_int - \l__iow_indent_int \__iow_sep:
          #1 #2 \s__iow_stop
      }
  }
\cs_new_protected:Npn \__iow_wrap_allow_break:n #1
  {
    \__kernel_tl_set:Nx \l__iow_line_tl
      { \l__iow_line_tl \__iow_wrap_trim:N \l__iow_line_part_tl }
    \bool_set_false:N \l__iow_line_break_bool
    \tl_if_empty:NTF \l__iow_line_part_tl
      { \__iow_wrap_chunk:nw {#1} }
      { \exp_args:Nf \__iow_wrap_chunk:nw { \int_eval:n { #1 + 1 } } }
  }
\cs_new_protected:Npn \__iow_wrap_indent:n #1
  {
    \tl_put_right:Ne \l__iow_line_tl { \l__iow_line_part_tl }
    \bool_set_false:N \l__iow_line_break_bool
    \int_add:Nn \l__iow_indent_int { \l__iow_one_indent_int }
    \tl_put_right:No \l__iow_indent_tl { \l__iow_one_indent_tl }
    \__iow_wrap_chunk:nw {#1}
  }
\cs_new_protected:Npn \__iow_wrap_unindent:n #1
  {
    \tl_put_right:Ne \l__iow_line_tl { \l__iow_line_part_tl }
    \bool_set_false:N \l__iow_line_break_bool
    \int_sub:Nn \l__iow_indent_int { \l__iow_one_indent_int }
    \__kernel_tl_set:Nx \l__iow_indent_tl
      { \exp_after:wN \__iow_unindent:w \l__iow_indent_tl }
    \__iow_wrap_chunk:nw {#1}
  }
\cs_new_protected:Npn \__iow_wrap_newline:n #1
  {
    \bool_if:NF \l__iow_line_break_bool
      { \__iow_wrap_store_do:n { \__iow_wrap_trim:N } }
    \bool_set_false:N \l__iow_line_break_bool
    \__iow_wrap_chunk:nw { \l__iow_line_target_int }
  }
\cs_new_protected:Npn \__iow_wrap_end:n #1
  {
    \bool_if:NF \l__iow_line_break_bool
      { \__iow_wrap_store_do:n { \__iow_wrap_trim:N } }
    \bool_set_false:N \l__iow_line_break_bool
  }
\cs_new_protected:Npn \__iow_wrap_store_do:n #1
  {
    \__kernel_tl_set:Nx \l__iow_line_tl
      { \l__iow_line_tl \l__iow_line_part_tl }
    \__kernel_tl_set:Nx \l__iow_wrap_tl
      {
        \l__iow_wrap_tl
        \l__iow_newline_tl
        #1 \l__iow_line_tl
      }
    \tl_clear:N \l__iow_line_tl
  }
\cs_set_protected:Npn \__iow_tmp:w #1
  {
    \cs_new:Npn \__iow_wrap_trim:N ##1
      { \exp_after:wN \__iow_wrap_trim:w ##1 \s__iow_mark #1 \s__iow_mark \s__iow_stop }
    \cs_new:Npn \__iow_wrap_trim:w ##1 #1 \s__iow_mark
      { \__iow_wrap_trim_aux:w ##1 \s__iow_mark }
    \cs_new:Npn \__iow_wrap_trim_aux:w ##1 \s__iow_mark ##2 \s__iow_stop {##1}
  }
\exp_args:NV \__iow_tmp:w \c_catcode_other_space_tl
\tl_new:N \l__file_internal_tl
\str_new:N \g_file_curr_dir_str
\str_new:N \g_file_curr_ext_str
\str_new:N \g_file_curr_name_str
\seq_new:N \g__file_stack_seq
\group_begin:
  \cs_set_protected:Npn \__file_tmp:w #1#2#3
    {
      \tl_if_blank:nTF {#1}
        {
          \cs_set:Npn \__file_tmp:w ##1 " ##2 " ##3 \s__file_stop
            { { } {##2} {  } }
          \seq_gput_right:Ne \g__file_stack_seq
            {
              \exp_after:wN \__file_tmp:w \tex_jobname:D
                " \tex_jobname:D " \s__file_stop
            }
        }
        {
          \seq_gput_right:Nn \g__file_stack_seq { { } {#1} {#2} }
          \__file_tmp:w
        }
    }
  \cs_if_exist:NT \@currnamestack
    {
      \tl_if_empty:NF \@currnamestack
        { \exp_after:wN \__file_tmp:w \@currnamestack }
    }
\group_end:
\seq_new:N \g__file_record_seq
\tl_new:N \l__file_base_name_tl
\tl_new:N \l__file_full_name_tl
\str_new:N \l__file_dir_str
\str_new:N \l__file_ext_str
\str_new:N \l__file_name_str
\seq_new:N \l_file_search_path_seq
```

```
\seq_new:N \l__file_tmp_seq
\scan_new:N \s__file_stop
\quark_new:N \q__file_nil
\__kernel_quark_new_conditional:Nn \__file_quark_if_nil:n { TF }
\quark_new:N \q__file_recursion_tail
\quark_new:N \q__file_recursion_stop
\__kernel_quark_new_test:N \__file_if_recursion_tail_stop:N
\__kernel_quark_new_test:N \__file_if_recursion_tail_stop_do:nn
\cs_new:Npn \__kernel_file_name_sanitize:n #1
  {
    \exp_args:Ne \__file_name_trim_spaces:n
      {
        \exp_args:Ne \__file_name_strip_quotes:n
          { \__file_name_expand:n {#1} }
      }
  }
\cs_new:Npn \__file_name_expand:n #1
  {
    \exp_after:wN \__file_name_expand_cleanup:Nw
      \cs:w __file_name = #1 \cs_end:
        \__file_name_expand_end:
  }
\cs_new:Npn \__file_name_expand_cleanup:Nw #1 #2 \__file_name_expand_end:
  {
    \tl_if_empty:nF {#2}
      { \__file_name_expand_error:Nw #2 \__file_name_expand_end: }
    \exp_after:wN \__file_name_expand_cleanup:w \token_to_str:N #1
  }
\exp_last_unbraced:NNNNo
\cs_new:Npn \__file_name_expand_cleanup:w #1 \tl_to_str:n { __file_name = } { }
\cs_new:Npn \__file_name_expand_end:
  {
    \msg_expandable_error:nn
      { kernel } { filename-missing-endcsname }
    \cs_end: \__file_name_expand_end:
  }
\cs_new:Npn \__file_name_expand_error:Nw #1 #2 \__file_name_expand_end:
  { \__file_name_expand_error_aux:Nw #1 #2 \cs_end: \__file_name_expand_end: }
\cs_new:Npn \__file_name_expand_error_aux:Nw #1 #2 \cs_end: #3
    \__file_name_expand_end:
  {
    \msg_expandable_error:nnff
      { kernel } { filename-chars-lost }
        { \token_to_str:N #1 } { \exp_stop_f: #2 }
  }
\cs_new:Npn \__file_name_strip_quotes:n #1
```

```
  {
    \__file_name_strip_quotes:nw { 0 }
      #1 " \q__file_recursion_tail " \q__file_recursion_stop {#1}
  }
\cs_new:Npn \__file_name_strip_quotes:nw #1#2 "
  {
    \if_meaning:w \q__file_recursion_tail #2
      \__file_name_strip_quotes_end:wnwn
    \fi:
    #2
    \__file_name_strip_quotes:nw { #1 + 1 }
  }
\cs_new:Npn \__file_name_strip_quotes_end:wnwn \fi: #1
    \__file_name_strip_quotes:nw #2 \q__file_recursion_stop #3
  {
    \fi:
    \int_if_odd:nT {#2}
      {
        \msg_expandable_error:nnn
          { kernel } { unbalanced-quote-in-filename } {#3}
      }
  }
\cs_new:Npn \__file_name_trim_spaces:n #1
  { \__file_name_trim_spaces:nw {#1} #1 . \q__file_nil . \s__file_stop }
\cs_new:Npn \__file_name_trim_spaces:nw #1#2 . #3 . #4 \s__file_stop
  {
    \__file_quark_if_nil:nTF {#3}
      {
        \tl_trim_spaces_apply:nN { #1 \s__file_stop }
          \__file_name_trim_spaces_aux:n
      }
      { \tl_trim_spaces:n {#1} }
  }
\cs_new:Npn \__file_name_trim_spaces_aux:n #1
  { \__file_name_trim_spaces_aux:w #1 }
\cs_new:Npn \__file_name_trim_spaces_aux:w #1 \s__file_stop {#1}
\cs_new:Npn \__kernel_file_name_quote:n #1
  { \__file_name_quote:nw {#1} #1 ~ \q__file_nil \s__file_stop }
\cs_new:Npn \__file_name_quote:nw #1 #2 ~ #3 \s__file_stop
  {
    \__file_quark_if_nil:nTF {#3}
      { #1 }
      { "#1" }
  }
\tl_const:Ne \c__file_marker_tl { : \token_to_str:N : }
\cs_new_protected:Npn \file_get:nnN #1#2#3
```

```
  {
    \file_get:nnNF {#1} {#2} #3
      { \tl_set:Nn #3 { \q_no_value } }
  }
\cs_generate_variant:Nn \file_get:nnN { V }
\prg_new_protected_conditional:Npnn \file_get:nnN #1#2#3 { T , F , TF }
  {
    \file_get_full_name:nNTF {#1} \l__file_full_name_tl
      {
        \exp_args:NV \__file_get_aux:nnN
          \l__file_full_name_tl
          {#2} #3
        \prg_return_true:
      }
      { \prg_return_false: }
  }
\prg_generate_conditional_variant:Nnn \file_get:nnN { V } { T , F , TF }
\cs_new_protected:Npe \__file_get_aux:nnN #1#2#3
  {
    \exp_not:N \if_false: { \exp_not:N \fi:
    \group_begin:
      \int_set_eq:NN \tex_tracingnesting:D \c_zero_int
      \exp_not:N \exp_args:No \tex_everyeof:D
        { \exp_not:N \c__file_marker_tl }
      #2 \scan_stop:
      \exp_not:N \exp_after:wN \exp_not:N \__file_get_do:Nw
      \exp_not:N \exp_after:wN #3
      \exp_not:N \exp_after:wN \exp_not:N \prg_do_nothing:
      \exp_not:N \tex_input:D
      \sys_if_engine_luatex:TF
        { {#1} }
        { \exp_not:N \__kernel_file_name_quote:n {#1} \scan_stop: }
    \exp_not:N \if_false: } \exp_not:N \fi:
  }
\exp_args:Nno \use:nn
  { \cs_new_protected:Npn \__file_get_do:Nw #1#2 }
  { \c__file_marker_tl }
  {
    \group_end:
    \tl_set:No #1 {#2}
  }
\cs_new_eq:NN \__file_size:n \tex_filesize:D
\cs_new:Npn \file_full_name:n #1
  {
    \exp_args:Ne \__file_full_name:n
      { \__kernel_file_name_sanitize:n {#1} }
  }
\cs_generate_variant:Nn \file_full_name:n { V }
\cs_new:Npn \__file_full_name:n #1
  {
    \tl_if_blank:nF {#1}
      { \exp_args:Nne \__file_full_name_auxii:nn {#1} { \__file_full_name_aux:n {#1} } }
  }
\cs_new:Npn \__file_full_name_aux:n #1
  {
    \if_cs_exist:w __file_seen_ \tl_to_str:n {#1} : \cs_end:
      -1
    \else:
      \exp_args:Ne \__file_full_name_auxi:nn { \__file_size:n {#1} } {#1}
    \fi:
  }
\cs_new:Npn \__file_full_name_auxi:nn #1#2
  {
    \if:w \scan_stop: #1 \scan_stop:
    \else:
      \exp_after:wN \use_none:n
        \cs:w __file_seen_ \tl_to_str:n {#2} : \cs_end:
      #1
    \fi:
  }
\cs_new:Npn \__file_full_name_auxii:nn #1 #2
  {
    \tl_if_blank:nTF {#2}
      {
        \seq_map_tokens:Nn \l_file_search_path_seq
          { \__file_full_name_aux:Nnn \seq_map_break:n {#1} }
        \cs_if_exist:NT \input@path
          {
            \tl_map_tokens:Nn \input@path
              { \__file_full_name_aux:Nnn \tl_map_break:n {#1} }
          }
        \__file_name_end:
      }
      { \__file_ext_check:nn {#1} {#2} }
  }
\cs_new:Npn \__file_full_name_aux:Nnn #1#2#3
  {
    \exp_args:Ne \__file_full_name_aux:nN
      { \__file_full_name_slash:n {#3} #2 }
      #1
  }
\cs_new:Npn \__file_full_name_slash:n #1
```

```
  {
    \__file_full_name_slash:nw {#1} #1 \q_nil / \q_nil / \q_nil \q_stop
  }
\cs_new:Npn \__file_full_name_slash:nw #1#2 / \q_nil / #3 \q_stop
  {
    \quark_if_nil:nTF {#3}
      { #1 / }
      { #2 / }
  }
\cs_new:Npn \__file_full_name_aux:nN #1
  { \exp_args:Nne \__file_full_name_aux:nnN {#1} { \__file_full_name_aux:n {#1} } }
\cs_new:Npn \__file_full_name_aux:nnN #1 #2 #3
  {
    \tl_if_blank:nF {#2}
      {
        #3
          {
            \__file_ext_check:nn {#1} {#2}
            \__file_name_cleanup:w
          }
      }
  }
\cs_new:Npn \__file_name_cleanup:w #1 \__file_name_end: { }
\cs_new:Npn \__file_name_end: { }
\cs_new:Npn \__file_ext_check:nn #1 #2
{ \__file_ext_check:nnw {#2} { / } #1 / \q__file_nil / \s__file_stop }
\cs_new:Npn \__file_ext_check:nnw #1 #2 #3 / #4 / #5 \s__file_stop
  {
    \__file_quark_if_nil:nTF {#4}
      {
        \exp_args:No \__file_ext_check:nnnw
          { \use_none:n #2 } {#1} {#3} #3 . \q__file_nil . \s__file_stop
      }
      { \__file_ext_check:nnw {#1} { #2 #3 / } #4 / #5 \s__file_stop }
  }
\cs_new:Npe \__file_ext_check:nnnw #1#2#3#4 . #5 . #6 \s__file_stop
  {
    \exp_not:N \__file_quark_if_nil:nTF {#5}
      {
        \exp_not:N \__file_ext_check:nnn
          { #1 #3 \tl_to_str:n { .tex } } { #1 #3 } {#2}
      }
      { #1 #3 }
  }
\cs_new:Npn \__file_ext_check:nnn #1
  { \exp_args:Nne \__file_ext_check:nnnn {#1} { \__file_full_name_aux:n {#1} } }
```

```
\cs_new:Npn \__file_ext_check:nnnn #1#2#3#4
  {
    \tl_if_blank:nTF {#2}
      {#3}
      {
        \bool_lazy_or:nnTF
          { \int_compare_p:nNn {#4} = {#2} }
          { \int_compare_p:nNn {#2} = { -1 } }
          {#1}
          {#3}
      }
  }
\cs_new_protected:Npn \file_forget:n #1
  { \cs_undefine:c { __file_seen_ \file_full_name:n {#1} : } }
\cs_new_protected:Npn \file_get_full_name:nN #1#2
  {
    \file_get_full_name:nNF {#1} #2
      { \tl_set:Nn #2 { \q_no_value } }
  }
\cs_generate_variant:Nn \file_get_full_name:nN { V }
\prg_new_protected_conditional:Npnn \file_get_full_name:nN #1#2 { T , F , TF }
  {
    \__kernel_tl_set:Nx #2
      { \file_full_name:n {#1} }
    \tl_if_empty:NTF #2
      { \prg_return_false: }
      { \prg_return_true: }
  }
\prg_generate_conditional_variant:Nnn \file_get_full_name:nN
  { V } { T , F ,  TF }
\ior_new:N \g__file_internal_ior
\cs_new:Npn \file_size:n #1
  { \__file_details:nn {#1} { size } }
\cs_generate_variant:Nn \file_size:n { V }
\cs_new:Npn \file_timestamp:n #1
  { \__file_details:nn {#1} { moddate } }
\cs_generate_variant:Nn \file_timestamp:n { V }
\cs_new:Npn \__file_details:nn #1#2
  {
    \exp_args:Ne \__file_details_aux:nn
      { \file_full_name:n {#1} } {#2}
  }
\cs_new:Npn \__file_details_aux:nn #1#2
  {
    \tl_if_blank:nF {#1}
      { \use:c { tex_file #2 :D } {#1} }
  }
\cs_new:Npn \file_mdfive_hash:n #1
  { \exp_args:Ne \__file_mdfive_hash:n { \file_full_name:n {#1} } }
\cs_generate_variant:Nn \file_mdfive_hash:n { V }
\cs_new:Npn \__file_mdfive_hash:n #1
  { \tex_mdfivesum:D file {#1} }
\cs_new:Npn \file_hex_dump:nnn #1#2#3
  {
    \exp_args:Neee \__file_hex_dump_auxi:nnn
      { \file_full_name:n {#1} }
      { \int_eval:n {#2} }
      { \int_eval:n {#3} }
  }
\cs_generate_variant:Nn \file_hex_dump:nnn { V }
\cs_new:Npn \__file_hex_dump_auxi:nnn #1#2#3
  {
    \bool_lazy_any:nF
      {
        { \tl_if_blank_p:n {#1} }
        { \int_compare_p:nNn {#2} = 0 }
        { \int_compare_p:nNn {#3} = 0 }
      }
      {
        \exp_args:Ne \__file_hex_dump_auxii:nnnn
          { \__file_details_aux:nn {#1} { size } }
          {#1} {#2} {#3}
      }
  }
\cs_new:Npn \__file_hex_dump_auxii:nnnn #1#2#3#4
  {
    \int_compare:nNnTF {#3} > 0
      { \__file_hex_dump_auxiii:nnnn {#3} }
      {
        \exp_args:Ne \__file_hex_dump_auxiii:nnnn
          { \int_eval:n { #1 + #3 } }
      }
        {#1} {#2} {#4}
  }
\cs_new:Npn \__file_hex_dump_auxiii:nnnn #1#2#3#4
  {
    \int_compare:nNnTF {#4} > 0
      { \__file_hex_dump_auxiv:nnn {#4} }
      {
        \exp_args:Ne \__file_hex_dump_auxiv:nnn
          { \int_eval:n { #2 + #4 } }
      }
```

```
        {#1} {#3}
  }
\cs_new:Npn \__file_hex_dump_auxiv:nnn #1#2#3
  {
    \tex_filedump:D
      offset ~ \int_eval:n { #2 - 1 } ~
      length ~ \int_eval:n { #1 - #2 + 1 }
      {#3}
  }
\cs_new:Npn \file_hex_dump:n #1
  { \exp_args:Ne \__file_hex_dump:n { \file_full_name:n {#1} } }
\cs_generate_variant:Nn \file_hex_dump:n { V }
\sys_if_engine_luatex:TF
  {
    \cs_new:Npn \__file_hex_dump:n #1
      {
        \tl_if_blank:nF {#1}
          { \tex_filedump:D whole {#1} {#1} }
      }
  }
  {
    \cs_new:Npn \__file_hex_dump:n #1
      {
        \tl_if_blank:nF {#1}
          { \tex_filedump:D length \tex_filesize:D {#1} {#1} }
      }
  }
\cs_new_protected:Npn \file_get_hex_dump:nN #1#2
  { \file_get_hex_dump:nNF {#1} #2 { \tl_set:Nn #2 { \q_no_value } } }
\cs_generate_variant:Nn \file_get_hex_dump:nN { V }
\cs_new_protected:Npn \file_get_mdfive_hash:nN #1#2
  { \file_get_mdfive_hash:nNF {#1} #2 { \tl_set:Nn #2 { \q_no_value } } }
\cs_generate_variant:Nn \file_get_mdfive_hash:nN { V }
\cs_new_protected:Npn \file_get_size:nN #1#2
  { \file_get_size:nNF {#1} #2 { \tl_set:Nn #2 { \q_no_value } } }
\cs_generate_variant:Nn \file_get_size:nN { V }
\cs_new_protected:Npn \file_get_timestamp:nN #1#2
  { \file_get_timestamp:nNF {#1} #2 { \tl_set:Nn #2 { \q_no_value } } }
\cs_generate_variant:Nn \file_get_timestamp:nN { V }
\prg_new_protected_conditional:Npnn \file_get_hex_dump:nN #1#2 { T , F , TF }
  { \__file_get_details:nnN {#1} { hex_dump } #2 }
\prg_generate_conditional_variant:Nnn \file_get_hex_dump:nN
  { V } { T , F , TF }
\prg_new_protected_conditional:Npnn \file_get_mdfive_hash:nN #1#2 { T , F , TF }
  { \__file_get_details:nnN {#1} { mdfive_hash } #2 }
\prg_generate_conditional_variant:Nnn \file_get_mdfive_hash:nN
```

```
  { V } { T , F , TF }
\prg_new_protected_conditional:Npnn \file_get_size:nN #1#2 { T , F , TF }
  { \__file_get_details:nnN {#1} { size } #2 }
\prg_generate_conditional_variant:Nnn \file_get_size:nN
  { V } { T , F , TF }
\prg_new_protected_conditional:Npnn \file_get_timestamp:nN #1#2 { T , F , TF }
  { \__file_get_details:nnN {#1} { timestamp } #2 }
\prg_generate_conditional_variant:Nnn \file_get_timestamp:nN
  { V } { T , F , TF }
\cs_new_protected:Npn \__file_get_details:nnN #1#2#3
  {
    \__kernel_tl_set:Nx #3
      { \use:c { file_ #2 :n } {#1} }
    \tl_if_empty:NTF #3
      { \prg_return_false: }
      { \prg_return_true: }
  }
\cs_new_protected:Npn \file_get_hex_dump:nnnN #1#2#3#4
  {
    \file_get_hex_dump:nnnNF {#1} {#2} {#3} #4
      { \tl_set:Nn #4 { \q_no_value } }
  }
\cs_generate_variant:Nn \file_get_hex_dump:nnnN { V }
\prg_new_protected_conditional:Npnn \file_get_hex_dump:nnnN #1#2#3#4
  { T , F , TF }
  {
    \__kernel_tl_set:Nx #4
      { \file_hex_dump:nnn {#1} {#2} {#3} }
    \tl_if_empty:NTF #4
      { \prg_return_false: }
      { \prg_return_true: }
  }
\prg_generate_conditional_variant:Nnn \file_get_hex_dump:nnnN
  { V } { T , F , TF }
\cs_new_eq:NN \__file_str_cmp:nn \tex_strcmp:D
\prg_new_conditional:Npnn \file_compare_timestamp:nNn #1#2#3
  { p , T , F , TF }
  {
    \exp_args:Nee \__file_compare_timestamp:nnN
      { \file_full_name:n {#1} }
      { \file_full_name:n {#3} }
      #2
  }
\prg_generate_conditional_variant:Nnn \file_compare_timestamp:nNn
  { nNV , V , VNV } { p , T , F , TF }
\cs_new:Npn \__file_compare_timestamp:nnN #1#2#3
```

```
  {
    \tl_if_blank:nTF {#1}
      {
        \if_charcode:w #3 <
          \prg_return_true:
        \else:
          \prg_return_false:
        \fi:
      }
      {
        \tl_if_blank:nTF {#2}
          {
            \if_charcode:w #3 >
              \prg_return_true:
            \else:
              \prg_return_false:
            \fi:
          }
          {
            \if_int_compare:w
              \__file_str_cmp:nn
                { \__file_timestamp:n {#1} }
                { \__file_timestamp:n {#2} }
                #3 \c_zero_int
              \prg_return_true:
            \else:
              \prg_return_false:
            \fi:
          }
      }
  }
\cs_new_eq:NN \__file_timestamp:n \tex_filemoddate:D
\prg_new_conditional:Npnn \file_if_exist:n #1 { p , T , F , TF }
  {
    \tl_if_blank:eTF { \file_full_name:n {#1} }
      { \prg_return_false: }
      { \prg_return_true: }
  }
\prg_generate_conditional_variant:Nnn \file_if_exist:n { V } { p , T , F , TF }
\cs_new_protected:Npn \file_if_exist_input:n #1
  {
    \file_get_full_name:nNT {#1} \l__file_full_name_tl
      { \__file_input:V \l__file_full_name_tl }
  }
\cs_generate_variant:Nn \file_if_exist_input:n { V }
\cs_new_protected:Npn \file_if_exist_input:nF #1#2
```

```
  {
    \file_get_full_name:nNTF {#1} \l__file_full_name_tl
      { \__file_input:V \l__file_full_name_tl }
      {#2}
  }
\cs_generate_variant:Nn \file_if_exist_input:nF { V }
\cs_new_protected:Npn \file_input_stop: { \tex_endinput:D }
\cs_new_protected:Npn \__kernel_file_missing:n #1
  {
    \msg_error:nne { kernel } { file-not-found }
      { \__kernel_file_name_sanitize:n {#1} }
  }
\cs_new_protected:Npn \file_input:n #1
  {
    \file_get_full_name:nNTF {#1} \l__file_full_name_tl
      { \__file_input:V \l__file_full_name_tl }
      { \__kernel_file_missing:n {#1} }
  }
\cs_generate_variant:Nn \file_input:n { V }
\cs_new_protected:Npe \__file_input:n #1
  {
    \exp_not:N \clist_if_exist:NTF \exp_not:N \@filelist
      { \exp_not:N \@addtofilelist {#1} }
      { \seq_gput_right:Nn \exp_not:N \g__file_record_seq {#1} }
    \exp_not:N \__file_input_push:n {#1}
    \exp_not:N \tex_input:D
    \sys_if_engine_luatex:TF
      { {#1} }
      { \exp_not:N \__kernel_file_name_quote:n {#1} \scan_stop: }
    \exp_not:N \__file_input_pop:
  }
\cs_generate_variant:Nn \__file_input:n { V }
\cs_new_protected:Npn \__file_input_push:n #1
  {
    \seq_gpush:Ne \g__file_stack_seq
      {
        { \g_file_curr_dir_str }
        { \g_file_curr_name_str }
        { \g_file_curr_ext_str }
      }
    \file_parse_full_name:nNNN {#1}
      \l__file_dir_str \l__file_name_str \l__file_ext_str
    \str_gset_eq:NN \g_file_curr_dir_str  \l__file_dir_str
    \str_gset_eq:NN \g_file_curr_name_str \l__file_name_str
    \str_gset_eq:NN \g_file_curr_ext_str  \l__file_ext_str
  }
```

```
\cs_new_eq:NN \__kernel_file_input_push:n \__file_input_push:n
\cs_new_protected:Npn \__file_input_pop:
  {
    \seq_gpop:NN \g__file_stack_seq \l__file_internal_tl
    \exp_after:wN \__file_input_pop:nnn \l__file_internal_tl
  }
\cs_new_eq:NN \__kernel_file_input_pop: \__file_input_pop:
\cs_new_protected:Npn \__file_input_pop:nnn #1#2#3
  {
    \str_gset:Nn \g_file_curr_dir_str  {#1}
    \str_gset:Nn \g_file_curr_name_str {#2}
    \str_gset:Nn \g_file_curr_ext_str  {#3}
  }
\cs_new:Npn \file_input_raw:n #1
  { \exp_args:Ne \__file_input_raw:nn { \file_full_name:n {#1} } {#1} }
\cs_generate_variant:Nn \file_input_raw:n { V }
\cs_new:Npe \__file_input_raw:nn #1#2
  {
    \exp_not:N \tl_if_blank:nTF {#1}
      {
        \exp_not:N \exp_args:Nnne \exp_not:N \msg_expandable_error:nnn
          { kernel } { file-not-found }
          { \exp_not:N \__kernel_file_name_sanitize:n {#2} }
      }
      {
        \exp_not:N \tex_input:D
          \sys_if_engine_luatex:TF
            { {#1} }
            { \exp_not:N \__kernel_file_name_quote:n {#1} \scan_stop: }
        }
  }
\exp_args_generate:n { nne }
\cs_new:Npn \file_parse_full_name:n #1
  {
    \file_parse_full_name_apply:nN {#1}
      \prg_do_nothing:
  }
\cs_generate_variant:Nn \file_parse_full_name:n { V }
\cs_new:Npn \file_parse_full_name_apply:nN #1
  {
    \exp_args:Ne \__file_parse_full_name_auxi:nN
      { \__kernel_file_name_sanitize:n {#1} }
  }
\cs_generate_variant:Nn \file_parse_full_name_apply:nN { V }
\cs_new:Npn \__file_parse_full_name_auxi:nN #1
  {
    \__file_parse_full_name_area:nw { } #1
      / \s__file_stop
  }
\cs_new:Npn \__file_parse_full_name_area:nw #1 #2 / #3 \s__file_stop
  {
    \tl_if_empty:nTF {#3}
      { \__file_parse_full_name_base:nw { } #2 . \s__file_stop {#1} }
      { \__file_parse_full_name_area:nw { #1 / #2 } #3 \s__file_stop }
  }
\cs_new:Npn \__file_parse_full_name_base:nw #1 #2 . #3 \s__file_stop
  {
    \tl_if_empty:nTF {#3}
      {
        \tl_if_empty:nTF {#1}
          {
            \tl_if_empty:nTF {#2}
              { \__file_parse_full_name_tidy:nnnN { } { } }
              { \__file_parse_full_name_tidy:nnnN { .#2 } { } }
          }
          { \__file_parse_full_name_tidy:nnnN {#1} { .#2 } }
      }
      { \__file_parse_full_name_base:nw { #1 . #2 } #3 \s__file_stop }
  }
\cs_new:Npn \__file_parse_full_name_tidy:nnnN #1 #2 #3 #4
  {
    \exp_args:Nee #4
      {
        \str_if_eq:nnF {#3} { / } { \use_none:n }
        #3 \prg_do_nothing:
      }
      { \use_none:n #1 \prg_do_nothing: }
      {#2}
  }
\cs_new_protected:Npn \file_parse_full_name:nNNN #1 #2 #3 #4
  {
    \file_parse_full_name_apply:nN {#1}
      \__file_full_name_assign:nnnNNN #2 #3 #4
  }
\cs_new_protected:Npn \__file_full_name_assign:nnnNNN #1 #2 #3 #4 #5 #6
  {
    \str_set:Nn #4 {#1}
    \str_set:Nn #5 {#2}
    \str_set:Nn #6 {#3}
  }
\cs_generate_variant:Nn \file_parse_full_name:nNNN { V }
\cs_new_protected:Npn \file_show_list: { \__file_list:N \msg_show:nneeee }
```

```
\cs_new_protected:Npn \file_log_list: { \__file_list:N \msg_log:nneeee }
\cs_new_protected:Npn \__file_list:N #1
  {
    \seq_clear:N \l__file_tmp_seq
    \clist_if_exist:NT \@filelist
      {
        \exp_args:NNe \seq_set_from_clist:Nn \l__file_tmp_seq
          { \tl_to_str:N \@filelist }
      }
    \seq_concat:NNN \l__file_tmp_seq \l__file_tmp_seq \g__file_record_seq
    \seq_remove_duplicates:N \l__file_tmp_seq
    #1 { kernel } { file-list }
      { \seq_map_function:NN \l__file_tmp_seq \__file_list_aux:n }
        { } { } { }
  }
\cs_new:Npn \__file_list_aux:n #1 { \iow_newline: #1 }
\cs_if_exist:NT \@filelist
  {
    \AtBeginDocument
      {
        \exp_args:NNe \seq_set_from_clist:Nn \l__file_tmp_seq
          { \tl_to_str:N \@filelist }
        \seq_gconcat:NNN
          \g__file_record_seq
          \g__file_record_seq
          \l__file_tmp_seq
      }
  }
\cs_new_protected:Npn \GetIdInfo
  {
    \tl_clear_new:N \ExplFileDescription
    \tl_clear_new:N \ExplFileDate
    \tl_clear_new:N \ExplFileName
    \tl_clear_new:N \ExplFileExtension
    \tl_clear_new:N \ExplFileVersion
    \group_begin:
    \char_set_catcode_space:n { 32 }
    \exp_after:wN
    \group_end:
    \__file_id_info_auxi:w
  }
\cs_new_protected:Npn \__file_id_info_auxi:w $ #1 $ #2
  {
    \tl_set:Nn \ExplFileDescription {#2}
    \str_if_eq:nnTF {#1} { Id }
      {
        \tl_set:Nn \ExplFileDate { 0000/00/00 }
        \tl_set:Nn \ExplFileName { [unknown] }
        \tl_set:Nn \ExplFileExtension { [unknown~extension] }
        \tl_set:Nn \ExplFileVersion {-1}
      }
      { \__file_id_info_auxii:w #1 ~ \s__file_stop }
  }
\cs_new_protected:Npn \__file_id_info_auxii:w
    #1 ~ #2.#3 ~ #4 ~ #5 ~ #6 \s__file_stop
  {
    \tl_set:Nn \ExplFileName {#2}
    \tl_set:Nn \ExplFileExtension {#3}
    \tl_set:Nn \ExplFileVersion {#4}
    \str_if_eq:nnTF {#4} {-1}
      { \tl_set:Nn \ExplFileDate { 0000/00/00 } }
      { \__file_id_info_auxiii:w #5 - 0 - 0 - \s__file_stop }
  }
\cs_new_protected:Npn \__file_id_info_auxiii:w #1 - #2 - #3 - #4 \s__file_stop
  { \tl_set:Nn \ExplFileDate { #1/#2/#3 } }
\cs_new_protected:Npn \__kernel_dependency_version_check:Nn #1
  { \exp_args:NV \__kernel_dependency_version_check:nn #1 }
\cs_new_protected:Npn \__kernel_dependency_version_check:nn #1
  {
    \cs_if_exist:NTF \c__kernel_expl_date_tl
      {
        \exp_args:NV \__file_kernel_dependency_compare:nnn
          \c__kernel_expl_date_tl {#1}
      }
      { \__file_kernel_dependency_compare:nnn { 0000-00-00 } {#1} }
  }
\cs_new_protected:Npn \__file_kernel_dependency_compare:nnn #1 #2 #3
  {
    \int_compare:nNnT
        { \__file_parse_version:w #1 \s__file_stop } <
        { \__file_parse_version:w #2 \s__file_stop }
      { \__file_mismatched_dependency_error:nn {#2} {#3} }
  }
\cs_new:Npn \__file_parse_version:w #1 - #2 - #3 \s__file_stop {#1#2#3}
\cs_new_protected:Npn \__file_mismatched_dependency_error:nn #1 #2
  {
    \exp_args:NNe \ior_shell_open:Nn \g__file_internal_ior
      {
        kpsewhich ~ --all ~
          --engine = \c_sys_engine_exec_str
          \c_space_tl \c_sys_engine_format_str
            \bool_lazy_and:nnT
```

```
                { \tl_if_exist_p:N \development@branch@name }
                { ! \tl_if_empty_p:N \development@branch@name }
              { -dev } .fmt
      }
    \seq_clear:N \l__file_tmp_seq
    \ior_map_inline:Nn \g__file_internal_ior
      { \seq_put_right:Nn \l__file_tmp_seq {##1} }
    \ior_close:N \g__file_internal_ior
    \msg_error:nnnn { kernel } { mismatched-support-file }
      {#1} {#2}
    \tex_endinput:D
  }
\msg_new:nnnn { kernel } { mismatched-support-file }
  {
    Mismatched~LaTeX~support~files~detected. \\
    Loading~'#2'~aborted!
    \tl_if_exist:NT \c__kernel_expl_date_tl
      {
        \\ \\
        The~L3~programming~layer~in~the~LaTeX~format \\
        is~dated~\c__kernel_expl_date_tl,~but~in~your~TeX~
        tree~the~files~require \\ at~least~#1.
      }
  }
  {
    \int_compare:nNnTF { \seq_count:N \l__file_tmp_seq } > 1
      {
        The~cause~seems~to~be~an~old~format~file~in~the~user~tree. \\
        LaTeX~found~these~files:
        \seq_map_tokens:Nn \l__file_tmp_seq { \\~~-~\use:n } \\
        Try~deleting~the~file~in~the~user~tree~then~run~LaTeX~again.
      }
      {
        The~most~likely~causes~are:
        \\~~-~A~recent~format~generation~failed;
        \\~~-~A~stray~format~file~in~the~user~tree~which~needs~
             to~be~removed~or~rebuilt;
        \\~~-~You~are~running~a~manually~installed~version~of~#2 \\
        \␣\␣\␣which~is~incompatible~with~the~version~in~LaTeX. \\
      }
    \\
    LaTeX~will~abort~loading~the~incompatible~support~files~
    but~this~may~lead~to \\ later~errors.~Please~ensure~that~
    your~LaTeX~format~is~correctly~regenerated.
  }
\msg_new:nnnn { kernel } { file-not-found }
```

```
    { File~'#1'~not~found. }
    {
      The~requested~file~could~not~be~found~in~the~current~directory,~
      in~the~TeX~search~path~or~in~the~LaTeX~search~path.
    }
  \msg_new:nnn { kernel } { file-list }
    {
      >~File~List~<
      #1 \\
      ..............
    }
  \msg_new:nnnn { kernel } { filename-chars-lost }
    { #1~invalid~in~file~name.~Lost:~#2. }
    {
      There~was~an~invalid~token~in~the~file~name~that~caused~
      the~characters~following~it~to~be~lost.
    }
  \msg_new:nnnn { kernel } { filename-missing-endcsname }
    { Missing~\iow_char:N\\endcsname~inserted~in~filename. }
    {
      The~file~name~had~more~\iow_char:N\\csname~commands~than~
      \iow_char:N\\endcsname~ones.~LaTeX~will~add~the~missing~
      \iow_char:N\\endcsname~and~try~to~continue~as~best~as~it~can.
    }
  \msg_new:nnnn { kernel } { unbalanced-quote-in-filename }
    { Unbalanced~quotes~in~file~name~'#1'. }
    {
      File~names~must~contain~balanced~numbers~of~quotes~(").
    }
  \msg_new:nnnn { kernel } { iow-indent }
    { Only~#1 allows~#2 }
    {
      The~command~#2 can~only~be~used~in~messages~
      which~will~be~wrapped~using~#1.
      \tl_if_empty:nF {#3} { ~ It~was~called~with~argument~'#3'. }
    }
  \sys_if_engine_luatex:TF
    {
      \str_const:Ne \c_sys_platform_str
        { \tex_directlua:D { tex.print(os.type) } }
    }
    {
      \file_if_exist:nTF { nul: }
        {
          \file_if_exist:nF { /dev/null }
            { \str_const:Nn \c_sys_platform_str { windows } }
        }
        {
          \file_if_exist:nT { /dev/null }
            { \str_const:Nn \c_sys_platform_str { unix } }
        }
    }
  \cs_if_exist:NF \c_sys_platform_str
    { \str_const:Nn \c_sys_platform_str { unknown }  }
  \clist_map_inline:nn { unix , windows }
    {
      \__sys_const:nn { sys_if_platform_ #1 }
        { \str_if_eq_p:Vn \c_sys_platform_str { #1 } }
    }
  %% File: l3skip.dtx
  \cs_new_eq:NN \if_dim:w      \tex_ifdim:D
  \cs_new_eq:NN \__dim_eval:w      \tex_dimexpr:D
  \cs_new_eq:NN \__dim_eval_end:   \tex_relax:D
  \scan_new:N \s__dim_mark
  \scan_new:N \s__dim_stop
  \cs_new:Npn \__dim_use_none_delimit_by_s_stop:w #1 \s__dim_stop { }
  \cs_new_protected:Npn \dim_new:N #1
    {
      \__kernel_chk_if_free_cs:N #1
      \cs:w newdimen \cs_end: #1
    }
  \cs_generate_variant:Nn \dim_new:N { c }
  \cs_new_protected:Npn \dim_const:Nn #1#2
    {
      \dim_new:N #1
      \tex_global:D #1 = \dim_eval:n {#2} \scan_stop:
    }
  \cs_generate_variant:Nn \dim_const:Nn { c }
  \cs_new_protected:Npn \dim_zero:N #1 { #1 = \c_zero_skip }
  \cs_new_protected:Npn \dim_gzero:N #1
    { \tex_global:D #1 = \c_zero_skip }
  \cs_generate_variant:Nn \dim_zero:N  { c }
  \cs_generate_variant:Nn \dim_gzero:N { c }
  \cs_new_protected:Npn \dim_zero_new:N  #1
    { \dim_if_exist:NTF #1 { \dim_zero:N #1 } { \dim_new:N #1 } }
  \cs_new_protected:Npn \dim_gzero_new:N #1
    { \dim_if_exist:NTF #1 { \dim_gzero:N #1 } { \dim_new:N #1 } }
  \cs_generate_variant:Nn \dim_zero_new:N  { c }
  \cs_generate_variant:Nn \dim_gzero_new:N { c }
  \prg_new_eq_conditional:NNn \dim_if_exist:N \cs_if_exist:N
    { TF , T , F , p }
  \prg_new_eq_conditional:NNn \dim_if_exist:c \cs_if_exist:c
```

```
  { TF , T , F , p }
\cs_new_protected:Npn \dim_set:Nn #1#2
  { #1 = \__dim_eval:w #2 \__dim_eval_end: \scan_stop: }
\cs_new_protected:Npn \dim_gset:Nn #1#2
  { \tex_global:D #1 = \__dim_eval:w #2 \__dim_eval_end: \scan_stop: }
\cs_generate_variant:Nn \dim_set:Nn  { NV , c , cV }
\cs_generate_variant:Nn \dim_gset:Nn { NV , c , cV }
\cs_new_protected:Npn \dim_set_eq:NN #1#2
  { #1 = #2 \scan_stop: }
\cs_generate_variant:Nn \dim_set_eq:NN { c , Nc , cc }
\cs_new_protected:Npn \dim_gset_eq:NN #1#2
  { \tex_global:D #1 = #2 \scan_stop: }
\cs_generate_variant:Nn \dim_gset_eq:NN { c , Nc , cc }
\cs_new_protected:Npn \dim_add:Nn #1#2
  { \tex_advance:D #1 \__dim_eval:w #2 \__dim_eval_end: \scan_stop: }
\cs_new_protected:Npn \dim_gadd:Nn #1#2
  {
    \tex_global:D \tex_advance:D #1
      \__dim_eval:w #2 \__dim_eval_end: \scan_stop:
  }
\cs_generate_variant:Nn \dim_add:Nn  { c }
\cs_generate_variant:Nn \dim_gadd:Nn { c }
\cs_new_protected:Npn \dim_sub:Nn #1#2
  { \tex_advance:D #1 - \__dim_eval:w #2 \__dim_eval_end: \scan_stop: }
\cs_new_protected:Npn \dim_gsub:Nn #1#2
  {
    \tex_global:D \tex_advance:D #1
      -\__dim_eval:w #2 \__dim_eval_end: \scan_stop:
  }
\cs_generate_variant:Nn \dim_sub:Nn  { c }
\cs_generate_variant:Nn \dim_gsub:Nn { c }
\cs_new_eq:NN \__dim_sep: \__kernel_int_sep:
\cs_new:Npn \dim_abs:n #1
  {
    \exp_after:wN \__dim_abs:N
    \dim_use:N \__dim_eval:w #1 \__dim_eval_end:
  }
\cs_new:Npn \__dim_abs:N #1
  { \if_meaning:w - #1 \else: \exp_after:wN #1 \fi: }
\cs_new:Npn \dim_max:nn #1#2
  {
    \dim_use:N \__dim_eval:w \exp_after:wN \__dim_maxmin:wwN
      \dim_use:N \__dim_eval:w #1 \exp_after:wN \__dim_sep:
      \dim_use:N \__dim_eval:w #2 \__dim_sep:
      >
    \__dim_eval_end:
  }
\cs_new:Npn \dim_min:nn #1#2
  {
    \dim_use:N \__dim_eval:w \exp_after:wN \__dim_maxmin:wwN
      \dim_use:N \__dim_eval:w #1 \exp_after:wN \__dim_sep:
      \dim_use:N \__dim_eval:w #2 \__dim_sep:
      <
    \__dim_eval_end:
  }
\cs_new:Npn \__dim_maxmin:wwN #1 \__dim_sep: #2 \__dim_sep: #3
  {
    \if_dim:w #1 #3 #2 ~
      #1
    \else:
      #2
    \fi:
  }
\cs_new:Npn \dim_ratio:nn #1#2
  { \__dim_ratio:n {#1} / \__dim_ratio:n {#2} }
\cs_new:Npn \__dim_ratio:n #1
  { \int_value:w \__dim_eval:w (#1) \__dim_eval_end: }
\prg_new_conditional:Npnn \dim_compare:nNn #1#2#3 { p , T , F , TF }
  {
    \if_dim:w \__dim_eval:w #1 #2 \__dim_eval:w #3 \__dim_eval_end:
      \prg_return_true: \else: \prg_return_false: \fi:
  }
\prg_new_conditional:Npnn \dim_compare:n #1 { p , T , F , TF }
  {
    \exp_after:wN \__dim_compare:w
    \dim_use:N \__dim_eval:w #1 \__dim_compare_error:
  }
\cs_new:Npn \__dim_compare:w #1 \__dim_compare_error:
  {
    \exp_after:wN \if_false: \exp:w \exp_end_continue_f:w
      \__dim_compare:wNN #1 ? { = \__dim_compare_end:w \else: } \s__dim_stop
  }
\exp_args:Nno \use:nn
  { \cs_new:Npn \__dim_compare:wNN #1 } { \tl_to_str:n {pt} #2#3 }
  {
      \if_meaning:w = #3
        \use:c { __dim_compare_#2:w }
      \fi:
        #1 pt \exp_stop_f:
      \prg_return_false:
      \exp_after:wN \__dim_use_none_delimit_by_s_stop:w
    \fi:
```

```
    \reverse_if:N \if_dim:w #1 pt #2
      \exp_after:wN \__dim_compare:wNN
      \dim_use:N \__dim_eval:w #3
  }
\cs_new:cpn { __dim_compare_ ! :w }
    #1 \reverse_if:N #2 ! #3 = { #1 #2 = #3 }
\cs_new:cpn { __dim_compare_ = :w }
    #1 \__dim_eval:w = { #1 \__dim_eval:w }
\cs_new:cpn { __dim_compare_ < :w }
    #1 \reverse_if:N #2 < #3 = { #1 #2 > #3 }
\cs_new:cpn { __dim_compare_ > :w }
    #1 \reverse_if:N #2 > #3 = { #1 #2 < #3 }
\cs_new:Npn \__dim_compare_end:w #1 \prg_return_false: #2 \s__dim_stop
  { #1 \prg_return_false: \else: \prg_return_true: \fi: }
\cs_new_protected:Npn \__dim_compare_error:
  {
    \if_int_compare:w \c_zero_int \c_zero_int \fi:
    =
    \__dim_compare_error:
  }
\cs_new:Npn \dim_case:nnTF #1
  {
    \exp:w
    \exp_args:Nf \__dim_case:nnTF { \dim_eval:n {#1} }
  }
\cs_new:Npn \dim_case:nnT #1#2#3
  {
    \exp:w
    \exp_args:Nf \__dim_case:nnTF { \dim_eval:n {#1} } {#2} {#3} { }
  }
\cs_new:Npn \dim_case:nnF #1#2
  {
    \exp:w
    \exp_args:Nf \__dim_case:nnTF { \dim_eval:n {#1} } {#2} { }
  }
\cs_new:Npn \dim_case:nn #1#2
  {
    \exp:w
    \exp_args:Nf \__dim_case:nnTF { \dim_eval:n {#1} } {#2} { } { }
  }
\cs_new:Npn \__dim_case:nnTF #1#2#3#4
  { \__dim_case:nw {#1} #2 {#1} { } \s__dim_mark {#3} \s__dim_mark {#4} \s__dim_stop }
\cs_new:Npn \__dim_case:nw #1#2#3
  {
    \dim_compare:nNnTF {#1} = {#2}
      { \__dim_case_end:nw {#3} }
      { \__dim_case:nw {#1} }
  }
\cs_new:Npn \__dim_case_end:nw #1#2#3 \s__dim_mark #4#5 \s__dim_stop
  { \exp_end: #1 #4 }
\cs_new:Npn \dim_while_do:nn #1#2
  {
    \dim_compare:nT {#1}
      {
        #2
        \dim_while_do:nn {#1} {#2}
      }
  }
\cs_new:Npn \dim_until_do:nn #1#2
  {
    \dim_compare:nF {#1}
      {
        #2
        \dim_until_do:nn {#1} {#2}
      }
  }
\cs_new:Npn \dim_do_while:nn #1#2
  {
    #2
    \dim_compare:nT {#1}
      { \dim_do_while:nn {#1} {#2} }
  }
\cs_new:Npn \dim_do_until:nn #1#2
  {
    #2
    \dim_compare:nF {#1}
      { \dim_do_until:nn {#1} {#2} }
  }
\cs_new:Npn \dim_while_do:nNnn #1#2#3#4
  {
    \dim_compare:nNnT {#1} #2 {#3}
      {
        #4
        \dim_while_do:nNnn {#1} #2 {#3} {#4}
      }
  }
\cs_new:Npn \dim_until_do:nNnn #1#2#3#4
  {
  \dim_compare:nNnF {#1} #2 {#3}
    {
      #4
      \dim_until_do:nNnn {#1} #2 {#3} {#4}
    }
  }
\cs_new:Npn \dim_do_while:nNnn #1#2#3#4
  {
    #4
    \dim_compare:nNnT {#1} #2 {#3}
      { \dim_do_while:nNnn {#1} #2 {#3} {#4} }
  }
\cs_new:Npn \dim_do_until:nNnn #1#2#3#4
  {
    #4
    \dim_compare:nNnF {#1} #2 {#3}
      { \dim_do_until:nNnn {#1} #2 {#3} {#4} }
  }
\cs_new:Npn \dim_step_function:nnnN #1#2#3
  {
    \exp_after:wN \__dim_step:wwwN
    \tex_the:D \__dim_eval:w #1 \exp_after:wN \__dim_sep:
    \tex_the:D \__dim_eval:w #2 \exp_after:wN \__dim_sep:
    \tex_the:D \__dim_eval:w #3 \__dim_sep:
  }
\cs_new:Npn \__dim_step:wwwN #1 \__dim_sep: #2 \__dim_sep: #3 \__dim_sep: #4
  {
    \dim_compare:nNnTF {#2} > \c_zero_dim
      { \__dim_step:NnnnN > }
      {
        \dim_compare:nNnTF {#2} = \c_zero_dim
          {
            \msg_expandable_error:nnn { kernel } { zero-step } {#4}
            \use_none:nnnn
          }
          { \__dim_step:NnnnN < }
      }
      {#1} {#2} {#3} #4
  }
\cs_new:Npn \__dim_step:NnnnN #1#2#3#4#5
  {
    \dim_compare:nNnF {#2} #1 {#4}
      {
        #5 {#2}
        \exp_args:NNf \__dim_step:NnnnN
          #1 { \dim_eval:n { #2 + #3 } } {#3} {#4} #5
      }
  }
\cs_new_protected:Npn \dim_step_inline:nnnn
  {
    \int_gincr:N \g__kernel_prg_map_int
    \exp_args:NNc \__dim_step:NNnnnn
      \cs_gset_protected:Npn
      { __dim_map_ \int_use:N \g__kernel_prg_map_int :w }
  }
\cs_new_protected:Npn \dim_step_variable:nnnNn #1#2#3#4#5
  {
    \int_gincr:N \g__kernel_prg_map_int
    \exp_args:NNc \__dim_step:NNnnnn
      \cs_gset_protected:Npe
      { __dim_map_ \int_use:N \g__kernel_prg_map_int :w }
      {#1}{#2}{#3}
      {
        \tl_set:Nn \exp_not:N #4 {##1}
        \exp_not:n {#5}
      }
  }
\cs_new_protected:Npn \__dim_step:NNnnnn #1#2#3#4#5#6
  {
    #1 #2 ##1 {#6}
    \dim_step_function:nnnN {#3} {#4} {#5} #2
    \prg_break_point:Nn \scan_stop: { \int_gdecr:N \g__kernel_prg_map_int }
  }
\cs_new:Npn \dim_eval:n #1
  { \dim_use:N \__dim_eval:w #1 \__dim_eval_end: }
\cs_new:Npn \dim_sign:n #1
  {
    \int_value:w \exp_after:wN \__dim_sign:Nw
      \dim_use:N \__dim_eval:w #1 \__dim_eval_end: \__dim_sep:
    \exp_stop_f:
  }
\cs_new:Npn \__dim_sign:Nw #1#2 \__dim_sep:
  {
    \if_dim:w #1#2 > \c_zero_dim
      1
    \else:
      \if_meaning:w - #1
        -1
      \else:
        0
      \fi:
    \fi:
  }
\cs_new_eq:NN \dim_use:N \tex_the:D
\cs_new:Npn \dim_use:c #1 { \tex_the:D \cs:w #1 \cs_end: }
\cs_new:Npn \dim_to_decimal:n #1
```

```
  {
    \exp_after:wN
      \__dim_to_decimal:w \dim_use:N \__dim_eval:w #1 \__dim_eval_end:
  }
\use:e
  {
    \cs_new:Npn \exp_not:N \__dim_to_decimal:w
      #1 . #2 \tl_to_str:n { pt }
  }
      {
        \int_compare:nNnTF {#2} > \c_zero_int
          { #1 . #2 }
          { #1 }
      }
\cs_new:Npn \dim_to_decimal_in_sp:n #1
  { \int_value:w \__dim_eval:w #1 \__dim_eval_end: }
\group_begin:
  \cs_set_protected:Npn \__dim_tmp:w #1#2
    {
      \cs_new:cpn { dim_to_decimal_in_ #1 :n } ##1
        {
          \exp_after:wN \__dim_to_decimal_aux:w
            \int_value:w \__dim_eval:w ##1 \__dim_eval_end: \__dim_sep: #2 \__dim_sep:
        }
    }
  \__dim_tmp:w { in } {   50 /  7227 } % delta = 7227/100
  \__dim_tmp:w { pc } {    1 /    24 } % delta = 12/1
  \__dim_tmp:w { cm } {  127 /  7227 } % delta = 7227/254
  \__dim_tmp:w { mm } { 1270 /  7227 } % delta = 7227/2540
  \__dim_tmp:w { bp } {  400 /   803 } % delta = 7227/7200
  \__dim_tmp:w { dd } { 1157 /  2476 } % delta = 1238/1157
  \__dim_tmp:w { cc } { 1157 / 29712 } % delta = 14856/1157
\group_end:
\cs_new:Npn \__dim_to_decimal_aux:w #1#2 \__dim_sep: #3 \__dim_sep:
  {
    \dim_to_decimal:n
      {
        \int_eval:n
          { ( 2 * #1#2 \if:w #1 - - \else: + \fi: 1 ) * #3 }
        sp
      }
  }
\cs_new:Npn \dim_to_decimal_in_unit:nn #1#2
  {
    \exp_after:wN \__dim_chk_unit:w
      \int_value:w \__dim_eval:w #2 \__dim_eval_end: \__dim_sep: {#1}
  }
\cs_new:Npn \__dim_chk_unit:w #1#2 \__dim_sep: #3
  {
    \token_if_eq_charcode:NNTF #1 0
      { \msg_expandable_error:nn { dim } { zero-unit } }
      {
        \exp_after:wN \__dim_branch_unit:w
          \int_value:w \if:w #1 - - \fi: \__dim_eval:w #3 \exp_after:wN \__dim_sep:
          \int_value:w \if:w #1 - - \fi: #1#2 \__dim_sep:
      }
  }
\cs_new:Npn \__dim_branch_unit:w #1 \__dim_sep: #2 \__dim_sep:
  {
    \int_compare:nNnTF {#2} > { 65536 }
      { \__dim_to_decimal_aux:w #1 \__dim_sep: 32768 / #2 \__dim_sep: }
      {
        \int_compare:nNnTF {#2} = { 65536 }
          { \dim_to_decimal:n { #1sp } }
          { \__dim_get_quotient:w #1 \__dim_sep: #2 \__dim_sep: }
      }
  }
\cs_new:Npn \__dim_get_quotient:w #1#2 \__dim_sep: #3 \__dim_sep:
  {
    \token_if_eq_charcode:NNTF #1 0
      { 0 }
      {
        \token_if_eq_charcode:NNTF #1 -
          {
            \exp_after:wN \exp_after:wN \exp_after:wN \__dim_get_remainder:w
              \int_eval:n {  ( 2 * #2 - #3 ) / ( 2 * #3 ) } \__dim_sep:
              #2 \__dim_sep: #3 \__dim_sep: - \__dim_sep:
          }
          {
            \exp_after:wN \exp_after:wN \exp_after:wN \__dim_get_remainder:w
              \int_eval:n { ( 2 * #1#2 - #3 ) / ( 2 * #3 ) } \__dim_sep:
              #1#2 \__dim_sep: #3 \__dim_sep: \__dim_sep:
          }
      }
  }
\cs_new:Npn \__dim_get_remainder:w #1 \__dim_sep: #2 \__dim_sep: #3 \__dim_sep:
  {
    \exp_after:wN \exp_after:wN \exp_after:wN \__dim_convert_remainder:w
      \int_eval:n { #2 - #1 * #3 } \__dim_sep:
      #3 \__dim_sep: #1 \__dim_sep:
  }
\cs_new:Npn \__dim_convert_remainder:w #1 \__dim_sep: #2 \__dim_sep:
  {
    \exp_after:wN \exp_after:wN \exp_after:wN \__dim_test_candidate:w
      \int_eval:n { #1 * 65536 / #2 } \__dim_sep:
      #1 \__dim_sep: #2 \__dim_sep:
  }
\cs_new:Npn \__dim_test_candidate:w #1 \__dim_sep: #2 \__dim_sep: #3 \__dim_sep:
  {
    \dim_compare:nNnTF { #2sp } =
      { \dim_to_decimal:n { #1sp } \__dim_eval:w #3sp \__dim_eval_end: }
      { \__dim_parse_decimal:w #1 \__dim_sep: }
      {
        \__dim_parse_decimal:w \int_eval:n { #1 + 1 } \__dim_sep:
      }
  }
\cs_new:Npn \__dim_parse_decimal:w #1 \__dim_sep: #2 \__dim_sep: #3 \__dim_sep:
  {
    \exp_after:wN \__dim_parse_decimal_aux:w
      \int_value:w #3 \int_eval:w #2 + \dim_to_decimal:n { #1sp } \__dim_sep:
  }
\cs_new:Npn \__dim_parse_decimal_aux:w #1 \__dim_sep: {#1}
\cs_new_eq:NN  \dim_show:N \__kernel_register_show:N
\cs_generate_variant:Nn \dim_show:N { c }
\cs_new_protected:Npn \dim_show:n
  { \__kernel_msg_show_eval:Nn \dim_eval:n }
\cs_new_eq:NN \dim_log:N \__kernel_register_log:N
\cs_new_eq:NN \dim_log:c \__kernel_register_log:c
\cs_new_protected:Npn \dim_log:n
  { \__kernel_msg_log_eval:Nn \dim_eval:n }
\dim_const:Nn \c_zero_dim { 0 pt }
\dim_const:Nn \c_max_dim { 16383.99999 pt }
\dim_new:N \l_tmpa_dim
\dim_new:N \l_tmpb_dim
\dim_new:N \g_tmpa_dim
\dim_new:N \g_tmpb_dim
\scan_new:N \s__skip_stop
\cs_new_protected:Npn \skip_new:N #1
  {
    \__kernel_chk_if_free_cs:N #1
    \cs:w newskip \cs_end: #1
  }
\cs_generate_variant:Nn \skip_new:N { c }
\cs_new_protected:Npn \skip_const:Nn #1#2
  {
    \skip_new:N #1
    \tex_global:D #1 = \skip_eval:n {#2} \scan_stop:
  }
```

```
\cs_generate_variant:Nn \skip_const:Nn { c }
\cs_new_eq:NN \skip_zero:N  \dim_zero:N
\cs_new_eq:NN \skip_gzero:N \dim_gzero:N
\cs_generate_variant:Nn \skip_zero:N  { c }
\cs_generate_variant:Nn \skip_gzero:N { c }
\cs_new_protected:Npn \skip_zero_new:N  #1
  { \skip_if_exist:NTF #1 { \skip_zero:N #1 } { \skip_new:N #1 } }
\cs_new_protected:Npn \skip_gzero_new:N #1
  { \skip_if_exist:NTF #1 { \skip_gzero:N #1 } { \skip_new:N #1 } }
\cs_generate_variant:Nn \skip_zero_new:N  { c }
\cs_generate_variant:Nn \skip_gzero_new:N { c }
\prg_new_eq_conditional:NNn \skip_if_exist:N \cs_if_exist:N
  { TF , T , F , p }
\prg_new_eq_conditional:NNn \skip_if_exist:c \cs_if_exist:c
  { TF , T , F , p }
\cs_new_protected:Npn \skip_set:Nn #1#2
  { #1 = \tex_glueexpr:D #2 \scan_stop: }
\cs_new_protected:Npn \skip_gset:Nn #1#2
  { \tex_global:D #1 = \tex_glueexpr:D #2 \scan_stop: }
\cs_generate_variant:Nn \skip_set:Nn  { NV , c , cV }
\cs_generate_variant:Nn \skip_gset:Nn { NV , c , cV }
\cs_new_protected:Npn \skip_set_eq:NN #1#2 { #1 = #2 }
\cs_generate_variant:Nn \skip_set_eq:NN { c , Nc , cc }
\cs_new_protected:Npn \skip_gset_eq:NN #1#2 { \tex_global:D #1 = #2 }
\cs_generate_variant:Nn \skip_gset_eq:NN { c , Nc , cc }
\cs_new_protected:Npn \skip_add:Nn #1#2
  { \tex_advance:D #1 \tex_glueexpr:D #2 \scan_stop: }
\cs_new_protected:Npn \skip_gadd:Nn #1#2
  { \tex_global:D \tex_advance:D #1 \tex_glueexpr:D #2 \scan_stop: }
\cs_generate_variant:Nn \skip_add:Nn  { c }
\cs_generate_variant:Nn \skip_gadd:Nn { c }
\cs_new_protected:Npn \skip_sub:Nn #1#2
  { \tex_advance:D #1 - \tex_glueexpr:D #2 \scan_stop: }
\cs_new_protected:Npn \skip_gsub:Nn #1#2
  { \tex_global:D \tex_advance:D #1 - \tex_glueexpr:D #2 \scan_stop: }
\cs_generate_variant:Nn \skip_sub:Nn  { c }
\cs_generate_variant:Nn \skip_gsub:Nn { c }
\prg_new_conditional:Npnn \skip_if_eq:nn #1#2 { p , T , F , TF }
  {
    \str_if_eq:eeTF { \skip_eval:n {#1} } { \skip_eval:n {#2} }
      { \prg_return_true: }
      { \prg_return_false: }
  }
\cs_new_eq:NN \__skip_sep: \__kernel_int_sep:
\cs_set_protected:Npn \__skip_tmp:w #1
  {
    \prg_new_conditional:Npnn \skip_if_finite:n ##1 { p , T , F , TF }
      {
        \exp_after:wN \__skip_if_finite:wwNw
        \skip_use:N \tex_glueexpr:D ##1 \__skip_sep: \prg_return_false:
        #1 \__skip_sep: \prg_return_true: \s__skip_stop
      }
    \cs_new:Npn \__skip_if_finite:wwNw ##1 #1 ##2 \__skip_sep: ##3 ##4 \s__skip_stop {##3}
  }
\exp_args:No \__skip_tmp:w { \tl_to_str:n { fil } }
\cs_new:Npn \skip_eval:n #1
  { \skip_use:N \tex_glueexpr:D #1 \scan_stop: }
\cs_new_eq:NN \skip_use:N \dim_use:N
\cs_new_eq:NN \skip_use:c \dim_use:c
\cs_new_eq:NN  \skip_horizontal:N \tex_hskip:D
\cs_new:Npn \skip_horizontal:n #1
  { \skip_horizontal:N \tex_glueexpr:D #1 \scan_stop: }
\cs_new_eq:NN  \skip_vertical:N \tex_vskip:D
\cs_new:Npn \skip_vertical:n #1
  { \skip_vertical:N \tex_glueexpr:D #1 \scan_stop: }
\cs_generate_variant:Nn \skip_horizontal:N { c }
\cs_generate_variant:Nn \skip_vertical:N { c }
\cs_new_eq:NN  \skip_show:N \__kernel_register_show:N
\cs_generate_variant:Nn \skip_show:N { c }
\cs_new_protected:Npn \skip_show:n
  { \__kernel_msg_show_eval:Nn \skip_eval:n }
\cs_new_eq:NN \skip_log:N \__kernel_register_log:N
\cs_new_eq:NN \skip_log:c \__kernel_register_log:c
\cs_new_protected:Npn \skip_log:n
  { \__kernel_msg_log_eval:Nn \skip_eval:n }
\skip_const:Nn \c_zero_skip { \c_zero_dim }
\skip_const:Nn \c_max_skip { \c_max_dim }
\skip_new:N \l_tmpa_skip
\skip_new:N \l_tmpb_skip
\skip_new:N \g_tmpa_skip
\skip_new:N \g_tmpb_skip
\cs_new_protected:Npn \muskip_new:N #1
  {
    \__kernel_chk_if_free_cs:N #1
    \cs:w newmuskip \cs_end: #1
  }
\cs_generate_variant:Nn \muskip_new:N { c }
\cs_new_protected:Npn \muskip_const:Nn #1#2
  {
    \muskip_new:N #1
    \tex_global:D #1 = \muskip_eval:n {#2} \scan_stop:
  }
\cs_generate_variant:Nn \muskip_const:Nn { c }
\cs_new_protected:Npn \muskip_zero:N #1
  { #1 = \c_zero_muskip }
\cs_new_protected:Npn \muskip_gzero:N #1
  { \tex_global:D #1 = \c_zero_muskip }
\cs_generate_variant:Nn \muskip_zero:N  { c }
\cs_generate_variant:Nn \muskip_gzero:N { c }
\cs_new_protected:Npn \muskip_zero_new:N  #1
  { \muskip_if_exist:NTF #1 { \muskip_zero:N #1 } { \muskip_new:N #1 } }
\cs_new_protected:Npn \muskip_gzero_new:N #1
  { \muskip_if_exist:NTF #1 { \muskip_gzero:N #1 } { \muskip_new:N #1 } }
\cs_generate_variant:Nn \muskip_zero_new:N  { c }
\cs_generate_variant:Nn \muskip_gzero_new:N { c }
\prg_new_eq_conditional:NNn \muskip_if_exist:N \cs_if_exist:N
  { TF , T , F , p }
\prg_new_eq_conditional:NNn \muskip_if_exist:c \cs_if_exist:c
  { TF , T , F , p }
\cs_new_protected:Npn \muskip_set:Nn #1#2
  { #1 = \tex_muexpr:D #2 \scan_stop: }
\cs_new_protected:Npn \muskip_gset:Nn #1#2
  { \tex_global:D #1 = \tex_muexpr:D #2 \scan_stop: }
\cs_generate_variant:Nn \muskip_set:Nn  { NV , c , cV }
\cs_generate_variant:Nn \muskip_gset:Nn { NV , c , cV }
\cs_new_protected:Npn \muskip_set_eq:NN #1#2 { #1 = #2 }
\cs_generate_variant:Nn \muskip_set_eq:NN { c , Nc , cc }
\cs_new_protected:Npn \muskip_gset_eq:NN #1#2 { \tex_global:D #1 = #2 }
\cs_generate_variant:Nn \muskip_gset_eq:NN { c , Nc , cc }
\cs_new_protected:Npn \muskip_add:Nn #1#2
  { \tex_advance:D #1 \tex_muexpr:D #2 \scan_stop: }
\cs_new_protected:Npn \muskip_gadd:Nn #1#2
  { \tex_global:D \tex_advance:D #1 \tex_muexpr:D #2 \scan_stop: }
\cs_generate_variant:Nn \muskip_add:Nn  { c }
\cs_generate_variant:Nn \muskip_gadd:Nn { c }
\cs_new_protected:Npn \muskip_sub:Nn #1#2
  { \tex_advance:D #1 - \tex_muexpr:D #2 \scan_stop: }
\cs_new_protected:Npn \muskip_gsub:Nn #1#2
  { \tex_global:D \tex_advance:D #1 - \tex_muexpr:D #2 \scan_stop: }
\cs_generate_variant:Nn \muskip_sub:Nn  { c }
\cs_generate_variant:Nn \muskip_gsub:Nn { c }
\cs_new:Npn \muskip_eval:n #1
  { \muskip_use:N \tex_muexpr:D #1 \scan_stop: }
\cs_new_eq:NN \muskip_use:N \dim_use:N
\cs_new_eq:NN \muskip_use:c \dim_use:c
\cs_new_eq:NN  \muskip_show:N \__kernel_register_show:N
\cs_generate_variant:Nn \muskip_show:N { c }
```

```
\cs_new_protected:Npn \muskip_show:n
  { \__kernel_msg_show_eval:Nn \muskip_eval:n }
\cs_new_eq:NN \muskip_log:N \__kernel_register_log:N
\cs_new_eq:NN \muskip_log:c \__kernel_register_log:c
\cs_new_protected:Npn \muskip_log:n
  { \__kernel_msg_log_eval:Nn \muskip_eval:n }
\muskip_const:Nn \c_zero_muskip { 0 mu }
\muskip_const:Nn \c_max_muskip  { 16383.99999 mu }
\muskip_new:N \l_tmpa_muskip
\muskip_new:N \l_tmpb_muskip
\muskip_new:N \g_tmpa_muskip
\muskip_new:N \g_tmpb_muskip
%% File: l3keys.dtx
\scan_new:N \s__keyval_nil
\scan_new:N \s__keyval_mark
\scan_new:N \s__keyval_stop
\scan_new:N \s__keyval_tail
\bool_new:N \l__kernel_keyval_allow_blank_keys_bool
\group_begin:
  \cs_set_protected:Npn \__keyval_tmp:w #1#2
    {
      \cs_new:Npn \keyval_parse:nnn ##1 ##2 ##3
        {
          \__kernel_exp_not:w \tex_expanded:D
            {
              {
                \__keyval_loop_active:nnw {##1} {##2}
                  \s__keyval_mark ##3 #1 \s__keyval_tail #1
              }
            }
        }
      \cs_new_eq:NN \keyval_parse:NNn \keyval_parse:nnn
      \cs_new:Npn \__keyval_loop_active:nnw ##1 ##2 ##3 #1
        {
          \__keyval_if_recursion_tail:w ##3
            \__keyval_end_loop_active:w \s__keyval_tail
          \__keyval_loop_other:nnw {##1} {##2} ##3 , \s__keyval_tail ,
        }
      \cs_new:Npn \__keyval_split_other:w ##1 = ##2 \s__keyval_mark ##3
        { ##3 ##1 \s__keyval_stop \s__keyval_mark ##2 }
      \cs_new:Npn \__keyval_split_active:w ##1 #2 ##2 \s__keyval_mark ##3
        { ##3 ##1 \s__keyval_stop \s__keyval_mark ##2 }
      \cs_new:Npn \__keyval_loop_other:nnw ##1 ##2 ##3 ,
        {
          \__keyval_if_recursion_tail:w ##3
            \__keyval_end_loop_other:w \s__keyval_tail
          \__keyval_split_active:w ##3 \s__keyval_nil
            \s__keyval_mark \__keyval_split_active_auxi:w
            #2 \s__keyval_mark \__keyval_clean_up_active:w
          {##1} {##2}
          \s__keyval_mark
        }
      \cs_new:Npn \__keyval_split_active_auxi:w ##1 \s__keyval_stop
        {
          \__keyval_split_other:w ##1 \s__keyval_nil
            \s__keyval_mark \__keyval_misplaced_equal_after_active_error:w
            = \s__keyval_mark \__keyval_split_active_auxii:w
        }
      \cs_new:Npn \__keyval_split_active_auxii:w
          ##1 \s__keyval_nil \s__keyval_mark \__keyval_misplaced_equal_after_active_error:w
          \s__keyval_stop \s__keyval_mark
          ##2 \s__keyval_nil #2 \s__keyval_mark \__keyval_clean_up_active:w
        { \__keyval_trim:nN {##1} \__keyval_split_active_auxiii:w ##2 \s__keyval_nil }
      \cs_new:Npn \__keyval_split_active_auxiii:w ##1 ##2 \s__keyval_nil
        {
          \__keyval_split_active:w ##2 \s__keyval_nil
            \s__keyval_mark \__keyval_misplaced_equal_in_split_error:w
            #2 \s__keyval_mark \__keyval_split_active_auxiv:w
            {##1}
        }
      \cs_new:Npn \__keyval_split_active_auxiv:w
          ##1 \s__keyval_nil \s__keyval_mark \__keyval_misplaced_equal_in_split_error:w
          \s__keyval_stop \s__keyval_mark
        {
          \__keyval_split_other:w ##1 \s__keyval_nil
            \s__keyval_mark \__keyval_misplaced_equal_in_split_error:w
            = \s__keyval_mark \__keyval_split_active_auxv:w
        }
      \cs_new:Npn \__keyval_split_active_auxv:w
          ##1 \s__keyval_nil \s__keyval_mark \__keyval_misplaced_equal_in_split_error:w
          \s__keyval_stop \s__keyval_mark
        { \__keyval_trim:nN { ##1 } \__keyval_pair:nnnn }
      \cs_new:Npn \__keyval_clean_up_active:w
          ##1 \s__keyval_nil \s__keyval_mark \__keyval_split_active_auxi:w \s__keyval_stop \s__keyval_mark
        {
          \__keyval_split_other:w ##1 \s__keyval_nil
            \s__keyval_mark \__keyval_split_other_auxi:w
            = \s__keyval_mark \__keyval_clean_up_other:w
        }
      \cs_new:Npn \__keyval_split_other_auxi:w ##1 \s__keyval_stop
```

```
        { \__keyval_trim:nN { ##1 } \__keyval_split_other_auxii:w }
      \cs_new:Npn \__keyval_split_other_auxii:w
          ##1 ##2 \s__keyval_nil = \s__keyval_mark \__keyval_clean_up_other:w
        {
          \__keyval_split_other:w ##2 \s__keyval_nil
            \s__keyval_mark \__keyval_misplaced_equal_in_split_error:w
            = \s__keyval_mark \__keyval_split_other_auxiii:w
            { ##1 }
        }
      \cs_new:Npn \__keyval_split_other_auxiii:w
          ##1 \s__keyval_nil \s__keyval_mark \__keyval_misplaced_equal_in_split_error:w
          \s__keyval_stop \s__keyval_mark
        { \__keyval_trim:nN { ##1 } \__keyval_pair:nnnn }
      \cs_new:Npn \__keyval_clean_up_other:w
          ##1 \s__keyval_nil \s__keyval_mark \__keyval_split_other_auxi:w \s__keyval_stop \s__keyval_mark
        {
          \__keyval_if_blank:w ##1 \s__keyval_nil \s__keyval_stop \__keyval_blank_true:w
            \s__keyval_mark \s__keyval_stop
            \__keyval_trim:nN { ##1 } \__keyval_key:nn
        }
      \cs_new:Npn \__keyval_misplaced_equal_after_active_error:w
          \s__keyval_mark ##1 \s__keyval_stop \s__keyval_mark ##2 \s__keyval_nil
          = \s__keyval_mark \__keyval_split_active_auxii:w
          \s__keyval_mark ##3 \s__keyval_nil
          #2 \s__keyval_mark \__keyval_clean_up_active:w
        {
          \msg_expandable_error:nn
            { keyval } { misplaced-equals-sign }
          \__keyval_loop_other:nnw
        }
      \cs_new:Npn \__keyval_misplaced_equal_in_split_error:w
          \s__keyval_mark ##1 \s__keyval_stop \s__keyval_mark ##2 \s__keyval_nil
          ##3 \s__keyval_mark ##4 ##5
        {
          \msg_expandable_error:nn
            { keyval } { misplaced-equals-sign }
          \__keyval_loop_other:nnw
        }
      \cs_new:Npn \__keyval_end_loop_other:w
          \s__keyval_tail
          \__keyval_split_active:w
          \s__keyval_mark \s__keyval_tail
          \s__keyval_nil \s__keyval_mark
          \__keyval_split_active_auxi:w
          #2 \s__keyval_mark \__keyval_clean_up_active:w
        { \__keyval_loop_active:nnw }
      \cs_new:Npn \__keyval_end_loop_active:w
          \s__keyval_tail
          \__keyval_loop_other:nnw ##1 \s__keyval_mark \s__keyval_tail , \s__keyval_tail ,
        { }
    }
  \char_set_catcode_active:n { `\, }
  \char_set_catcode_active:n { `\= }
  \__keyval_tmp:w , =
\group_end:
\cs_generate_variant:Nn \keyval_parse:NNn { NNV , NNv }
\cs_generate_variant:Nn \keyval_parse:nnn { nnV , nnv }
\group_begin:
  \cs_set_protected:Npn \__keyval_tmp:w #1#2
    {
      \cs_new:Npn \__keyval_pair:nnnn ##1 ##2 ##3 ##4
        {
          \__keyval_if_blank:w \s__keyval_mark ##2 \s__keyval_nil \s__keyval_stop \__keyval_blank_key_error:w
            \s__keyval_mark \s__keyval_stop
          #1
          \exp_not:n { ##4 {##2} {##1} }
          #2
          \__keyval_loop_other:nnw {##3} {##4}
        }
      \cs_new:Npn \__keyval_key:nn ##1 ##2
        {
          \__keyval_if_blank:w \s__keyval_mark ##1 \s__keyval_nil \s__keyval_stop \__keyval_blank_key_error:w
            \s__keyval_mark \s__keyval_stop
          #1
          \exp_not:n { ##2 {##1} }
          #2
          \__keyval_loop_other:nnw {##2}
        }
    }
  \__keyval_tmp:w { } { }
\group_end:
\cs_new:Npn \__keyval_if_empty:w #1 \s__keyval_mark \s__keyval_stop { }
\cs_new:Npn \__keyval_if_blank:w \s__keyval_mark #1 { \__keyval_if_empty:w \s__keyval_mark }
\cs_new:Npn \__keyval_if_recursion_tail:w \s__keyval_mark #1 \s__keyval_tail { }
\cs_new:Npn \__keyval_blank_true:w \s__keyval_mark \s__keyval_stop \__keyval_trim:nN #1 \__keyval_key:nn
  { \__keyval_loop_other:nnw }
```

```
\cs_new:Npn \__keyval_blank_key_error:w \s__keyval_mark \s__keyval_stop #1 \__keyval_loop_other:nnw
  {
    \bool_if:NTF \l__kernel_keyval_allow_blank_keys_bool
      { #1 }
      { \msg_expandable_error:nn { keyval } { blank-key-name } }
    \__keyval_loop_other:nnw
  }
\msg_new:nnn { keyval } { misplaced-equals-sign }
  { Misplaced~'='~in~key-value~input~\msg_line_context: }
\msg_new:nnn { keyval } { blank-key-name }
  { Blank~key~name~in~key-value~input~\msg_line_context: }
\prop_gput:Nnn \g_msg_module_name_prop { keyval } { LaTeX }
\prop_gput:Nnn \g_msg_module_type_prop { keyval } { }
\group_begin:
  \cs_set_protected:Npn \__keyval_tmp:w #1
    {
      \cs_new:Npn \__keyval_trim:nN ##1
        {
          \__keyval_trim_auxi:w
            ##1
            \s__keyval_nil
            \s__keyval_mark #1 { }
            \s__keyval_mark \__keyval_trim_auxii:w
            \__keyval_trim_auxiii:w
            #1 \s__keyval_nil
            \__keyval_trim_auxiv:w
        }
      \cs_new:Npn \__keyval_trim_auxi:w ##1 \s__keyval_mark #1 ##2 \s__keyval_mark ##3
        {
          ##3
          \__keyval_trim_auxi:w
          \s__keyval_mark
          ##2
          \s__keyval_mark #1 {##1}
        }
      \cs_new:Npn \__keyval_trim_auxii:w \__keyval_trim_auxi:w \s__keyval_mark \s__keyval_mark ##1
        {
          \__keyval_trim_auxiii:w
          ##1
        }
      \cs_new:Npn \__keyval_trim_auxiii:w ##1 #1 \s__keyval_nil ##2
        {
          ##2
          ##1 \s__keyval_nil
          \__keyval_trim_auxiii:w
        }
      \cs_new:Npn \__keyval_trim_auxiv:w
          \s__keyval_mark ##1 \s__keyval_nil
          \__keyval_trim_auxiii:w \s__keyval_nil \__keyval_trim_auxiii:w
          ##2
        { ##2 { ##1 } }
    }
  \__keyval_tmp:w { ~ }
\group_end:
\str_const:Nn \c__keys_code_root_str     { key~code~>~ }
\str_const:Nn \c__keys_check_root_str    { key~check~>~ }
\str_const:Nn \c__keys_default_root_str  { key~default~>~ }
\str_const:Nn \c__keys_groups_root_str   { key~groups~>~ }
\str_const:Nn \c__keys_inherit_root_str  { key~inherit~>~ }
\str_const:Nn \c__keys_type_root_str     { key~type~>~ }
\str_const:Nn \c__keys_props_root_str { key~prop~>~ }
\int_new:N \l_keys_choice_int
\tl_new:N \l_keys_choice_tl
\clist_new:N \l__keys_groups_clist
\bool_new:N \l__keys_inherit_bool
\clist_new:N \l__keys_inherit_clist
\str_new:N \l_keys_key_str
\tl_new:N \l_keys_key_tl
\str_new:N \l__keys_module_str
\bool_new:N \l__keys_no_value_bool
\bool_new:N \l__keys_only_known_bool
\str_new:N \l_keys_path_str
\tl_new:N \l_keys_path_tl
\str_new:N \l__keys_inherit_str
\tl_new:N \l__keys_relative_tl
\tl_set:Nn \l__keys_relative_tl { \q__keys_no_value }
\str_new:N \l__keys_property_str
\bool_new:N \l__keys_selective_bool
\bool_new:N \l__keys_exclude_bool
\clist_new:N \l__keys_selective_clist
\clist_new:N \l__keys_tmp_clist
\clist_new:N \l__keys_unused_clist
\tl_new:N \l_keys_value_tl
\bool_new:N \l__keys_tmp_bool
\tl_new:N \l__keys_tmpa_tl
\tl_new:N \l__keys_tmpb_tl
\bool_new:N \l__keys_precompile_bool
\tl_new:N \l__keys_precompile_tl
\prop_new:N \l_keys_usage_load_prop
\prop_new:N \l_keys_usage_preamble_prop
```

```
\scan_new:N \s__keys_nil
\scan_new:N \s__keys_mark
\scan_new:N \s__keys_stop
\quark_new:N \q__keys_no_value
\__kernel_quark_new_conditional:Nn \__keys_quark_if_no_value:N { TF }
\cs_new_protected:Npn \__keys_precompile:n #1
  {
    \bool_if:NTF \l__keys_precompile_bool
      { \tl_put_right:Nn \l__keys_precompile_tl }
      { \use:n }
        {#1}
  }
\cs_new_protected:Npn \__keys_cs_undefine:c #1
  {
    \if_cs_exist:w #1 \cs_end:
    \else:
      \use_i:nnnn
    \fi:
    \cs_set_eq:cN {#1} \tex_undefined:D
  }
\cs_new_protected:Npn \keys_define:nn #1#2
  {
    \use:e
      {
        \exp_not:n
          {
            \str_set:Ne \l__keys_module_str { \__keys_trim_spaces:n {#1} }
            \keyval_parse:NNn \__keys_define:n \__keys_define:nn {#2}
          }
        \__keys_reset_var:N \l__keys_module_str
        \__keys_reset_var:N \l__keys_inherit_str
        \__keys_reset_var:N \l_keys_choice_tl
        \__keys_reset_var:N \l_keys_key_tl
        \__keys_reset_var:N \l_keys_key_str
        \__keys_reset_var:N \l_keys_path_tl
        \__keys_reset_var:N \l_keys_path_str
        \__keys_reset_var:N \l_keys_value_tl
        \int_set:Nn \l_keys_choice_int { \int_use:N \l_keys_choice_int }
      }
  }
\cs_generate_variant:Nn \keys_define:nn { ne , nx }
\cs_new_protected:Npn \__keys_define:n #1
  {
    \bool_set_true:N \l__keys_no_value_bool
    \__keys_define_aux:nn {#1} { }
  }
```

```
\cs_new_protected:Npn \__keys_define:nn #1#2
  {
    \bool_set_false:N \l__keys_no_value_bool
    \__keys_define_aux:nn {#1} {#2}
  }
\cs_new_protected:Npn \__keys_define_aux:nn #1#2
  {
    \__keys_property_find:n {#1}
    \cs_if_exist:cTF { \c__keys_props_root_str \l__keys_property_str }
      { \__keys_define_code:n {#2} }
      {
        \str_if_empty:NF \l__keys_property_str
          {
            \msg_error:nnee { keys } { property-unknown }
              \l__keys_property_str \l_keys_path_str
          }
      }
  }
\cs_new_protected:Npn \__keys_property_find:n #1
  {
    \exp_after:wN \__keys_property_find_auxi:w \tl_to_str:n {#1}
      \s__keys_nil \__keys_property_find_auxii:w
      . \s__keys_nil \__keys_property_find_err:w
  }
\cs_new:Npn \__keys_property_find_auxi:w #1 . #2 \s__keys_nil #3
  {
    #3 #1 \s__keys_mark #2 \s__keys_nil #3
  }
\cs_new_protected:Npn \__keys_property_find_auxii:w
    #1 \s__keys_mark #2 \s__keys_nil \__keys_property_find_auxii:w . \s__keys_nil
    \__keys_property_find_err:w
  {
    \cs_set_nopar:Npe \l_keys_path_str
      {
        \str_if_empty:NF \l__keys_module_str { \l__keys_module_str / }
        \exp_after:wN \__keys_trim_spaces:n \tex_expanded:D {{
        #1
        \if_false: }}} \fi:
        \__keys_property_find_auxi:w #2 \s__keys_nil \__keys_property_find_auxiii:w
          . \s__keys_nil \__keys_property_find_auxiv:w
  }
\cs_new:Npn \__keys_property_find_auxiii:w #1 \s__keys_mark #2 . #3 \s__keys_nil #4
  {
    . #1 #4 #2 \s__keys_mark #3 \s__keys_nil #4
  }
\cs_new:Npn \__keys_property_find_auxiv:w
```

```
    #1 \s__keys_nil \__keys_property_find_auxiii:w
    \s__keys_mark \s__keys_nil \__keys_property_find_auxiv:w
  {
    \if_false: {{{ \fi: }}}
    \cs_set_nopar:Npe \l__keys_property_str { . #1 }
    \tl_set_eq:NN \l_keys_path_tl \l_keys_path_str
  }
\cs_new_protected:Npn \__keys_property_find_err:w
    #1 \s__keys_nil #2 \__keys_property_find_err:w
  {
    \str_clear:N \l__keys_property_str
    \msg_error:nnn { keys } { no-property } {#1}
  }
\cs_new_protected:Npn \__keys_define_code:n #1
  {
    \bool_if:NTF \l__keys_no_value_bool
      {
        \__keys_define_code:nnn
          { \use:c { \c__keys_props_root_str \l__keys_property_str } {#1} }
          { \use:c { \c__keys_props_root_str \l__keys_property_str } }
          {
            \msg_error:nnee { keys } { property-requires-value }
              \l__keys_property_str \l_keys_path_str
          }
      }
      { \use:c { \c__keys_props_root_str \l__keys_property_str } {#1} }
  }
\cs_new:Npe \__keys_define_code:nnn
  {
    \exp_not:N \exp_after:wN \exp_not:N \__keys_define_code:w
      \exp_not:N \l__keys_property_str
      \c_colon_str \c_colon_str
      \exp_not:N \s__keys_stop
  }
\use:e
  {
    \cs_new:Npn \exp_not:N \__keys_define_code:w
      #1 \c_colon_str #2 \c_colon_str #3 \exp_not:N \s__keys_stop
  }
    {
      \tl_if_empty:nTF {#3}
        { \use_i:nnn }
        {
          \tl_if_empty:nTF {#2}
            { \use_ii:nnn }
            { \use_iii:nnn }
        }
    }
\cs_new_protected:Npn \__keys_bool_set:Nn #1#2
  { \__keys_bool_set:Nnnn #1 {#2} { true } { false } }
\cs_generate_variant:Nn \__keys_bool_set:Nn { c }
\cs_new_protected:Npn \__keys_bool_set_inverse:Nn #1#2
  { \__keys_bool_set:Nnnn #1 {#2} { false } { true } }
\cs_generate_variant:Nn \__keys_bool_set_inverse:Nn { c }
\cs_new_protected:Npn \__keys_bool_set:Nnnn #1#2#3#4
  {
    \bool_if_exist:NF #1 { \bool_new:N #1 }
    \__keys_choice_make:
    \__keys_cmd_set:ne { \l_keys_path_str / true }
      { \exp_not:c { bool_ #2 set_ #3 :N } \exp_not:N #1 }
    \__keys_cmd_set:ne { \l_keys_path_str / false }
      { \exp_not:c { bool_ #2 set_ #4 :N } \exp_not:N #1 }
    \__keys_cmd_set_direct:nn { \l_keys_path_str / unknown }
      {
        \msg_error:nne { keys } { boolean-values-only }
          \l_keys_path_str
      }
    \__keys_default_set:n { true }
  }
\cs_generate_variant:Nn \__keys_bool_set:Nn { c }
\cs_new_protected:Npn \__keys_choice_make:
  { \__keys_choice_make:N \__keys_choice_find:n }
\cs_new_protected:Npn \__keys_multichoice_make:
  { \__keys_choice_make:N \__keys_multichoice_find:n }
\cs_new_protected:Npn \__keys_choice_make:N #1
  {
    \cs_if_exist:cTF
      { \c__keys_type_root_str \__keys_parent:o \l_keys_path_str }
      {
        \str_if_eq:vnTF
          { \c__keys_type_root_str \__keys_parent:o \l_keys_path_str }
          { choice }
          {
            \msg_error:nnee { keys } { nested-choice-key }
              \l_keys_path_tl { \__keys_parent:o \l_keys_path_str }
          }
          { \__keys_choice_make_aux:N #1 }
      }
      { \__keys_choice_make_aux:N #1 }
  }
\cs_new_protected:Npn \__keys_choice_make_aux:N #1
  {
    \cs_set_nopar:cpn { \c__keys_type_root_str \l_keys_path_str }
      { choice }
    \__keys_cmd_set_direct:nn \l_keys_path_str { #1 {##1} }
    \__keys_cmd_set_direct:nn { \l_keys_path_str / unknown }
      {
        \msg_error:nnee { keys } { choice-unknown }
          \l_keys_path_str {##1}
      }
  }
\cs_new_protected:Npn \__keys_choices_make:nn
  { \__keys_choices_make:Nnn \__keys_choice_make: }
\cs_new_protected:Npn \__keys_multichoices_make:nn
  { \__keys_choices_make:Nnn \__keys_multichoice_make: }
\cs_new_protected:Npn \__keys_choices_make:Nnn #1#2#3
  {
    #1
    \int_zero:N \l_keys_choice_int
    \clist_map_inline:nn {#2}
      {
        \int_incr:N \l_keys_choice_int
        \__keys_cmd_set:ne
          { \l_keys_path_str / \__keys_trim_spaces:n {##1} }
          {
            \tl_set:Nn \exp_not:N \l_keys_choice_tl {##1}
            \int_set:Nn \exp_not:N \l_keys_choice_int
              { \int_use:N \l_keys_choice_int }
            \exp_not:n {#3}
          }
      }
  }
\cs_new_protected:Npn \__keys_cmd_set:nn #1#2
  {  \__keys_cmd_set_direct:nn {#1} { \__keys_precompile:n {#2} } }
\cs_generate_variant:Nn \__keys_cmd_set:nn { ne , Vn , Vo }
\cs_new_protected:Npn \__keys_cmd_set_direct:nn #1#2
  { \cs_set_protected:cpn { \c__keys_code_root_str #1 } ##1 {#2} }
\cs_new_protected:Npn \__keys_cs_set:NNpn #1#2#3#
  {
    \cs_set_protected:cpe { \c__keys_code_root_str \l_keys_path_str } ##1
      {
        \__keys_precompile:n
          { #1 \exp_not:N #2 \exp_not:n {#3} {##1} }
      }
    \use_none:n
  }
\cs_generate_variant:Nn \__keys_cs_set:NNpn { Nc }
\cs_new_protected:Npn \__keys_default_set:n #1
```

```
  {
    \tl_if_empty:nTF {#1}
      {
        \__keys_cs_undefine:c
          { \c__keys_default_root_str \l_keys_path_str }
      }
      {
        \cs_set_nopar:cpe
          { \c__keys_default_root_str \l_keys_path_str }
          { \exp_not:n {#1} }
        \__keys_value_requirement:nn { required } { false }
      }
  }
\cs_new_protected:Npn \__keys_groups_set:n #1
  {
    \clist_set:Ne \l__keys_groups_clist { \tl_to_str:n {#1} }
    \clist_if_empty:NTF \l__keys_groups_clist
      {
        \__keys_cs_undefine:c
          { \c__keys_groups_root_str \l_keys_path_str }
      }
      {
        \cs_set_eq:cN { \c__keys_groups_root_str \l_keys_path_str }
          \l__keys_groups_clist
      }
  }
\cs_new_protected:Npn \__keys_inherit:n #1
  {
    \__keys_undefine:
    \clist_set:Nn \l__keys_inherit_clist {#1}
    \cs_set_eq:cN { \c__keys_inherit_root_str \l_keys_path_str }
      \l__keys_inherit_clist
  }
\cs_new_protected:Npn \__keys_initialise:n #1
  {
    \typeout{------------------------------->}
    \exp_after:wN \__keys_find_key_module:wNN
      \l_keys_path_str \s__keys_stop
      \l_keys_key_tl \l_keys_key_str
    \tl_set_eq:NN \l_keys_key_tl \l_keys_key_str
    \tl_set:Nn \l_keys_value_tl {#1}
    \cs_if_exist:cTF { \c__keys_code_root_str \l_keys_path_str }
      {
        \str_clear:N \l__keys_inherit_str
        \__keys_execute:nn \l_keys_path_str {#1}
      }
      {
        \cs_if_exist:cT
          { \c__keys_inherit_root_str \__keys_parent:o \l_keys_path_str }
          { \__keys_execute_inherit: }
      }
  }
\cs_new_protected:Npn \__keys_legacy_if_set:nn #1#2
  { \__keys_legacy_if_set:nnnn {#1} {#2} { true } { false } }
\cs_new_protected:Npn \__keys_legacy_if_set_inverse:nn #1#2
  { \__keys_legacy_if_set:nnnn {#1} {#2} { false } { true } }
\cs_new_protected:Npn \__keys_legacy_if_set:nnnn #1#2#3#4
  {
    \__keys_choice_make:
    \__keys_cmd_set:ne { \l_keys_path_str / true }
      { \exp_not:c { legacy_if_#2  set_ #3 :n } { \exp_not:n {#1} } }
    \__keys_cmd_set:ne { \l_keys_path_str / false }
      { \exp_not:c { legacy_if_#2  set_ #4 :n } { \exp_not:n {#1} } }
    \__keys_cmd_set:nn { \l_keys_path_str / unknown }
      {
        \msg_error:nne { keys } { boolean-values-only }
          \l_keys_path_str
      }
    \__keys_default_set:n { true }
    \cs_if_exist:cF { if#1 }
      {
        \cs:w newif \exp_after:wN \cs_end:
          \cs:w if#1 \cs_end:
      }
  }
\cs_new_protected:Npn \__keys_meta_make:n
  { \exp_args:NV \__keys_meta_make:nn \l__keys_module_str }
\cs_new_protected:Npn \__keys_meta_make:nn #1#2
  {
    \exp_args:NV \__keys_cmd_set_direct:nn
      \l_keys_path_str { \__keys_set:nn {#1} {#2} }
  }
\cs_new_protected:Npn \__keys_prop_put:Nn #1#2
  {
    \prop_if_exist:NF #1 { \prop_new:N #1 }
    \exp_after:wN \__keys_find_key_module:wNN \l_keys_path_str \s__keys_stop
      \l__keys_tmpa_tl \l__keys_tmpb_tl
    \__keys_cmd_set:ne \l_keys_path_str
      {
        \exp_not:c { prop_ #2 put:Nnn }
        \exp_not:N #1
        { \l__keys_tmpb_tl }
        \exp_not:n { {##1} }
      }
  }
\cs_generate_variant:Nn \__keys_prop_put:Nn { c }
\cs_new_protected:Npn \__keys_undefine:
  {
    \clist_map_inline:nn
      { code , default , groups , inherit , type , check }
      {
        \__keys_cs_undefine:c
          { \tl_use:c { c__keys_ ##1 _root_str } \l_keys_path_str }
      }
  }
\cs_new_protected:Npn \__keys_value_requirement:nn #1#2
  {
    \str_case:nnF {#2}
      {
        { true }
          {
            \cs_set_eq:cc
              { \c__keys_check_root_str \l_keys_path_str }
              { __keys_check_ #1 : }
          }
        { false }
          {
            \cs_if_eq:ccT
              { \c__keys_check_root_str \l_keys_path_str }
              { __keys_check_ #1 : }
              {
                \__keys_cs_undefine:c
                  { \c__keys_check_root_str \l_keys_path_str }
              }
          }
      }
      {
        \msg_error:nne { keys }
          { boolean-values-only }
          { .value_ #1 :n }
      }
  }
\cs_new_protected:Npn \__keys_check_forbidden:
  {
    \bool_if:NF \l__keys_no_value_bool
      {
        \msg_error:nnee { keys } { value-forbidden }
          \l_keys_path_str \l_keys_value_tl
        \use_none:nnn
      }
  }
\cs_new_protected:Npn \__keys_check_required:
  {
    \bool_if:NT \l__keys_no_value_bool
      {
        \msg_error:nne { keys } { value-required }
          \l_keys_path_str
        \use_none:nnn
      }
  }
\cs_new_protected:Npn \__keys_usage:n #1
  {
    \str_case:nnF {#1}
      {
        { general }
          {
            \__keys_usage:NN \l_keys_usage_load_prop
              \c_false_bool
            \__keys_usage:NN \l_keys_usage_preamble_prop
              \c_false_bool
          }
        { load }
          {
            \__keys_usage:NN \l_keys_usage_load_prop
              \c_true_bool
            \__keys_usage:NN \l_keys_usage_preamble_prop
              \c_false_bool
          }
        { preamble }
          {
            \__keys_usage:NN \l_keys_usage_load_prop
              \c_false_bool
            \__keys_usage:NN \l_keys_usage_preamble_prop
              \c_true_bool
          }
      }
      {
        \msg_error:nnnn { keys }
          { choice-unknown }
          { .usage:n }
          {#1}
      }
  }
\cs_new_protected:Npn \__keys_usage:NN #1#2
```

```
  {
    \prop_get:NVNF #1 \l__keys_module_str \l__keys_tmpa_tl
      { \tl_clear:N \l__keys_tmpa_tl }
    \tl_set:Ne \l__keys_tmpb_tl
      {
        \exp_after:wN \exp_after:wN \exp_after:wN \__keys_usage:w \exp_after:wN
          \l_keys_path_str \exp_after:wN / \exp_after:wN \s__keys_stop
          \exp_after:wN { \l_keys_path_str }
      }
    \bool_if:NTF #2
      { \clist_put_right:NV \l__keys_tmpa_tl \l__keys_tmpb_tl }
      { \clist_remove_all:NV \l__keys_tmpa_tl \l__keys_tmpb_tl }
    \prop_put:NVV #1 \l__keys_module_str
      \l__keys_tmpa_tl
  }
\cs_new:Npn \__keys_usage:w #1 / #2 \s__keys_stop #3
  {
    \tl_if_blank:nTF {#2}
     {#1}
     { \__keys_usage_aux:w #3 \s__keys_stop }
  }
\cs_new:Npn \__keys_usage_aux:w #1 / #2 \s__keys_stop {#2}
\cs_new_protected:Npn \__keys_variable_set:NnnN #1#2#3#4
  {
    \use:c { #2_if_exist:NF } #1 { \use:c { #2 _new:N } #1 }
    \__keys_cmd_set:ne \l_keys_path_str
      {
        \exp_not:c { #2 _ #3 set:N #4 }
        \exp_not:N #1
        \exp_not:n  { {##1} }
      }
  }
\cs_generate_variant:Nn \__keys_variable_set:NnnN { c }
\cs_new_protected:Npn \__keys_variable_set_required:NnnN #1#2#3#4
  {
    \__keys_variable_set:NnnN #1 {#2} {#3} #4
    \__keys_value_requirement:nn { required } { true }
  }
\cs_generate_variant:Nn \__keys_variable_set_required:NnnN { c }
\cs_new_protected:cpn { \c__keys_props_root_str .bool_set:N } #1
  { \__keys_bool_set:Nn #1 { } }
\cs_new_protected:cpn { \c__keys_props_root_str .bool_set:c } #1
  { \__keys_bool_set:cn {#1} { } }
\cs_new_protected:cpn { \c__keys_props_root_str .bool_gset:N } #1
  { \__keys_bool_set:Nn #1 { g } }
\cs_new_protected:cpn { \c__keys_props_root_str .bool_gset:c } #1
```

```
  { \__keys_bool_set:cn {#1} { g } }
\cs_new_protected:cpn { \c__keys_props_root_str .bool_set_inverse:N } #1
  { \__keys_bool_set_inverse:Nn #1 { } }
\cs_new_protected:cpn { \c__keys_props_root_str .bool_set_inverse:c } #1
  { \__keys_bool_set_inverse:cn {#1} { } }
\cs_new_protected:cpn { \c__keys_props_root_str .bool_gset_inverse:N } #1
  { \__keys_bool_set_inverse:Nn #1 { g } }
\cs_new_protected:cpn { \c__keys_props_root_str .bool_gset_inverse:c } #1
  { \__keys_bool_set_inverse:cn {#1} { g } }
\cs_new_protected:cpn { \c__keys_props_root_str .choice: }
  { \__keys_choice_make: }
\cs_new_protected:cpn { \c__keys_props_root_str .choices:nn } #1
  { \__keys_choices_make:nn #1 }
\cs_new_protected:cpn { \c__keys_props_root_str .choices:Vn } #1
  { \exp_args:NV \__keys_choices_make:nn #1 }
\cs_new_protected:cpn { \c__keys_props_root_str .choices:en } #1
  { \exp_args:Ne \__keys_choices_make:nn #1 }
\cs_new_protected:cpn { \c__keys_props_root_str .choices:on } #1
  { \exp_args:No \__keys_choices_make:nn #1 }
\cs_new_protected:cpn { \c__keys_props_root_str .choices:xn } #1
  { \exp_args:Nx \__keys_choices_make:nn #1 }
\cs_new_protected:cpn { \c__keys_props_root_str .code:n } #1
  { \__keys_cmd_set:nn \l_keys_path_str {#1} }
\cs_new_protected:cpn { \c__keys_props_root_str .clist_set:N } #1
  { \__keys_variable_set:NnnN #1 { clist } { } n }
\cs_new_protected:cpn { \c__keys_props_root_str .clist_set:c } #1
  { \__keys_variable_set:cnnN {#1} { clist } { } n }
\cs_new_protected:cpn { \c__keys_props_root_str .clist_gset:N } #1
  { \__keys_variable_set:NnnN #1 { clist } { g } n }
\cs_new_protected:cpn { \c__keys_props_root_str .clist_gset:c } #1
  { \__keys_variable_set:cnnN {#1} { clist } { g } n }
\cs_new_protected:cpn { \c__keys_props_root_str .cs_set:Np } #1
  { \__keys_cs_set:NNpn \cs_set:Npn #1 { } }
\cs_new_protected:cpn { \c__keys_props_root_str .cs_set:cp } #1
  { \__keys_cs_set:Ncpn \cs_set:Npn #1 { } }
\cs_new_protected:cpn { \c__keys_props_root_str .cs_set_protected:Np } #1
  { \__keys_cs_set:NNpn \cs_set_protected:Npn #1 { } }
\cs_new_protected:cpn { \c__keys_props_root_str .cs_set_protected:cp } #1
  { \__keys_cs_set:Ncpn \cs_set_protected:Npn #1 { } }
\cs_new_protected:cpn { \c__keys_props_root_str .cs_gset:Np } #1
  { \__keys_cs_set:NNpn \cs_gset:Npn #1 { } }
\cs_new_protected:cpn { \c__keys_props_root_str .cs_gset:cp } #1
  { \__keys_cs_set:Ncpn \cs_gset:Npn #1 { } }
\cs_new_protected:cpn { \c__keys_props_root_str .cs_gset_protected:Np } #1
  { \__keys_cs_set:NNpn \cs_gset_protected:Npn #1 { } }
\cs_new_protected:cpn { \c__keys_props_root_str .cs_gset_protected:cp } #1
```

```
  { \__keys_cs_set:Ncpn \cs_gset_protected:Npn #1 { } }
\cs_new_protected:cpn { \c__keys_props_root_str .default:n } #1
  { \__keys_default_set:n {#1} }
\cs_new_protected:cpn { \c__keys_props_root_str .default:V } #1
  { \exp_args:NV \__keys_default_set:n #1 }
\cs_new_protected:cpn { \c__keys_props_root_str .default:e } #1
  { \exp_args:Ne \__keys_default_set:n {#1} }
\cs_new_protected:cpn { \c__keys_props_root_str .default:o } #1
  { \exp_args:No \__keys_default_set:n {#1} }
\cs_new_protected:cpn { \c__keys_props_root_str .default:x } #1
  { \exp_args:Nx \__keys_default_set:n {#1} }
\cs_new_protected:cpn { \c__keys_props_root_str .dim_set:N } #1
  { \__keys_variable_set_required:NnnN #1 { dim } { } n }
\cs_new_protected:cpn { \c__keys_props_root_str .dim_set:c } #1
  { \__keys_variable_set_required:cnnN {#1} { dim } { } n }
\cs_new_protected:cpn { \c__keys_props_root_str .dim_gset:N } #1
  { \__keys_variable_set_required:NnnN #1 { dim } { g } n }
\cs_new_protected:cpn { \c__keys_props_root_str .dim_gset:c } #1
  { \__keys_variable_set_required:cnnN {#1} { dim } { g } n }
\cs_new_protected:cpn { \c__keys_props_root_str .fp_set:N } #1
  { \__keys_variable_set_required:NnnN #1 { fp } { } n }
\cs_new_protected:cpn { \c__keys_props_root_str .fp_set:c } #1
  { \__keys_variable_set_required:cnnN {#1} { fp } { } n }
\cs_new_protected:cpn { \c__keys_props_root_str .fp_gset:N } #1
  { \__keys_variable_set_required:NnnN #1 { fp } { g } n }
\cs_new_protected:cpn { \c__keys_props_root_str .fp_gset:c } #1
  { \__keys_variable_set_required:cnnN {#1} { fp } { g } n }
\cs_new_protected:cpn { \c__keys_props_root_str .groups:n } #1
  { \__keys_groups_set:n {#1} }
\cs_new_protected:cpn { \c__keys_props_root_str .inherit:n } #1
  { \__keys_inherit:n {#1} }
\cs_new_protected:cpn { \c__keys_props_root_str .initial:n } #1
  { \__keys_initialise:n {#1} }
\cs_new_protected:cpn { \c__keys_props_root_str .initial:V } #1
  { \exp_args:NV \__keys_initialise:n #1 }
\cs_new_protected:cpn { \c__keys_props_root_str .initial:e } #1
  { \exp_args:Ne \__keys_initialise:n {#1} }
\cs_new_protected:cpn { \c__keys_props_root_str .initial:o } #1
  { \exp_args:No \__keys_initialise:n {#1} }
\cs_new_protected:cpn { \c__keys_props_root_str .initial:x } #1
  { \exp_args:Nx \__keys_initialise:n {#1} }
\cs_new_protected:cpn { \c__keys_props_root_str .int_set:N } #1
  { \__keys_variable_set_required:NnnN #1 { int } { } n }
\cs_new_protected:cpn { \c__keys_props_root_str .int_set:c } #1
  { \__keys_variable_set_required:cnnN {#1} { int } { } n }
\cs_new_protected:cpn { \c__keys_props_root_str .int_gset:N } #1
```

```
  { \__keys_variable_set_required:NnnN #1 { int } { g } n }
\cs_new_protected:cpn { \c__keys_props_root_str .int_gset:c } #1
  { \__keys_variable_set_required:cnnN {#1} { int } { g } n }
\cs_new_protected:cpn { \c__keys_props_root_str .legacy_if_set:n } #1
  { \__keys_legacy_if_set:nn {#1} { } }
\cs_new_protected:cpn { \c__keys_props_root_str .legacy_if_gset:n } #1
  { \__keys_legacy_if_set:nn {#1} { g } }
\cs_new_protected:cpn { \c__keys_props_root_str .legacy_if_set_inverse:n } #1
  { \__keys_legacy_if_set_inverse:nn {#1} { } }
\cs_new_protected:cpn { \c__keys_props_root_str .legacy_if_gset_inverse:n } #1
  { \__keys_legacy_if_set_inverse:nn {#1} { g } }
\cs_new_protected:cpn { \c__keys_props_root_str .meta:n } #1
  { \__keys_meta_make:n {#1} }
\cs_new_protected:cpn { \c__keys_props_root_str .meta:nn } #1
  { \__keys_meta_make:nn #1 }
\cs_new_protected:cpn { \c__keys_props_root_str .multichoice: }
  { \__keys_multichoice_make: }
\cs_new_protected:cpn { \c__keys_props_root_str .multichoices:nn } #1
  { \__keys_multichoices_make:nn #1 }
\cs_new_protected:cpn { \c__keys_props_root_str .multichoices:Vn } #1
  { \exp_args:NV \__keys_multichoices_make:nn #1 }
\cs_new_protected:cpn { \c__keys_props_root_str .multichoices:en } #1
  { \exp_args:Ne \__keys_multichoices_make:nn #1 }
\cs_new_protected:cpn { \c__keys_props_root_str .multichoices:on } #1
  { \exp_args:No \__keys_multichoices_make:nn #1 }
\cs_new_protected:cpn { \c__keys_props_root_str .multichoices:xn } #1
  { \exp_args:Nx \__keys_multichoices_make:nn #1 }
\cs_new_protected:cpn { \c__keys_props_root_str .muskip_set:N } #1
  { \__keys_variable_set_required:NnnN #1 { muskip } { } n }
\cs_new_protected:cpn { \c__keys_props_root_str .muskip_set:c } #1
  { \__keys_variable_set_required:cnnN {#1} { muskip } { } n }
\cs_new_protected:cpn { \c__keys_props_root_str .muskip_gset:N } #1
  { \__keys_variable_set_required:NnnN #1 { muskip } { g } n }
\cs_new_protected:cpn { \c__keys_props_root_str .muskip_gset:c } #1
  { \__keys_variable_set_required:cnnN {#1} { muskip } { g } n }
\cs_new_protected:cpn { \c__keys_props_root_str .prop_put:N } #1
  { \__keys_prop_put:Nn #1 { } }
\cs_new_protected:cpn { \c__keys_props_root_str .prop_put:c } #1
  { \__keys_prop_put:cn {#1} { } }
\cs_new_protected:cpn { \c__keys_props_root_str .prop_gput:N } #1
  { \__keys_prop_put:Nn #1 { g } }
\cs_new_protected:cpn { \c__keys_props_root_str .prop_gput:c } #1
  { \__keys_prop_put:cn {#1} { g } }
\cs_new_protected:cpn { \c__keys_props_root_str .skip_set:N } #1
  { \__keys_variable_set_required:NnnN #1 { skip } { } n }
\cs_new_protected:cpn { \c__keys_props_root_str .skip_set:c } #1
```

```
  { \__keys_variable_set_required:cnnN {#1} { skip } { } n }
\cs_new_protected:cpn { \c__keys_props_root_str .skip_gset:N } #1
  { \__keys_variable_set_required:NnnN #1 { skip } { g } n }
\cs_new_protected:cpn { \c__keys_props_root_str .skip_gset:c } #1
  { \__keys_variable_set_required:cnnN {#1} { skip } { g } n }
\cs_new_protected:cpn { \c__keys_props_root_str .str_set:N } #1
  { \__keys_variable_set:NnnN #1 { str } { } n }
\cs_new_protected:cpn { \c__keys_props_root_str .str_set:c } #1
  { \__keys_variable_set:cnnN {#1} { str } { } n }
\cs_new_protected:cpn { \c__keys_props_root_str .str_set_e:N } #1
  { \__keys_variable_set:NnnN #1 { str } { } e }
\cs_new_protected:cpn { \c__keys_props_root_str .str_set_e:c } #1
  { \__keys_variable_set:cnnN {#1} { str } { } e }
\cs_new_protected:cpn { \c__keys_props_root_str .str_gset:N } #1
  { \__keys_variable_set:NnnN #1 { str } { g } n }
\cs_new_protected:cpn { \c__keys_props_root_str .str_gset:c } #1
  { \__keys_variable_set:cnnN {#1} { str } { g } n }
\cs_new_protected:cpn { \c__keys_props_root_str .str_gset_e:N } #1
  { \__keys_variable_set:NnnN #1 { str } { g } e }
\cs_new_protected:cpn { \c__keys_props_root_str .str_gset_e:c } #1
  { \__keys_variable_set:cnnN {#1} { str } { g } e }
\cs_new_protected:cpn { \c__keys_props_root_str .tl_set:N } #1
  { \__keys_variable_set:NnnN #1 { tl } { } n }
\cs_new_protected:cpn { \c__keys_props_root_str .tl_set:c } #1
  { \__keys_variable_set:cnnN {#1} { tl } { } n }
\cs_new_protected:cpn { \c__keys_props_root_str .tl_set_e:N } #1
  { \__keys_variable_set:NnnN #1 { tl } { } e }
\cs_new_protected:cpn { \c__keys_props_root_str .tl_set_e:c } #1
  { \__keys_variable_set:cnnN {#1} { tl } { } e }
\cs_new_protected:cpn { \c__keys_props_root_str .tl_gset:N } #1
  { \__keys_variable_set:NnnN #1 { tl } { g } n }
\cs_new_protected:cpn { \c__keys_props_root_str .tl_gset:c } #1
  { \__keys_variable_set:cnnN {#1} { tl } { g } n }
\cs_new_protected:cpn { \c__keys_props_root_str .tl_gset_e:N } #1
  { \__keys_variable_set:NnnN #1 { tl } { g } e }
\cs_new_protected:cpn { \c__keys_props_root_str .tl_gset_e:c } #1
  { \__keys_variable_set:cnnN {#1} { tl } { g } e }
\cs_new_protected:cpn { \c__keys_props_root_str .undefine: }
  { \__keys_undefine: }
\cs_new_protected:cpn { \c__keys_props_root_str .usage:n } #1
  { \__keys_usage:n {#1} }
\cs_new_protected:cpn { \c__keys_props_root_str .value_forbidden:n } #1
  { \__keys_value_requirement:nn { forbidden } {#1} }
\cs_new_protected:cpn { \c__keys_props_root_str .value_required:n } #1
  { \__keys_value_requirement:nn { required } {#1} }
\cs_new_protected:Npn \__keys_set:nnnnNn #1#2#3#4#5#6
```

```
  {
    \use:e
      {
        \exp_not:n
          {
            \clist_clear:N \l__keys_unused_clist
            \clist_set:Ne \l__keys_selective_clist { \tl_to_str:n {#2} }
            \tl_set:Nn \l__keys_relative_tl {#4}
            #6
            \__keys_set:nn {#1} {#3}
            \clist_set_eq:NN #5 \l__keys_unused_clist
          }
        \__keys_reset_bool:N \l__keys_only_known_bool
        \__keys_reset_bool:N \l__keys_exclude_bool
        \__keys_reset_bool:N \l__keys_selective_bool
        \__keys_reset_var:N \l__keys_unused_clist
        \__keys_reset_var:N \l__keys_selective_clist
        \__keys_reset_var:N \l__keys_relative_tl
        \__keys_reset_var:N \l__keys_inherit_str
        \__keys_reset_var:N \l_keys_choice_tl
        \__keys_reset_var:N \l_keys_key_tl
        \__keys_reset_var:N \l_keys_key_str
        \__keys_reset_var:N \l_keys_path_tl
        \__keys_reset_var:N \l_keys_path_str
        \__keys_reset_var:N \l_keys_value_tl
        \int_set:Nn \l_keys_choice_int { \int_use:N \l_keys_choice_int }
      }
  }
\cs_new:Npn \__keys_reset_bool:N #1
  {
    \exp_not:c
      { bool_set_ \bool_if:NTF #1 { true } { false } :N }
        \exp_not:N #1
  }
\cs_new:Npn \__keys_reset_var:N #1
  {
    \exp_not:n
      { \__kernel_tl_set:Nx #1 }
        { \exp_not:N \exp_not:n { \exp_not:o { #1 } } }
  }
\cs_new_protected:Npn \__keys_set:nn #1#2
  { \exp_args:No \__keys_set:nnn \l__keys_module_str {#1} {#2} }
\cs_new_protected:Npn \__keys_set:nnn #1#2#3
  {
    \str_set:Ne \l__keys_module_str { \__keys_trim_spaces:n {#2} }
    \keyval_parse:NNn \__keys_set_keyval:n \__keys_set_keyval:nn {#3}
    \str_set:Nn \l__keys_module_str {#1}
  }
\cs_new_protected:Npn \keys_set:nn #1#2
  {
    \__keys_set:nnnnNn
      {#1} { } {#2} { \q__keys_no_value } \l__keys_tmp_clist
      {
        \bool_set_false:N \l__keys_only_known_bool
        \bool_set_false:N \l__keys_exclude_bool
        \bool_set_false:N \l__keys_selective_bool
      }
  }
\cs_generate_variant:Nn \keys_set:nn { nV , nv , ne , no , nx }
\cs_new_protected:Npn \keys_set_known:nnnN #1#2#3#4
  {
    \__keys_set:nnnnNn
      {#1} { } {#2} {#3} #4
      {
        \bool_set_true:N \l__keys_only_known_bool
        \bool_set_false:N \l__keys_exclude_bool
        \bool_set_false:N \l__keys_selective_bool
      }
  }
\cs_generate_variant:Nn \keys_set_known:nnnN { nV , nv , ne , no }
\cs_new_protected:Npn \keys_set_known:nnN #1#2#3
  { \keys_set_known:nnnN {#1} {#2} { \q__keys_no_value } #3 }
\cs_generate_variant:Nn \keys_set_known:nnN { nV , nv , ne , no }
\cs_new_protected:Npn \keys_set_known:nn #1#2
  { \keys_set_known:nnnN {#1} {#2} { \q__keys_no_value } \l__keys_tmp_clist }
\cs_generate_variant:Nn \keys_set_known:nn { nV , nv , ne , no }
\cs_new_protected:Npn \keys_set_exclude_groups:nnnnN #1#2#3#4#5
  {
    \__keys_set:nnnnNn
      {#1} {#2} {#3} {#4} #5
      {
        \bool_set_false:N \l__keys_only_known_bool
        \bool_set_true:N \l__keys_exclude_bool
        \bool_set_true:N \l__keys_selective_bool
      }
  }
\cs_generate_variant:Nn \keys_set_exclude_groups:nnnnN { nnV , nnv , nno }
\cs_new_protected:Npn \keys_set_exclude_groups:nnnN #1#2#3#4
  { \keys_set_exclude_groups:nnnnN {#1} {#2} {#3} { \q__keys_no_value } #4 }
\cs_generate_variant:Nn \keys_set_exclude_groups:nnnN { nnV , nnv , nno }
\cs_new_protected:Npn \keys_set_exclude_groups:nnn #1#2#3
  {
    \keys_set_exclude_groups:nnnnN {#1} {#2} {#3}
      { \q__keys_no_value } \l__keys_tmp_clist
  }
\cs_generate_variant:Nn \keys_set_exclude_groups:nnn { nnV , nnv , nno }
\cs_new_protected:Npn \keys_set_groups:nnnnN #1#2#3#4#5
  {
    \__keys_set:nnnnNn
      {#1} {#2} {#3} {#4} #5
      {
        \bool_set_false:N \l__keys_only_known_bool
        \bool_set_false:N \l__keys_exclude_bool
        \bool_set_true:N \l__keys_selective_bool
      }
  }
\cs_generate_variant:Nn \keys_set_groups:nnnnN { nnV , nnv , nno }
\cs_new_protected:Npn \keys_set_groups:nnnN #1#2#3#4
  { \keys_set_groups:nnnnN {#1} {#2} {#3} { \q__keys_no_value } #4 }
\cs_generate_variant:Nn \keys_set_groups:nnnN { nnV , nnv , nno }
\cs_new_protected:Npn \keys_set_groups:nnn #1#2#3
  {
    \keys_set_groups:nnnnN {#1} {#2} {#3}
      { \q__keys_no_value } \l__keys_tmp_clist
  }
\cs_generate_variant:Nn \keys_set_groups:nnn { nnV , nnv , nno }
\cs_new_protected:Npn \keys_precompile:nnN #1#2#3
  {
    \bool_set_true:N \l__keys_precompile_bool
    \tl_clear:N \l__keys_precompile_tl
    \keys_set:nn {#1} {#2}
    \bool_set_false:N \l__keys_precompile_bool
    \tl_set_eq:NN #3 \l__keys_precompile_tl
  }
\cs_new_protected:Npn \__keys_set_keyval:n #1
  {
    \bool_set_true:N \l__keys_no_value_bool
    \__keys_set_keyval:onn \l__keys_module_str {#1} { }
  }
\cs_new_protected:Npn \__keys_set_keyval:nn #1#2
  {
    \bool_set_false:N \l__keys_no_value_bool
    \__keys_set_keyval:onn \l__keys_module_str {#1} {#2}
  }
\cs_new_protected:Npn \__keys_set_keyval:nnn #1#2#3
  {
    \__kernel_tl_set:Nx \l_keys_path_str
      {
        \tl_if_blank:nF {#1}
          { #1 / }
        \__keys_trim_spaces:n {#2}
      }
    \str_clear:N \l__keys_module_str
    \str_clear:N \l__keys_inherit_str
    \exp_after:wN \__keys_find_key_module:wNN \l_keys_path_str \s__keys_stop
      \l__keys_module_str \l_keys_key_str
    \tl_set_eq:NN \l_keys_key_tl \l_keys_key_str
    \__keys_value_or_default:n {#3}
    \bool_if:NTF \l__keys_selective_bool
      \__keys_set_selective:
      \__keys_execute:
    \str_set:Nn \l__keys_module_str {#1}
  }
\cs_generate_variant:Nn \__keys_set_keyval:nnn { o }
\cs_new_protected:Npn \__keys_find_key_module:wNN #1 \s__keys_stop #2 #3
  {
    \__keys_find_key_module_auxi:Nw #2 #1 \s__keys_nil \__keys_find_key_module_auxii:Nw
      / \s__keys_nil \__keys_find_key_module_auxiv:Nw #3
  }
\cs_new_protected:Npn \__keys_find_key_module_auxi:Nw #1 #2 / #3 \s__keys_nil #4
  {
    #4 #1 #2 \s__keys_mark #3 \s__keys_nil #4
  }
\cs_new_protected:Npn \__keys_find_key_module_auxii:Nw
    #1 #2 \s__keys_mark #3 \s__keys_nil \__keys_find_key_module_auxii:Nw
  {
    \cs_set_nopar:Npe #1 { \tl_if_empty:NF #1 { #1 / } #2 }
    \__keys_find_key_module_auxi:Nw #1 #3 \s__keys_nil \__keys_find_key_module_auxiii:Nw
  }
\cs_new_protected:Npn \__keys_find_key_module_auxiii:Nw #1 #2 \s__keys_mark
  {
    \cs_set_nopar:Npe #1 { \tl_if_empty:NF #1 { #1 / } #2 }
    \__keys_find_key_module_auxi:Nw #1
  }
\cs_new_protected:Npn \__keys_find_key_module_auxiv:Nw
    #1 #2 \s__keys_nil #3 \s__keys_mark
    \s__keys_nil \__keys_find_key_module_auxiv:Nw #4
  {
    \cs_set_nopar:Npn #4 { #2 }
  }
\cs_new_protected:Npn \__keys_set_selective:
  {
    \cs_if_exist:cTF { \c__keys_groups_root_str \l_keys_path_str }
      {
        \clist_set_eq:Nc \l__keys_groups_clist
          { \c__keys_groups_root_str \l_keys_path_str }
        \__keys_check_groups:
      }
      {
        \bool_if:NTF \l__keys_exclude_bool
          \__keys_execute:
          \__keys_store_unused:
      }
  }
\cs_new_protected:Npn \__keys_check_groups:
  {
    \bool_set_false:N \l__keys_tmp_bool
    \clist_map_inline:Nn \l__keys_selective_clist
      {
        \clist_if_in:NnT \l__keys_groups_clist {##1}
          {
            \bool_set_true:N \l__keys_tmp_bool
            \clist_map_break:
          }
      }
    \bool_if:NTF \l__keys_tmp_bool
      {
        \bool_if:NTF \l__keys_exclude_bool
          \__keys_store_unused:
          \__keys_execute:
      }
      {
        \bool_if:NTF \l__keys_exclude_bool
          \__keys_execute:
          \__keys_store_unused:
      }
  }
\cs_new_protected:Npn \__keys_value_or_default:n #1
  {
    \bool_if:NTF \l__keys_no_value_bool
      {
        \cs_if_exist:cTF { \c__keys_default_root_str \l_keys_path_str }
          {
            \tl_set_eq:Nc
              \l_keys_value_tl
              { \c__keys_default_root_str \l_keys_path_str }
          }
          {
            \tl_clear:N \l_keys_value_tl
            \cs_if_exist:cT
```

```
              { \c__keys_inherit_root_str \__keys_parent:o \l_keys_path_str }
              { \__keys_default_inherit: }
          }
      }
      { \tl_set:Nn \l_keys_value_tl {#1} }
  }
\cs_new_protected:Npn \__keys_default_inherit:
  {
    \clist_map_inline:cn
      { \c__keys_inherit_root_str \__keys_parent:o \l_keys_path_str }
      {
        \cs_if_exist:cT
          { \c__keys_default_root_str ##1 / \l_keys_key_str }
          {
            \tl_set_eq:Nc
              \l_keys_value_tl
              { \c__keys_default_root_str ##1 / \l_keys_key_str }
            \clist_map_break:
          }
      }
  }
\cs_new_protected:Npn \__keys_execute:
  {
    \cs_if_exist:cTF { \c__keys_code_root_str \l_keys_path_str }
      {
        \cs_if_exist_use:c { \c__keys_check_root_str \l_keys_path_str }
        \__keys_execute:no \l_keys_path_str \l_keys_value_tl
      }
      {
        \cs_if_exist:cTF
          { \c__keys_inherit_root_str \__keys_parent:o \l_keys_path_str }
          { \__keys_execute_inherit: }
          { \__keys_execute_unknown: }
      }
  }
\cs_new_protected:Npn \__keys_execute_inherit:
  {
    \bool_set_false:N \l__keys_inherit_bool
    \clist_map_inline:cn
      { \c__keys_inherit_root_str \__keys_parent:o \l_keys_path_str }
      { \__keys_excute_inherit:n {##1} }
    \bool_if:NF \l__keys_inherit_bool
      { \__keys_execute_unknown: }
  }
\cs_new_protected:Npn \__keys_excute_inherit:n #1
  {
    \cs_if_exist:cTF
      { \c__keys_inherit_root_str #1 }
      {
        \clist_map_inline:cn { \c__keys_inherit_root_str #1 }
          { \__keys_excute_inherit:n {##1} }
        \bool_if:NT \l__keys_inherit_bool
          { \clist_map_break: }
      }
      {
        \cs_if_exist:cT
          { \c__keys_code_root_str #1 / \l_keys_key_str }
          {
            \str_set:Nn \l__keys_inherit_str {#1}
            \cs_if_exist_use:c { \c__keys_check_root_str #1 / \l_keys_key_str }
            \__keys_execute:no { #1 / \l_keys_key_str } \l_keys_value_tl
            \clist_map_break:n
              { \bool_set_true:N \l__keys_inherit_bool }
          }
      }
  }
\cs_new_protected:Npn \__keys_execute_unknown:
  {
    \bool_if:NTF \l__keys_only_known_bool
      { \__keys_store_unused: }
      {
        \cs_if_exist:cTF
          { \c__keys_code_root_str \l__keys_module_str / unknown }
          {
            \bool_if:NT \l__keys_no_value_bool
              {
                \cs_if_exist:cT
                  { \c__keys_default_root_str \l__keys_module_str / unknown }
                  {
                    \tl_set_eq:Nc
                      \l_keys_value_tl
                      { \c__keys_default_root_str \l__keys_module_str / unknown }
                  }
              }
            \__keys_execute:no { \l__keys_module_str / unknown } \l_keys_value_tl
          }
          {
            \msg_error:nnee { keys } { unknown }
              \l_keys_path_str \l__keys_module_str
          }
      }
  }
```

```
\cs_new:Npn \__keys_execute:nn #1#2
  { \__keys_execute:no {#1} { \prg_do_nothing: #2 } }
\cs_new:Npn \__keys_execute:no #1#2
  {
    \exp_args:NNo \exp_args:No \use:n
      {
        \cs:w \c__keys_code_root_str #1 \exp_after:wN \cs_end:
        \exp_after:wN {#2}
      }
  }
\cs_new_protected:Npn \__keys_store_unused:
  {
    \__keys_quark_if_no_value:NTF \l__keys_relative_tl
      {
        \clist_put_right:Ne \l__keys_unused_clist
          {
            \l_keys_key_str
            \bool_if:NF \l__keys_no_value_bool
              { = { \exp_not:o \l_keys_value_tl } }
          }
      }
      {
        \tl_if_empty:NTF \l__keys_relative_tl
          {
            \clist_put_right:Ne \l__keys_unused_clist
              {
                \l_keys_path_str
                \bool_if:NF \l__keys_no_value_bool
                  { = { \exp_not:o \l_keys_value_tl } }
              }
          }
          { \__keys_store_unused_aux: }
      }
  }
\cs_new_protected:Npn \__keys_store_unused_aux:
  {
    \__kernel_tl_set:Nx \l__keys_relative_tl
      { \exp_args:No \__keys_trim_spaces:n \l__keys_relative_tl }
    \use:e
      {
        \cs_set_protected:Npn \__keys_store_unused:w
          ##1 \l__keys_relative_tl /
          ##2 \l__keys_relative_tl /
          ##3 \s__keys_stop
      }
        {
          \tl_if_blank:nF {##1}
            {
              \msg_error:nnee { keys } { bad-relative-key-path }
                \l_keys_path_str
                \l__keys_relative_tl
            }
          \clist_put_right:Ne \l__keys_unused_clist
            {
              \exp_not:n {##2}
              \bool_if:NF \l__keys_no_value_bool
                { = { \exp_not:o \l_keys_value_tl } }
            }
        }
    \use:e
      {
        \__keys_store_unused:w \l_keys_path_str
          \l__keys_relative_tl / \l__keys_relative_tl /
          \s__keys_stop
      }
  }
\cs_new_protected:Npn \__keys_store_unused:w { }
\cs_new:Npn \__keys_choice_find:n #1
  {
    \str_if_empty:NTF \l__keys_inherit_str
      { \__keys_choice_find:nn \l_keys_path_str {#1} }
      {
        \__keys_choice_find:nn
          { \l__keys_inherit_str / \l_keys_key_str } {#1}
      }
  }
\cs_new:Npn \__keys_choice_find:nn #1#2
  {
    \cs_if_exist:cTF { \c__keys_code_root_str #1 / \__keys_trim_spaces:n {#2} }
      { \__keys_execute:nn { #1 / \__keys_trim_spaces:n {#2} } {#2} }
      { \__keys_execute:nn { #1 / unknown } {#2} }
  }
\cs_new:Npn \__keys_multichoice_find:n #1
  { \clist_map_function:nN {#1} \__keys_choice_find:n }
\cs_new:Npn \__keys_parent:o #1
  {
    \exp_after:wN \__keys_parent_auxi:w #1 \q_nil \__keys_parent_auxii:w
      / \q_nil \__keys_parent_auxiv:w
  }
\cs_new:Npn \__keys_parent_auxi:w #1 / #2 \q_nil #3
  {
    #3 { #1 } #2 \q_nil #3
  }
\cs_new:Npn \__keys_parent_auxii:w #1 #2 \q_nil \__keys_parent_auxii:w
  {
    #1 \__keys_parent_auxi:w #2 \q_nil \__keys_parent_auxiii:n
  }
\cs_new:Npn \__keys_parent_auxiii:n #1
  {
    / #1 \__keys_parent_auxi:w
  }
\cs_new:Npn \__keys_parent_auxiv:w #1 \q_nil \__keys_parent_auxiv:w
  {
  }
\group_begin:
  \cs_set:Npn \__keys_tmp:w #1
    {
      \cs_new:Npn \__keys_trim_spaces:n ##1
        {
          \exp_after:wN \__keys_trim_spaces_auxi:w \tl_to_str:n { / ##1 } /
            \s__keys_nil  \__keys_trim_spaces_auxi:w
            \s__keys_mark \__keys_trim_spaces_auxii:w
            #1 / #1
            \s__keys_nil  \__keys_trim_spaces_auxii:w
            \s__keys_mark \__keys_trim_spaces_auxiii:w
        }
    }
  \__keys_tmp:w { ~ }
\group_end:
\cs_new:Npn \__keys_trim_spaces_auxi:w #1 ~ / #2 \s__keys_nil #3
  {
    #3 #1 / #2 \s__keys_nil #3
  }
\cs_new:Npn \__keys_trim_spaces_auxii:w #1 / ~ #2 \s__keys_mark #3
  {
    #3 #1 / #2 \s__keys_mark #3
  }
\cs_new:Npn \__keys_trim_spaces_auxiii:w
    / #1 /
    \s__keys_nil  \__keys_trim_spaces_auxi:w
    \s__keys_mark \__keys_trim_spaces_auxii:w
    /
    \s__keys_nil  \__keys_trim_spaces_auxii:w
    \s__keys_mark \__keys_trim_spaces_auxiii:w
  {
    #1
  }
\prg_new_conditional:Npnn \keys_if_exist:nn #1#2 { p , T , F , TF }
```

```
  {
    \__keys_if_exist:ee
      { \__keys_trim_spaces:n {#1} }
      { \__keys_trim_spaces:n {#2} }
  }
\prg_generate_conditional_variant:Nnn \keys_if_exist:nn { ne } { p , T , F , TF }
\cs_new:Npn \__keys_if_exist:nn #1#2
  {
    \cs_if_exist:cTF
      { \c__keys_code_root_str #1 \tl_if_blank:nF {#1} { / } #2 }
      { \prg_return_true: }
      { \prg_return_false: }
  }
\cs_generate_variant:Nn \__keys_if_exist:nn { ee }
\prg_new_conditional:Npnn \keys_if_choice_exist:nnn #1#2#3
  { p , T , F , TF }
  {
    \__keys_if_exist:eee
      { \__keys_trim_spaces:n {#1} }
      { \__keys_trim_spaces:n {#2} }
      { \__keys_trim_spaces:n {#3} }
  }
\cs_new:Npn \__keys_if_exist:nnn #1#2#3
  {
    \cs_if_exist:cTF
      {
        \c__keys_code_root_str
          #1 \tl_if_blank:nF {#1} { / }
          #2 \tl_if_blank:nF {#2} { / }
          #3
      }
      { \prg_return_true: }
      { \prg_return_false: }
  }
\cs_generate_variant:Nn \__keys_if_exist:nnn { eee }
\cs_new_protected:Npn \keys_show:nn
  { \__keys_show:Nnn \msg_show:nneeee }
\cs_new_protected:Npn \keys_log:nn
  { \__keys_show:Nnn \msg_log:nneeee }
\cs_new_protected:Npn \__keys_show:Nnn #1#2#3
  {
    \__keys_show_aux:Nee
      #1
      { \__keys_trim_spaces:n {#2} }
      { \__keys_trim_spaces:n {#3} }
  }
```

```
\cs_new_protected:Npn \__keys_show_aux:Nnn #1#2#3
  {
    #1 { keys } { show-key }
      { #2 \tl_if_blank:nF {#2} { / } #3 }
      {
        \keys_if_exist:nnT {#2} {#3}
          {
            \exp_args:Nnf \msg_show_item_unbraced:nn { code }
              {
                \exp_args:Ne \__keys_show:n
                  {
                    \exp_args:Nc \cs_replacement_spec:N
                    {
                      \c__keys_code_root_str
                      #2 \tl_if_blank:nF {#2} { / } #3
                    }
                  }
              }
          }
      }
      { } { }
  }
\cs_generate_variant:Nn \__keys_show_aux:Nnn { Nee }
\cs_new:Npe \__keys_show:n #1
  {
    \exp_not:N \__keys_show:w
      #1
      \tl_to_str:n { \__keys_precompile:n }
      #1
      \tl_to_str:n { \__keys_precompile:n }
      \exp_not:N \s__keys_stop
  }
\use:e
  {
    \cs_new:Npn \exp_not:N \__keys_show:w
      #1 \tl_to_str:n { \__keys_precompile:n }
      #2 \tl_to_str:n { \__keys_precompile:n }
      #3 \exp_not:N \s__keys_stop
  }
  {
    \tl_if_blank:nTF {#2}
      {#1}
      { \__keys_show:Nw #2 \s__keys_stop }
  }
\use:e
  {
    \cs_new:Npn \exp_not:N \__keys_show:Nw #1#2
      \c_right_brace_str \exp_not:N \s__keys_stop
  }
  {#2}
\msg_new:nnnn { keys } { bad-relative-key-path }
  { The~key~'#1'~is~not~inside~the~'#2'~path. }
  { The~key~'#1'~cannot~be~expressed~relative~to~path~'#2'. }
\msg_new:nnnn { keys } { boolean-values-only }
  { Key~'#1'~accepts~boolean~values~only. }
  { The~key~'#1'~only~accepts~the~values~'true'~and~'false'. }
\msg_new:nnnn { keys } { choice-unknown }
  { Key~'#1'~accepts~only~a~fixed~set~of~choices. }
  {
    The~key~'#1'~only~accepts~predefined~values,~
    and~'#2'~is~not~one~of~these.
  }
\msg_new:nnnn { keys } { unknown }
  { The~key~'#1'~is~unknown~and~is~being~ignored. }
  {
    The~module~'#2'~does~not~have~a~key~called~'#1'.\\
    Check~that~you~have~spelled~the~key~name~correctly.
  }
\msg_new:nnnn { keys } { nested-choice-key }
  { Attempt~to~define~'#1'~as~a~nested~choice~key. }
  {
    The~key~'#1'~cannot~be~defined~as~a~choice~as~the~parent~key~'#2'~is~
    itself~a~choice.
  }
\msg_new:nnnn { keys } { value-forbidden }
  { The~key~'#1'~does~not~take~a~value. }
  {
    The~key~'#1'~should~be~given~without~a~value.\\
    The~value~'#2'~was~present:~the~key~will~be~ignored.
  }
\msg_new:nnnn { keys } { value-required }
  { The~key~'#1'~requires~a~value. }
  {
    The~key~'#1'~must~have~a~value.\\
    No~value~was~present:~the~key~will~be~ignored.
  }
\msg_new:nnn { keys } { show-key }
  {
    The~key~#1~
    \tl_if_empty:nTF {#2}
      { is~undefined. }
      { has~the~properties: #2 . }
  }
\prop_gput:Nnn \g_msg_module_name_prop { keys } { LaTeX }
\prop_gput:Nnn \g_msg_module_type_prop { keys } { }
%% File: l3intarray.dtx
\msg_new:nnn { kernel } { negative-array-size }
  { Size~of~array~may~not~be~negative:~#1 }
\cs_new_eq:NN \__intarray_sep: \__kernel_int_sep:
\int_new:N \l__intarray_loop_int
\cs_if_exist:NTF \__intarray_gset_count:Nw
  {
    \int_new:N \g__intarray_table_int
    \int_new:N \l__intarray_bad_index_int
    \cs_new_protected:Npn \__intarray_new:N #1
      {
        \__kernel_chk_if_free_cs:N #1
        \int_gincr:N \g__intarray_table_int
        \cs_gset_nopar:Npe #1 { \__intarray:w \int_use:N \g__intarray_table_int \c_space_tl }
      }
    \cs_new_protected:Npn \intarray_new:Nn #1#2
      {
        \__intarray_new:N #1
        \__intarray_gset_count:Nw #1 \int_eval:n {#2} \scan_stop:
        \int_compare:nNnT { \intarray_count:N #1 } < 0
          {
            \msg_error:nne { kernel } { negative-array-size }
              { \intarray_count:N #1 }
          }
      }
    \cs_generate_variant:Nn \intarray_new:Nn { c }
    \cs_generate_variant:Nn \intarray_count:N { c }
    \cs_new_protected:Npn \__kernel_intarray_gset:Nnn #1#2#3
      { \__intarray_gset:w #2 #1 #3 \scan_stop: }
    \cs_new_protected:Npn \intarray_gset:Nnn #1#2#3
      {
        \__intarray_gset:wF \int_eval:n {#2} #1 \int_eval:n{#3}
          {
            \msg_error:nneee { kernel } { out-of-bounds }
              { \token_to_str:N #1 } { \int_use:N \l__intarray_bad_index_int } { \intarray_count:N #1 }
          }
      }
    \cs_generate_variant:Nn \intarray_gset:Nnn { c }
    \cs_generate_variant:Nn \intarray_gzero:N { c }
    \cs_new:Npn \__kernel_intarray_item:Nn #1#2
      { \__intarray_item:w #2 #1 }
    \cs_new:Npn \intarray_item:Nn #1#2
      {
        \__intarray_item:wF \int_eval:n {#2} #1
          {
            \msg_expandable_error:nnfff { kernel } { out-of-bounds }
              { \token_to_str:N #1 } { \int_use:N \l__intarray_bad_index_int } { \intarray_count:N #1 }
            0
          }
      }
    \cs_generate_variant:Nn \intarray_item:Nn { c }
    \cs_new:Npn \intarray_rand_item:N #1
      { \intarray_item:Nn #1 { \int_rand:n { \intarray_count:N #1 } } }
    \cs_generate_variant:Nn \intarray_rand_item:N { c }
    \cs_new_protected:Npn \intarray_const_from_clist:Nn #1#2
      {
        \__intarray_new:N #1
        \int_zero:N \l__intarray_loop_int
        \clist_map_inline:nn {#2}
          {
            \int_incr:N \l__intarray_loop_int
            \__kernel_intarray_gset:Nnn #1 \l__intarray_loop_int { \int_eval:n {##1} } }
      }
    \cs_generate_variant:Nn \intarray_const_from_clist:Nn { c }
    \cs_new:Npn \__kernel_intarray_range_to_clist:Nnn #1#2#3
      {
        \__intarray_range_to_clist:w #1
        \int_eval:n {#2} ~ \int_eval:n {#3} ~
      }
    \cs_new_protected:Npn \__kernel_intarray_gset_range_from_clist:Nnn #1#2#3
      {
        \__intarray_gset_range:w \int_eval:w #2 #1 #3 , , \scan_stop:
      }
    \cs_new_protected:Npn \__intarray_gset_overflow_test:nw #1
    {
    }
  }
  {
    \cs_new_eq:NN \__intarray_entry:w \tex_fontdimen:D
    \cs_new_eq:NN \__intarray_count:w \tex_hyphenchar:D
    \dim_const:Nn \c__intarray_sp_dim { 1 sp }
    \int_new:N \g__intarray_font_int
    \cs_new_protected:Npn \__intarray_new:N #1
      {
        \__kernel_chk_if_free_cs:N #1
        \int_gincr:N \g__intarray_font_int
        \tex_global:D \tex_font:D #1
          = cmr10~at~ \g__intarray_font_int \c__intarray_sp_dim \scan_stop:
        \int_step_inline:nn { 8 }
          { \__kernel_intarray_gset:Nnn #1 {##1} \c_zero_int }
      }
    \cs_new_protected:Npn \intarray_new:Nn #1#2
      {
        \__intarray_new:N #1
        \__intarray_count:w #1 = \int_eval:n {#2} \scan_stop:
        \int_compare:nNnT { \intarray_count:N #1 } < 0
          {
            \msg_error:nne { kernel } { negative-array-size }
              { \intarray_count:N #1 }
          }
        \int_compare:nNnT { \intarray_count:N #1 } > 0
          { \__kernel_intarray_gset:Nnn #1 { \intarray_count:N #1 } { 0 } }
      }
    \cs_generate_variant:Nn \intarray_new:Nn { c }
    \cs_new:Npn \intarray_count:N #1 { \int_value:w \__intarray_count:w #1 }
    \cs_generate_variant:Nn \intarray_count:N { c }
    \cs_new:Npn \__intarray_signed_max_dim:n #1
      { \int_value:w \int_compare:nNnT {#1} < 0 { - } \c_max_dim }
    \cs_new:Npn \__intarray_bounds:NNnTF #1#2#3
      {
        \if_int_compare:w 1 > #3 \exp_stop_f:
          \__intarray_bounds_error:NNnw #1 #2 {#3}
        \else:
          \if_int_compare:w #3 > \intarray_count:N #2 \exp_stop_f:
            \__intarray_bounds_error:NNnw #1 #2 {#3}
          \fi:
        \fi:
        \use_i:nn
      }
    \cs_new:Npn \__intarray_bounds_error:NNnw #1#2#3#4 \use_i:nn #5#6
      {
        #4
        #1 { kernel } { out-of-bounds }
          { \token_to_str:N #2 } {#3} { \intarray_count:N #2 }
        #6
      }
    \cs_new_protected:Npn \__kernel_intarray_gset:Nnn #1#2#3
      { \__intarray_entry:w #2 #1 #3 \c__intarray_sp_dim }
    \cs_new_protected:Npn \intarray_gset:Nnn #1#2#3
      {
        \exp_after:wN \__intarray_gset:Nww
        \exp_after:wN #1
        \int_value:w \int_eval:n {#2} \exp_after:wN \__intarray_sep:
        \int_value:w \int_eval:n {#3} \__intarray_sep:
      }
    \cs_generate_variant:Nn \intarray_gset:Nnn { c }
    \cs_new_protected:Npn \__intarray_gset:Nww #1#2 \__intarray_sep: #3 \__intarray_sep:
      {
        \__intarray_bounds:NNnTF \msg_error:nneee #1 {#2}
          {
            \__intarray_gset_overflow_test:nw {#3}
            \__kernel_intarray_gset:Nnn #1 {#2} {#3}
          }
          { }
      }
    \cs_if_exist:NTF \tex_ifabsnum:D
      {
        \cs_new_protected:Npn \__intarray_gset_overflow_test:nw #1
          {
            \tex_ifabsnum:D #1 > \c_max_dim
              \exp_after:wN \__intarray_gset_overflow:NNnn
            \fi:
          }
      }
      {
        \cs_new_protected:Npn \__intarray_gset_overflow_test:nw #1
          {
            \if_int_compare:w \int_abs:n {#1} > \c_max_dim
              \exp_after:wN \__intarray_gset_overflow:NNnn
            \fi:
          }
      }
    \cs_new_protected:Npn \__intarray_gset_overflow:NNnn #1#2#3#4
      {
        \msg_error:nneeee { kernel } { overflow }
          { \token_to_str:N #2 } {#3} {#4} {  \__intarray_signed_max_dim:n {#4} }
        #1 #2 {#3} { \__intarray_signed_max_dim:n {#4} }
      }
    \cs_new_protected:Npn \intarray_gzero:N #1
      {
        \int_zero:N \l__intarray_loop_int
        \prg_replicate:nn { \intarray_count:N #1 }
          {
            \int_incr:N \l__intarray_loop_int
            \__intarray_entry:w \l__intarray_loop_int #1 \c_zero_dim
          }
      }
    \cs_generate_variant:Nn \intarray_gzero:N { c }
```

```
\cs_new:Npn \__kernel_intarray_item:Nn #1#2
  { \int_value:w \__intarray_entry:w #2 #1 }
\cs_new:Npn \intarray_item:Nn #1#2
  {
    \exp_after:wN \__intarray_item:Nw
    \exp_after:wN #1
    \int_value:w \int_eval:n {#2} \__intarray_sep:
  }
\cs_generate_variant:Nn \intarray_item:Nn { c }
\cs_new:Npn \__intarray_item:Nw #1#2 \__intarray_sep:
  {
    \__intarray_bounds:NNnTF \msg_expandable_error:nnfff #1 {#2}
      { \__kernel_intarray_item:Nn #1 {#2} }
      { 0 }
  }
\cs_new:Npn \intarray_rand_item:N #1
  { \intarray_item:Nn #1 { \int_rand:n { \intarray_count:N #1 } } }
\cs_generate_variant:Nn \intarray_rand_item:N { c }
\cs_new_protected:Npn \intarray_const_from_clist:Nn #1#2
  {
    \__intarray_new:N #1
    \int_zero:N \l__intarray_loop_int
    \clist_map_inline:nn {#2}
      { \exp_args:Nf \__intarray_const_from_clist:nN { \int_eval:n {##1} } #1 }
    \__intarray_count:w #1 \l__intarray_loop_int
  }
\cs_generate_variant:Nn \intarray_const_from_clist:Nn { c }
\cs_new_protected:Npn \__intarray_const_from_clist:nN #1#2
  {
    \int_incr:N \l__intarray_loop_int
    \__intarray_gset_overflow_test:nw {#1}
    \__kernel_intarray_gset:Nnn #2 \l__intarray_loop_int {#1}
  }
\cs_new:Npn \__intarray_to_clist:Nn #1#2
  {
    \int_compare:nNnF { \intarray_count:N #1 } = \c_zero_int
      {
        \exp_last_unbraced:Nf \use_none:n
          { \__intarray_to_clist:w 1 \__intarray_sep: #1 {#2} \prg_break_point: }
      }
  }
\cs_new:Npn \__intarray_to_clist:w #1 \__intarray_sep: #2#3
  {
    \if_int_compare:w #1 > \__intarray_count:w #2
      \prg_break:n
    \fi:
```

```
    #3 \__kernel_intarray_item:Nn #2 {#1}
    \exp_after:wN \__intarray_to_clist:w
    \int_value:w \int_eval:w #1 + \c_one_int \__intarray_sep: #2 {#3}
   }
  \cs_new:Npn \__kernel_intarray_range_to_clist:Nnn #1#2#3
   {
    \exp_last_unbraced:Nf \use_none:n
     {
      \exp_after:wN \__intarray_range_to_clist:ww
      \int_value:w \int_eval:w #2 \exp_after:wN \__intarray_sep:
      \int_value:w \int_eval:w #3 \__intarray_sep:
      #1 \prg_break_point:
     }
   }
  \cs_new:Npn \__intarray_range_to_clist:ww #1 \__intarray_sep: #2 \__intarray_sep: #3
   {
    \if_int_compare:w #1 > #2 \exp_stop_f:
     \prg_break:n
    \fi:
    , \__kernel_intarray_item:Nn #3 {#1}
    \exp_after:wN \__intarray_range_to_clist:ww
    \int_value:w \int_eval:w #1 + \c_one_int \__intarray_sep: #2 \__intarray_sep: #3
   }
  \cs_new_protected:Npn \__kernel_intarray_gset_range_from_clist:Nnn #1#2#3
   {
    \int_set:Nn \l__intarray_loop_int {#2}
    \__intarray_gset_range:Nw #1 #3 , , \prg_break_point:
   }
  \cs_new_protected:Npn \__intarray_gset_range:Nw #1 #2 ,
   {
    \if_catcode:w \scan_stop: \tl_to_str:n {#2} \scan_stop:
     \prg_break:n
    \fi:
    \__kernel_intarray_gset:Nnn #1 \l__intarray_loop_int {#2}
    \int_incr:N \l__intarray_loop_int
    \__intarray_gset_range:Nw #1
   }
 }
\prg_new_eq_conditional:NNn \intarray_if_exist:N \cs_if_exist:N
 { TF , T , F , p }
\prg_new_eq_conditional:NNn \intarray_if_exist:c \cs_if_exist:c
 { TF , T , F , p }
\cs_new_protected:Npn \intarray_show:N { \__intarray_show:NN \msg_show:nneeee }
\cs_generate_variant:Nn \intarray_show:N { c }
\cs_new_protected:Npn \intarray_log:N { \__intarray_show:NN \msg_log:nneeee }
\cs_generate_variant:Nn \intarray_log:N { c }
```

```
\cs_new_protected:Npn \__intarray_show:NN #1#2
  {
    \__kernel_chk_defined:NT #2
      {
        #1 { intarray } { show }
          { \token_to_str:N #2 }
          { \intarray_count:N #2 }
          { >~ \__intarray_to_clist:Nn #2 { , ~ } }
          { }
      }
  }
%% File: l3fp.dtx
%% File: l3fp-aux.dtx
\cs_new_eq:NN \__fp_int_eval:w \tex_numexpr:D
\cs_new_eq:NN \__fp_int_eval_end: \scan_stop:
\cs_new_eq:NN \__fp_int_to_roman:w \tex_romannumeral:D
\cs_new_eq:NN \__fp_sep: \__kernel_int_sep:
\cs_new:Npn \__fp_use_none_stop_f:n #1 { \exp_stop_f: }
\cs_new:Npn \__fp_use_s:n #1 { #1\__fp_sep: }
\cs_new:Npn \__fp_use_s:nn #1#2 { #1#2\__fp_sep: }
\cs_new:Npn \__fp_use_none_until_s:w #1\__fp_sep: { }
\cs_new:Npn \__fp_use_i_until_s:nw #1#2\__fp_sep: {#1}
\cs_new:Npn \__fp_use_ii_until_s:nnw #1#2#3\__fp_sep: {#2}
\cs_new:Npn \__fp_reverse_args:Nww #1 #2\__fp_sep: #3\__fp_sep:
  { #1 #3\__fp_sep: #2\__fp_sep: }
\cs_new:Npn \__fp_rrot:www #1\__fp_sep: #2\__fp_sep: #3\__fp_sep:
  { #2\__fp_sep: #3\__fp_sep: #1\__fp_sep: }
\cs_new:Npn \__fp_use_i:ww #1\__fp_sep: #2\__fp_sep: { #1\__fp_sep: }
\cs_new:Npn \__fp_use_i:www #1\__fp_sep: #2\__fp_sep: #3\__fp_sep: { #1\__fp_sep: }
\cs_new_protected:Npn \__fp_misused:n #1
  { \msg_error:nne { fp } { misused } { \fp_to_tl:n {#1} } }
\scan_new:N \s__fp
\cs_new_protected:Npn \__fp_chk:w #1 \__fp_sep:
  { \__fp_misused:n { \s__fp \__fp_chk:w #1 \__fp_sep: } }
\scan_new:N \s__fp_expr_mark
\scan_new:N \s__fp_expr_stop
\scan_new:N \s__fp_mark
\scan_new:N \s__fp_stop
\cs_new:Npn \__fp_use_i_delimit_by_s_stop:nw #1 #2 \s__fp_stop {#1}
\scan_new:N \s__fp_invalid
\scan_new:N \s__fp_underflow
\scan_new:N \s__fp_overflow
\scan_new:N \s__fp_division
\scan_new:N \s__fp_exact
\tl_const:Nn \c_zero_fp       { \s__fp \__fp_chk:w 0 0 \s__fp_exact \__fp_sep: }
\tl_const:Nn \c_minus_zero_fp { \s__fp \__fp_chk:w 0 2 \s__fp_exact \__fp_sep: }
```

```
\tl_const:Nn \c_inf_fp        { \s__fp \__fp_chk:w 2 0 \s__fp_exact \__fp_sep: }
\tl_const:Nn \c_minus_inf_fp  { \s__fp \__fp_chk:w 2 2 \s__fp_exact \__fp_sep: }
\tl_const:Nn \c_nan_fp        { \s__fp \__fp_chk:w 3 1 \s__fp_exact \__fp_sep: }
\int_const:Nn \c__fp_prec_int { 16 }
\int_const:Nn \c__fp_half_prec_int { 8 }
\int_const:Nn \c__fp_block_int { 4 }
\int_const:Nn \c__fp_myriad_int { 10000 }
\int_const:Nn \c__fp_minus_min_exponent_int { 10000 }
\int_const:Nn \c__fp_max_exponent_int { 10000 }
\int_const:Nn \c__fp_max_exp_exponent_int { 5 }
\tl_const:Ne \c__fp_overflowing_fp
  {
    \s__fp \__fp_chk:w 1 0
      { \int_eval:n { \c__fp_max_exponent_int + 1 } }
      {1000} {0000} {0000} {0000} \__fp_sep:
  }
\cs_new:Npn \__fp_zero_fp:N #1
  { \s__fp \__fp_chk:w 0 #1 \s__fp_underflow \__fp_sep: }
\cs_new:Npn \__fp_inf_fp:N #1
  { \s__fp \__fp_chk:w 2 #1 \s__fp_overflow \__fp_sep: }
\cs_new:Npn \__fp_exponent:w \s__fp \__fp_chk:w #1
  {
    \if_meaning:w 1 #1
      \exp_after:wN \__fp_use_ii_until_s:nnw
    \else:
      \exp_after:wN \__fp_use_i_until_s:nw
      \exp_after:wN 0
    \fi:
  }
\cs_new:Npn \__fp_neg_sign:N #1
  { \__fp_int_eval:w 2 - #1 \__fp_int_eval_end: }
\cs_new:Npn \__fp_kind:w #1
  {
    \__fp_if_type_fp:NTwFw
      #1 \__fp_use_ii_until_s:nnw
      \s__fp { \__fp_use_i_until_s:nw 4 }
      \s__fp_stop
  }
\cs_new:Npn \__fp_sanitize:Nw #1 #2\__fp_sep:
  {
    \if_case:w
        \if_int_compare:w #2 > \c__fp_max_exponent_int 1 ~ \else:
        \if_int_compare:w #2 < - \c__fp_minus_min_exponent_int 2 ~ \else:
        \if_meaning:w 1 #1 3 ~ \fi: \fi: \fi: 0 ~
    \or: \exp_after:wN \__fp_overflow:w
    \or: \exp_after:wN \__fp_underflow:w
    \or: \exp_after:wN \__fp_sanitize_zero:w
    \fi:
    \s__fp \__fp_chk:w 1 #1 {#2}
  }
\cs_new:Npn \__fp_sanitize:wN #1\__fp_sep: #2 { \__fp_sanitize:Nw #2 #1\__fp_sep: }
\cs_new:Npn \__fp_sanitize_zero:w \s__fp \__fp_chk:w #1 #2 #3\__fp_sep:
  { \c_zero_fp }
\cs_new:Npn \__fp_exp_after_o:w \s__fp \__fp_chk:w #1
  {
    \if_meaning:w 1 #1
      \exp_after:wN \__fp_exp_after_normal:nNNw
    \else:
      \exp_after:wN \__fp_exp_after_special:nNNw
    \fi:
    { }
    #1
  }
\cs_new:Npn \__fp_exp_after_f:nw #1 \s__fp \__fp_chk:w #2
  {
    \if_meaning:w 1 #2
      \exp_after:wN \__fp_exp_after_normal:nNNw
    \else:
      \exp_after:wN \__fp_exp_after_special:nNNw
    \fi:
    { \exp:w \exp_end_continue_f:w #1 }
    #2
  }
\cs_new:Npn \__fp_exp_after_special:nNNw #1#2#3#4\__fp_sep:
  {
    \exp_after:wN \s__fp
    \exp_after:wN \__fp_chk:w
    \exp_after:wN #2
    \exp_after:wN #3
    \exp_after:wN #4
    \exp_after:wN \__fp_sep:
    #1
  }
\cs_new:Npn \__fp_exp_after_normal:nNNw #1 1 #2 #3 #4#5#6#7\__fp_sep:
  {
    \exp_after:wN \__fp_exp_after_normal:Nwwwww
    \exp_after:wN #2
    \int_value:w #3   \exp_after:wN \__fp_sep:
    \int_value:w 1 #4 \exp_after:wN \__fp_sep:
    \int_value:w 1 #5 \exp_after:wN \__fp_sep:
    \int_value:w 1 #6 \exp_after:wN \__fp_sep:
    \int_value:w 1 #7 \exp_after:wN \__fp_sep: #1
  }
\cs_new:Npn \__fp_exp_after_normal:Nwwwww
    #1 #2\__fp_sep: 1 #3 \__fp_sep: 1 #4 \__fp_sep: 1 #5 \__fp_sep: 1 #6 \__fp_sep:
  { \s__fp \__fp_chk:w 1 #1 {#2} {#3} {#4} {#5} {#6} \__fp_sep: }
\scan_new:N \s__fp_tuple
\cs_new_protected:Npn \__fp_tuple_chk:w #1 \__fp_sep:
  { \__fp_misused:n { \s__fp_tuple \__fp_tuple_chk:w #1 \__fp_sep: } }
\tl_const:Nn \c__fp_empty_tuple_fp
  { \s__fp_tuple \__fp_tuple_chk:w { } \__fp_sep: }
\cs_new:Npn \__fp_array_count:n #1
  { \__fp_tuple_count:w \s__fp_tuple \__fp_tuple_chk:w {#1} \__fp_sep: }
\cs_new:Npn \__fp_tuple_count:w \s__fp_tuple \__fp_tuple_chk:w #1 \__fp_sep:
  {
    \int_value:w \__fp_int_eval:w 0
      \__fp_tuple_count_loop:Nw #1 { ? \prg_break: } \__fp_sep:
      \prg_break_point:
    \__fp_int_eval_end:
  }
\cs_new:Npn \__fp_tuple_count_loop:Nw #1#2\__fp_sep:
  { \use_none:n #1 + 1 \__fp_tuple_count_loop:Nw }
\cs_new:Npn \__fp_if_type_fp:NTwFw #1 \s__fp #2 #3 \s__fp_stop {#2}
\cs_new:Npn \__fp_array_if_all_fp:nTF #1
  {
    \__fp_array_if_all_fp_loop:w #1 { \s__fp \prg_break: } \__fp_sep:
    \prg_break_point: \use_i:nn
  }
\cs_new:Npn \__fp_array_if_all_fp_loop:w #1#2 \__fp_sep:
  {
    \__fp_if_type_fp:NTwFw
      #1 \__fp_array_if_all_fp_loop:w
      \s__fp { \prg_break:n \use_iii:nnn }
      \s__fp_stop
  }
\cs_new:Npn \__fp_type_from_scan:N #1
  {
    \__fp_if_type_fp:NTwFw
      #1 { }
      \s__fp { \__fp_type_from_scan_other:N #1 }
      \s__fp_stop
  }
\cs_new:Npe \__fp_type_from_scan_other:N #1
  {
    \exp_not:N \exp_after:wN \exp_not:N \__fp_type_from_scan:w
    \exp_not:N \token_to_str:N #1 \s__fp_mark
      \tl_to_str:n { s__fp _? } \s__fp_mark \s__fp_stop
  }
```

```
\exp_last_unbraced:NNNNo
  \cs_new:Npn \__fp_type_from_scan:w #1
    { \tl_to_str:n { s__fp } } #2 \s__fp_mark #3 \s__fp_stop {#2}
\cs_new:Npn \__fp_change_func_type:NNN #1#2#3
  {
    \__fp_if_type_fp:NTwFw
      #1 #2
      \s__fp
        {
          \exp_after:wN \__fp_change_func_type_chk:NNN
          \cs:w
            __fp \__fp_type_from_scan_other:N #1
            \exp_after:wN \__fp_change_func_type_aux:w \token_to_str:N #2
          \cs_end:
          #2 #3
        }
      \s__fp_stop
  }
\exp_last_unbraced:NNNNo
  \cs_new:Npn \__fp_change_func_type_aux:w #1 { \tl_to_str:n { __fp } } { }
\cs_new:Npn \__fp_change_func_type_chk:NNN #1#2#3
  {
    \if_meaning:w \scan_stop: #1
      \exp_after:wN #3 \exp_after:wN #2
    \else:
      \exp_after:wN #1
    \fi:
  }
\cs_new:Npn \__fp_exp_after_any_f:Nnw #1
  { \cs:w __fp_exp_after \__fp_type_from_scan_other:N #1 _f:nw \cs_end: }
\cs_new:Npn \__fp_exp_after_any_f:nw #1#2
  {
    \__fp_if_type_fp:NTwFw
      #2 \__fp_exp_after_f:nw
      \s__fp { \__fp_exp_after_any_f:Nnw #2 }
      \s__fp_stop
    {#1} #2
  }
\cs_new_eq:NN \__fp_exp_after_expr_stop_f:nw \use_none:nn
\cs_new:Npn \__fp_exp_after_tuple_o:w
  { \__fp_exp_after_tuple_f:nw { \exp_after:wN \exp_stop_f: } }
\cs_new:Npn \__fp_exp_after_tuple_f:nw
  #1 \s__fp_tuple \__fp_tuple_chk:w #2 \__fp_sep:
  {
    \exp_after:wN \s__fp_tuple
    \exp_after:wN \__fp_tuple_chk:w
    \exp_after:wN {
      \exp:w \exp_end_continue_f:w
      \__fp_exp_after_array_f:w #2 \s__fp_expr_stop
    \exp_after:wN }
    \exp_after:wN \__fp_sep:
    \exp:w \exp_end_continue_f:w #1
  }
\cs_new:Npn \__fp_exp_after_array_f:w
  { \__fp_exp_after_any_f:nw { \__fp_exp_after_array_f:w } }
\int_const:Nn \c__fp_leading_shift_int  { - 5 0000 }
\int_const:Nn \c__fp_middle_shift_int   { 5 0000 *  9999 }
\int_const:Nn \c__fp_trailing_shift_int { 5 0000 * 10000 }
\cs_new:Npn \__fp_pack:NNNNNw #1 #2#3#4#5 #6\__fp_sep:
  { + #1#2#3#4#5 \__fp_sep: {#6} }
\int_const:Nn \c__fp_big_leading_shift_int  { - 15 2374 }
\int_const:Nn \c__fp_big_middle_shift_int   { 15 2374 *  9999 }
\int_const:Nn \c__fp_big_trailing_shift_int { 15 2374 * 10000 }
\cs_new:Npn \__fp_pack_big:NNNNNNw #1#2 #3#4#5#6 #7\__fp_sep:
  { + #1#2#3#4#5#6 \__fp_sep: {#7} }
\int_const:Nn \c__fp_Bigg_leading_shift_int  { - 20 0000 }
\int_const:Nn \c__fp_Bigg_middle_shift_int   { 20 0000 *  9999 }
\int_const:Nn \c__fp_Bigg_trailing_shift_int { 20 0000 * 10000 }
\cs_new:Npn \__fp_pack_Bigg:NNNNNNw #1#2 #3#4#5#6 #7\__fp_sep:
  { + #1#2#3#4#5#6 \__fp_sep: {#7} }
\cs_new:Npn \__fp_pack_twice_four:wNNNNNNNN #1\__fp_sep: #2#3#4#5 #6#7#8#9
  { #1 {#2#3#4#5} {#6#7#8#9} \__fp_sep: }
\cs_new:Npn \__fp_pack_eight:wNNNNNNNN #1\__fp_sep: #2#3#4#5 #6#7#8#9
  { #1 {#2#3#4#5#6#7#8#9} \__fp_sep: }
\cs_new:Npn \__fp_basics_pack_low:NNNNNw #1 #2#3#4#5 #6\__fp_sep:
  { + #1 - 1 \__fp_sep: {#2#3#4#5} {#6} \__fp_sep: }
\cs_new:Npn \__fp_basics_pack_high:NNNNNw #1 #2#3#4#5 #6\__fp_sep:
  {
    \if_meaning:w 2 #1
      \__fp_basics_pack_high_carry:w
    \fi:
    \__fp_sep: {#2#3#4#5} {#6}
  }
\cs_new:Npn \__fp_basics_pack_high_carry:w \fi: \__fp_sep: #1
  { \fi: + 1 \__fp_sep: {1000} }
\cs_new:Npn \__fp_basics_pack_weird_low:NNNNw #1 #2#3#4 #5\__fp_sep:
  {
    \if_meaning:w 2 #1
      + 1
    \fi:
    \__fp_int_eval_end:
    #2#3#4\__fp_sep: {#5} \__fp_sep:
  }
\cs_new:Npn \__fp_basics_pack_weird_high:NNNNNNNNw
  1 #1#2#3#4 #5#6#7#8 #9\__fp_sep: { \__fp_sep: {#1#2#3#4} {#5#6#7#8} {#9} }
\cs_new:Npn \__fp_decimate:nNnnnn #1
  {
    \cs:w
      __fp_decimate_
      \if_int_compare:w \__fp_int_eval:w #1 > \c__fp_prec_int
        tiny
      \else:
        \__fp_int_to_roman:w \__fp_int_eval:w #1
      \fi:
      :Nnnnn
    \cs_end:
  }
\cs_new:Npn \__fp_decimate_:Nnnnn #1 #2#3#4#5
  { #1 0 {#2#3} {#4#5} \__fp_sep: }
\cs_new:Npn \__fp_decimate_tiny:Nnnnn #1 #2#3#4#5
  { #1 1 { 0000 0000 } { 0000 0000 } 0 #2#3#4#5 \__fp_sep: }
\cs_new:Npn \__fp_tmp:w #1 #2 #3
  {
    \cs_new:cpn { __fp_decimate_ #1 :Nnnnn } ##1 ##2##3##4##5
      {
        \exp_after:wN ##1
        \int_value:w
          \exp_after:wN \__fp_round_digit:Nw #2 \__fp_sep:
        \__fp_decimate_pack:nnnnnnnnnnw #3 \__fp_sep:
      }
  }
\__fp_tmp:w {i}   {\use_none:nnn      #50}{    0{#2}#3{#4}#5               }
\__fp_tmp:w {ii}  {\use_none:nn       #5 }{    00{#2}#3{#4}#5              }
\__fp_tmp:w {iii} {\use_none:n        #5 }{    000{#2}#3{#4}#5             }
\__fp_tmp:w {iv}  {                   #5 }{   {0000}#2{#3}#4 #5            }
\__fp_tmp:w {v}   {\use_none:nnn    #4#5 }{   0{0000}#2{#3}#4 #5           }
\__fp_tmp:w {vi}  {\use_none:nn     #4#5 }{   00{0000}#2{#3}#4 #5          }
\__fp_tmp:w {vii} {\use_none:n      #4#5 }{   000{0000}#2{#3}#4 #5         }
\__fp_tmp:w {viii}{                 #4#5 }{  {0000}0000{#2}#3 #4 #5        }
\__fp_tmp:w {ix}  {\use_none:nnn  #3#4+#5}{  0{0000}0000{#2}#3 #4 #5       }
\__fp_tmp:w {x}   {\use_none:nn   #3#4+#5}{  00{0000}0000{#2}#3 #4 #5      }
\__fp_tmp:w {xi}  {\use_none:n    #3#4+#5}{  000{0000}0000{#2}#3 #4 #5     }
\__fp_tmp:w {xii} {               #3#4+#5}{ {0000}0000{0000}#2 #3 #4 #5    }
\__fp_tmp:w {xiii}{\use_none:nnn#2#3+#4#5}{ 0{0000}0000{0000}#2 #3 #4 #5   }
\__fp_tmp:w {xiv} {\use_none:nn #2#3+#4#5}{ 00{0000}0000{0000}#2 #3 #4 #5  }
\__fp_tmp:w {xv}  {\use_none:n  #2#3+#4#5}{ 000{0000}0000{0000}#2 #3 #4 #5 }
\__fp_tmp:w {xvi} {             #2#3+#4#5}{{0000}0000{0000}0000 #2 #3 #4 #5}
\cs_new:Npn \__fp_decimate_pack:nnnnnnnnnnw #1#2#3#4#5
```

```
  { \__fp_decimate_pack:nnnnnnnw { #1#2#3#4#5 } }
\cs_new:Npn \__fp_decimate_pack:nnnnnnnw #1 #2#3#4#5#6
  { {#1} {#2#3#4#5#6} }
\cs_new:Npn \__fp_case_use:nw #1#2 \fi: #3 \s__fp { \fi: #1 \s__fp }
\cs_new:Npn \__fp_case_return:nw #1#2 \fi: #3 \__fp_sep: { \fi: #1 }
\cs_new:Npn \__fp_case_return_o:Nw #1#2 \fi: #3 \s__fp #4 \__fp_sep:
  { \fi: \exp_after:wN #1 }
\cs_new:Npn \__fp_case_return_same_o:w #1 \fi: #2 \s__fp
  { \fi: \__fp_exp_after_o:w \s__fp }
\cs_new:Npn \__fp_case_return_o:Nww #1#2 \fi: #3 \s__fp #4 \__fp_sep: #5 \__fp_sep:
  { \fi: \exp_after:wN #1 }
\cs_new:Npn \__fp_case_return_i_o:ww
    #1 \fi: #2 \s__fp #3 \__fp_sep: \s__fp #4 \__fp_sep:
  { \fi: \__fp_exp_after_o:w \s__fp #3 \__fp_sep: }
\cs_new:Npn \__fp_case_return_ii_o:ww #1 \fi: #2 \s__fp #3 \__fp_sep:
  { \fi: \__fp_exp_after_o:w }
\prg_new_conditional:Npnn \__fp_int:w \s__fp \__fp_chk:w #1 #2 #3 #4\__fp_sep:
  { TF , T , F , p }
  {
    \if_case:w #1 \exp_stop_f:
           \prg_return_true:
    \or:
      \if_charcode:w 0
        \__fp_decimate:nNnnnn { \c__fp_prec_int - #3 }
          \__fp_use_i_until_s:nw #4
        \prg_return_true:
      \else:
        \prg_return_false:
      \fi:
    \else: \prg_return_false:
    \fi:
  }
\cs_new:Npn \__fp_small_int:wTF \s__fp \__fp_chk:w #1#2
  {
    \if_case:w #1 \exp_stop_f:
           \__fp_case_return:nw { \__fp_small_int_true:wTF 0 \__fp_sep: }
    \or:   \exp_after:wN \__fp_small_int_normal:NnwTF
    \or:
      \__fp_case_return:nw
        {
          \exp_after:wN \__fp_small_int_true:wTF \int_value:w
            \if_meaning:w 2 #2 - \fi: 1 0000 0000 \__fp_sep:
        }
    \else: \__fp_case_return:nw \use_ii:nn
    \fi:
    #2
  }
\cs_new:Npn \__fp_small_int_true:wTF #1\__fp_sep: #2#3 { #2 {#1} }
\cs_new:Npn \__fp_small_int_normal:NnwTF #1#2#3\__fp_sep:
  {
    \__fp_decimate:nNnnnn { \c__fp_prec_int - #2 }
      \__fp_small_int_test:NnnwNw
      #3 #1
  }
\cs_new:Npn \__fp_small_int_test:NnnwNw #1#2#3#4\__fp_sep: #5
  {
    \if_meaning:w 0 #1
      \exp_after:wN \__fp_small_int_true:wTF
      \int_value:w \if_meaning:w 2 #5 - \fi:
        \if_int_compare:w #2 > \c_zero_int
          1 0000 0000
        \else:
          #3
        \fi:
      \exp_after:wN \__fp_sep:
    \else:
      \exp_after:wN \use_ii:nn
    \fi:
  }
\cs_new_eq:NN \__fp_str_if_eq:nn \tex_strcmp:D
\cs_new:Npn \__fp_func_to_name:N #1
  {
    \exp_last_unbraced:Nf
      \__fp_func_to_name_aux:w { \cs_to_str:N #1 } X
  }
\cs_set_protected:Npn \__fp_tmp:w #1 #2
  { \cs_new:Npn \__fp_func_to_name_aux:w ##1 #1 ##2 #2 ##3 X {##2} }
\exp_args:Nff \__fp_tmp:w { \tl_to_str:n { __fp_ } }
  { \tl_to_str:n { _o: } }
\msg_new:nnnn { fp } { misused }
  { A~floating~point~with~value~'#1'~was~misused. }
  {
    To~obtain~the~value~of~a~floating~point~variable,~use~
    '\token_to_str:N \fp_to_decimal:N',~
    '\token_to_str:N \fp_to_tl:N',~or~other~
    conversion~functions.
  }
\prop_gput:Nnn \g_msg_module_name_prop { fp } { LaTeX }
\prop_gput:Nnn \g_msg_module_type_prop { fp } { }
%% File: l3fp-traps.dtx
\flag_new:N \l_fp_invalid_operation_flag
\flag_new:N \l_fp_division_by_zero_flag
```

```
\flag_new:N \l_fp_overflow_flag
\flag_new:N \l_fp_underflow_flag
\cs_new_protected:Npn \fp_trap:nn #1#2
  {
    \cs_if_exist_use:cF { __fp_trap_#1_set_#2: }
      {
        \clist_if_in:nnTF
          { invalid_operation , division_by_zero , overflow , underflow }
          {#1}
          {
            \msg_error:nnee { fp }
              { unknown-fpu-trap-type } {#1} {#2}
          }
          {
            \msg_error:nne
              { fp } { unknown-fpu-exception } {#1}
          }
      }
  }
\cs_new_protected:Npn \__fp_trap_invalid_operation_set_error:
  { \__fp_trap_invalid_operation_set:N \prg_do_nothing: }
\cs_new_protected:Npn \__fp_trap_invalid_operation_set_flag:
  { \__fp_trap_invalid_operation_set:N \use_none:nnnnn }
\cs_new_protected:Npn \__fp_trap_invalid_operation_set_none:
  { \__fp_trap_invalid_operation_set:N \use_none:nnnnnnn }
\cs_new_protected:Npn \__fp_trap_invalid_operation_set:N #1
  {
    \exp_args:Nno \use:n
      { \cs_set:Npn \__fp_invalid_operation:nnw ##1##2##3\__fp_sep: }
      {
        #1
        \__fp_error:nnfn { invalid } {##2} { \fp_to_tl:n { ##3\__fp_sep: } } { }
        \flag_ensure_raised:N \l_fp_invalid_operation_flag
        ##1
      }
    \exp_args:Nno \use:n
      { \cs_set:Npn \__fp_invalid_operation_o:Nww ##1##2\__fp_sep: ##3\__fp_sep: }
      {
        #1
        \__fp_error:nffn { invalid-ii }
          { \fp_to_tl:n { ##2\__fp_sep: } }
          { \fp_to_tl:n { ##3\__fp_sep: } }
          {##1}
        \flag_ensure_raised:N \l_fp_invalid_operation_flag
        \exp_after:wN \c_nan_fp
      }
    \exp_args:Nno \use:n
      { \cs_set:Npn \__fp_invalid_operation_tl_o:ff ##1##2 }
      {
        #1
        \__fp_error:nffn { invalid } {##1} {##2} { }
        \flag_ensure_raised:N \l_fp_invalid_operation_flag
        \exp_after:wN \c_nan_fp
      }
  }
\cs_new_protected:Npn \__fp_trap_division_by_zero_set_error:
  { \__fp_trap_division_by_zero_set:N \prg_do_nothing: }
\cs_new_protected:Npn \__fp_trap_division_by_zero_set_flag:
  { \__fp_trap_division_by_zero_set:N \use_none:nnnnn }
\cs_new_protected:Npn \__fp_trap_division_by_zero_set_none:
  { \__fp_trap_division_by_zero_set:N \use_none:nnnnnnn }
\cs_new_protected:Npn \__fp_trap_division_by_zero_set:N #1
  {
    \exp_args:Nno \use:n
      { \cs_set:Npn \__fp_division_by_zero_o:Nnw ##1##2##3\__fp_sep: }
      {
        #1
        \__fp_error:nnfn { zero-div } {##2} { \fp_to_tl:n { ##3\__fp_sep: } } { }
        \flag_ensure_raised:N \l_fp_division_by_zero_flag
        \exp_after:wN ##1
      }
    \exp_args:Nno \use:n
      {
        \cs_set:Npn \__fp_division_by_zero_o:NNww ##1##2##3\__fp_sep: ##4\__fp_sep:
      }
      {
        #1
        \__fp_error:nffn { zero-div-ii }
          { \fp_to_tl:n { ##3\__fp_sep: } }
          { \fp_to_tl:n { ##4\__fp_sep: } }
          {##2}
        \flag_ensure_raised:N \l_fp_division_by_zero_flag
        \exp_after:wN ##1
      }
  }
\cs_new_protected:Npn \__fp_trap_overflow_set_error:
  { \__fp_trap_overflow_set:N \prg_do_nothing: }
\cs_new_protected:Npn \__fp_trap_overflow_set_flag:
  { \__fp_trap_overflow_set:N \use_none:nnnnn }
\cs_new_protected:Npn \__fp_trap_overflow_set_none:
  { \__fp_trap_overflow_set:N \use_none:nnnnnnn }
\cs_new_protected:Npn \__fp_trap_overflow_set:N #1
```

```
  { \__fp_trap_overflow_set:NnNn #1 { overflow } \__fp_inf_fp:N { inf } }
\cs_new_protected:Npn \__fp_trap_underflow_set_error:
  { \__fp_trap_underflow_set:N \prg_do_nothing: }
\cs_new_protected:Npn \__fp_trap_underflow_set_flag:
  { \__fp_trap_underflow_set:N \use_none:nnnnn }
\cs_new_protected:Npn \__fp_trap_underflow_set_none:
  { \__fp_trap_underflow_set:N \use_none:nnnnnnn }
\cs_new_protected:Npn \__fp_trap_underflow_set:N #1
  { \__fp_trap_overflow_set:NnNn #1 { underflow } \__fp_zero_fp:N { 0 } }
\cs_new_protected:Npn \__fp_trap_overflow_set:NnNn #1#2#3#4
  {
    \exp_args:Nno \use:n
      { \cs_set:cpn { __fp_ #2 :w } \s__fp \__fp_chk:w ##1##2##3\__fp_sep: }
      {
        #1
        \__fp_error:nffn
          { flow \if_meaning:w 1 ##1 -to \fi: }
          { \fp_to_tl:n { \s__fp \__fp_chk:w ##1##2##3\__fp_sep: } }
          { \token_if_eq_meaning:NNF 0 ##2 { - } #4 }
          {#2}
        \flag_ensure_raised:c { l_fp_#2_flag }
        #3 ##2
      }
  }
\cs_new:Npn \__fp_invalid_operation:nnw #1#2#3\__fp_sep: { }
\cs_new:Npn \__fp_invalid_operation_o:Nww #1#2\__fp_sep: #3\__fp_sep: { }
\cs_new:Npn \__fp_invalid_operation_tl_o:ff #1 #2 { }
\cs_new:Npn \__fp_division_by_zero_o:Nnw #1#2#3\__fp_sep: { }
\cs_new:Npn \__fp_division_by_zero_o:NNww #1#2#3\__fp_sep: #4\__fp_sep: { }
\cs_new:Npn \__fp_overflow:w { }
\cs_new:Npn \__fp_underflow:w { }
\fp_trap:nn { invalid_operation } { error }
\fp_trap:nn { division_by_zero } { flag }
\fp_trap:nn { overflow } { flag }
\fp_trap:nn { underflow } { flag }
\cs_new:Npn \__fp_invalid_operation_o:nw
  { \__fp_invalid_operation:nnw { \exp_after:wN \c_nan_fp } }
\cs_generate_variant:Nn \__fp_invalid_operation_o:nw { f }
\cs_new:Npn \__fp_error:nnnn
  { \msg_expandable_error:nnnnn { fp } }
\cs_generate_variant:Nn \__fp_error:nnnn { nnf, nff , nfff }
\cs_new:Npn \__fp_error_num_args:nnnn #1#2#3#4
  {
    \int_compare:nNnTF {#2} = {#3}
      { \msg_expandable_error:nnnnn { fp } { num-args-eq } {#1} {#2} {#4} }
      { \msg_expandable_error:nnnnnn { fp } { num-args } {#1} {#2} {#3} {#4} }
  }
\cs_generate_variant:Nn \__fp_error_num_args:nnnn { ffff }
\msg_new:nnn { fp } { num-args-eq }
  { #1()~needs~#2~arguments,~got~#3. }
\msg_new:nnn { fp } { num-args }
  { #1()~needs~#2~to~#3~arguments,~got~#4. }
\msg_new:nnnn { fp } { unknown-fpu-exception }
  {
    The~FPU~exception~'#1'~is~not~known:~
    that~trap~will~never~be~triggered.
  }
  {
    The~only~exceptions~to~which~traps~can~be~attached~are \\
    \iow_indent:n
      {
        * ~ invalid_operation \\
        * ~ division_by_zero \\
        * ~ overflow \\
        * ~ underflow
      }
  }
\msg_new:nnnn { fp } { unknown-fpu-trap-type }
  { The~FPU~trap~type~'#2'~is~not~known. }
  {
    The~trap~type~must~be~one~of \\
    \iow_indent:n
      {
        * ~ error \\
        * ~ flag \\
        * ~ none
      }
  }
\msg_new:nnn { fp } { flow }
  { An ~ #3 ~ occurred. }
\msg_new:nnn { fp } { flow-to }
  { #1 ~ #3 ed ~ to ~ #2 . }
\msg_new:nnn { fp } { zero-div }
  { Division~by~zero~in~ #1 (#2) }
\msg_new:nnn { fp } { zero-div-ii }
  { Division~by~zero~in~ (#1) #3 (#2) }
\msg_new:nnn { fp } { invalid }
  { Invalid~operation~ #1 (#2) }
\msg_new:nnn { fp } { invalid-ii }
  { Invalid~operation~ (#1) #3 (#2) }
\msg_new:nnn { fp } { unknown-type }
  { Unknown~type~for~'#1' }
```

```
%% File: l3fp-round.dtx
\cs_new:Npn \__fp_parse_word_trunc:N
  { \__fp_parse_function:NNN \__fp_round_o:Nw \__fp_round_to_zero:NNN }
\cs_new:Npn \__fp_parse_word_floor:N
  { \__fp_parse_function:NNN \__fp_round_o:Nw \__fp_round_to_ninf:NNN }
\cs_new:Npn \__fp_parse_word_ceil:N
  { \__fp_parse_function:NNN \__fp_round_o:Nw \__fp_round_to_pinf:NNN }
\cs_new:Npn \__fp_parse_word_round:N #1#2
  {
    \__fp_parse_function:NNN
      \__fp_round_o:Nw \__fp_round_to_nearest:NNN #1
    #2
  }
\cs_new:Npn \__fp_parse_round:Nw #1 #2 \__fp_round_to_nearest:NNN #3#4
  { #2 #1 #3 }
\int_const:Nn \c__fp_five_int { 5 }
\cs_new:Npn \__fp_round_return_one:
  { \exp_after:wN 1 \exp_after:wN \exp_stop_f: \exp:w }
\cs_new:Npn \__fp_round_to_ninf:NNN #1 #2 #3
  {
    \if_meaning:w 2 #1
      \if_int_compare:w #3 > \c_zero_int
        \__fp_round_return_one:
      \fi:
    \fi:
    \c_zero_int
  }
\cs_new:Npn \__fp_round_to_zero:NNN #1 #2 #3 { \c_zero_int }
\cs_new:Npn \__fp_round_to_pinf:NNN #1 #2 #3
  {
    \if_meaning:w 0 #1
      \if_int_compare:w #3 > \c_zero_int
        \__fp_round_return_one:
      \fi:
    \fi:
    \c_zero_int
  }
\cs_new:Npn \__fp_round_to_nearest:NNN #1 #2 #3
  {
    \if_int_compare:w #3 > \c__fp_five_int
      \__fp_round_return_one:
    \else:
      \if_meaning:w 5 #3
        \if_int_odd:w #2 \exp_stop_f:
          \__fp_round_return_one:
        \fi:
```

```
      \fi:
    \fi:
    \c_zero_int
  }
\cs_new:Npn \__fp_round_to_nearest_ninf:NNN #1 #2 #3
  {
    \if_int_compare:w #3 > \c__fp_five_int
      \__fp_round_return_one:
    \else:
      \if_meaning:w 5 #3
        \if_meaning:w 2 #1
            \__fp_round_return_one:
        \fi:
      \fi:
    \fi:
    \c_zero_int
  }
\cs_new:Npn \__fp_round_to_nearest_zero:NNN #1 #2 #3
  {
    \if_int_compare:w #3 > \c__fp_five_int
      \__fp_round_return_one:
    \fi:
    \c_zero_int
  }
\cs_new:Npn \__fp_round_to_nearest_pinf:NNN #1 #2 #3
  {
    \if_int_compare:w #3 > \c__fp_five_int
      \__fp_round_return_one:
    \else:
      \if_meaning:w 5 #3
        \if_meaning:w 0 #1
            \__fp_round_return_one:
        \fi:
      \fi:
    \fi:
    \c_zero_int
  }
\cs_new_eq:NN \__fp_round:NNN \__fp_round_to_nearest:NNN
\cs_new:Npn \__fp_round_s:NNNw #1 #2 #3 #4\__fp_sep:
  {
    \exp_after:wN \__fp_round:NNN
    \exp_after:wN #1
    \exp_after:wN #2
    \int_value:w \__fp_int_eval:w
      \if_int_odd:w 0 \if_meaning:w 0 #3 1 \fi:
                      \if_meaning:w 5 #3 1 \fi:
```

```
        \exp_stop_f:
    \if_int_compare:w \__fp_int_eval:w #4 > \c_zero_int
     1 +
    \fi:
   \fi:
   #3
  \__fp_sep:
 }
\cs_new:Npn \__fp_round_digit:Nw #1 #2\__fp_sep:
 {
  \if_int_odd:w \if_meaning:w 0 #1 1 \else:
         \if_meaning:w 5 #1 1 \else:
         0 \fi: \fi: \exp_stop_f:
   \if_int_compare:w \__fp_int_eval:w #2 > \c_zero_int
    \__fp_int_eval:w 1 +
   \fi:
  \fi:
  #1
 }
\cs_new_eq:NN \__fp_round_to_ninf_neg:NNN \__fp_round_to_pinf:NNN
\cs_new:Npn \__fp_round_to_zero_neg:NNN #1 #2 #3
 {
  \if_int_compare:w #3 > \c_zero_int
   \__fp_round_return_one:
  \fi:
  \c_zero_int
 }
\cs_new_eq:NN \__fp_round_to_pinf_neg:NNN \__fp_round_to_ninf:NNN
\cs_new_eq:NN \__fp_round_to_nearest_neg:NNN \__fp_round_to_nearest:NNN
\cs_new_eq:NN \__fp_round_to_nearest_ninf_neg:NNN
 \__fp_round_to_nearest_pinf:NNN
\cs_new:Npn \__fp_round_to_nearest_zero_neg:NNN #1 #2 #3
 {
  \if_int_compare:w #3 < \c__fp_five_int \else:
   \__fp_round_return_one:
  \fi:
  \c_zero_int
 }
\cs_new_eq:NN \__fp_round_to_nearest_pinf_neg:NNN
 \__fp_round_to_nearest_ninf:NNN
\cs_new_eq:NN \__fp_round_neg:NNN \__fp_round_to_nearest_neg:NNN
\cs_new:Npn \__fp_round_o:Nw #1
 {
  \__fp_parse_function_all_fp_o:fnw
   { \__fp_round_name_from_cs:N #1 }
   { \__fp_round_aux_o:Nw #1 }
  }
\cs_new:Npn \__fp_round_aux_o:Nw #1#2 @
  {
    \if_case:w
      \__fp_int_eval:w \__fp_array_count:n {#2} \__fp_int_eval_end:
         \__fp_round_no_arg_o:Nw #1 \exp:w
    \or: \__fp_round:Nwn #1 #2 {0} \exp:w
    \or: \__fp_round:Nww #1 #2 \exp:w
    \else: \__fp_round:Nwww #1 #2 @ \exp:w
    \fi:
    \exp_after:wN \exp_end:
  }
\cs_new:Npn \__fp_round_no_arg_o:Nw #1
  {
    \cs_if_eq:NNTF #1 \__fp_round_to_nearest:NNN
      { \__fp_error_num_args:nnnn { round } { 1 } { 3 } { 0 } }
      {
        \__fp_error_num_args:ffff
          { \__fp_round_name_from_cs:N #1 } { 1 } { 2 } { 0 }
      }
    \exp_after:wN \c_nan_fp
  }
\cs_new:Npn \__fp_round:Nwww
    #1#2 \__fp_sep: #3 \__fp_sep: \s__fp \__fp_chk:w #4#5#6 \__fp_sep: #7 @
  {
    \cs_if_eq:NNTF #1 \__fp_round_to_nearest:NNN
      {
        \tl_if_empty:nTF {#7}
          {
            \exp_args:Nc \__fp_round:Nww
              {
                __fp_round_to_nearest
                \if_meaning:w 0 #4 _zero \else:
                \if_case:w #5 \exp_stop_f: _pinf \or: \else: _ninf \fi: \fi:
                :NNN
              }
            #2 \__fp_sep: #3 \__fp_sep:
          }
          {
            \__fp_error_num_args:ffff { round } { 1 } { 3 }
              { \int_eval:n { 3 + \__fp_array_count:n {#7} } }
            \exp_after:wN \c_nan_fp
          }
      }
      {
        \__fp_error_num_args:ffff
```

```
          { \__fp_round_name_from_cs:N #1 } { 1 } { 2 }
          { \int_eval:n { 3 + \__fp_array_count:n {#7} } }
        \exp_after:wN \c_nan_fp
      }
  }
\cs_new:Npn \__fp_round_name_from_cs:N #1
  {
    \cs_if_eq:NNTF #1 \__fp_round_to_zero:NNN { trunc }
      {
        \cs_if_eq:NNTF #1 \__fp_round_to_ninf:NNN { floor }
          {
            \cs_if_eq:NNTF #1 \__fp_round_to_pinf:NNN { ceil }
              { round }
          }
      }
  }
\cs_new:Npn \__fp_round:Nww #1#2 \__fp_sep: #3 \__fp_sep:
  {
    \__fp_small_int:wTF #3\__fp_sep: { \__fp_round:Nwn #1#2\__fp_sep: }
      {
        \if:w 3 \__fp_kind:w #3 \__fp_sep:
          \exp_after:wN \use_i:nn
        \else:
          \exp_after:wN \use_ii:nn
        \fi:
        { \exp_after:wN \c_nan_fp }
        {
          \__fp_invalid_operation_tl_o:ff
            { \__fp_round_name_from_cs:N #1 }
            { \__fp_array_to_clist:n { #2\__fp_sep: #3\__fp_sep: } }
        }
      }
  }
\cs_new:Npn \__fp_round:Nwn #1 \s__fp \__fp_chk:w #2#3#4\__fp_sep: #5
  {
    \if_meaning:w 1 #2
      \exp_after:wN \__fp_round_normal:NwNNnw
      \exp_after:wN #1
      \int_value:w #5
    \else:
      \exp_after:wN \__fp_exp_after_o:w
    \fi:
    \s__fp \__fp_chk:w #2#3#4\__fp_sep:
  }
\cs_new:Npn \__fp_round_normal:NwNNnw #1#2 \s__fp \__fp_chk:w 1#3#4#5\__fp_sep:
  {
    \__fp_decimate:nNnnnn { \c__fp_prec_int - #4 - #2 }
      \__fp_round_normal:NnnwNNnn #5 #1 #3 {#4} {#2}
  }
\cs_new:Npn \__fp_round_normal:NnnwNNnn #1#2#3#4\__fp_sep: #5#6
  {
    \exp_after:wN \__fp_round_normal:NNwNnn
    \int_value:w \__fp_int_eval:w
      \if_int_compare:w #2 > \c_zero_int
        1 \int_value:w #2
        \exp_after:wN \__fp_round_pack:Nw
        \int_value:w \__fp_int_eval:w 1#3 +
      \else:
        \if_int_compare:w #3 > \c_zero_int
          1 \int_value:w #3 +
        \fi:
      \fi:
      \exp_after:wN #5
      \exp_after:wN #6
      \use_none:nnnnnnn #3
      #1
      \__fp_int_eval_end:
      0000 0000 0000 0000 \__fp_sep: #6
  }
\cs_new:Npn \__fp_round_pack:Nw #1
  { \if_meaning:w 2 #1 + 1 \fi: \__fp_int_eval_end: }
\cs_new:Npn \__fp_round_normal:NNwNnn #1 #2
  {
    \if_meaning:w 0 #2
      \exp_after:wN \__fp_round_special:NwwNnn
      \exp_after:wN #1
    \fi:
    \__fp_pack_twice_four:wNNNNNNNN
    \__fp_pack_twice_four:wNNNNNNNN
    \__fp_round_normal_end:wwNnn
    \__fp_sep: #2
  }
\cs_new:Npn \__fp_round_normal_end:wwNnn #1\__fp_sep:#2\__fp_sep:#3#4#5
  {
    \exp_after:wN \__fp_exp_after_o:w \exp:w \exp_end_continue_f:w
    \__fp_sanitize:Nw #3 #4 \__fp_sep: #1 \__fp_sep:
  }
\cs_new:Npn \__fp_round_special:NwwNnn #1#2\__fp_sep:#3\__fp_sep:#4#5#6
  {
    \if_meaning:w 0 #1
      \__fp_case_return:nw
        { \exp_after:wN \__fp_zero_fp:N \exp_after:wN #4 }
    \else:
      \exp_after:wN \__fp_round_special_aux:Nw
      \exp_after:wN #4
      \int_value:w \__fp_int_eval:w 1
        \if_meaning:w 1 #1 -#6 \else: +#5 \fi:
    \fi:
    \__fp_sep:
  }
\cs_new:Npn \__fp_round_special_aux:Nw #1#2\__fp_sep:
  {
    \exp_after:wN \__fp_exp_after_o:w \exp:w \exp_end_continue_f:w
    \__fp_sanitize:Nw #1#2\__fp_sep: {1000}{0000}{0000}{0000}\__fp_sep:
  }
%% File: l3fp-parse.dtx
\int_const:Nn \c__fp_prec_func_int   { 16 }
\int_const:Nn \c__fp_prec_hatii_int  { 14 }
\int_const:Nn \c__fp_prec_hat_int    { 13 }
\int_const:Nn \c__fp_prec_not_int    { 12 }
\int_const:Nn \c__fp_prec_juxt_int   { 11 }
\int_const:Nn \c__fp_prec_times_int  { 10 }
\int_const:Nn \c__fp_prec_plus_int   { 9 }
\int_const:Nn \c__fp_prec_comp_int   { 7 }
\int_const:Nn \c__fp_prec_and_int    { 6 }
\int_const:Nn \c__fp_prec_or_int     { 5 }
\int_const:Nn \c__fp_prec_quest_int  { 4 }
\int_const:Nn \c__fp_prec_colon_int  { 3 }
\int_const:Nn \c__fp_prec_comma_int  { 2 }
\int_const:Nn \c__fp_prec_tuple_int  { 1 }
\int_const:Nn \c__fp_prec_end_int    { 0 }
\cs_new:Npn \__fp_parse_expand:w #1 { \exp_end_continue_f:w #1 }
\cs_new:Npn \__fp_parse_return_sep:w
    #1 \fi: \__fp_parse_expand:w { \fi: \__fp_sep: #1 }
\cs_set_protected:Npn \__fp_tmp:w #1 #2 #3
  {
    \cs_new:cpn { __fp_parse_digits_ #1 :N } ##1
      {
        \if_int_compare:w 9 < 1 \token_to_str:N ##1 \exp_stop_f:
          \token_to_str:N ##1 \exp_after:wN #2 \exp:w
        \else:
          \__fp_parse_return_sep:w #3 ##1
        \fi:
        \__fp_parse_expand:w
      }
  }
\__fp_tmp:w {vii}  \__fp_parse_digits_vi:N   { 0000000 \__fp_sep: 7 }
\__fp_tmp:w {vi}   \__fp_parse_digits_v:N    { 000000 \__fp_sep: 6 }
```

```
\__fp_tmp:w {v}    \__fp_parse_digits_iv:N   { 00000 \__fp_sep: 5 }
\__fp_tmp:w {iv}   \__fp_parse_digits_iii:N  { 0000 \__fp_sep: 4 }
\__fp_tmp:w {iii}  \__fp_parse_digits_ii:N   { 000 \__fp_sep: 3 }
\__fp_tmp:w {ii}   \__fp_parse_digits_i:N    { 00 \__fp_sep: 2 }
\__fp_tmp:w {i}    \__fp_parse_digits_:N     { 0 \__fp_sep: 1 }
\cs_new:Npn \__fp_parse_digits_:N { \__fp_sep: \__fp_sep: 0 }
\cs_new:Npn \__fp_parse_one:Nw #1 #2
  {
    \if_catcode:w \scan_stop: \exp_not:N #2
      \exp_after:wN \if_meaning:w \exp_not:N #2 #2 \else:
        \exp_after:wN \reverse_if:N
      \fi:
      \if_meaning:w \scan_stop: #2
        \exp_after:wN \exp_after:wN
        \exp_after:wN \__fp_parse_one_fp:NN
      \else:
        \exp_after:wN \exp_after:wN
        \exp_after:wN \__fp_parse_one_register:NN
      \fi:
    \else:
      \if_int_compare:w 9 < 1 \token_to_str:N #2 \exp_stop_f:
        \exp_after:wN \exp_after:wN
        \exp_after:wN \__fp_parse_one_digit:NN
      \else:
        \exp_after:wN \exp_after:wN
        \exp_after:wN \__fp_parse_one_other:NN
      \fi:
    \fi:
    #1 #2
  }
\cs_new:Npn \__fp_parse_one_fp:NN #1
  {
    \__fp_exp_after_any_f:nw
      {
        \exp_after:wN \__fp_parse_infix:NN
        \exp_after:wN #1 \exp:w \__fp_parse_expand:w
      }
  }
\cs_new:Npn \__fp_exp_after_expr_mark_f:nw #1
  {
    \int_case:nnF { \exp_after:wN \use_i:nnn \use_none:nnn #1 }
      {
        \c__fp_prec_comma_int { }
        \c__fp_prec_tuple_int { }
        \c__fp_prec_end_int
          {
            \exp_after:wN \c__fp_empty_tuple_fp
            \exp:w \exp_end_continue_f:w
          }
      }
      {
        \msg_expandable_error:nn { fp } { early-end }
        \exp_after:wN \c_nan_fp \exp:w \exp_end_continue_f:w
      }
    #1
  }
\cs_new:cpn { __fp_exp_after_?_f:nw } #1#2
  {
    \msg_expandable_error:nnn { kernel } { bad-variable }
      {#2}
    \exp_after:wN \c_nan_fp \exp:w \exp_end_continue_f:w #1
  }
\cs_set_protected:Npn \__fp_tmp:w #1
  {
    \cs_if_exist:NT #1
      {
        \cs_gset:cpn { __fp_exp_after_?_f:nw } ##1##2
          {
            \exp_after:wN \c_nan_fp \exp:w \exp_end_continue_f:w ##1
            \str_if_eq:nnTF {##2} { \protect }
              {
                \cs_if_eq:NNTF ##2 #1 { \use_i:nn } { \use:n }
                {
                  \msg_expandable_error:nnn { fp }
                    { robust-cmd }
                }
              }
              {
                \msg_expandable_error:nnn { kernel }
                  { bad-variable } {##2}
              }
          }
      }
  }
\exp_args:Nc \__fp_tmp:w { @unexpandable@protect }
\cs_new:Npn \__fp_parse_one_register:NN #1#2
  {
    \exp_after:wN \__fp_parse_infix_after_operand:NwN
    \exp_after:wN #1
    \exp:w \exp_end_continue_f:w
      \__fp_parse_one_register_special:N #2
      \exp_after:wN \__fp_parse_one_register_aux:Nw
      \exp_after:wN #2
      \int_value:w
        \exp_after:wN \__fp_parse_exponent:N
        \exp:w \__fp_parse_expand:w
  }
\cs_new:Npe \__fp_parse_one_register_aux:Nw #1
  {
    \exp_not:n
      {
        \exp_after:wN \use:nn
        \exp_after:wN \__fp_parse_one_register_auxii:wwwNw
      }
    \exp_not:N \exp_after:wN { \exp_not:N \tex_the:D #1 }
      \__fp_sep: \exp_not:N \__fp_parse_one_register_dim:ww
      \tl_to_str:n { pt } \__fp_sep: \exp_not:N \__fp_parse_one_register_mu:www
      . \tl_to_str:n { pt } \__fp_sep: \exp_not:N \__fp_parse_one_register_int:www
      \s__fp_stop
  }
\exp_args:Nno \use:nn
  { \cs_new:Npn \__fp_parse_one_register_auxii:wwwNw #1 . #2 }
    { \tl_to_str:n { pt } #3 \__fp_sep: #4#5 \s__fp_stop }
    { #4 #1.#2\__fp_sep: }
\exp_args:Nno \use:nn
  { \cs_new:Npn \__fp_parse_one_register_mu:www #1 }
    { \tl_to_str:n { mu } \__fp_sep: #2 \__fp_sep: }
    { \__fp_parse_one_register_dim:ww #1 \__fp_sep: }
\cs_new:Npn \__fp_parse_one_register_int:www #1\__fp_sep: #2.\__fp_sep: #3\__fp_sep:
  { \__fp_parse:n { #1 e #3 } }
\cs_new:Npn \__fp_parse_one_register_dim:ww #1\__fp_sep: #2\__fp_sep:
  {
    \exp_after:wN \__fp_from_dim_test:ww
    \int_value:w #2 \exp_after:wN ,
    \int_value:w \dim_to_decimal_in_sp:n { #1 pt } \__fp_sep:
  }
\cs_new:Npn \__fp_parse_one_register_special:N #1
  {
    \if_meaning:w \box_wd:N #1 \__fp_parse_one_register_wd:w \fi:
    \if_meaning:w \box_ht:N #1 \__fp_parse_one_register_wd:w \fi:
    \if_meaning:w \box_dp:N #1 \__fp_parse_one_register_wd:w \fi:
    \if_meaning:w \infty #1
      \__fp_parse_one_register_math:NNw \infty #1
    \fi:
    \if_meaning:w \pi #1
      \__fp_parse_one_register_math:NNw \pi #1
    \fi:
  }
```

```
\cs_new:Npn \__fp_parse_one_register_math:NNw
    #1#2#3#4 \__fp_parse_expand:w
  {
    #3
    \str_if_eq:nnTF {#1} {#2}
      {
        \msg_expandable_error:nnn
          { fp } { infty-pi } {#1}
        \c_nan_fp
      }
      { #4 \__fp_parse_expand:w }
  }
\cs_new:Npn \__fp_parse_one_register_wd:w
    #1#2 \exp_after:wN #3#4 \__fp_parse_expand:w
  {
    #1
    \exp_after:wN \__fp_parse_one_register_wd:Nw
    #4 \__fp_parse_expand:w e
  }
\cs_new:Npn \__fp_parse_one_register_wd:Nw #1#2 \__fp_sep:
  {
    \exp_after:wN \__fp_from_dim_test:ww
    \exp_after:wN 0 \exp_after:wN ,
    \int_value:w \dim_to_decimal_in_sp:n { #1 #2 } \__fp_sep:
  }
\cs_new:Npn \__fp_parse_one_digit:NN #1
  {
    \exp_after:wN \__fp_parse_infix_after_operand:NwN
    \exp_after:wN #1
    \exp:w \exp_end_continue_f:w
      \exp_after:wN \__fp_sanitize:wN
      \int_value:w \__fp_int_eval:w 0 \__fp_parse_trim_zeros:N
  }
\cs_new:Npn \__fp_parse_one_other:NN #1 #2
  {
    \if_int_compare:w
        \__fp_int_eval:w
          ( `#2 \if_int_compare:w `#2 > `Z - 32 \fi: ) / 26
        = 3 \exp_stop_f:
      \exp_after:wN \__fp_parse_word:Nw
      \exp_after:wN #1
      \exp_after:wN #2
      \exp:w \exp_after:wN \__fp_parse_letters:N
      \exp:w
    \else:
      \exp_after:wN \__fp_parse_prefix:NNN
      \exp_after:wN #1
      \exp_after:wN #2
      \cs:w
        __fp_parse_prefix_ \token_to_str:N #2 :Nw
        \exp_after:wN
      \cs_end:
      \exp:w
    \fi:
    \__fp_parse_expand:w
  }
\cs_new:Npn \__fp_parse_word:Nw #1#2\__fp_sep:
  {
    \cs_if_exist_use:cF { __fp_parse_word_#2:N }
      {
        \cs_if_exist_use:cF
          { __fp_parse_caseless_ \str_casefold:n {#2} :N }
          {
            \msg_expandable_error:nnn
              { fp } { unknown-fp-word } {#2}
            \exp_after:wN \c_nan_fp \exp:w \exp_end_continue_f:w
            \__fp_parse_infix:NN
          }
      }
      #1
  }
\cs_new:Npn \__fp_parse_letters:N #1
  {
    \exp_end_continue_f:w
    \if_int_compare:w
        \if_catcode:w \scan_stop: \exp_not:N #1
          0
        \else:
          \__fp_int_eval:w
            ( `#1 \if_int_compare:w `#1 > `Z - 32 \fi: ) / 26
        \fi:
        = 3 \exp_stop_f:
      \exp_after:wN #1
      \exp:w \exp_after:wN \__fp_parse_letters:N
      \exp:w
    \else:
      \__fp_parse_return_sep:w #1
    \fi:
    \__fp_parse_expand:w
  }
\cs_new:Npn \__fp_parse_prefix:NNN #1#2#3
  {
    \if_meaning:w \scan_stop: #3
      \exp_after:wN \__fp_parse_prefix_unknown:NNN
      \exp_after:wN #2
    \fi:
    #3 #1
  }
\cs_new:Npn \__fp_parse_prefix_unknown:NNN #1#2#3
  {
    \cs_if_exist:cTF { __fp_parse_infix_ \token_to_str:N #1 :N }
      {
        \msg_expandable_error:nnn
          { fp } { missing-number } {#1}
        \exp_after:wN \c_nan_fp \exp:w \exp_end_continue_f:w
        \__fp_parse_infix:NN #3 #1
      }
      {
        \msg_expandable_error:nnn
          { fp } { unknown-symbol } {#1}
        \__fp_parse_one:Nw #3
      }
  }
\cs_new:Npn \__fp_parse_trim_zeros:N #1
  {
    \if:w 0 \exp_not:N #1
      \exp_after:wN \__fp_parse_trim_zeros:N
      \exp:w
    \else:
      \if:w . \exp_not:N #1
        \exp_after:wN \__fp_parse_strim_zeros:N
        \exp:w
      \else:
        \__fp_parse_trim_end:w #1
      \fi:
    \fi:
    \__fp_parse_expand:w
  }
\cs_new:Npn \__fp_parse_trim_end:w #1 \fi: \fi: \__fp_parse_expand:w
  {
      \fi:
    \fi:
    \if_int_compare:w 9 < 1 \token_to_str:N #1 \exp_stop_f:
      \exp_after:wN \__fp_parse_large:N
    \else:
      \exp_after:wN \__fp_parse_zero:
    \fi:
    #1
  }
\cs_new:Npn \__fp_parse_strim_zeros:N #1
  {
    \if:w 0 \exp_not:N #1
      - 1
      \exp_after:wN \__fp_parse_strim_zeros:N \exp:w
    \else:
      \__fp_parse_strim_end:w #1
    \fi:
    \__fp_parse_expand:w
  }
\cs_new:Npn \__fp_parse_strim_end:w #1 \fi: \__fp_parse_expand:w
  {
    \fi:
    \if_int_compare:w 9 < 1 \token_to_str:N #1 \exp_stop_f:
      \exp_after:wN \__fp_parse_small:N
    \else:
      \exp_after:wN \__fp_parse_zero:
    \fi:
    #1
  }
\cs_new:Npn \__fp_parse_zero:
  {
    \exp_after:wN \__fp_sep: \exp_after:wN 1
    \int_value:w \__fp_parse_exponent:N
  }
\cs_new:Npn \__fp_parse_small:N #1
  {
    \exp_after:wN \__fp_parse_pack_leading:NNNNNww
    \int_value:w \__fp_int_eval:w 1 \token_to_str:N #1
      \exp_after:wN \__fp_parse_small_leading:wwNN
      \int_value:w 1
        \exp_after:wN \__fp_parse_digits_vii:N
        \exp:w \__fp_parse_expand:w
  }
\cs_new:Npn \__fp_parse_small_leading:wwNN 1 #1 \__fp_sep: #2\__fp_sep: #3 #4
  {
    #1 #2
    \exp_after:wN \__fp_parse_pack_trailing:NNNNNNww
    \exp_after:wN 0
    \int_value:w \__fp_int_eval:w 1
      \if_int_compare:w 9 < 1 \token_to_str:N #4 \exp_stop_f:
        \token_to_str:N #4
        \exp_after:wN \__fp_parse_small_trailing:wwNN
        \int_value:w 1
          \exp_after:wN \__fp_parse_digits_vi:N
          \exp:w
      \else:
        0000 0000 \__fp_parse_exponent:Nw #4
      \fi:
      \__fp_parse_expand:w
  }
\cs_new:Npn \__fp_parse_small_trailing:wwNN 1 #1 \__fp_sep: #2\__fp_sep: #3 #4
  {
    #1 #2
    \if_int_compare:w 9 < 1 \token_to_str:N #4 \exp_stop_f:
      \token_to_str:N #4
      \exp_after:wN \__fp_parse_small_round:NN
      \exp_after:wN #4
      \exp:w
    \else:
      0 \__fp_parse_exponent:Nw #4
    \fi:
    \__fp_parse_expand:w
  }
\cs_new:Npn \__fp_parse_pack_trailing:NNNNNNww
    #1 #2 #3#4#5#6 #7\__fp_sep: #8 \__fp_sep:
  {
    \if_meaning:w 2 #2 + 1 \fi:
    \__fp_sep: #8 + #1 \__fp_sep: {#3#4#5#6} {#7}\__fp_sep:
  }
\cs_new:Npn \__fp_parse_pack_leading:NNNNNww #1 #2#3#4#5 #6\__fp_sep: #7\__fp_sep:
  {
    + #7
    \if_meaning:w 2 #1 \__fp_parse_pack_carry:w \fi:
    \__fp_sep: 0 {#2#3#4#5} {#6}
  }
\cs_new:Npn \__fp_parse_pack_carry:w \fi: \__fp_sep: 0 #1
  { \fi: + 1 \__fp_sep: 0 {1000} }
\cs_new:Npn \__fp_parse_large:N #1
  {
    \exp_after:wN \__fp_parse_large_leading:wwNN
    \int_value:w 1 \token_to_str:N #1
      \exp_after:wN \__fp_parse_digits_vii:N
      \exp:w \__fp_parse_expand:w
  }
\cs_new:Npn \__fp_parse_large_leading:wwNN 1 #1 \__fp_sep: #2\__fp_sep: #3 #4
  {
    + \c__fp_half_prec_int - #3
    \exp_after:wN \__fp_parse_pack_leading:NNNNNww
    \int_value:w \__fp_int_eval:w 1 #1
      \if_int_compare:w 9 < 1 \token_to_str:N #4 \exp_stop_f:
        \exp_after:wN \__fp_parse_large_trailing:wwNN
        \int_value:w 1 \token_to_str:N #4
          \exp_after:wN \__fp_parse_digits_vi:N
          \exp:w
      \else:
        \if:w . \exp_not:N #4
          \exp_after:wN \__fp_parse_small_leading:wwNN
          \int_value:w 1
            \cs:w
              __fp_parse_digits_
              \__fp_int_to_roman:w #3
              :N \exp_after:wN
            \cs_end:
            \exp:w
        \else:
          #2
          \exp_after:wN \__fp_parse_pack_trailing:NNNNNNww
          \exp_after:wN 0
          \int_value:w 1 0000 0000
          \__fp_parse_exponent:Nw #4
        \fi:
      \fi:
      \__fp_parse_expand:w
  }
\cs_new:Npn \__fp_parse_large_trailing:wwNN 1 #1 \__fp_sep: #2\__fp_sep: #3 #4
  {
    \if_int_compare:w 9 < 1 \token_to_str:N #4 \exp_stop_f:
      \exp_after:wN \__fp_parse_pack_trailing:NNNNNNww
      \exp_after:wN \c__fp_half_prec_int
      \int_value:w \__fp_int_eval:w 1 #1 \token_to_str:N #4
        \exp_after:wN \__fp_parse_large_round:NN
        \exp_after:wN #4
        \exp:w
    \else:
      \exp_after:wN \__fp_parse_pack_trailing:NNNNNNww
      \int_value:w \__fp_int_eval:w 7 - #3 \exp_stop_f:
      \int_value:w \__fp_int_eval:w 1 #1
        \if:w . \exp_not:N #4
          \exp_after:wN \__fp_parse_small_trailing:wwNN
          \int_value:w 1
            \cs:w
              __fp_parse_digits_
              \__fp_int_to_roman:w #3
              :N \exp_after:wN
            \cs_end:
            \exp:w
        \else:
          #2 0 \__fp_parse_exponent:Nw #4
        \fi:
    \fi:
    \__fp_parse_expand:w
  }
\cs_new:Npn \__fp_parse_round_loop:N #1
  {
    \if_int_compare:w 9 < 1 \token_to_str:N #1 \exp_stop_f:
      + 1
      \if:w 0 \token_to_str:N #1
        \exp_after:wN \__fp_parse_round_loop:N
        \exp:w
      \else:
        \exp_after:wN \__fp_parse_round_up:N
        \exp:w
      \fi:
    \else:
      \__fp_parse_return_sep:w 0 #1
    \fi:
    \__fp_parse_expand:w
  }
\cs_new:Npn \__fp_parse_round_up:N #1
  {
    \if_int_compare:w 9 < 1 \token_to_str:N #1 \exp_stop_f:
      + 1
      \exp_after:wN \__fp_parse_round_up:N
      \exp:w
    \else:
      \__fp_parse_return_sep:w 1 #1
    \fi:
    \__fp_parse_expand:w
  }
\cs_new:Npn \__fp_parse_round_after:wN #1\__fp_sep: #2
  {
    + #2 \exp_after:wN \__fp_sep:
    \int_value:w \__fp_int_eval:w #1 + \__fp_parse_exponent:N
  }
\cs_new:Npn \__fp_parse_small_round:NN #1#2
  {
    \if_int_compare:w 9 < 1 \token_to_str:N #2 \exp_stop_f:
      +
      \exp_after:wN \__fp_round_s:NNNw
      \exp_after:wN 0
      \exp_after:wN #1
      \exp_after:wN #2
      \int_value:w \__fp_int_eval:w
        \exp_after:wN \__fp_parse_round_after:wN
        \int_value:w \__fp_int_eval:w 0 * \__fp_int_eval:w 0
          \exp_after:wN \__fp_parse_round_loop:N
          \exp:w
    \else:
      \__fp_parse_exponent:Nw #2
    \fi:
    \__fp_parse_expand:w
  }
\cs_new:Npn \__fp_parse_large_round:NN #1#2
  {
    \if_int_compare:w 9 < 1 \token_to_str:N #2 \exp_stop_f:
      +
      \exp_after:wN \__fp_round_s:NNNw
      \exp_after:wN 0
      \exp_after:wN #1
      \exp_after:wN #2
      \int_value:w \__fp_int_eval:w
        \exp_after:wN \__fp_parse_large_round_aux:wNN
        \int_value:w \__fp_int_eval:w 1
          \exp_after:wN \__fp_parse_round_loop:N
    \else: %^^A could be dot, or e, or other
      \exp_after:wN \__fp_parse_large_round_test:NN
      \exp_after:wN #1
      \exp_after:wN #2
    \fi:
  }
\cs_new:Npn \__fp_parse_large_round_test:NN #1#2
  {
    \if:w . \exp_not:N #2
      \exp_after:wN \__fp_parse_small_round:NN
      \exp_after:wN #1
      \exp:w
    \else:
      \__fp_parse_exponent:Nw #2
    \fi:
    \__fp_parse_expand:w
  }
\cs_new:Npn \__fp_parse_large_round_aux:wNN #1 \__fp_sep: #2 #3
  {
    + #2
    \exp_after:wN \__fp_parse_round_after:wN
    \int_value:w \__fp_int_eval:w #1
      \if:w . \exp_not:N #3
        + 0 * \__fp_int_eval:w 0
```

```
          \exp_after:wN \__fp_parse_round_loop:N
          \exp:w \exp_after:wN \__fp_parse_expand:w
      \else:
        \exp_after:wN \__fp_sep:
        \exp_after:wN 0
        \exp_after:wN #3
      \fi:
  }
\cs_new:Npn \__fp_parse_exponent:Nw #1 #2 \__fp_parse_expand:w
  {
    \exp_after:wN \__fp_sep:
    \int_value:w #2 \__fp_parse_exponent:N #1
  }
\cs_new:Npn \__fp_parse_exponent:N #1
  {
    \if:w e \if:w E \exp_not:N #1 e \else: \exp_not:N #1 \fi:
      \exp_after:wN \__fp_parse_exponent_aux:NN
      \exp_after:wN #1
      \exp:w
    \else:
      0 \__fp_parse_return_sep:w #1
    \fi:
    \__fp_parse_expand:w
  }
\cs_new:Npn \__fp_parse_exponent_aux:NN #1#2
  {
    \if_int_compare:w \if_catcode:w \scan_stop: \exp_not:N #2
                0 \else: `#2 \fi: > `9 \exp_stop_f:
      0 \exp_after:wN \__fp_sep: \exp_after:wN #1
    \else:
      \exp_after:wN \__fp_parse_exponent_sign:N
    \fi:
    #2
  }
\cs_new:Npn \__fp_parse_exponent_sign:N #1
  {
    \if:w + \if:w - \exp_not:N #1 + \fi: \token_to_str:N #1
      \exp_after:wN \__fp_parse_exponent_sign:N
      \exp:w \exp_after:wN \__fp_parse_expand:w
    \else:
      \exp_after:wN \__fp_parse_exponent_body:N
      \exp_after:wN #1
    \fi:
  }
\cs_new:Npn \__fp_parse_exponent_body:N #1
  {
    \if_int_compare:w 9 < 1 \token_to_str:N #1 \exp_stop_f:
      \token_to_str:N #1
      \exp_after:wN \__fp_parse_exponent_digits:N
      \exp:w
    \else:
      \__fp_parse_exponent_keep:NTF #1
        { \__fp_parse_return_sep:w #1 }
        {
          \exp_after:wN \__fp_sep:
          \exp:w
        }
    \fi:
    \__fp_parse_expand:w
  }
\cs_new:Npn \__fp_parse_exponent_digits:N #1
  {
    \if_int_compare:w 9 < 1 \token_to_str:N #1 \exp_stop_f:
      \token_to_str:N #1
      \exp_after:wN \__fp_parse_exponent_digits:N
      \exp:w
    \else:
      \__fp_parse_return_sep:w #1
    \fi:
    \__fp_parse_expand:w
  }
\prg_new_conditional:Npnn \__fp_parse_exponent_keep:N #1 { TF }
  {
    \if_catcode:w \scan_stop: \exp_not:N #1
      \if_meaning:w \scan_stop: #1
        \if:w 0 \__fp_str_if_eq:nn { \s__fp } { \exp_not:N #1 }
          0
          \msg_expandable_error:nnn
            { fp } { after-e } { floating~point~ }
          \prg_return_true:
        \else:
          0
          \msg_expandable_error:nnn
            { kernel } { bad-variable } {#1}
          \prg_return_false:
        \fi:
      \else:
        \if:w 0 \__fp_str_if_eq:nn { \int_value:w #1 } { \tex_the:D #1 }
          \int_value:w #1
        \else:
          0
          \msg_expandable_error:nnn
```

```
            { fp } { after-e } { dimension~#1 }
        \fi:
        \prg_return_false:
      \fi:
    \else:
      0
      \msg_expandable_error:nnn
        { fp } { missing } { exponent }
      \prg_return_true:
    \fi:
  }
\cs_new_eq:cN { __fp_parse_prefix_+:Nw } \__fp_parse_one:Nw
\cs_new:Npn \__fp_parse_apply_function:NNNwN #1#2#3#4@#5
  {
    #3 #2 #4 @
    \exp:w \exp_end_continue_f:w #5 #1
  }
\cs_new:Npn \__fp_parse_apply_unary:NNNwN #1#2#3#4@#5
  {
    \__fp_parse_apply_unary_chk:NwNw #4 @ \__fp_sep: . \s__fp_stop
    \__fp_parse_apply_unary_type:NNN
    #3 #2 #4 @
    \exp:w \exp_end_continue_f:w #5 #1
  }
\cs_new:Npn \__fp_parse_apply_unary_chk:NwNw #1#2 \__fp_sep: #3#4 \s__fp_stop
  {
    \if_meaning:w @ #3 \else:
      \token_if_eq_meaning:NNTF . #3
        { \__fp_parse_apply_unary_chk:nNNNNw { no } }
        { \__fp_parse_apply_unary_chk:nNNNNw { multi } }
    \fi:
  }
\cs_new:Npn \__fp_parse_apply_unary_chk:nNNNNw #1#2#3#4#5#6 @
  {
    #2
    \__fp_error:nffn { #1-arg } { \__fp_func_to_name:N #4 } { } { }
    \exp_after:wN #4 \exp_after:wN #5 \c_nan_fp @
  }
\cs_new:Npn \__fp_parse_apply_unary_type:NNN #1#2#3
  {
    \__fp_change_func_type:NNN #3 #1 \__fp_parse_apply_unary_error:NNw
    #2 #3
  }
\cs_new:Npn \__fp_parse_apply_unary_error:NNw #1#2#3 @
  { \__fp_invalid_operation_o:fw { \__fp_func_to_name:N #1 } #3 }
\cs_set_protected:Npn \__fp_tmp:w #1#2#3#4
```

```
  {
    \cs_new:cpn { __fp_parse_prefix_ #1 :Nw } ##1
      {
        \exp_after:wN \__fp_parse_apply_unary:NNNwN
        \exp_after:wN ##1
        \exp_after:wN #4
        \exp_after:wN #3
        \exp:w
        \if_int_compare:w #2 < ##1
          \__fp_parse_operand:Nw ##1
        \else:
          \__fp_parse_operand:Nw #2
        \fi:
        \__fp_parse_expand:w
      }
  }
\__fp_tmp:w - \c__fp_prec_not_int \__fp_set_sign_o:w 2
\__fp_tmp:w ! \c__fp_prec_not_int \__fp_not_o:w ?
\cs_new:cpn { __fp_parse_prefix_.:Nw } #1
  {
    \exp_after:wN \__fp_parse_infix_after_operand:NwN
    \exp_after:wN #1
    \exp:w \exp_end_continue_f:w
      \exp_after:wN \__fp_sanitize:wN
      \int_value:w \__fp_int_eval:w 0 \__fp_parse_strim_zeros:N
  }
\cs_new:cpn { __fp_parse_prefix_(:Nw } #1
  {
    \exp_after:wN \__fp_parse_lparen_after:NwN
    \exp_after:wN #1
    \exp:w
    \if_int_compare:w #1 = \c__fp_prec_func_int
      \__fp_parse_operand:Nw \c__fp_prec_comma_int
    \else:
      \__fp_parse_operand:Nw \c__fp_prec_tuple_int
    \fi:
    \__fp_parse_expand:w
  }
\cs_new:Npe \__fp_parse_lparen_after:NwN #1#2 @ #3
  {
    \exp_not:N \token_if_eq_meaning:NNTF #3
      \exp_not:c { __fp_parse_infix_):N }
      {
        \exp_not:N \__fp_exp_after_array_f:w #2 \s__fp_expr_stop
        \exp_not:N \exp_after:wN
        \exp_not:N \__fp_parse_infix_after_paren:NN
        \exp_not:N \exp_after:wN #1
        \exp_not:N \exp:w
        \exp_not:N \__fp_parse_expand:w
       }
       {
        \exp_not:N \msg_expandable_error:nnn
         { fp } { missing } { ) }
        \exp_not:N \tl_if_empty:nT {#2} \exp_not:N \c__fp_empty_tuple_fp
        #2 @
        \exp_not:N \use_none:n #3
       }
     }
\cs_new:cpn { __fp_parse_prefix_):Nw } #1
     {
      \if_int_compare:w #1 = \c__fp_prec_comma_int
      \else:
      \if_int_compare:w #1 = \c__fp_prec_tuple_int
        \exp_after:wN \c__fp_empty_tuple_fp \exp:w
      \else:
        \msg_expandable_error:nnn
          { fp } { missing-number } { ) }
        \exp_after:wN \c_nan_fp \exp:w
      \fi:
      \exp_end_continue_f:w
      \fi:
    \__fp_parse_infix_after_paren:NN #1 )
     }
\cs_set_protected:Npn \__fp_tmp:w #1 #2
     {
    \cs_new:cpn { __fp_parse_word_#1:N }
      { \exp_after:wN #2 \exp:w \exp_end_continue_f:w \__fp_parse_infix:NN }
     }
\__fp_tmp:w { inf } \c_inf_fp
\__fp_tmp:w { nan } \c_nan_fp
\__fp_tmp:w { pi  } \c_pi_fp
\__fp_tmp:w { deg } \c_one_degree_fp
\__fp_tmp:w { true } \c_one_fp
\__fp_tmp:w { false } \c_zero_fp
\cs_new_eq:NN \__fp_parse_caseless_inf:N \__fp_parse_word_inf:N
\cs_new_eq:NN \__fp_parse_caseless_infinity:N \__fp_parse_word_inf:N
\cs_new_eq:NN \__fp_parse_caseless_nan:N \__fp_parse_word_nan:N
\cs_set_protected:Npn \__fp_tmp:w #1 #2
     {
    \cs_new:cpn { __fp_parse_word_#1:N }
      {
        \__fp_exp_after_f:nw { \__fp_parse_infix:NN }
```

```
        \s__fp \__fp_chk:w 10 #2 \__fp_sep:
      }
  }
\__fp_tmp:w {pt} { {1} {1000} {0000} {0000} {0000} }
\__fp_tmp:w {in} { {2} {7227} {0000} {0000} {0000} }
\__fp_tmp:w {pc} { {2} {1200} {0000} {0000} {0000} }
\__fp_tmp:w {cm} { {2} {2845} {2755} {9055} {1181} }
\__fp_tmp:w {mm} { {1} {2845} {2755} {9055} {1181} }
\__fp_tmp:w {dd} { {1} {1070} {0085} {6496} {0630} }
\__fp_tmp:w {cc} { {2} {1284} {0102} {7795} {2756} }
\__fp_tmp:w {nd} { {1} {1066} {9783} {4645} {6693} }
\__fp_tmp:w {nc} { {2} {1280} {3740} {1574} {8031} }
\__fp_tmp:w {bp} { {1} {1003} {7500} {0000} {0000} }
\__fp_tmp:w {sp} { {-4} {1525} {8789} {0625} {0000} }
\tl_map_inline:nn { {em} {ex} }
  {
    \cs_new:cpn { __fp_parse_word_#1:N }
      {
        \exp_after:wN \__fp_from_dim_test:ww
        \exp_after:wN 0 \exp_after:wN ,
        \int_value:w \dim_to_decimal_in_sp:n { 1 #1 } \exp_after:wN \__fp_sep:
        \exp:w \exp_end_continue_f:w \__fp_parse_infix:NN
      }
  }
\cs_new:Npn \__fp_parse_unary_function:NNN #1#2#3
  {
    \exp_after:wN \__fp_parse_apply_unary:NNNwN
    \exp_after:wN #3
    \exp_after:wN #2
    \exp_after:wN #1
    \exp:w
    \__fp_parse_operand:Nw \c__fp_prec_func_int \__fp_parse_expand:w
  }
\cs_new:Npn \__fp_parse_function:NNN #1#2#3
  {
    \exp_after:wN \__fp_parse_apply_function:NNNwN
    \exp_after:wN #3
    \exp_after:wN #2
    \exp_after:wN #1
    \exp:w
    \__fp_parse_operand:Nw \c__fp_prec_func_int \__fp_parse_expand:w
  }
\cs_new:Npn \__fp_parse:n #1
  {
    \exp:w
      \exp_after:wN \__fp_parse_after:ww
      \exp:w
        \__fp_parse_operand:Nw \c__fp_prec_end_int
        \__fp_parse_expand:w #1
        \s__fp_expr_mark \__fp_parse_infix_end:N
      \s__fp_expr_stop
    \exp_end:
  }
\cs_new:Npn \__fp_parse_after:ww
    #1@ \__fp_parse_infix_end:N \s__fp_expr_stop #2 { #2 #1 }
\cs_new:Npn \__fp_parse_o:n #1
  {
    \exp:w
      \exp_after:wN \__fp_parse_after:ww
      \exp:w
        \__fp_parse_operand:Nw \c__fp_prec_end_int
        \__fp_parse_expand:w #1
        \s__fp_expr_mark \__fp_parse_infix_end:N
      \s__fp_expr_stop
    {
      \exp_end_continue_f:w
      \__fp_exp_after_any_f:nw { \exp_after:wN \exp_stop_f: }
    }
  }
\cs_new:Npn \__fp_parse_operand:Nw #1
  {
    \exp_end_continue_f:w
    \exp_after:wN \__fp_parse_continue:NwN
    \exp_after:wN #1
    \exp:w \exp_end_continue_f:w
    \exp_after:wN \__fp_parse_one:Nw
    \exp_after:wN #1
    \exp:w
  }
\cs_new:Npn \__fp_parse_continue:NwN #1 #2 @ #3 { #3 #1 #2 @ }
\cs_new:Npn \__fp_parse_apply_binary:NwNwN #1 #2#3@ #4 #5#6@ #7
  {
    \exp_after:wN \__fp_parse_continue:NwN
    \exp_after:wN #1
    \exp:w \exp_end_continue_f:w
      \exp_after:wN \__fp_parse_apply_binary_chk:NN
        \cs:w
          __fp
          \__fp_type_from_scan:N #2
          _#4
          \__fp_type_from_scan:N #5
          _o:ww
        \cs_end:
        #4
      #2#3 #5#6
    \exp:w \exp_end_continue_f:w #7 #1
  }
\cs_new:Npn \__fp_parse_apply_binary_chk:NN #1#2
  {
    \if_meaning:w \scan_stop: #1
      \__fp_parse_apply_binary_error:NNN #2
    \fi:
    #1
  }
\cs_new:Npn \__fp_parse_apply_binary_error:NNN #1#2#3
  {
    #2
    \__fp_invalid_operation_o:Nww #1
  }
\cs_new:Npn \__fp_binary_type_o:Nww #1 #2#3 \__fp_sep: #4
  {
    \exp_after:wN \__fp_parse_apply_binary_chk:NN
      \cs:w
        __fp
        \__fp_type_from_scan:N #2
        _ #1
        \__fp_type_from_scan:N #4
        _o:ww
      \cs_end:
      #1
    #2 #3 \__fp_sep: #4
  }
\cs_new:Npn \__fp_binary_rev_type_o:Nww #1 #2#3 \__fp_sep: #4#5 \__fp_sep:
  {
    \exp_after:wN \__fp_parse_apply_binary_chk:NN
      \cs:w
        __fp
        \__fp_type_from_scan:N #4
        _ #1
        \__fp_type_from_scan:N #2
        _o:ww
      \cs_end:
      #1
    #4 #5 \__fp_sep: #2 #3 \__fp_sep:
  }
\cs_new:Npn \__fp_parse_infix_after_operand:NwN #1 #2\__fp_sep:
  {
    \__fp_exp_after_f:nw { \__fp_parse_infix:NN #1 }
    #2\__fp_sep:
  }
\cs_new:Npn \__fp_parse_infix:NN #1 #2
  {
    \if_catcode:w \scan_stop: \exp_not:N #2
      \if:w 0 \__fp_str_if_eq:nn { \s__fp_expr_mark } { \exp_not:N #2 }
        \exp_after:wN \exp_after:wN
        \exp_after:wN \__fp_parse_infix_mark:NNN
      \else:
        \exp_after:wN \exp_after:wN
        \exp_after:wN \__fp_parse_infix_juxt:N
      \fi:
    \else:
      \if_int_compare:w
          \__fp_int_eval:w
            ( `#2 \if_int_compare:w `#2 > `Z - 32 \fi: ) / 26
          = 3 \exp_stop_f:
        \exp_after:wN \exp_after:wN
        \exp_after:wN \__fp_parse_infix_juxt:N
      \else:
        \exp_after:wN \__fp_parse_infix_check:NNN
        \cs:w
          __fp_parse_infix_ \token_to_str:N #2 :N
          \exp_after:wN \exp_after:wN \exp_after:wN
        \cs_end:
      \fi:
    \fi:
    #1
    #2
  }
\cs_new:Npn \__fp_parse_infix_check:NNN #1#2#3
  {
    \if_meaning:w \scan_stop: #1
      \msg_expandable_error:nnn
        { fp } { missing } { * }
      \exp_after:wN \__fp_parse_infix_mul:N
      \exp_after:wN #2
      \exp_after:wN #3
    \else:
      \exp_after:wN #1
      \exp_after:wN #2
      \exp:w \exp_after:wN \__fp_parse_expand:w
    \fi:
  }
\cs_new:Npn \__fp_parse_infix_after_paren:NN #1 #2
  {
    \if_catcode:w \scan_stop: \exp_not:N #2
      \if:w 0 \__fp_str_if_eq:nn { \s__fp_expr_mark } { \exp_not:N #2 }
        \exp_after:wN \exp_after:wN
        \exp_after:wN \__fp_parse_infix_mark:NNN
      \else:
        \exp_after:wN \exp_after:wN
        \exp_after:wN \__fp_parse_infix_mul:N
      \fi:
    \else:
      \if_int_compare:w
          \__fp_int_eval:w
            ( `#2 \if_int_compare:w `#2 > `Z - 32 \fi: ) / 26
          = 3 \exp_stop_f:
        \exp_after:wN \exp_after:wN
        \exp_after:wN \__fp_parse_infix_mul:N
      \else:
        \exp_after:wN \__fp_parse_infix_check:NNN
        \cs:w
          __fp_parse_infix_ \token_to_str:N #2 :N
          \exp_after:wN \exp_after:wN \exp_after:wN
        \cs_end:
      \fi:
    \fi:
    #1
    #2
  }
\cs_new:Npn \__fp_parse_infix_mark:NNN #1#2#3 { #3 #1 }
\cs_new:Npn \__fp_parse_infix_end:N #1
  { @ \use_none:n \__fp_parse_infix_end:N }
\cs_set_protected:Npn \__fp_tmp:w #1
  {
    \cs_new:Npn #1 ##1
      {
        \if_int_compare:w ##1 > \c__fp_prec_end_int
          \exp_after:wN @
          \exp_after:wN \use_none:n
          \exp_after:wN #1
        \else:
          \msg_expandable_error:nnn { fp } { extra } { ) }
          \exp_after:wN \__fp_parse_infix:NN
          \exp_after:wN ##1
          \exp:w \exp_after:wN \__fp_parse_expand:w
        \fi:
      }
  }
\exp_args:Nc \__fp_tmp:w { __fp_parse_infix_):N }
```

```
\cs_set_protected:Npn \__fp_tmp:w #1
  {
    \cs_new:Npn #1 ##1
      {
        \if_int_compare:w ##1 > \c__fp_prec_comma_int
          \exp_after:wN @
          \exp_after:wN \use_none:n
          \exp_after:wN #1
        \else:
          \if_int_compare:w ##1 < \c__fp_prec_comma_int
            \exp_after:wN @
            \exp_after:wN \__fp_parse_apply_comma:NwNwN
            \exp_after:wN ,
            \exp:w
          \else:
            \exp_after:wN \__fp_parse_infix_comma:w
            \exp:w
          \fi:
          \__fp_parse_operand:Nw \c__fp_prec_comma_int
          \exp_after:wN \__fp_parse_expand:w
        \fi:
      }
  }
\exp_args:Nc \__fp_tmp:w { __fp_parse_infix_,:N }
\cs_new:Npn \__fp_parse_infix_comma:w #1 @
  { #1 @ \use_none:n }
\cs_new:Npn \__fp_parse_apply_comma:NwNwN #1 #2@ #3 #4@ #5
  {
    \exp_after:wN \__fp_parse_continue:NwN
    \exp_after:wN #1
    \exp:w \exp_end_continue_f:w
    \__fp_exp_after_tuple_f:nw { }
      \s__fp_tuple \__fp_tuple_chk:w { #2 #4 } \__fp_sep:
    #5 #1
  }
\cs_set_protected:Npn \__fp_tmp:w #1#2#3#4
  {
    \cs_new:Npn #1 ##1
      {
        \if_int_compare:w ##1 < #3
          \exp_after:wN @
          \exp_after:wN \__fp_parse_apply_binary:NwNwN
          \exp_after:wN #2
          \exp:w
          \__fp_parse_operand:Nw #4
          \exp_after:wN \__fp_parse_expand:w
        \else:
          \exp_after:wN @
          \exp_after:wN \use_none:n
          \exp_after:wN #1
        \fi:
      }
  }
\exp_args:Nc \__fp_tmp:w { __fp_parse_infix_^:N }   ^
  \c__fp_prec_hatii_int \c__fp_prec_hat_int
\exp_args:Nc \__fp_tmp:w { __fp_parse_infix_juxt:N } *
  \c__fp_prec_juxt_int \c__fp_prec_juxt_int
\exp_args:Nc \__fp_tmp:w { __fp_parse_infix_/:N }   /
  \c__fp_prec_times_int \c__fp_prec_times_int
\exp_args:Nc \__fp_tmp:w { __fp_parse_infix_mul:N } *
  \c__fp_prec_times_int \c__fp_prec_times_int
\exp_args:Nc \__fp_tmp:w { __fp_parse_infix_-:N }   -
  \c__fp_prec_plus_int  \c__fp_prec_plus_int
\exp_args:Nc \__fp_tmp:w { __fp_parse_infix_+:N }   +
  \c__fp_prec_plus_int  \c__fp_prec_plus_int
\exp_args:Nc \__fp_tmp:w { __fp_parse_infix_and:N } &
  \c__fp_prec_and_int   \c__fp_prec_and_int
\exp_args:Nc \__fp_tmp:w { __fp_parse_infix_or:N }  |
  \c__fp_prec_or_int    \c__fp_prec_or_int
\cs_new:cpn { __fp_parse_infix_(:N } #1
  { \__fp_parse_infix_mul:N #1 ( }
\cs_set_protected:Npn \__fp_tmp:w #1
  {
    \cs_new:cpn { __fp_parse_infix_*:N } ##1##2
      {
        \if:w * \exp_not:N ##2
          \exp_after:wN #1
          \exp_after:wN ##1
        \else:
          \exp_after:wN \__fp_parse_infix_mul:N
          \exp_after:wN ##1
          \exp_after:wN ##2
        \fi:
      }
  }
\exp_args:Nc \__fp_tmp:w { __fp_parse_infix_^:N }
\cs_set_protected:Npn \__fp_tmp:w #1#2#3
  {
    \cs_new:Npn #1 ##1##2
      {
        \if:w #2 \exp_not:N ##2
          \exp_after:wN #1
          \exp_after:wN ##1
          \exp:w \exp_after:wN \__fp_parse_expand:w
        \else:
          \exp_after:wN #3
          \exp_after:wN ##1
          \exp_after:wN ##2
        \fi:
      }
  }
\exp_args:Nc \__fp_tmp:w { __fp_parse_infix_|:N } | \__fp_parse_infix_or:N
\exp_args:Nc \__fp_tmp:w { __fp_parse_infix_&:N } & \__fp_parse_infix_and:N
\cs_set_protected:Npn \__fp_tmp:w #1#2#3#4
  {
    \cs_new:Npn #1 ##1
      {
        \if_int_compare:w ##1 < \c__fp_prec_quest_int
          #4
          \exp_after:wN @
          \exp_after:wN #2
          \exp:w
          \__fp_parse_operand:Nw #3
          \exp_after:wN \__fp_parse_expand:w
        \else:
          \exp_after:wN @
          \exp_after:wN \use_none:n
          \exp_after:wN #1
        \fi:
      }
  }
\exp_args:Nc \__fp_tmp:w { __fp_parse_infix_?:N }
  \__fp_ternary:NwwN \c__fp_prec_quest_int { }
\exp_args:Nc \__fp_tmp:w { __fp_parse_infix_::N }
  \__fp_ternary_auxii:NwwN \c__fp_prec_colon_int
  {
    \msg_expandable_error:nnnn
      { fp } { missing } { ? } { ~for~?: }
  }
\cs_new:cpn { __fp_parse_infix_<:N } #1
  { \__fp_parse_compare:NNNNNNN #1 1 0 0 0 0 < }
\cs_new:cpn { __fp_parse_infix_=:N } #1
  { \__fp_parse_compare:NNNNNNN #1 1 0 0 0 0 = }
\cs_new:cpn { __fp_parse_infix_>:N } #1
  { \__fp_parse_compare:NNNNNNN #1 1 0 0 0 0 > }
\cs_new:cpn { __fp_parse_infix_!:N } #1
  {
    \exp_after:wN \__fp_parse_compare:NNNNNNN
    \exp_after:wN #1
    \exp_after:wN 0
    \exp_after:wN 1
    \exp_after:wN 1
    \exp_after:wN 1
    \exp_after:wN 1
  }
\cs_new:Npn \__fp_parse_excl_error:
  {
    \msg_expandable_error:nnnn
      { fp } { missing } { = } { ~after~!. }
  }
\cs_new:Npn \__fp_parse_compare:NNNNNNN #1
  {
    \if_int_compare:w #1 < \c__fp_prec_comp_int
      \exp_after:wN \__fp_parse_compare_auxi:NNNNNNN
      \exp_after:wN \__fp_parse_excl_error:
    \else:
      \exp_after:wN @
      \exp_after:wN \use_none:n
      \exp_after:wN \__fp_parse_compare:NNNNNNN
    \fi:
  }
\cs_new:Npn \__fp_parse_compare_auxi:NNNNNNN #1#2#3#4#5#6#7
  {
    \if_case:w
      \__fp_int_eval:w \exp_after:wN ` \token_to_str:N #7 - `<
        \__fp_int_eval_end:
         \__fp_parse_compare_auxii:NNNNN #2#2#4#5#6
    \or: \__fp_parse_compare_auxii:NNNNN #2#3#2#5#6
    \or: \__fp_parse_compare_auxii:NNNNN #2#3#4#2#6
    \or: \__fp_parse_compare_auxii:NNNNN #2#3#4#5#2
    \else: #1 \__fp_parse_compare_end:NNNNw #3#4#5#6#7
    \fi:
  }
\cs_new:Npn \__fp_parse_compare_auxii:NNNNN #1#2#3#4#5
  {
    \exp_after:wN \__fp_parse_compare_auxi:NNNNNNN
    \exp_after:wN \prg_do_nothing:
    \exp_after:wN #1
    \exp_after:wN #2
    \exp_after:wN #3
    \exp_after:wN #4
    \exp_after:wN #5
    \exp:w \exp_after:wN \__fp_parse_expand:w
  }
```

```
\cs_new:Npn \__fp_parse_compare_end:NNNNw #1#2#3#4#5 \fi:
  {
    \fi:
    \exp_after:wN @
    \exp_after:wN \__fp_parse_apply_compare:NwNNNNNwN
    \exp_after:wN \c_one_fp
    \exp_after:wN #1
    \exp_after:wN #2
    \exp_after:wN #3
    \exp_after:wN #4
    \exp:w
    \__fp_parse_operand:Nw \c__fp_prec_comp_int \__fp_parse_expand:w #5
  }
\cs_new:Npn \__fp_parse_apply_compare:NwNNNNNwN
    #1 #2@ #3 #4#5#6#7 #8@ #9
  {
    \if_int_odd:w
        \if_meaning:w \c_zero_fp #3
          0
        \else:
          \if_case:w \__fp_compare_back_any:ww #8 #2 \exp_stop_f:
            #5 \or: #6 \or: #7 \else: #4
          \fi:
        \fi:
        \exp_stop_f:
      \exp_after:wN \__fp_parse_apply_compare_aux:NNwN
      \exp_after:wN \c_one_fp
    \else:
      \exp_after:wN \__fp_parse_apply_compare_aux:NNwN
      \exp_after:wN \c_zero_fp
    \fi:
    #1 #8 #9
  }
\cs_new:Npn \__fp_parse_apply_compare_aux:NNwN #1 #2 #3\__fp_sep: #4
  {
    \if_meaning:w \__fp_parse_compare:NNNNNNN #4
      \exp_after:wN \__fp_parse_continue_compare:NNwNN
      \exp_after:wN #1
      \exp_after:wN #2
      \exp:w \exp_end_continue_f:w
      \__fp_exp_after_o:w #3\__fp_sep:
      \exp:w \exp_end_continue_f:w
    \else:
      \exp_after:wN \__fp_parse_continue:NwN
      \exp_after:wN #2
      \exp:w \exp_end_continue_f:w
      \exp_after:wN #1
      \exp:w \exp_end_continue_f:w
    \fi:
    #4 #2
  }
\cs_new:Npn \__fp_parse_continue_compare:NNwNN #1#2 #3@ #4#5
  { #4 #2 #3@ #1 }
\cs_new:Npn \__fp_parse_function_all_fp_o:fnw #1#2#3 @
  {
    \__fp_array_if_all_fp:nTF {#3}
      { #2 #3 @ }
      {
        \__fp_error:nffn { bad-args }
          {#1}
          { \fp_to_tl:n { \s__fp_tuple \__fp_tuple_chk:w {#3} \__fp_sep: } }
          { }
        \exp_after:wN \c_nan_fp
      }
  }
\cs_new:Npn \__fp_parse_function_one_two:nnw #1#2#3
  {
    \__fp_if_type_fp:NTwFw
      #3 { } \s__fp \__fp_parse_function_one_two_error_o:w \s__fp_stop
    \__fp_parse_function_one_two_aux:nnw {#1} {#2} #3
  }
\cs_new:Npn \__fp_parse_function_one_two_error_o:w #1#2#3#4 @
  {
    \__fp_error:nffn { bad-args }
      {#2}
      { \fp_to_tl:n { \s__fp_tuple \__fp_tuple_chk:w {#4} \__fp_sep: } }
      { }
    \exp_after:wN \c_nan_fp
  }
\cs_new:Npn \__fp_parse_function_one_two_aux:nnw #1#2 #3\__fp_sep: #4
  {
    \__fp_if_type_fp:NTwFw
      #4 { }
      \s__fp
      {
        \if_meaning:w @ #4
          \exp_after:wN \use_iv:nnnn
        \fi:
        \__fp_parse_function_one_two_error_o:w
      }
      \s__fp_stop
    \__fp_parse_function_one_two_auxii:nnw {#1} {#2} #3\__fp_sep: #4
  }
\cs_new:Npn \__fp_parse_function_one_two_auxii:nnw #1#2#3\__fp_sep: #4\__fp_sep: #5
  {
    \if_meaning:w @ #5 \else:
      \exp_after:wN \__fp_parse_function_one_two_error_o:w
    \fi:
    \use_ii:nn {#1} { \use_none:n #2 } #3\__fp_sep: #4\__fp_sep: #5
  }
\cs_new:Npn \__fp_tuple_map_o:nw #1 \s__fp_tuple \__fp_tuple_chk:w #2 \__fp_sep:
  {
    \exp_after:wN \s__fp_tuple
    \exp_after:wN \__fp_tuple_chk:w
    \exp_after:wN {
      \exp:w \exp_end_continue_f:w
      \__fp_tuple_map_loop_o:nw {#1} #2
        { \s__fp \prg_break: } \__fp_sep:
      \prg_break_point:
    \exp_after:wN } \exp_after:wN \__fp_sep:
  }
\cs_new:Npn \__fp_tuple_map_loop_o:nw #1#2#3 \__fp_sep:
  {
    \use_none:n #2
    #1 #2 #3 \__fp_sep:
    \exp:w \exp_end_continue_f:w
    \__fp_tuple_map_loop_o:nw {#1}
  }
\cs_new:Npn \__fp_tuple_mapthread_o:nww #1
    \s__fp_tuple \__fp_tuple_chk:w #2 \__fp_sep:
    \s__fp_tuple \__fp_tuple_chk:w #3 \__fp_sep:
  {
    \exp_after:wN \s__fp_tuple
    \exp_after:wN \__fp_tuple_chk:w
    \exp_after:wN {
      \exp:w \exp_end_continue_f:w
      \__fp_tuple_mapthread_loop_o:nw {#1}
        #2 { \s__fp \prg_break: } \__fp_sep: @
        #3 { \s__fp \prg_break: } \__fp_sep:
      \prg_break_point:
    \exp_after:wN } \exp_after:wN \__fp_sep:
  }
\cs_new:Npn \__fp_tuple_mapthread_loop_o:nw #1#2#3 \__fp_sep: #4 @ #5#6 \__fp_sep:
  {
    \use_none:n #2
    \use_none:n #5
    #1 #2 #3 \__fp_sep: #5 #6 \__fp_sep:
    \exp:w \exp_end_continue_f:w
    \__fp_tuple_mapthread_loop_o:nw {#1} #4 @
  }
\msg_new:nnn { fp } { deprecated }
  { '#1'~deprecated;~use~'#2' }
\msg_new:nnn { fp } { unknown-fp-word }
  { Unknown~fp~word~#1. }
\msg_new:nnn { fp } { missing }
  { Missing~#1~inserted #2. }
\msg_new:nnn { fp } { extra }
  { Extra~#1~ignored. }
\msg_new:nnn { fp } { early-end }
  { Premature~end~in~fp~expression. }
\msg_new:nnn { fp } { after-e }
  { Cannot~use~#1 after~'e'. }
\msg_new:nnn { fp } { missing-number }
  { Missing~number~before~'#1'. }
\msg_new:nnn { fp } { unknown-symbol }
  { Unknown~symbol~#1~ignored. }
\msg_new:nnn { fp } { extra-comma }
  { Unexpected~comma~turned~to~nan~result. }
\msg_new:nnn { fp } { no-arg }
  { #1~got~no~argument;~used~nan. }
\msg_new:nnn { fp } { multi-arg }
  { #1~got~more~than~one~argument;~used~nan. }
\msg_new:nnn { fp } { bad-args }
  { Arguments~in~#1#2~are~invalid. }
\msg_new:nnn { fp } { infty-pi }
  { Math~command~#1 is~not~an~fp }
\cs_if_exist:cT { @unexpandable@protect }
  {
    \msg_new:nnn { fp } { robust-cmd }
      { Robust~command~#1 invalid~in~fp~expression! }
  }
%% File: l3fp-assign.dtx
\cs_new_protected:Npn \fp_new:N #1
  { \cs_new_eq:NN #1 \c_zero_fp }
\cs_generate_variant:Nn \fp_new:N {c}
\cs_new_protected:Npn \fp_set:Nn   #1#2
  { \__kernel_tl_set:Nx #1 { \exp_not:f { \__fp_parse:n {#2} } } }
\cs_new_protected:Npn \fp_gset:Nn  #1#2
  { \__kernel_tl_gset:Nx #1 { \exp_not:f { \__fp_parse:n {#2} } } }
\cs_new_protected:Npn \fp_const:Nn #1#2
  { \tl_const:Ne #1 { \exp_not:f { \__fp_parse:n {#2} } } }
\cs_generate_variant:Nn \fp_set:Nn { NV , c , cV }
\cs_generate_variant:Nn \fp_gset:Nn { NV , c , cV }
\cs_generate_variant:Nn \fp_const:Nn {c}
```

```
\cs_new_eq:NN \fp_set_eq:NN  \tl_set_eq:NN
\cs_new_eq:NN \fp_gset_eq:NN \tl_gset_eq:NN
\cs_generate_variant:Nn \fp_set_eq:NN  { c , Nc , cc }
\cs_generate_variant:Nn \fp_gset_eq:NN { c , Nc , cc }
\cs_new_protected:Npn \fp_zero:N #1 { \fp_set_eq:NN #1 \c_zero_fp }
\cs_new_protected:Npn \fp_gzero:N #1 { \fp_gset_eq:NN #1 \c_zero_fp }
\cs_generate_variant:Nn \fp_zero:N  { c }
\cs_generate_variant:Nn \fp_gzero:N { c }
\cs_new_protected:Npn \fp_zero_new:N #1
  { \fp_if_exist:NTF #1 { \fp_zero:N #1 } { \fp_new:N #1 } }
\cs_new_protected:Npn \fp_gzero_new:N #1
  { \fp_if_exist:NTF #1 { \fp_gzero:N #1 } { \fp_new:N #1 } }
\cs_generate_variant:Nn \fp_zero_new:N  { c }
\cs_generate_variant:Nn \fp_gzero_new:N { c }
\cs_new_protected:Npn \fp_add:Nn  { \__fp_add:NNNn \fp_set:Nn  + }
\cs_new_protected:Npn \fp_gadd:Nn { \__fp_add:NNNn \fp_gset:Nn + }
\cs_new_protected:Npn \fp_sub:Nn  { \__fp_add:NNNn \fp_set:Nn  - }
\cs_new_protected:Npn \fp_gsub:Nn { \__fp_add:NNNn \fp_gset:Nn - }
\cs_new_protected:Npn \__fp_add:NNNn #1#2#3#4
  { #1 #3 { #3 #2 \__fp_parse:n {#4} } }
\cs_generate_variant:Nn \fp_add:Nn  { c }
\cs_generate_variant:Nn \fp_gadd:Nn { c }
\cs_generate_variant:Nn \fp_sub:Nn  { c }
\cs_generate_variant:Nn \fp_gsub:Nn { c }
\cs_new_protected:Npn \fp_show:N { \__fp_show:NN \tl_show:n }
\cs_generate_variant:Nn \fp_show:N { c }
\cs_new_protected:Npn \fp_log:N { \__fp_show:NN \tl_log:n }
\cs_generate_variant:Nn \fp_log:N { c }
\cs_new_protected:Npn \__fp_show:NN #1#2
  {
    \__kernel_chk_tl_type:NnnT #2 { fp }
      { \exp_args:No \__fp_show_validate:n #2 }
      { \exp_args:Ne #1 { \token_to_str:N #2 = \fp_to_tl:N #2 } }
  }
\cs_new:Npn \__fp_show_validate:n #1
  {
    \__fp_show_validate:nn { #1 }
      {
        \s__fp \__fp_chk:w ??? \__fp_sep: or \iow_newline:
        \s__fp_tuple \__fp_tuple_chk:w ? \__fp_sep: or \iow_newline:
        \s__fp_symbolic \__fp_symbolic_chk:w ? , ? \__fp_sep:
      }
  }
\cs_new:Npn \__fp_show_validate_aux:n #1
  {
    \__fp_show_validate:nn { #1 } { }
  }
\cs_new:Npn \__fp_show_validate:nn #1#2
  {
    \tl_if_empty:nF { #1 }
      {
        \str_case:enF { \tl_head:n { #1 } }
          {
            { \s__fp }
              {
                \__fp_show_validate:w #1 \s__fp
                \__fp_chk:w ??? \__fp_sep: \s__fp_stop
              }
            { \s__fp_tuple }
              {
                \__fp_tuple_show_validate:w #1
                \s__fp_tuple \__fp_tuple_chk:w ?? \__fp_sep: \s__fp_stop
              }
            { \s__fp_symbolic }
              {
                \__fp_symbolic_show_validate:w #1
                \s__fp_symbolic \__fp_symbolic_chk:w ? , ?? \__fp_sep: \s__fp_stop
              }
          }
          { #2 }
      }
  }
\cs_new:Npn \__fp_show_validate:w
    #1 \s__fp \__fp_chk:w #2#3#4#5 \__fp_sep: #6 \s__fp_stop
  {
    \str_if_eq:nnF { #2 } {?}
      {
        \token_if_eq_meaning:NNTF #2 1
          { \s__fp \__fp_chk:w #2 #3 { #4 } #5 \__fp_sep: }
          { \s__fp \__fp_chk:w #2 #3 #4 #5 \__fp_sep: }
        \__fp_show_validate_aux:n { #6 }
      }
  }
\cs_new:Npn \__fp_tuple_show_validate:w
    #1 \s__fp_tuple \__fp_tuple_chk:w #2#3 \__fp_sep: #4 \s__fp_stop
  {
    \str_if_eq:nnF { #2 } {?}
      {
        \s__fp_tuple \__fp_tuple_chk:w { \__fp_show_validate_aux:n { #2 } }
          \__fp_sep:
      }
  }
```

```
\cs_new:Npn \__fp_symbolic_show_validate:w
    #1 \s__fp_symbolic \__fp_symbolic_chk:w #2 , #3#4 \__fp_sep: #5 \s__fp_stop
  {
    \str_if_eq:nnF { #2 } {?}
      {
        \s__fp_symbolic \__fp_symbolic_chk:w \exp_not:n { #2 } ,
        { \__fp_show_validate_aux:n { #3 } }\__fp_sep:
        \__fp_show_validate_aux:n { #5 }
      }
  }
\cs_new_protected:Npn \fp_show:n
  { \__kernel_msg_show_eval:Nn \fp_to_tl:n }
\cs_new_protected:Npn \fp_log:n
  { \__kernel_msg_log_eval:Nn \fp_to_tl:n }
\fp_const:Nn \c_e_fp          { 2.718 2818 2845 9045 }
\fp_const:Nn \c_one_fp        { 1 }
\fp_const:Nn \c_pi_fp         { 3.141 5926 5358 9793 }
\fp_const:Nn \c_one_degree_fp { 0.0 1745 3292 5199 4330 }
\fp_new:N \l_tmpa_fp
\fp_new:N \l_tmpb_fp
\fp_new:N \g_tmpa_fp
\fp_new:N \g_tmpb_fp
%% File: l3fp-logic.dtx
\cs_new:Npn \__fp_parse_word_max:N
  { \__fp_parse_function:NNN \__fp_minmax_o:Nw 2 }
\cs_new:Npn \__fp_parse_word_min:N
  { \__fp_parse_function:NNN \__fp_minmax_o:Nw 0 }
\prg_new_eq_conditional:NNn \fp_if_exist:N \cs_if_exist:N { TF , T , F , p }
\prg_new_eq_conditional:NNn \fp_if_exist:c \cs_if_exist:c { TF , T , F , p }
\prg_new_conditional:Npnn \fp_if_nan:n #1 { TF , T , F , p }
  {
    \if:w 3 \exp_last_unbraced:Nf \__fp_kind:w { \__fp_parse:n {#1} }
      \prg_return_true:
    \else:
      \prg_return_false:
    \fi:
  }
\prg_new_conditional:Npnn \fp_compare:n #1 { p , T , F , TF }
  {
    \exp_after:wN \__fp_compare_return:w
    \exp:w \exp_end_continue_f:w \__fp_parse:n {#1}
  }
\cs_new:Npn \__fp_compare_return:w #1#2#3\__fp_sep:
  {
    \if_charcode:w 0
          \__fp_if_type_fp:NTwFw
```

```
            #1 { \__fp_use_i_delimit_by_s_stop:nw #3 \s__fp_stop }
            \s__fp 1 \s__fp_stop
      \prg_return_false:
    \else:
      \prg_return_true:
    \fi:
  }
\prg_new_conditional:Npnn \fp_compare:nNn #1#2#3 { p , T , F , TF }
  {
    \if_int_compare:w
        \exp_after:wN \__fp_compare_aux:wn
          \exp:w \exp_end_continue_f:w \__fp_parse:n {#1} {#3}
        = \__fp_int_eval:w `#2 - `= \__fp_int_eval_end:
      \prg_return_true:
    \else:
      \prg_return_false:
    \fi:
  }
\cs_new:Npn \__fp_compare_aux:wn #1\__fp_sep: #2
  {
    \exp_after:wN \__fp_compare_back_any:ww
      \exp:w \exp_end_continue_f:w \__fp_parse:n {#2} #1\__fp_sep:
  }
\cs_new:Npn \__fp_compare_back:ww #1#2\__fp_sep: #3#4\__fp_sep:
  {
    \cs:w
      __fp
      \__fp_type_from_scan:N #1
      _bcmp
      \__fp_type_from_scan:N #3
      :ww
    \cs_end:
    #1#2\__fp_sep: #3#4\__fp_sep:
  }
\cs_new:Npn \__fp_compare_back_any:ww #1#2\__fp_sep: #3
  {
    \__fp_if_type_fp:NTwFw
      #1 { \__fp_if_type_fp:NTwFw #3 \use_i:nn \s__fp \use_ii:nn \s__fp_stop }
      \s__fp \use_ii:nn \s__fp_stop
    \__fp_compare_back:ww
    {
      \cs:w
        __fp
        \__fp_type_from_scan:N #1
        _compare_back
        \__fp_type_from_scan:N #3
        :ww
      \cs_end:
    }
    #1#2 \__fp_sep: #3
  }
\cs_new:Npn \__fp_bcmp:ww
    \s__fp \__fp_chk:w #1 #2 #3\__fp_sep:
    \s__fp \__fp_chk:w #4 #5 #6\__fp_sep:
  {
    \int_value:w
      \if_meaning:w 3 #1 \exp_after:wN \__fp_compare_nan:w \fi:
      \if_meaning:w 3 #4 \exp_after:wN \__fp_compare_nan:w \fi:
      \if_meaning:w 2 #5 - \fi:
      \if_meaning:w #2 #5
        \if_meaning:w #1 #4
          \if_meaning:w 1 #1
            \__fp_compare_npos:nwnw #6\__fp_sep: #3\__fp_sep:
          \else:
            0
          \fi:
        \else:
          \if_int_compare:w #4 < #1 - \fi: 1
        \fi:
      \else:
        \if_int_compare:w #1#4 = \c_zero_int
          0
        \else:
          1
        \fi:
      \fi:
    \exp_stop_f:
  }
\cs_new:Npn \__fp_compare_nan:w #1 \fi: \exp_stop_f: { 2 \exp_stop_f: }
\cs_new:Npn \__fp_compare_back_tuple:ww #1\__fp_sep: #2\__fp_sep: { 2 }
\cs_new:Npn \__fp_tuple_compare_back:ww #1\__fp_sep: #2\__fp_sep: { 2 }
\cs_new:Npn \__fp_tuple_compare_back_tuple:ww
  \s__fp_tuple \__fp_tuple_chk:w #1\__fp_sep:
  \s__fp_tuple \__fp_tuple_chk:w #2\__fp_sep:
  {
    \int_compare:nNnTF { \__fp_array_count:n {#1} } =
      { \__fp_array_count:n {#2} }
      {
        \int_value:w 0
          \__fp_tuple_compare_back_loop:w
              #1 { \s__fp \prg_break: } \__fp_sep: @
              #2 { \s__fp \prg_break: } \__fp_sep:
```

```
            \prg_break_point:
        \exp_stop_f:
      }
      { 2 }
  }
\cs_new:Npn \__fp_tuple_compare_back_loop:w #1#2 \__fp_sep: #3 @ #4#5 \__fp_sep:
  {
    \use_none:n #1
    \use_none:n #4
    \if_int_compare:w
        \__fp_compare_back_any:ww #1 #2 \__fp_sep: #4 #5 \__fp_sep: = \c_zero_int
    \else:
      2 \exp_after:wN \prg_break:
    \fi:
    \__fp_tuple_compare_back_loop:w #3 @
  }
\cs_new:Npn \__fp_compare_npos:nwnw #1#2\__fp_sep: #3#4\__fp_sep:
  {
    \if_int_compare:w #1 = #3 \exp_stop_f:
      \__fp_compare_significand:nnnnnnnn #2 #4
    \else:
      \if_int_compare:w #1 < #3 - \fi: 1
    \fi:
  }
\cs_new:Npn \__fp_compare_significand:nnnnnnnn #1#2#3#4#5#6#7#8
  {
    \if_int_compare:w #1#2 = #5#6 \exp_stop_f:
      \if_int_compare:w #3#4 = #7#8 \exp_stop_f:
        0
      \else:
        \if_int_compare:w #3#4 < #7#8 - \fi: 1
      \fi:
    \else:
      \if_int_compare:w #1#2 < #5#6 - \fi: 1
    \fi:
  }
\cs_new:Npn \fp_do_until:nn #1#2
  {
    #2
    \fp_compare:nF {#1}
      { \fp_do_until:nn {#1} {#2} }
  }
\cs_new:Npn \fp_do_while:nn #1#2
  {
    #2
    \fp_compare:nT {#1}
```

```
      { \fp_do_while:nn {#1} {#2} }
  }
\cs_new:Npn \fp_until_do:nn #1#2
  {
    \fp_compare:nF {#1}
      {
        #2
        \fp_until_do:nn {#1} {#2}
      }
  }
\cs_new:Npn \fp_while_do:nn #1#2
  {
    \fp_compare:nT {#1}
      {
        #2
        \fp_while_do:nn {#1} {#2}
      }
  }
\cs_new:Npn \fp_do_until:nNnn #1#2#3#4
  {
    #4
    \fp_compare:nNnF {#1} #2 {#3}
      { \fp_do_until:nNnn {#1} #2 {#3} {#4} }
  }
\cs_new:Npn \fp_do_while:nNnn #1#2#3#4
  {
    #4
    \fp_compare:nNnT {#1} #2 {#3}
      { \fp_do_while:nNnn {#1} #2 {#3} {#4} }
  }
\cs_new:Npn \fp_until_do:nNnn #1#2#3#4
  {
    \fp_compare:nNnF {#1} #2 {#3}
      {
        #4
        \fp_until_do:nNnn {#1} #2 {#3} {#4}
      }
  }
\cs_new:Npn \fp_while_do:nNnn #1#2#3#4
  {
    \fp_compare:nNnT {#1} #2 {#3}
      {
        #4
        \fp_while_do:nNnn {#1} #2 {#3} {#4}
      }
  }
```

```
\cs_new:Npn \fp_step_function:nnnN #1#2#3
  {
    \exp_after:wN \__fp_step:wwwN
      \exp:w \exp_end_continue_f:w \__fp_parse_o:n {#1}
      \exp:w \exp_end_continue_f:w \__fp_parse_o:n {#2}
      \exp:w \exp_end_continue_f:w \__fp_parse:n {#3}
  }
\cs_generate_variant:Nn \fp_step_function:nnnN { nnnc }
\cs_new:Npn \__fp_step:wwwN #1#2\__fp_sep: #3#4\__fp_sep: #5#6\__fp_sep: #7
  {
    \__fp_if_type_fp:NTwFw #1 { } \s__fp \prg_break: \s__fp_stop
    \__fp_if_type_fp:NTwFw #3 { } \s__fp \prg_break: \s__fp_stop
    \__fp_if_type_fp:NTwFw #5 { } \s__fp \prg_break: \s__fp_stop
    \use_i:nnnn
      { \__fp_step_fp:wwwN #1#2\__fp_sep: #3#4\__fp_sep: #5#6\__fp_sep: #7 }
    \prg_break_point:
    \use:n
      {
        \__fp_error:nfff { step-tuple } { \fp_to_tl:n { #1#2 \__fp_sep: } }
          { \fp_to_tl:n { #3#4 \__fp_sep: } } { \fp_to_tl:n { #5#6 \__fp_sep: } }
      }
  }
\cs_new:Npn \__fp_step_fp:wwwN
    #1 \__fp_sep: \s__fp \__fp_chk:w #2#3#4 \__fp_sep: #5\__fp_sep: #6
  {
    \token_if_eq_meaning:NNTF #2 1
      {
        \token_if_eq_meaning:NNTF #3 0
          { \__fp_step:NnnnnN > }
          { \__fp_step:NnnnnN < }
      }
      {
        \token_if_eq_meaning:NNTF #2 0
          {
            \msg_expandable_error:nnn { kernel }
              { zero-step } {#6}
          }
          {
            \__fp_error:nnfn { bad-step } { }
              { \fp_to_tl:n { \s__fp \__fp_chk:w #2#3#4 \__fp_sep: } } {#6}
          }
        \use_none:nnnnn
      }
        { #1 \__fp_sep: }
        { \c_nan_fp }
        { \s__fp \__fp_chk:w #2#3#4 \__fp_sep: }
        { #5 \__fp_sep: }
        #6
  }
\cs_new:Npn \__fp_step:NnnnnN #1#2#3#4#5#6
  {
    \fp_compare:nNnTF {#2} = {#3}
      {
        \__fp_error:nffn { tiny-step }
          { \fp_to_tl:n {#3} } { \fp_to_tl:n {#4} } {#6}
      }
      {
        \fp_compare:nNnF {#2} #1 {#5}
          {
            \exp_args:Nf #6 { \__fp_to_decimal_dispatch:w #2 }
            \__fp_step:NfnnnN
              #1 { \__fp_parse:n { #2 + #4 } } {#2} {#4} {#5} #6
          }
      }
  }
\cs_generate_variant:Nn \__fp_step:NnnnnN { Nf }
\cs_new_protected:Npn \fp_step_inline:nnnn
  {
    \int_gincr:N \g__kernel_prg_map_int
    \exp_args:NNc \__fp_step:NNnnnn
      \cs_gset_protected:Npn
      { __fp_map_ \int_use:N \g__kernel_prg_map_int :w }
  }
\cs_new_protected:Npn \fp_step_variable:nnnNn #1#2#3#4#5
  {
    \int_gincr:N \g__kernel_prg_map_int
    \exp_args:NNc \__fp_step:NNnnnn
      \cs_gset_protected:Npe
      { __fp_map_ \int_use:N \g__kernel_prg_map_int :w }
      {#1} {#2} {#3}
      {
        \tl_set:Nn \exp_not:N #4 {##1}
        \exp_not:n {#5}
      }
  }
\cs_new_protected:Npn \__fp_step:NNnnnn #1#2#3#4#5#6
  {
    #1 #2 ##1 {#6}
    \fp_step_function:nnnN {#3} {#4} {#5} #2
    \prg_break_point:Nn \scan_stop: { \int_gdecr:N \g__kernel_prg_map_int }
  }
\msg_new:nnn { fp } { step-tuple }
```

```
  { Tuple~argument~in~fp_step_...~{#1}{#2}{#3}. }
\msg_new:nnn { fp } { bad-step }
  { Invalid~step~size~#2~for~function~#3. }
\msg_new:nnn { fp } { tiny-step }
  { Tiny~step~size~(#1+#2=#1)~for~function~#3. }
\cs_new:Npn \__fp_minmax_o:Nw #1
  {
    \__fp_parse_function_all_fp_o:fnw
      { \token_if_eq_meaning:NNTF 0 #1 { min } { max } }
      { \__fp_minmax_aux_o:Nw #1 }
  }
\cs_new:Npn \__fp_minmax_aux_o:Nw #1#2 @
  {
    \if_meaning:w 0 #1
      \exp_after:wN \__fp_minmax_loop:Nww \exp_after:wN +
    \else:
      \exp_after:wN \__fp_minmax_loop:Nww \exp_after:wN -
    \fi:
    #2
    \s__fp \__fp_chk:w 2 #1 \s__fp_exact \__fp_sep:
    \s__fp \__fp_chk:w { 3 \__fp_minmax_break_o:w } \__fp_sep:
  }
\cs_new:Npn \__fp_minmax_loop:Nww
    #1 \s__fp \__fp_chk:w #2#3\__fp_sep: \s__fp \__fp_chk:w #4#5\__fp_sep:
  {
    \if_meaning:w 3 #4
      \if_meaning:w 3 #2
        \__fp_minmax_auxi:ww
      \else:
        \__fp_minmax_auxii:ww
      \fi:
    \else:
      \if_int_compare:w
          \__fp_compare_back:ww
            \s__fp \__fp_chk:w #4#5\__fp_sep:
            \s__fp \__fp_chk:w #2#3\__fp_sep:
          = #1 1 \exp_stop_f:
        \__fp_minmax_auxii:ww
      \else:
        \__fp_minmax_auxi:ww
      \fi:
    \fi:
    \__fp_minmax_loop:Nww #1
      \s__fp \__fp_chk:w #2#3\__fp_sep:
      \s__fp \__fp_chk:w #4#5\__fp_sep:
  }
```

```
\cs_new:Npn \__fp_minmax_auxi:ww
    #1 \fi: \fi: #2 \s__fp #3 \__fp_sep: \s__fp #4\__fp_sep:
  { \fi: \fi: #2 \s__fp #3 \__fp_sep: }
\cs_new:Npn \__fp_minmax_auxii:ww #1 \fi: \fi: #2 \s__fp #3 \__fp_sep:
  { \fi: \fi: #2 }
\cs_new:Npn \__fp_minmax_break_o:w #1 \fi: \fi: #2 \s__fp #3\__fp_sep: #4\__fp_sep:
  { \fi: \__fp_exp_after_o:w \s__fp #3\__fp_sep: }
\cs_new:Npn \__fp_not_o:w #1 \s__fp \__fp_chk:w #2#3\__fp_sep: @
  {
    \if_meaning:w 0 #2
      \exp_after:wN \exp_after:wN \exp_after:wN \c_one_fp
    \else:
      \exp_after:wN \exp_after:wN \exp_after:wN \c_zero_fp
    \fi:
  }
\cs_new:Npn \__fp_tuple_not_o:w #1 @ { \exp_after:wN \c_zero_fp }
\group_begin:
  \char_set_catcode_letter:N &
  \char_set_catcode_letter:N |
  \cs_new:Npn \__fp_&_o:ww #1 \s__fp \__fp_chk:w #2#3\__fp_sep:
    {
      \if_meaning:w 0 #2 #1
        \__fp_and_return:wNw \s__fp \__fp_chk:w #2#3\__fp_sep:
      \fi:
      \__fp_exp_after_o:w
    }
  \cs_new:Npn \__fp_&_tuple_o:ww #1 \s__fp \__fp_chk:w #2#3\__fp_sep:
    {
      \if_meaning:w 0 #2 #1
        \__fp_and_return:wNw \s__fp \__fp_chk:w #2#3\__fp_sep:
      \fi:
      \__fp_exp_after_tuple_o:w
    }
  \cs_new:Npn \__fp_tuple_&_o:ww #1\__fp_sep: { \__fp_exp_after_o:w }
  \cs_new:Npn \__fp_tuple_&_tuple_o:ww #1\__fp_sep: { \__fp_exp_after_tuple_o:w }
  \cs_new:Npn \__fp_|_o:ww { \__fp_&_o:ww \else: }
  \cs_new:Npn \__fp_|_tuple_o:ww { \__fp_&_tuple_o:ww \else: }
  \cs_new:Npn \__fp_tuple_|_o:ww #1\__fp_sep: #2\__fp_sep:
    { \__fp_exp_after_tuple_o:w #1\__fp_sep: }
  \cs_new:Npn \__fp_tuple_|_tuple_o:ww #1\__fp_sep: #2\__fp_sep:
    { \__fp_exp_after_tuple_o:w #1\__fp_sep: }
\group_end:
\cs_new:Npn \__fp_and_return:wNw #1\__fp_sep: \fi: #2\__fp_sep:
  { \fi: \__fp_exp_after_o:w #1\__fp_sep: }
\cs_new:Npn \__fp_ternary:NwwN #1 #2#3@ #4@ #5
  {
    \if_meaning:w \__fp_parse_infix_::N #5
      \if_charcode:w 0
            \__fp_if_type_fp:NTwFw
              #2 { \use_i:nn \__fp_use_i_delimit_by_s_stop:nw #3 \s__fp_stop }
              \s__fp 1 \s__fp_stop
        \exp_after:wN \exp_after:wN \exp_after:wN \__fp_ternary_auxii:NwwN
      \else:
        \exp_after:wN \exp_after:wN \exp_after:wN \__fp_ternary_auxi:NwwN
      \fi:
      \exp_after:wN #1
      \exp:w \exp_end_continue_f:w
      \__fp_exp_after_array_f:w #4 \s__fp_expr_stop
      \exp_after:wN @
      \exp:w
        \__fp_parse_operand:Nw \c__fp_prec_colon_int
        \__fp_parse_expand:w
    \else:
      \msg_expandable_error:nnnn
        { fp } { missing } { : } { ~for~?: }
      \exp_after:wN \__fp_parse_continue:NwN
      \exp_after:wN #1
      \exp:w \exp_end_continue_f:w
      \__fp_exp_after_array_f:w #4 \s__fp_expr_stop
      \exp_after:wN #5
      \exp_after:wN #1
    \fi:
  }
\cs_new:Npn \__fp_ternary_auxi:NwwN #1#2@#3@#4
  {
    \exp_after:wN \__fp_parse_continue:NwN
    \exp_after:wN #1
    \exp:w \exp_end_continue_f:w
    \__fp_exp_after_array_f:w #2 \s__fp_expr_stop
    #4 #1
  }
\cs_new:Npn \__fp_ternary_auxii:NwwN #1#2@#3@#4
  {
    \exp_after:wN \__fp_parse_continue:NwN
    \exp_after:wN #1
    \exp:w \exp_end_continue_f:w
    \__fp_exp_after_array_f:w #3 \s__fp_expr_stop
    #4 #1
  }
%% File: l3fp-basics.dtx
\cs_new:Npn \__fp_parse_word_abs:N
  { \__fp_parse_unary_function:NNN \__fp_set_sign_o:w 0 }
```

```
\cs_new:Npn \__fp_parse_word_logb:N
  { \__fp_parse_unary_function:NNN \__fp_logb_o:w ? }
\cs_new:Npn \__fp_parse_word_sign:N
  { \__fp_parse_unary_function:NNN \__fp_sign_o:w ? }
\cs_new:Npn \__fp_parse_word_sqrt:N
  { \__fp_parse_unary_function:NNN \__fp_sqrt_o:w ? }
\cs_new:cpe { __fp_-_o:ww } \s__fp
  {
    \exp_not:c { __fp_+_o:ww }
    \exp_not:n { \s__fp \__fp_neg_sign:N }
  }
\cs_new:cpn { __fp_+_o:ww }
    \s__fp #1 \__fp_chk:w #2 #3 \__fp_sep: \s__fp \__fp_chk:w #4 #5
  {
    \if_case:w
      \if_meaning:w #2 #4
        #2
      \else:
        \if_int_compare:w #2 > #4 \exp_stop_f:
          3
        \else:
          4
        \fi:
      \fi:
      \exp_stop_f:
           \exp_after:wN \__fp_add_zeros_o:Nww \int_value:w
    \or:   \exp_after:wN \__fp_add_normal_o:Nww \int_value:w
    \or:   \exp_after:wN \__fp_add_inf_o:Nww \int_value:w
    \or:   \__fp_case_return_i_o:ww
    \else: \exp_after:wN \__fp_add_return_ii_o:Nww \int_value:w
    \fi:
    #1 #5
    \s__fp \__fp_chk:w #2 #3 \__fp_sep:
    \s__fp \__fp_chk:w #4 #5
  }
\cs_new:Npn \__fp_add_return_ii_o:Nww #1 #2 \__fp_sep: \s__fp \__fp_chk:w #3 #4
  { \__fp_exp_after_o:w \s__fp \__fp_chk:w #3 #1 }
\cs_new:Npn \__fp_add_zeros_o:Nww #1 \s__fp \__fp_chk:w 0 #2
  {
    \if_int_compare:w #2 #1 = 20 \exp_stop_f:
      \exp_after:wN \__fp_add_return_ii_o:Nww
    \else:
      \__fp_case_return_i_o:ww
    \fi:
    #1
    \s__fp \__fp_chk:w 0 #2
  }
\cs_new:Npn \__fp_add_inf_o:Nww
    #1 \s__fp \__fp_chk:w 2 #2 #3\__fp_sep: \s__fp \__fp_chk:w 2 #4
  {
    \if_meaning:w #1 #2
      \__fp_case_return_i_o:ww
    \else:
      \__fp_case_use:nw
        {
          \exp_last_unbraced:Nf \__fp_invalid_operation_o:Nww
            { \token_if_eq_meaning:NNTF #1 #4 + - }
        }
    \fi:
    \s__fp \__fp_chk:w 2 #2 #3\__fp_sep:
    \s__fp \__fp_chk:w 2 #4
  }
\cs_new:Npn \__fp_add_normal_o:Nww #1 \s__fp \__fp_chk:w 1 #2
  {
    \if_meaning:w #1#2
      \exp_after:wN \__fp_add_npos_o:NnwNnw
    \else:
      \exp_after:wN \__fp_sub_npos_o:NnwNnw
    \fi:
    #2
  }
\cs_new:Npn \__fp_add_npos_o:NnwNnw #1#2#3 \__fp_sep: \s__fp \__fp_chk:w 1 #4 #5
  {
    \exp_after:wN \__fp_sanitize:Nw
    \exp_after:wN #1
    \int_value:w \__fp_int_eval:w
      \if_int_compare:w #2 > #5 \exp_stop_f:
        #2
        \exp_after:wN \__fp_add_big_i_o:wNww \int_value:w -
      \else:
        #5
        \exp_after:wN \__fp_add_big_ii_o:wNww \int_value:w
      \fi:
      \__fp_int_eval:w #5 - #2 \__fp_sep: #1 #3\__fp_sep:
  }
\cs_new:Npn \__fp_add_big_i_o:wNww #1\__fp_sep: #2 #3\__fp_sep: #4\__fp_sep:
  {
    \__fp_decimate:nNnnnn {#1}
      \__fp_add_significand_o:NnnwnnnnN
      #4
    #3
    #2
  }
\cs_new:Npn \__fp_add_big_ii_o:wNww #1\__fp_sep: #2 #3\__fp_sep: #4\__fp_sep:
  {
    \__fp_decimate:nNnnnn {#1}
      \__fp_add_significand_o:NnnwnnnnN
      #3
    #4
    #2
  }
\cs_new:Npn \__fp_add_significand_o:NnnwnnnnN #1 #2#3 #4\__fp_sep: #5#6#7#8
  {
    \exp_after:wN \__fp_add_significand_test_o:N
    \int_value:w \__fp_int_eval:w 1#5#6 + #2
      \exp_after:wN \__fp_add_significand_pack:NNNNNNN
      \int_value:w \__fp_int_eval:w 1#7#8 + #3 \__fp_sep: #1
  }
\cs_new:Npn \__fp_add_significand_pack:NNNNNNN #1 #2#3#4#5#6#7
  {
    \if_meaning:w 2 #1
      + 1
    \fi:
    \__fp_sep: #2 #3 #4 #5 #6 #7 \__fp_sep:
  }
\cs_new:Npn \__fp_add_significand_test_o:N #1
  {
    \if_meaning:w 2 #1
      \exp_after:wN \__fp_add_significand_carry_o:wwwNN
    \else:
      \exp_after:wN \__fp_add_significand_no_carry_o:wwwNN
    \fi:
  }
\cs_new:Npn \__fp_add_significand_no_carry_o:wwwNN
    #1\__fp_sep: #2\__fp_sep: #3#4 \__fp_sep: #5#6
  {
    \exp_after:wN \__fp_basics_pack_high:NNNNNw
    \int_value:w \__fp_int_eval:w 1 #1
      \exp_after:wN \__fp_basics_pack_low:NNNNNw
      \int_value:w \__fp_int_eval:w 1 #2 #3#4
        + \__fp_round:NNN #6 #4 #5
        \exp_after:wN \__fp_sep:
  }
\cs_new:Npn \__fp_add_significand_carry_o:wwwNN
    #1\__fp_sep: #2\__fp_sep: #3#4\__fp_sep: #5#6
  {
    + 1
    \exp_after:wN \__fp_basics_pack_weird_high:NNNNNNNNw
    \int_value:w \__fp_int_eval:w 1 1 #1
      \exp_after:wN \__fp_basics_pack_weird_low:NNNNw
      \int_value:w \__fp_int_eval:w 1 #2#3 +
        \exp_after:wN \__fp_round:NNN
        \exp_after:wN #6
        \exp_after:wN #3
        \int_value:w \__fp_round_digit:Nw #4 #5 \__fp_sep:
        \exp_after:wN \__fp_sep:
  }
\cs_new:Npn \__fp_sub_npos_o:NnwNnw
    #1#2#3\__fp_sep: \s__fp \__fp_chk:w 1 #4#5#6\__fp_sep:
  {
    \if_case:w
        \__fp_compare_npos:nwnw {#2} #3\__fp_sep: {#5} #6\__fp_sep: \exp_stop_f:
      \exp_after:wN \__fp_sub_eq_o:Nnwnw
    \or:
      \exp_after:wN \__fp_sub_npos_i_o:Nnwnw
    \else:
      \exp_after:wN \__fp_sub_npos_ii_o:Nnwnw
    \fi:
    #1 {#2} #3\__fp_sep: {#5} #6\__fp_sep:
  }
\cs_new:Npn \__fp_sub_eq_o:Nnwnw #1#2\__fp_sep: #3\__fp_sep:
  { \exp_after:wN \c_zero_fp }
\cs_new:Npn \__fp_sub_npos_ii_o:Nnwnw #1 #2\__fp_sep: #3\__fp_sep:
  {
    \exp_after:wN \__fp_sub_npos_i_o:Nnwnw
      \int_value:w \__fp_neg_sign:N #1
      #3\__fp_sep: #2\__fp_sep:
  }
\cs_new:Npn \__fp_sub_npos_i_o:Nnwnw #1 #2#3\__fp_sep: #4#5\__fp_sep:
  {
    \exp_after:wN \__fp_sanitize:Nw
    \exp_after:wN #1
    \int_value:w \__fp_int_eval:w
      #2
      \if_int_compare:w #2 = #4 \exp_stop_f:
        \exp_after:wN \__fp_sub_back_near_o:nnnnnnnnN
      \else:
        \exp_after:wN \__fp_decimate:nNnnnn \exp_after:wN
          { \int_value:w \__fp_int_eval:w #2 - #4 - 1 \exp_after:wN }
          \exp_after:wN \__fp_sub_back_far_o:NnnwnnnnN
      \fi:
        #5
      #3
      #1
  }
\cs_new:Npn \__fp_sub_back_near_o:nnnnnnnnN #1#2#3#4 #5#6#7#8 #9
  {
    \exp_after:wN \__fp_sub_back_near_after:wNNNNw
    \int_value:w \__fp_int_eval:w 10#5#6 - #1#2 - 11
      \exp_after:wN \__fp_sub_back_near_pack:NNNNNNw
      \int_value:w \__fp_int_eval:w 11#7#8 - #3#4 \exp_after:wN \__fp_sep:
  }
\cs_new:Npn \__fp_sub_back_near_pack:NNNNNNw #1#2#3#4#5#6#7 \__fp_sep:
  { + #1#2 \__fp_sep: {#3#4#5#6} {#7} \__fp_sep: }
\cs_new:Npn \__fp_sub_back_near_after:wNNNNw 10 #1#2#3#4 #5 \__fp_sep:
  {
    \if_meaning:w 0 #1
      \exp_after:wN \__fp_sub_back_shift:wnnnn
    \fi:
    \__fp_sep: {#1#2#3#4} {#5}
  }
\cs_new:Npn \__fp_sub_back_shift:wnnnn \__fp_sep: #1#2
  {
    \exp_after:wN \__fp_sub_back_shift_ii:ww
    \int_value:w #1 #2 0 \__fp_sep:
  }
\cs_new:Npn \__fp_sub_back_shift_ii:ww #1 0 \__fp_sep: #2#3 \__fp_sep:
  {
    \if_meaning:w @ #1 @
      - 7
      - \exp_after:wN \use_i:nnn
        \exp_after:wN \__fp_sub_back_shift_iii:NNNNNNNNw
        \int_value:w #2#3 0 ~ 123456789\__fp_sep:
    \else:
      - \__fp_sub_back_shift_iii:NNNNNNNNw #1 123456789\__fp_sep:
    \fi:
    \exp_after:wN \__fp_pack_twice_four:wNNNNNNNN
    \exp_after:wN \__fp_pack_twice_four:wNNNNNNNN
    \exp_after:wN \__fp_sub_back_shift_iv:nnnnw
    \exp_after:wN \__fp_sep:
    \int_value:w
    #1 ~ #2#3 0 ~ 0000 0000 0000 000 \__fp_sep:
  }
\cs_new:Npn \__fp_sub_back_shift_iii:NNNNNNNNw #1#2#3#4#5#6#7#8#9\__fp_sep: {#8}
\cs_new:Npn \__fp_sub_back_shift_iv:nnnnw #1 \__fp_sep: #2 \__fp_sep:
  { \__fp_sep: #1 \__fp_sep: }
\cs_new:Npn \__fp_sub_back_far_o:NnnwnnnnN #1 #2#3 #4\__fp_sep: #5#6#7#8
  {
    \if_case:w
      \if_int_compare:w 1 #2 = #5#6 \use_i:nnnn #7 \exp_stop_f:
        \if_int_compare:w #3 = \use_none:n #7#8 0 \exp_stop_f:
          0
        \else:
          \if_int_compare:w #3 > \use_none:n #7#8 0 - \fi: 1
        \fi:
      \else:
        \if_int_compare:w 1 #2 > #5#6 \use_i:nnnn #7 - \fi: 1
      \fi:
      \exp_stop_f:
           \exp_after:wN \__fp_sub_back_quite_far_o:wwNN
    \or:   \exp_after:wN \__fp_sub_back_very_far_o:wwwwNN
    \else: \exp_after:wN \__fp_sub_back_not_far_o:wwwwNN
    \fi:
    #2 ~ #3 \__fp_sep: #5 #6 ~ #7 #8 \__fp_sep: #1
  }
\cs_new:Npn \__fp_sub_back_quite_far_o:wwNN #1\__fp_sep: #2\__fp_sep: #3#4
  {
    \exp_after:wN \__fp_sub_back_quite_far_ii:NN
    \exp_after:wN #3
    \exp_after:wN #4
  }
\cs_new:Npn \__fp_sub_back_quite_far_ii:NN #1#2
  {
    \if_case:w \__fp_round_neg:NNN #2 0 #1
      \exp_after:wN \use_i:nn
    \else:
      \exp_after:wN \use_ii:nn
    \fi:
      { \__fp_sep: {1000} {0000} {0000} {0000} \__fp_sep: }
      { - 1 \__fp_sep: {9999} {9999} {9999} {9999} \__fp_sep: }
  }
\cs_new:Npn \__fp_sub_back_not_far_o:wwwwNN #1 ~ #2\__fp_sep: #3 ~ #4\__fp_sep: #5#6
  {
    - 1
    \exp_after:wN \__fp_sub_back_near_after:wNNNNw
    \int_value:w \__fp_int_eval:w 1#30 - #1 - 11
      \exp_after:wN \__fp_sub_back_near_pack:NNNNNNw
      \int_value:w \__fp_int_eval:w 11 0000 0000 + #40 - #2
        - \exp_after:wN \__fp_round_neg:NNN
          \exp_after:wN #6
          \use_none:nnnnnnn #2 #5
        \exp_after:wN \__fp_sep:
  }
\cs_new:Npn \__fp_sub_back_very_far_o:wwwwNN #1#2#3#4#5#6#7
  {
    \__fp_pack_eight:wNNNNNNNN
    \__fp_sub_back_very_far_ii_o:nnNwwNN
    { 0 #1#2#3 #4#5#6#7 }
    \__fp_sep:
  }
\cs_new:Npn \__fp_sub_back_very_far_ii_o:nnNwwNN
    #1#2 \__fp_sep: #3 \__fp_sep: #4 ~ #5\__fp_sep: #6#7
  {
    \exp_after:wN \__fp_basics_pack_high:NNNNNw
    \int_value:w \__fp_int_eval:w 1#4 - #1 - 1
      \exp_after:wN \__fp_basics_pack_low:NNNNNw
      \int_value:w \__fp_int_eval:w 2#5 - #2
        - \exp_after:wN \__fp_round_neg:NNN
          \exp_after:wN #7
          \int_value:w
            \if_int_odd:w \__fp_int_eval:w #5 - #2 \__fp_int_eval_end:
              1 \else: 2 \fi:
          \int_value:w \__fp_round_digit:Nw #3 #6 \__fp_sep:
      \exp_after:wN \__fp_sep:
  }
\cs_new:cpn { __fp_*_o:ww }
  {
    \__fp_mul_cases_o:NnNnww
      *
      { - 2 + }
      \__fp_mul_npos_o:Nww
      { }
  }
\cs_new:Npn \__fp_mul_cases_o:NnNnww
    #1#2#3#4 \s__fp \__fp_chk:w #5#6#7\__fp_sep: \s__fp \__fp_chk:w #8#9
  {
    \if_case:w \__fp_int_eval:w
                 \if_int_compare:w #5 #8 = 11 ~
                   1
                 \else:
                   \if_meaning:w 3 #8
                     3
                   \else:
                     \if_meaning:w 3 #5
                       2
                     \else:
                       \if_int_compare:w #5 #8 = 10 ~
                         9 #2 - 2
                       \else:
                         (#5 #2 #8) / 2 * 2 + 7
                       \fi:
                     \fi:
```

```
                   \fi:
                 \fi:
                 \if_meaning:w #6 #9 - 1 \fi:
               \__fp_int_eval_end:
         \__fp_case_use:nw { #3 0 }
    \or: \__fp_case_use:nw { #3 2 }
    \or: \__fp_case_return_i_o:ww
    \or: \__fp_case_return_ii_o:ww
    \or: \__fp_case_return_o:Nww \c_zero_fp
    \or: \__fp_case_return_o:Nww \c_minus_zero_fp
    \or: \__fp_case_use:nw { \__fp_invalid_operation_o:Nww #1 }
    \or: \__fp_case_use:nw { \__fp_invalid_operation_o:Nww #1 }
    \or: \__fp_case_return_o:Nww \c_inf_fp
    \or: \__fp_case_return_o:Nww \c_minus_inf_fp
    #4
    \fi:
    \s__fp \__fp_chk:w #5 #6 #7\__fp_sep:
    \s__fp \__fp_chk:w #8 #9
  }
\cs_new:Npn \__fp_mul_npos_o:Nww
    #1 \s__fp \__fp_chk:w #2 #3 #4 #5 \__fp_sep:
       \s__fp \__fp_chk:w #6 #7 #8 #9 \__fp_sep:
  {
    \exp_after:wN \__fp_sanitize:Nw
    \exp_after:wN #1
    \int_value:w \__fp_int_eval:w
      #4 + #8
      \__fp_mul_significand_o:nnnnNnnnn #5 #1 #9
  }
\cs_new:Npn \__fp_mul_significand_o:nnnnNnnnn #1#2#3#4 #5 #6#7#8#9
  {
    \exp_after:wN \__fp_mul_significand_test_f:NNN
    \exp_after:wN #5
    \int_value:w \__fp_int_eval:w 99990000 + #1*#6 +
      \exp_after:wN \__fp_mul_significand_keep:NNNNNw
      \int_value:w \__fp_int_eval:w 99990000 + #1*#7 + #2*#6 +
        \exp_after:wN \__fp_mul_significand_keep:NNNNNw
        \int_value:w \__fp_int_eval:w 99990000 + #1*#8 + #2*#7 + #3*#6 +
          \exp_after:wN \__fp_mul_significand_drop:NNNNNw
          \int_value:w \__fp_int_eval:w 99990000 + #1*#9 + #2*#8 +
            #3*#7 + #4*#6 +
            \exp_after:wN \__fp_mul_significand_drop:NNNNNw
            \int_value:w \__fp_int_eval:w 99990000 + #2*#9 + #3*#8 +
              #4*#7 +
              \exp_after:wN \__fp_mul_significand_drop:NNNNNw
              \int_value:w \__fp_int_eval:w 99990000 + #3*#9 + #4*#8 +
                \exp_after:wN \__fp_mul_significand_drop:NNNNNw
                \int_value:w \__fp_int_eval:w 100000000 + #4*#9 \__fp_sep:
    \__fp_sep: \exp_after:wN \__fp_sep:
  }
\cs_new:Npn \__fp_mul_significand_drop:NNNNNw #1#2#3#4#5 #6\__fp_sep:
  { #1#2#3#4#5 \__fp_sep: + #6 }
\cs_new:Npn \__fp_mul_significand_keep:NNNNNw #1#2#3#4#5 #6\__fp_sep:
  { #1#2#3#4#5 \__fp_sep: #6 \__fp_sep: }
\cs_new:Npn \__fp_mul_significand_test_f:NNN #1 #2 #3
  {
    \if_meaning:w 0 #3
      \exp_after:wN \__fp_mul_significand_small_f:NNwwwN
    \else:
      \exp_after:wN \__fp_mul_significand_large_f:NwwNNNN
    \fi:
    #1 #3
  }
\cs_new:Npn \__fp_mul_significand_large_f:NwwNNNN
    #1 #2\__fp_sep: #3\__fp_sep: #4#5#6#7\__fp_sep: +
  {
    \exp_after:wN \__fp_basics_pack_high:NNNNNw
    \int_value:w \__fp_int_eval:w 1#2
      \exp_after:wN \__fp_basics_pack_low:NNNNNw
      \int_value:w \__fp_int_eval:w 1#3#4#5#6#7
        + \exp_after:wN \__fp_round:NNN
          \exp_after:wN #1
          \exp_after:wN #7
          \int_value:w \__fp_round_digit:Nw
  }
\cs_new:Npn \__fp_mul_significand_small_f:NNwwwN
    #1 #2#3\__fp_sep: #4#5\__fp_sep: #6\__fp_sep: + #7
  {
    - 1
    \exp_after:wN \__fp_basics_pack_high:NNNNNw
    \int_value:w \__fp_int_eval:w 1#3#4
      \exp_after:wN \__fp_basics_pack_low:NNNNNw
      \int_value:w \__fp_int_eval:w 1#5#6#7
        + \exp_after:wN \__fp_round:NNN
          \exp_after:wN #1
          \exp_after:wN #7
          \int_value:w \__fp_round_digit:Nw
  }
\cs_new:cpn { __fp_/_o:ww }
  {
    \__fp_mul_cases_o:NnNnww
      /
      { - }
      \__fp_div_npos_o:Nww
      {
        \or:
          \__fp_case_use:nw
            { \__fp_division_by_zero_o:NNww \c_inf_fp / }
        \or:
          \__fp_case_use:nw
            { \__fp_division_by_zero_o:NNww \c_minus_inf_fp / }
      }
  }
\cs_new:Npn \__fp_div_npos_o:Nww
    #1 \s__fp \__fp_chk:w 1 #2 #3 #4 \__fp_sep:
       \s__fp \__fp_chk:w 1 #5 #6 #7#8#9\__fp_sep:
  {
    \exp_after:wN \__fp_sanitize:Nw
    \exp_after:wN #1
    \int_value:w \__fp_int_eval:w
      #3 - #6
      \exp_after:wN \__fp_div_significand_i_o:wnnw
        \int_value:w \__fp_int_eval:w #7 \use_i:nnnn #8 + 1 \__fp_sep:
        #4
        {#7}{#8}#9 \__fp_sep:
        #1
  }
\cs_new:Npn \__fp_div_significand_i_o:wnnw #1 \__fp_sep: #2#3 #4 \__fp_sep:
  {
    \exp_after:wN \__fp_div_significand_test_o:w
    \int_value:w \__fp_int_eval:w
      \exp_after:wN \__fp_div_significand_calc:wwnnnnnnn
      \int_value:w \__fp_int_eval:w 999999 + #2 #3 0 / #1 \__fp_sep:
        #2 #3 \__fp_sep:
        #4
        { \exp_after:wN \__fp_div_significand_ii:wwn \int_value:w #1 }
        { \exp_after:wN \__fp_div_significand_ii:wwn \int_value:w #1 }
        { \exp_after:wN \__fp_div_significand_ii:wwn \int_value:w #1 }
        { \exp_after:wN \__fp_div_significand_iii:wwnnnnn \int_value:w #1 }
  }
\cs_new:Npn \__fp_div_significand_calc:wwnnnnnnn 1#1
  {
    \if_meaning:w 1 #1
      \exp_after:wN \__fp_div_significand_calc_i:wwnnnnnnn
    \else:
      \exp_after:wN \__fp_div_significand_calc_ii:wwnnnnnnn
    \fi:
  }
```

```
\cs_new:Npn \__fp_div_significand_calc_i:wwnnnnnnn
  #1\__fp_sep: #2\__fp_sep:#3#4 #5#6#7#8 #9
  {
    1 1 #1
    #9 \exp_after:wN \__fp_sep:
    \int_value:w \__fp_int_eval:w \c__fp_Bigg_leading_shift_int
      + #2 - #1 * #5 - #5#60
      \exp_after:wN \__fp_pack_Bigg:NNNNNNw
      \int_value:w \__fp_int_eval:w \c__fp_Bigg_middle_shift_int
        + #3 - #1 * #6 - #70
        \exp_after:wN \__fp_pack_Bigg:NNNNNNw
        \int_value:w \__fp_int_eval:w \c__fp_Bigg_middle_shift_int
          + #4 - #1 * #7 - #80
          \exp_after:wN \__fp_pack_Bigg:NNNNNNw
          \int_value:w \__fp_int_eval:w \c__fp_Bigg_trailing_shift_int
            - #1 * #8 \__fp_sep:
    {#5}{#6}{#7}{#8}
  }
\cs_new:Npn \__fp_div_significand_calc_ii:wwnnnnnnn
  #1\__fp_sep: #2\__fp_sep:#3#4 #5#6#7#8 #9
  {
    1 0 #1
    #9 \exp_after:wN \__fp_sep:
    \int_value:w \__fp_int_eval:w \c__fp_Bigg_leading_shift_int
      + #2 - #1 * #5
      \exp_after:wN \__fp_pack_Bigg:NNNNNNw
      \int_value:w \__fp_int_eval:w \c__fp_Bigg_middle_shift_int
        + #3 - #1 * #6
        \exp_after:wN \__fp_pack_Bigg:NNNNNNw
        \int_value:w \__fp_int_eval:w \c__fp_Bigg_middle_shift_int
          + #4 - #1 * #7
          \exp_after:wN \__fp_pack_Bigg:NNNNNNw
          \int_value:w \__fp_int_eval:w \c__fp_Bigg_trailing_shift_int
            - #1 * #8 \__fp_sep:
    {#5}{#6}{#7}{#8}
  }
\cs_new:Npn \__fp_div_significand_ii:wwn #1\__fp_sep: #2\__fp_sep:#3
  {
    \exp_after:wN \__fp_div_significand_pack:NNN
    \int_value:w \__fp_int_eval:w
      \exp_after:wN \__fp_div_significand_calc:wwnnnnnnn
      \int_value:w \__fp_int_eval:w
        999999 + #2 #3 0 / #1 \__fp_sep: #2 #3 \__fp_sep:
  }
\cs_new:Npn \__fp_div_significand_iii:wwnnnnn #1\__fp_sep: #2\__fp_sep:#3#4#5 #6#7
  {
    0
    \exp_after:wN \__fp_div_significand_iv:wwnnnnnnn
    \int_value:w \__fp_int_eval:w ( 2 * #2 #3) / #6 #7 \__fp_sep: % <- P
      #2 \__fp_sep: {#3} {#4} {#5}
      {#6} {#7}
  }
\cs_new:Npn \__fp_div_significand_iv:wwnnnnnnn
    #1\__fp_sep: #2\__fp_sep:#3#4#5 #6#7#8#9
  {
    + 5 * #1
    \exp_after:wN \__fp_div_significand_vi:Nw
    \int_value:w \__fp_int_eval:w -50 + 2*#2#3 - #1*#6#7 +
      \exp_after:wN \__fp_div_significand_v:NN
      \int_value:w \__fp_int_eval:w 499950 + 2*#4 - #1*#8 +
        \exp_after:wN \__fp_div_significand_v:NN
        \int_value:w \__fp_int_eval:w 500000 + 2*#5 - #1*#9 \__fp_sep:
  }
\cs_new:Npn \__fp_div_significand_v:NN #1#2 { #1#2 \__fp_int_eval_end: + }
\cs_new:Npn \__fp_div_significand_vi:Nw #1#2\__fp_sep:
  {
    \if_meaning:w 0 #1
      \if_int_compare:w \__fp_int_eval:w #2 > 0 + 1 \fi:
    \else:
      \if_meaning:w - #1 - \else: + \fi: 1
    \fi:
    \__fp_sep:
  }
\cs_new:Npn \__fp_div_significand_pack:NNN 1 #1 #2 { + #1 #2 \__fp_sep: }
\cs_new:Npn \__fp_div_significand_test_o:w 10 #1
  {
    \if_meaning:w 0 #1
      \exp_after:wN \__fp_div_significand_small_o:wwwNNNNwN
    \else:
      \exp_after:wN \__fp_div_significand_large_o:wwwNNNNwN
    \fi:
    #1
  }
\cs_new:Npn \__fp_div_significand_small_o:wwwNNNNwN
    0 #1\__fp_sep: #2\__fp_sep: #3\__fp_sep: #4#5#6#7#8\__fp_sep: #9
  {
    \exp_after:wN \__fp_basics_pack_high:NNNNNw
    \int_value:w \__fp_int_eval:w 1 #1#2
      \exp_after:wN \__fp_basics_pack_low:NNNNNw
      \int_value:w \__fp_int_eval:w 1 #3#4#5#6#7
        + \__fp_round:NNN #9 #7 #8
        \exp_after:wN \__fp_sep:
  }
\cs_new:Npn \__fp_div_significand_large_o:wwwNNNNwN
    #1\__fp_sep: #2\__fp_sep: #3\__fp_sep: #4#5#6#7#8\__fp_sep: #9
  {
    + 1
    \exp_after:wN \__fp_basics_pack_weird_high:NNNNNNNNw
    \int_value:w \__fp_int_eval:w 1 #1 #2
      \exp_after:wN \__fp_basics_pack_weird_low:NNNNw
      \int_value:w \__fp_int_eval:w 1 #3 #4 #5 #6 +
        \exp_after:wN \__fp_round:NNN
        \exp_after:wN #9
        \exp_after:wN #6
        \int_value:w \__fp_round_digit:Nw #7 #8 \__fp_sep:
      \exp_after:wN \__fp_sep:
  }
\cs_new:Npn \__fp_sqrt_o:w #1 \s__fp \__fp_chk:w #2#3#4\__fp_sep: @
  {
    \if_meaning:w 0 #2 \__fp_case_return_same_o:w \fi:
    \if_meaning:w 2 #3
      \__fp_case_use:nw { \__fp_invalid_operation_o:nw { sqrt } }
    \fi:
    \if_meaning:w 1 #2 \else: \__fp_case_return_same_o:w \fi:
    \__fp_sqrt_npos_o:w
    \s__fp \__fp_chk:w #2 #3 #4\__fp_sep:
  }
\cs_new:Npn \__fp_sqrt_npos_o:w \s__fp \__fp_chk:w 1 0 #1#2#3#4#5\__fp_sep:
  {
    \exp_after:wN \__fp_sanitize:Nw
    \exp_after:wN 0
    \int_value:w \__fp_int_eval:w
      \if_int_odd:w #1 \exp_stop_f:
        \exp_after:wN \__fp_sqrt_npos_auxi_o:wwnnN
      \fi:
      #1 / 2
      \__fp_sqrt_Newton_o:wwn 56234133\__fp_sep: 0\__fp_sep: {#2#3} {#4#5} 0
  }
\cs_new:Npn \__fp_sqrt_npos_auxi_o:wwnnN #1 / 2 #2\__fp_sep: 0\__fp_sep: #3#4#5
  {
    ( #1 + 1 ) / 2
    \__fp_pack_eight:wNNNNNNNN
    \__fp_sqrt_npos_auxii_o:wNNNNNNNN
    \__fp_sep:
    0 #3 #4
  }
\cs_new:Npn \__fp_sqrt_npos_auxii_o:wNNNNNNNN #1\__fp_sep: #2#3#4#5#6#7#8#9
  { \__fp_sqrt_Newton_o:wwn 17782794\__fp_sep: 0\__fp_sep: {#1} {#2#3#4#5#6#7#8#9} }
```

```
\cs_new:Npn \__fp_sqrt_Newton_o:wwn #1\__fp_sep: #2\__fp_sep: #3
  {
    \if_int_compare:w #1 = #2 \exp_stop_f:
      \exp_after:wN \__fp_sqrt_auxi_o:NNNNwnnN
      \int_value:w \__fp_int_eval:w 9999 9999 +
        \exp_after:wN \__fp_use_none_until_s:w
    \fi:
    \exp_after:wN \__fp_sqrt_Newton_o:wwn
    \int_value:w \__fp_int_eval:w (#1 + #3 * 1 0000 0000 / #1) / 2 \__fp_sep:
    #1\__fp_sep: {#3}
  }
\cs_new:Npn \__fp_sqrt_auxi_o:NNNNwnnN 1 #1#2#3#4#5\__fp_sep:
  {
    \__fp_sqrt_auxii_o:NnnnnnnnN
      \__fp_sqrt_auxiii_o:wnnnnnnnn
      {#1#2#3#4} {#5} {2499} {9988} {7500}
  }
\cs_new:Npn \__fp_sqrt_auxii_o:NnnnnnnnN #1 #2#3#4#5#6 #7#8#9
  {
    \exp_after:wN #1
    \int_value:w \__fp_int_eval:w \c__fp_big_leading_shift_int
      + #7 - #2 * #2
      \exp_after:wN \__fp_pack_big:NNNNNNw
      \int_value:w \__fp_int_eval:w \c__fp_big_middle_shift_int
        - 2 * #2 * #3
        \exp_after:wN \__fp_pack_big:NNNNNNw
        \int_value:w \__fp_int_eval:w \c__fp_big_middle_shift_int
          + #8 - #3 * #3 - 2 * #2 * #4
          \exp_after:wN \__fp_pack_big:NNNNNNw
          \int_value:w \__fp_int_eval:w \c__fp_big_middle_shift_int
            - 2 * #3 * #4 - 2 * #2 * #5
            \exp_after:wN \__fp_pack_big:NNNNNNw
            \int_value:w \__fp_int_eval:w \c__fp_big_middle_shift_int
              + #9 000 0000 - #4 * #4 - 2 * #3 * #5 - 2 * #2 * #6
              \exp_after:wN \__fp_pack_big:NNNNNNw
              \int_value:w \__fp_int_eval:w \c__fp_big_middle_shift_int
                - 2 * #4 * #5 - 2 * #3 * #6
                \exp_after:wN \__fp_pack_big:NNNNNNw
                \int_value:w \__fp_int_eval:w \c__fp_big_middle_shift_int
                  - #5 * #5 - 2 * #4 * #6
                  \exp_after:wN \__fp_pack_big:NNNNNNw
                  \int_value:w \__fp_int_eval:w
                    \c__fp_big_middle_shift_int
                    - 2 * #5 * #6
                    \exp_after:wN \__fp_pack_big:NNNNNNw
                    \int_value:w \__fp_int_eval:w
```

```
           \c__fp_big_trailing_shift_int
           - #6 * #6 \__fp_sep:
  % (
  - 257 ) * 5000 0000 / (#2#3 + 1) + 10 0000 0000 \__fp_sep:
  {#2}{#3}{#4}{#5}{#6} {#7}{#8}#9
 }
\cs_new:Npn \__fp_sqrt_auxiii_o:wnnnnnnnn
  #1\__fp_sep: #2#3#4#5#6#7#8#9
 {
  \if_int_compare:w #1 > \c_one_int
   \exp_after:wN \__fp_sqrt_auxiv_o:NNNNNw
   \int_value:w \__fp_int_eval:w (#1#2 %)
  \else:
   \if_int_compare:w #1#2 > \c_one_int
    \exp_after:wN \__fp_sqrt_auxv_o:NNNNNw
    \int_value:w \__fp_int_eval:w (#1#2#3 %)
   \else:
    \if_int_compare:w #1#2#3 > \c_one_int
     \exp_after:wN \__fp_sqrt_auxvi_o:NNNNNw
     \int_value:w \__fp_int_eval:w (#1#2#3#4 %)
    \else:
     \exp_after:wN \__fp_sqrt_auxvii_o:NNNNNw
     \int_value:w \__fp_int_eval:w (#1#2#3#4#5 %)
    \fi:
   \fi:
  \fi:
 }
\cs_new:Npn \__fp_sqrt_auxiv_o:NNNNNw 1#1#2#3#4#5#6\__fp_sep:
 { \__fp_sqrt_auxviii_o:nnnnnnn {#1#2#3#4#5#6} {00000000} }
\cs_new:Npn \__fp_sqrt_auxv_o:NNNNNw 1#1#2#3#4#5#6\__fp_sep:
 { \__fp_sqrt_auxviii_o:nnnnnnn {000#1#2#3#4#5} {#60000} }
\cs_new:Npn \__fp_sqrt_auxvi_o:NNNNNw 1#1#2#3#4#5#6\__fp_sep:
 { \__fp_sqrt_auxviii_o:nnnnnnn {0000000#1} {#2#3#4#5#6} }
\cs_new:Npn \__fp_sqrt_auxvii_o:NNNNNw 1#1#2#3#4#5#6\__fp_sep:
 {
  \if_int_compare:w #1#2 = \c_zero_int
   \exp_after:wN \__fp_sqrt_auxx_o:Nnnnnnnn
  \fi:
  \__fp_sqrt_auxviii_o:nnnnnnn {00000000} {000#1#2#3#4#5}
 }
\cs_new:Npn \__fp_sqrt_auxviii_o:nnnnnnn #1#2 #3#4#5#6#7
 {
  \exp_after:wN \__fp_sqrt_auxix_o:wnwnw
  \int_value:w \__fp_int_eval:w #3
   \exp_after:wN \__fp_basics_pack_low:NNNNNw
   \int_value:w \__fp_int_eval:w #1 + 1#4#5
```

```
        \exp_after:wN \__fp_basics_pack_low:NNNNNw
        \int_value:w \__fp_int_eval:w #2 + 1#6#7 \__fp_sep:
  }
\cs_new:Npn \__fp_sqrt_auxix_o:wnwnw #1\__fp_sep: #2#3\__fp_sep: #4#5\__fp_sep:
  {
    \__fp_sqrt_auxii_o:NnnnnnnnN
      \__fp_sqrt_auxiii_o:wnnnnnnnn {#1}{#2}{#3}{#4}{#5}
  }
\cs_new:Npn \__fp_sqrt_auxx_o:Nnnnnnnn #1#2#3 #4#5#6#7#8
  {
    \exp_after:wN \__fp_sqrt_auxxi_o:wwnnN
    \int_value:w \__fp_int_eval:w
      (#8 + 2499) / 5000 * 5000 \__fp_sep:
      {#4} {#5} {#6} {#7} \__fp_sep:
  }
\cs_new:Npn \__fp_sqrt_auxxi_o:wwnnN #1\__fp_sep: #2\__fp_sep: #3#4#5
  {
    \__fp_sqrt_auxii_o:NnnnnnnnN
      \__fp_sqrt_auxxii_o:nnnnnnnnw
      #2 {#1}
      {#3} { #4 + 1 } #5
  }
\cs_new:Npn \__fp_sqrt_auxxii_o:nnnnnnnnw 0\__fp_sep: #1#2#3#4#5#6#7#8 #9\__fp_sep:
  {
    \if_int_compare:w #1#2 > \c_zero_int
      \if_int_compare:w #1#2 = \c_one_int
        \if_int_compare:w #3#4 = \c_zero_int
          \if_int_compare:w #5#6 = \c_zero_int
            \if_int_compare:w #7#8 = \c_zero_int
              \__fp_sqrt_auxxiii_o:w
            \fi:
          \fi:
        \fi:
      \fi:
      \exp_after:wN \__fp_sqrt_auxxiv_o:wnnnnnnnN
      \int_value:w 9998
    \else:
      \exp_after:wN \__fp_sqrt_auxxiv_o:wnnnnnnnN
      \int_value:w 10000
    \fi:
    \__fp_sep:
  }
\cs_new:Npn \__fp_sqrt_auxxiii_o:w \fi: \fi: \fi: \fi: #1 \fi: \__fp_sep:
  {
    \fi: \fi: \fi: \fi: \fi:
    \__fp_sqrt_auxxiv_o:wnnnnnnnN 9999 \__fp_sep:
  }
\cs_new:Npn \__fp_sqrt_auxxiv_o:wnnnnnnnN #1\__fp_sep: #2#3#4#5#6 #7#8#9
  {
    \exp_after:wN \__fp_basics_pack_high:NNNNNw
    \int_value:w \__fp_int_eval:w 1 0000 0000 + #2#3
      \exp_after:wN \__fp_basics_pack_low:NNNNNw
      \int_value:w \__fp_int_eval:w 1 0000 0000
        + #4#5
        \if_int_compare:w #6 > #1 \exp_stop_f: + 1 \fi:
        + \exp_after:wN \__fp_round:NNN
          \exp_after:wN 0
          \exp_after:wN 0
          \int_value:w
            \exp_after:wN \use_i:nn
            \exp_after:wN \__fp_round_digit:Nw
            \int_value:w \__fp_int_eval:w #6 + 19999 - #1 \__fp_sep:
    \exp_after:wN \__fp_sep:
  }
\cs_new:Npn \__fp_logb_o:w ? \s__fp \__fp_chk:w #1#2\__fp_sep: @
  {
    \if_case:w #1 \exp_stop_f:
           \__fp_case_use:nw
             { \__fp_division_by_zero_o:Nnw \c_minus_inf_fp { logb } }
    \or:   \exp_after:wN \__fp_logb_aux_o:w
    \or:   \__fp_case_return_o:Nw \c_inf_fp
    \else: \__fp_case_return_same_o:w
    \fi:
    \s__fp \__fp_chk:w #1 #2\__fp_sep:
  }
\cs_new:Npn \__fp_logb_aux_o:w \s__fp \__fp_chk:w #1 #2 #3 #4 \__fp_sep:
  {
    \exp_after:wN \__fp_parse:n \exp_after:wN
      { \int_value:w \int_eval:w #3 - 1 \exp_after:wN }
  }
\cs_new:Npn \__fp_sign_o:w ? \s__fp \__fp_chk:w #1#2\__fp_sep: @
  {
    \if_case:w #1 \exp_stop_f:
           \__fp_case_return_same_o:w
    \or:   \exp_after:wN \__fp_sign_aux_o:w
    \or:   \exp_after:wN \__fp_sign_aux_o:w
    \else: \__fp_case_return_same_o:w
    \fi:
    \s__fp \__fp_chk:w #1 #2\__fp_sep:
  }
\cs_new:Npn \__fp_sign_aux_o:w \s__fp \__fp_chk:w #1 #2 #3 \__fp_sep:
  { \exp_after:wN \__fp_set_sign_o:w \exp_after:wN #2 \c_one_fp @ }
```

```
\cs_new:Npn \__fp_set_sign_o:w #1 \s__fp \__fp_chk:w #2#3#4\__fp_sep: @
  {
    \exp_after:wN \__fp_exp_after_o:w
    \exp_after:wN \s__fp
    \exp_after:wN \__fp_chk:w
    \exp_after:wN #2
    \int_value:w
      \if_case:w #3 \exp_stop_f: #1 \or: 1 \or: 0 \fi: \exp_stop_f:
    #4\__fp_sep:
  }
\cs_new:Npn \__fp_tuple_set_sign_o:w #1#2 @
  {
    \if_meaning:w 2 #1
      \exp_after:wN \__fp_tuple_set_sign_aux_o:Nnw
    \fi:
    \__fp_invalid_operation_o:nw { abs }
    #2
  }
\cs_new:Npn \__fp_tuple_set_sign_aux_o:Nnw #1#2
  { \__fp_tuple_map_o:nw \__fp_tuple_set_sign_aux_o:w }
\cs_new:Npn \__fp_tuple_set_sign_aux_o:w #1#2 \__fp_sep:
  {
    \__fp_change_func_type:NNN #1 \__fp_set_sign_o:w
      \__fp_parse_apply_unary_error:NNw
    2 #1 #2 \__fp_sep: @
  }
\cs_new:cpn { __fp_*_tuple_o:ww } #1 \__fp_sep:
  { \__fp_tuple_map_o:nw { \__fp_binary_type_o:Nww * #1 \__fp_sep: } }
\cs_new:cpn { __fp_tuple_*_o:ww } #1 \__fp_sep: #2 \__fp_sep:
  {
    \__fp_tuple_map_o:nw { \__fp_binary_rev_type_o:Nww * #2 \__fp_sep: }
      #1 \__fp_sep:
  }
\cs_new:cpn { __fp_tuple_/_o:ww } #1 \__fp_sep: #2 \__fp_sep:
  {
    \__fp_tuple_map_o:nw { \__fp_binary_rev_type_o:Nww / #2 \__fp_sep: }
      #1 \__fp_sep:
  }
\cs_set_protected:Npn \__fp_tmp:w #1
  {
    \cs_new:cpn { __fp_tuple_#1_tuple_o:ww }
        \s__fp_tuple \__fp_tuple_chk:w ##1 \__fp_sep:
        \s__fp_tuple \__fp_tuple_chk:w ##2 \__fp_sep:
      {
        \int_compare:nNnTF
          { \__fp_array_count:n {##1} } = { \__fp_array_count:n {##2} }
          { \__fp_tuple_mapthread_o:nww { \__fp_binary_type_o:Nww #1 } }
          { \__fp_invalid_operation_o:nww #1 }
        \s__fp_tuple \__fp_tuple_chk:w {##1} \__fp_sep:
        \s__fp_tuple \__fp_tuple_chk:w {##2} \__fp_sep:
      }
  }
\__fp_tmp:w +
\__fp_tmp:w -
%% File: l3fp-extended.dtx
\tl_const:Nn \c__fp_one_fixed_tl
  { {10000} {0000} {0000} {0000} {0000} {0000} \__fp_sep: }
\cs_new:Npn \__fp_fixed_continue:wn #1\__fp_sep: #2 { #2 #1\__fp_sep: }
\cs_new:Npn \__fp_fixed_add_one:wN #1#2\__fp_sep: #3
  {
    \exp_after:wN #3 \exp_after:wN
      { \int_value:w \__fp_int_eval:w \c__fp_myriad_int + #1 } #2 \__fp_sep:
  }
\cs_new:Npn \__fp_fixed_div_myriad:wn #1#2#3#4#5#6\__fp_sep:
  {
    \exp_after:wN \__fp_fixed_mul_after:wwn
    \int_value:w \__fp_int_eval:w \c__fp_leading_shift_int
      \exp_after:wN \__fp_pack:NNNNNw
      \int_value:w \__fp_int_eval:w \c__fp_trailing_shift_int
        + #1 \__fp_sep: {#2}{#3}{#4}{#5}\__fp_sep:
  }
\cs_new:Npn \__fp_fixed_mul_after:wwn #1\__fp_sep: #2\__fp_sep: #3
  { #3 {#1} #2\__fp_sep: }
\cs_new:Npn \__fp_fixed_mul_short:wwn #1#2#3#4#5#6\__fp_sep: #7#8#9\__fp_sep:
  {
    \exp_after:wN \__fp_fixed_mul_after:wwn
    \int_value:w \__fp_int_eval:w \c__fp_leading_shift_int
      + #1*#7
      \exp_after:wN \__fp_pack:NNNNNw
      \int_value:w \__fp_int_eval:w \c__fp_middle_shift_int
        + #1*#8 + #2*#7
        \exp_after:wN \__fp_pack:NNNNNw
        \int_value:w \__fp_int_eval:w \c__fp_middle_shift_int
          + #1*#9 + #2*#8 + #3*#7
          \exp_after:wN \__fp_pack:NNNNNw
          \int_value:w \__fp_int_eval:w \c__fp_middle_shift_int
            + #2*#9 + #3*#8 + #4*#7
            \exp_after:wN \__fp_pack:NNNNNw
            \int_value:w \__fp_int_eval:w \c__fp_middle_shift_int
              + #3*#9 + #4*#8 + #5*#7
              \exp_after:wN \__fp_pack:NNNNNw
              \int_value:w \__fp_int_eval:w \c__fp_trailing_shift_int
```

```
                + #4*#9 + #5*#8 + #6*#7
                + ( #5*#9 + #6*#8 + #6*#9 / \c__fp_myriad_int )
                / \c__fp_myriad_int \__fp_sep: \__fp_sep:
  }
\cs_new:Npn \__fp_fixed_div_int:wwN #1#2#3#4#5#6 \__fp_sep: #7 \__fp_sep: #8
  {
    \exp_after:wN \__fp_fixed_div_int_after:Nw
    \exp_after:wN #8
    \int_value:w \__fp_int_eval:w - 1
      \__fp_fixed_div_int:wnN
      #1\__fp_sep: {#7} \__fp_fixed_div_int_auxi:wnn
      #2\__fp_sep: {#7} \__fp_fixed_div_int_auxi:wnn
      #3\__fp_sep: {#7} \__fp_fixed_div_int_auxi:wnn
      #4\__fp_sep: {#7} \__fp_fixed_div_int_auxi:wnn
      #5\__fp_sep: {#7} \__fp_fixed_div_int_auxi:wnn
      #6\__fp_sep: {#7} \__fp_fixed_div_int_auxii:wnn \__fp_sep:
  }
\cs_new:Npn \__fp_fixed_div_int:wnN #1\__fp_sep: #2 #3
  {
    \exp_after:wN #3
    \int_value:w \__fp_int_eval:w #1 / #2 - 1 \__fp_sep:
    {#2}
    {#1}
  }
\cs_new:Npn \__fp_fixed_div_int_auxi:wnn #1\__fp_sep: #2 #3
  {
    + #1
    \exp_after:wN \__fp_fixed_div_int_pack:Nw
    \int_value:w \__fp_int_eval:w 9999
      \exp_after:wN \__fp_fixed_div_int:wnN
      \int_value:w \__fp_int_eval:w #3 - #1*#2 \__fp_int_eval_end:
  }
\cs_new:Npn \__fp_fixed_div_int_auxii:wnn #1\__fp_sep: #2 #3 { + #1 + 2 \__fp_sep: }
\cs_new:Npn \__fp_fixed_div_int_pack:Nw #1 #2\__fp_sep: { + #1\__fp_sep: {#2} }
\cs_new:Npn \__fp_fixed_div_int_after:Nw #1 #2\__fp_sep: { #1 {#2} }
\cs_new:Npn \__fp_fixed_add:wwn { \__fp_fixed_add:Nnnnnwnn + }
\cs_new:Npn \__fp_fixed_sub:wwn { \__fp_fixed_add:Nnnnnwnn - }
\cs_new:Npn \__fp_fixed_add:Nnnnnwnn #1 #2#3#4#5 #6\__fp_sep: #7#8
  {
    \exp_after:wN \__fp_fixed_add_after:NNNNNwn
    \int_value:w \__fp_int_eval:w 9 9999 9998 + #2#3 #1 #7#8
      \exp_after:wN \__fp_fixed_add_pack:NNNNNwn
      \int_value:w \__fp_int_eval:w 1 9999 9998 + #4#5
        \__fp_fixed_add:nnNnnnwn #6 #1
  }
\cs_new:Npn \__fp_fixed_add:nnNnnnwn #1#2 #3 #4#5 #6#7 \__fp_sep: #8
```

```
  {
    #3 #4#5
    \exp_after:wN \__fp_fixed_add_pack:NNNNNwn
    \int_value:w \__fp_int_eval:w
      2 0000 0000 #3 #6#7 + #1#2 \__fp_sep: {#8} \__fp_sep:
  }
\cs_new:Npn \__fp_fixed_add_pack:NNNNNwn #1 #2#3#4#5 #6\__fp_sep: #7
  { + #1 \__fp_sep: {#7} {#2#3#4#5} {#6} }
\cs_new:Npn \__fp_fixed_add_after:NNNNNwn 1 #1 #2#3#4#5 #6\__fp_sep: #7
  { #7 {#1#2#3#4#5} {#6} }
\cs_new:Npn \__fp_fixed_mul:wwn #1#2#3#4 #5\__fp_sep: #6#7#8#9
  {
    \exp_after:wN \__fp_fixed_mul_after:wwn
    \int_value:w \__fp_int_eval:w \c__fp_leading_shift_int
      \exp_after:wN \__fp_pack:NNNNNw
      \int_value:w \__fp_int_eval:w \c__fp_middle_shift_int
        + #1*#6
        \exp_after:wN \__fp_pack:NNNNNw
        \int_value:w \__fp_int_eval:w \c__fp_middle_shift_int
          + #1*#7 + #2*#6
          \exp_after:wN \__fp_pack:NNNNNw
          \int_value:w \__fp_int_eval:w \c__fp_middle_shift_int
            + #1*#8 + #2*#7 + #3*#6
            \exp_after:wN \__fp_pack:NNNNNw
            \int_value:w \__fp_int_eval:w \c__fp_middle_shift_int
              + #1*#9 + #2*#8 + #3*#7 + #4*#6
              \exp_after:wN \__fp_pack:NNNNNw
              \int_value:w \__fp_int_eval:w \c__fp_trailing_shift_int
                + #2*#9 + #3*#8 + #4*#7
                + ( #3*#9 + #4*#8
                  + \__fp_fixed_mul:nnnnnnnw #5 {#6}{#7}  {#1}{#2}
  }
\cs_new:Npn \__fp_fixed_mul:nnnnnnnw #1#2 #3#4 #5#6 #7#8 \__fp_sep:
  {
    #1*#4 + #2*#3 + #5*#8 + #6*#7 ) / \c__fp_myriad_int
    + #1*#3 + #5*#7 \__fp_sep: \__fp_sep:
  }
\cs_new:Npn \__fp_fixed_mul_add:wwwn #1\__fp_sep: #2\__fp_sep: #3#4#5#6#7#8\__fp_sep:
  {
    \exp_after:wN \__fp_fixed_mul_after:wwn
    \int_value:w \__fp_int_eval:w \c__fp_big_leading_shift_int
      \exp_after:wN \__fp_pack_big:NNNNNNw
      \int_value:w \__fp_int_eval:w \c__fp_big_middle_shift_int + #3 #4
        \__fp_fixed_mul_add:Nwnnnwnnn +
          + #5 #6 \__fp_sep: #2 \__fp_sep: #1 \__fp_sep: #2 \__fp_sep: +
          + #7 #8 \__fp_sep: \__fp_sep:
  }
\cs_new:Npn \__fp_fixed_mul_sub_back:wwwn
    #1\__fp_sep: #2\__fp_sep: #3#4#5#6#7#8\__fp_sep:
  {
    \exp_after:wN \__fp_fixed_mul_after:wwn
    \int_value:w \__fp_int_eval:w \c__fp_big_leading_shift_int
      \exp_after:wN \__fp_pack_big:NNNNNNw
      \int_value:w \__fp_int_eval:w \c__fp_big_middle_shift_int + #3 #4
        \__fp_fixed_mul_add:Nwnnnwnnn -
          + #5 #6 \__fp_sep: #2 \__fp_sep: #1 \__fp_sep: #2 \__fp_sep: -
          + #7 #8 \__fp_sep: \__fp_sep:
  }
\cs_new:Npn \__fp_fixed_one_minus_mul:wwn #1\__fp_sep: #2\__fp_sep:
  {
    \exp_after:wN \__fp_fixed_mul_after:wwn
    \int_value:w \__fp_int_eval:w \c__fp_big_leading_shift_int
      \exp_after:wN \__fp_pack_big:NNNNNNw
      \int_value:w \__fp_int_eval:w \c__fp_big_middle_shift_int +
        1 0000 0000
        \__fp_fixed_mul_add:Nwnnnwnnn -
          \__fp_sep: #2 \__fp_sep: #1 \__fp_sep: #2 \__fp_sep: -
          \__fp_sep: \__fp_sep:
  }
\cs_new:Npn \__fp_fixed_mul_add:Nwnnnwnnn #1 #2\__fp_sep: #3#4#5#6\__fp_sep: #7#8#9
  {
    #1 #7*#3
    \exp_after:wN \__fp_pack_big:NNNNNNw
    \int_value:w \__fp_int_eval:w \c__fp_big_middle_shift_int
      #1 #7*#4 #1 #8*#3
      \exp_after:wN \__fp_pack_big:NNNNNNw
      \int_value:w \__fp_int_eval:w \c__fp_big_middle_shift_int
        #1 #7*#5 #1 #8*#4 #1 #9*#3 #2
        \exp_after:wN \__fp_pack_big:NNNNNNw
        \int_value:w \__fp_int_eval:w \c__fp_big_middle_shift_int
          #1 \__fp_fixed_mul_add:nnnnwnnnn {#7}{#8}{#9}
  }
\cs_new:Npn \__fp_fixed_mul_add:nnnnwnnnn #1#2#3#4#5\__fp_sep: #6#7#8#9
  {
    ( #1*#9 + #2*#8 + #3*#7 + #4*#6 )
    \exp_after:wN \__fp_pack_big:NNNNNNw
    \int_value:w \__fp_int_eval:w \c__fp_big_trailing_shift_int
      \__fp_fixed_mul_add:nnnnwnnwN
        { #6 + #4*#7 + #3*#8 + #2*#9 + #1 }
        { #7 + #4*#8 + #3*#9 + #2 }
        {#1} #5\__fp_sep:
        {#6}
  }
\cs_new:Npn \__fp_fixed_mul_add:nnnnwnnnwN #1#2 #3#4#5\__fp_sep: #6#7#8\__fp_sep: #9
  {
    #9 (#4* #1 *#7)
    #9 (#5*#6+#4* #2 *#7+#3*#8) / \c__fp_myriad_int
  }
\cs_new:Npn \__fp_ep_to_fixed:wwn #1,#2
  {
    \exp_after:wN \__fp_ep_to_fixed_auxi:www
    \int_value:w \__fp_int_eval:w 1 0000 0000 + #2 \exp_after:wN \__fp_sep:
    \exp:w \exp_end_continue_f:w
    \prg_replicate:nn { 4 - \int_max:nn {#1} { -32 } } { 0 } \__fp_sep:
  }
\cs_new:Npn \__fp_ep_to_fixed_auxi:www
    1#1\__fp_sep: #2\__fp_sep: #3#4#5#6#7\__fp_sep:
  {
    \__fp_pack_eight:wNNNNNNNN
    \__fp_pack_twice_four:wNNNNNNNN
    \__fp_pack_twice_four:wNNNNNNNN
    \__fp_pack_twice_four:wNNNNNNNN
    \__fp_ep_to_fixed_auxii:nnnnnnnwn \__fp_sep:
    #2 #1#3#4#5#6#7 0000 !
  }
\cs_new:Npn \__fp_ep_to_fixed_auxii:nnnnnnnwn #1#2#3#4#5#6#7\__fp_sep: #8! #9
  { #9 {#1#2}{#3}{#4}{#5}{#6}{#7}\__fp_sep: }
\cs_new:Npn \__fp_ep_to_ep:wwN #1,#2#3#4#5#6#7\__fp_sep: #8
  {
    \exp_after:wN #8
    \int_value:w \__fp_int_eval:w #1 + 4
      \exp_after:wN \use_i:nn
      \exp_after:wN \__fp_ep_to_ep_loop:N
      \int_value:w \__fp_int_eval:w 1 0000 0000 + #2 \__fp_int_eval_end:
      #3#4#5#6#7 \__fp_sep: \__fp_sep: !
  }
\cs_new:Npn \__fp_ep_to_ep_loop:N #1
  {
    \if_meaning:w 0 #1
      - 1
    \else:
      \__fp_ep_to_ep_end:www #1
    \fi:
    \__fp_ep_to_ep_loop:N
  }
\cs_new:Npn \__fp_ep_to_ep_end:www
    #1 \fi: \__fp_ep_to_ep_loop:N #2\__fp_sep: #3!
  {
    \fi:
    \if_meaning:w \__fp_sep: #1
      - 2 * \c__fp_max_exponent_int
      \__fp_ep_to_ep_zero:ww
    \fi:
    \__fp_pack_twice_four:wNNNNNNNN
    \__fp_pack_twice_four:wNNNNNNNN
    \__fp_pack_twice_four:wNNNNNNNN
    \__fp_use_i:ww , \__fp_sep:
    #1 #2 0000 0000 0000 0000 0000 0000 \__fp_sep:
  }
\cs_new:Npn \__fp_ep_to_ep_zero:ww \fi: #1\__fp_sep: #2\__fp_sep: #3\__fp_sep:
  { \fi: , {1000}{0000}{0000}{0000}{0000}{0000} \__fp_sep: }
\cs_new:Npn \__fp_ep_compare:wwww #1,#2#3#4#5#6#7\__fp_sep:
  { \__fp_ep_compare_aux:wwww {#1}{#2}{#3}{#4}{#5}\__fp_sep: #6#7\__fp_sep: }
\cs_new:Npn \__fp_ep_compare_aux:wwww
    #1\__fp_sep:#2\__fp_sep:#3,#4#5#6#7#8#9\__fp_sep:
  {
    \if_case:w
      \__fp_compare_npos:nwnw
        #1\__fp_sep: {#3}{#4}{#5}{#6}{#7}\__fp_sep: \exp_stop_f:
            \if_int_compare:w #2 = #8#9 \exp_stop_f:
              0
            \else:
              \if_int_compare:w #2 < #8#9 - \fi: 1
            \fi:
    \or:    1
    \else: -1
    \fi:
  }
\cs_new:Npn \__fp_ep_mul:wwwwn #1,#2\__fp_sep: #3,#4\__fp_sep:
  {
    \__fp_ep_to_ep:wwN #3,#4\__fp_sep:
    \__fp_fixed_continue:wn
    {
      \__fp_ep_to_ep:wwN #1,#2\__fp_sep:
      \__fp_ep_mul_raw:wwwwN
    }
    \__fp_fixed_continue:wn
  }
\cs_new:Npn \__fp_ep_mul_raw:wwwwN #1,#2\__fp_sep: #3,#4\__fp_sep: #5
  {
    \__fp_fixed_mul:wwn #2\__fp_sep: #4\__fp_sep:
    { \exp_after:wN #5 \int_value:w \__fp_int_eval:w #1 + #3 , }
  }
\cs_new:Npn \__fp_ep_div:wwwwn #1,#2\__fp_sep: #3,#4\__fp_sep:
  {
    \__fp_ep_to_ep:wwN #1,#2\__fp_sep:
    \__fp_fixed_continue:wn
    {
      \__fp_ep_to_ep:wwN #3,#4\__fp_sep:
      \__fp_ep_div_esti:wwwwn
    }
  }
\cs_new:Npn \__fp_ep_div_esti:wwwwn #1,#2#3\__fp_sep: #4,
  {
    \exp_after:wN \__fp_ep_div_estii:wwnnwwn
    \int_value:w \__fp_int_eval:w 10 0000 0000 / ( #2 + 1 )
      \exp_after:wN \__fp_sep:
    \int_value:w \__fp_int_eval:w #4 - #1 + 1 ,
    {#2} #3\__fp_sep:
  }
\cs_new:Npn \__fp_ep_div_estii:wwnnwwn
    #1\__fp_sep: #2,#3#4#5\__fp_sep: #6\__fp_sep: #7
  {
    \exp_after:wN \__fp_ep_div_estiii:NNNNNwwwn
    \int_value:w \__fp_int_eval:w 10 0000 0000 - 1750
      + #1 000 + (10 0000 0000 / #3 - #1) * (1000 - #4 / 10) \__fp_sep:
    {#3}{#4}#5\__fp_sep: #6\__fp_sep: { #7 #2, }
  }
\cs_new:Npn \__fp_ep_div_estiii:NNNNNwwwn 1#1#2#3#4#5#6\__fp_sep: #7\__fp_sep:
  {
    \__fp_fixed_mul_short:wwn #7\__fp_sep: {#1}{#2#3#4#5}{#6}\__fp_sep:
    \__fp_ep_div_epsi:wnNNNNNn {#1#2#3#4}#5#6
    \__fp_fixed_mul:wwn
  }
\cs_new:Npn \__fp_ep_div_epsi:wnNNNNNn #1#2#3#4#5#6\__fp_sep:
  {
    \exp_after:wN \__fp_ep_div_epsii:wwnNNNNNn
    \int_value:w \__fp_int_eval:w 1 9998 - #2
      \exp_after:wN \__fp_ep_div_eps_pack:NNNNNw
      \int_value:w \__fp_int_eval:w 1 9999 9998 - #3#4
        \exp_after:wN \__fp_ep_div_eps_pack:NNNNNw
        \int_value:w \__fp_int_eval:w 2 0000 0000 - #5#6 \__fp_sep: \__fp_sep:
  }
\cs_new:Npn \__fp_ep_div_eps_pack:NNNNNw #1#2#3#4#5#6\__fp_sep:
  { + #1 \__fp_sep: {#2#3#4#5} {#6} }
\cs_new:Npn \__fp_ep_div_epsii:wwnNNNNNn 1#1\__fp_sep: #2\__fp_sep: #3#4#5#6#7#8
  {
    \__fp_fixed_mul:wwn {0000}{#1}#2\__fp_sep: {0000}{#1}#2\__fp_sep:
    \__fp_fixed_add_one:wN
    \__fp_fixed_mul:wwn {10000} {#1} #2 \__fp_sep:
```

```
    {
      \__fp_fixed_mul_short:wwn
        {0000}{#1}#2\__fp_sep: {#3}{#4#5#6#7}{#8000}\__fp_sep:
      \__fp_fixed_div_myriad:wn
      \__fp_fixed_mul:wwn
    }
    \__fp_fixed_add:wwn {#3}{#4#5#6#7}{#8000}{0000}{0000}{0000}\__fp_sep:
  }
\cs_new:Npn \__fp_ep_isqrt:wwn #1,#2\__fp_sep:
  {
    \__fp_ep_to_ep:wwN #1,#2\__fp_sep:
    \__fp_ep_isqrt_auxi:wwn
  }
\cs_new:Npn \__fp_ep_isqrt_auxi:wwn #1,
  {
    \exp_after:wN \__fp_ep_isqrt_auxii:wwnnnwn
    \int_value:w \__fp_int_eval:w
      \int_if_odd:nTF {#1}
        { (1 - #1) / 2 , 535 , { 0 } { } }
        { 1 - #1 / 2 , 168 , { } { 0 } }
  }
\cs_new:Npn \__fp_ep_isqrt_auxii:wwnnnwn #1, #2, #3#4 #5#6\__fp_sep: #7
  {
    \__fp_ep_isqrt_esti:wwwnnwn #2, 0, #5, {#3} {#4}
      {#5} #6 \__fp_sep: { #7 #1 , }
  }
\cs_new:Npn \__fp_ep_isqrt_esti:wwwnnwn #1, #2, #3, #4
  {
    \if_int_compare:w #1 = #2 \exp_stop_f:
      \exp_after:wN \__fp_ep_isqrt_estii:wwwnnwn
    \fi:
    \exp_after:wN \__fp_ep_isqrt_esti:wwwnnwn
    \int_value:w \__fp_int_eval:w
      (#1 + 1 0050 0000 #4 / (#1 * #3)) / 2 ,
    #1, #3, {#4}
  }
\cs_new:Npn \__fp_ep_isqrt_estii:wwwnnwn #1, #2, #3, #4#5
  {
    \exp_after:wN \__fp_ep_isqrt_estiii:NNNNNwwwn
    \int_value:w \__fp_int_eval:w 1000 0000 + #2 * #2 #5 * 5
      \exp_after:wN , \int_value:w \__fp_int_eval:w 10000 + #2 \__fp_sep:
  }
\cs_new:Npn \__fp_ep_isqrt_estiii:NNNNNwwwn
    1#1#2#3#4#5#6, 1#7#8\__fp_sep: #9\__fp_sep:
  {
    \__fp_fixed_mul_short:wwn #9\__fp_sep: {#1} {#2#3#4#5} {#600} \__fp_sep:
```

```
    \__fp_ep_isqrt_epsi:wN
    \__fp_fixed_mul_short:wwn {#7} {#80} {0000} \__fp_sep:
  }
\cs_new:Npn \__fp_ep_isqrt_epsi:wN #1\__fp_sep:
  {
    \__fp_fixed_sub:wwn {15000}{0000}{0000}{0000}{0000}{0000}\__fp_sep: #1\__fp_sep:
    \__fp_ep_isqrt_epsii:wwN #1\__fp_sep:
    \__fp_ep_isqrt_epsii:wwN #1\__fp_sep:
    \__fp_ep_isqrt_epsii:wwN #1\__fp_sep:
  }
\cs_new:Npn \__fp_ep_isqrt_epsii:wwN #1\__fp_sep: #2\__fp_sep:
  {
    \__fp_fixed_mul:wwn #1\__fp_sep: #1\__fp_sep:
    \__fp_fixed_mul_sub_back:wwwn #2\__fp_sep:
      {15000}{0000}{0000}{0000}{0000}{0000}\__fp_sep:
    \__fp_fixed_mul:wwn #1\__fp_sep:
  }
\cs_new:Npn \__fp_ep_to_float_o:wwN #1,
  { + \__fp_int_eval:w #1 \__fp_fixed_to_float_o:wN }
\cs_new:Npn \__fp_ep_inv_to_float_o:wwN #1,#2\__fp_sep:
  {
    \__fp_ep_div:wwwwn
      1,{1000}{0000}{0000}{0000}{0000}{0000}\__fp_sep: #1,#2\__fp_sep:
    \__fp_ep_to_float_o:wwN
  }
\cs_new:Npn \__fp_fixed_inv_to_float_o:wN
  { \__fp_ep_inv_to_float_o:wwN 0, }
\cs_new:Npn \__fp_fixed_to_float_rad_o:wN #1\__fp_sep:
  {
    \__fp_fixed_mul:wwn #1\__fp_sep: {5729}{5779}{5130}{8232}{0876}{7981}\__fp_sep:
    { \__fp_ep_to_float_o:wwN 2, }
  }
\cs_new:Npn \__fp_fixed_to_float_o:Nw #1#2\__fp_sep:
  { \__fp_fixed_to_float_o:wN #2\__fp_sep: #1 }
\cs_new:Npn \__fp_fixed_to_float_o:wN #1#2#3#4#5#6\__fp_sep: #7
  { % for the 8-digit-at-the-start thing
    + \__fp_int_eval:w \c__fp_block_int
    \exp_after:wN \exp_after:wN
    \exp_after:wN \__fp_fixed_to_loop:N
    \exp_after:wN \use_none:n
    \int_value:w \__fp_int_eval:w
      1 0000 0000 + #1   \exp_after:wN \__fp_use_none_stop_f:n
      \int_value:w   1#2 \exp_after:wN \__fp_use_none_stop_f:n
      \int_value:w 1#3#4 \exp_after:wN \__fp_use_none_stop_f:n
      \int_value:w 1#5#6
    \exp_after:wN \__fp_sep:
```

```
    \exp_after:wN \__fp_sep:
  }
\cs_new:Npn \__fp_fixed_to_loop:N #1
  {
    \if_meaning:w 0 #1
      - 1
      \exp_after:wN \__fp_fixed_to_loop:N
    \else:
      \exp_after:wN \__fp_fixed_to_loop_end:w
      \exp_after:wN #1
    \fi:
  }
\cs_new:Npn \__fp_fixed_to_loop_end:w #1 #2 \__fp_sep:
  {
    \if_meaning:w \__fp_sep: #1
      \exp_after:wN \__fp_fixed_to_float_zero:w
    \else:
      \exp_after:wN \__fp_pack_twice_four:wNNNNNNNN
      \exp_after:wN \__fp_pack_twice_four:wNNNNNNNN
      \exp_after:wN \__fp_fixed_to_float_pack:ww
      \exp_after:wN \__fp_sep:
    \fi:
    #1 #2 0000 0000 0000 0000 \__fp_sep:
  }
\cs_new:Npn \__fp_fixed_to_float_zero:w \__fp_sep: 0000 0000 0000 0000 \__fp_sep:
  {
    - 2 * \c__fp_max_exponent_int \__fp_sep:
    {0000} {0000} {0000} {0000} \__fp_sep:
  }
\cs_new:Npn \__fp_fixed_to_float_pack:ww #1 \__fp_sep: #2#3 \__fp_sep: \__fp_sep:
  {
    \if_int_compare:w #2 > 4 \exp_stop_f:
      \exp_after:wN \__fp_fixed_to_float_round_up:wnnnnw
    \fi:
    \__fp_sep: #1 \__fp_sep:
  }
\cs_new:Npn \__fp_fixed_to_float_round_up:wnnnnw \__fp_sep: #1#2#3#4 \__fp_sep:
  {
    \exp_after:wN \__fp_basics_pack_high:NNNNNw
    \int_value:w \__fp_int_eval:w 1 #1#2
      \exp_after:wN \__fp_basics_pack_low:NNNNNw
      \int_value:w \__fp_int_eval:w 1 #3#4 + 1 \__fp_sep:
  }
%% File: l3fp-expo.dtx
\cs_new:Npn \__fp_parse_word_exp:N
  { \__fp_parse_unary_function:NNN \__fp_exp_o:w ? }
```

```
\cs_new:Npn \__fp_parse_word_ln:N
  { \__fp_parse_unary_function:NNN \__fp_ln_o:w ? }
\cs_new:Npn \__fp_parse_word_fact:N
  { \__fp_parse_unary_function:NNN \__fp_fact_o:w ? }
\tl_const:Nn \c__fp_ln_i_fixed_tl
  { {0000}{0000}{0000}{0000}{0000}{0000}\__fp_sep:}
\tl_const:Nn \c__fp_ln_ii_fixed_tl
  { {6931}{4718}{0559}{9453}{0941}{7232}\__fp_sep:}
\tl_const:Nn \c__fp_ln_iii_fixed_tl
  {{10986}{1228}{8668}{1096}{9139}{5245}\__fp_sep:}
\tl_const:Nn \c__fp_ln_iv_fixed_tl
  {{13862}{9436}{1119}{8906}{1883}{4464}\__fp_sep:}
\tl_const:Nn \c__fp_ln_vi_fixed_tl
  {{17917}{5946}{9228}{0550}{0081}{2477}\__fp_sep:}
\tl_const:Nn \c__fp_ln_vii_fixed_tl
  {{19459}{1014}{9055}{3133}{0510}{5353}\__fp_sep:}
\tl_const:Nn \c__fp_ln_viii_fixed_tl
  {{20794}{4154}{1679}{8359}{2825}{1696}\__fp_sep:}
\tl_const:Nn \c__fp_ln_ix_fixed_tl
  {{21972}{2457}{7336}{2193}{8279}{0490}\__fp_sep:}
\tl_const:Nn \c__fp_ln_x_fixed_tl
  {{23025}{8509}{2994}{0456}{8401}{7991}\__fp_sep:}
\cs_new:Npn \__fp_ln_o:w #1 \s__fp \__fp_chk:w #2#3#4\__fp_sep: @
  {
    \if_meaning:w 2 #3
      \__fp_case_use:nw { \__fp_invalid_operation_o:nw { ln } }
    \fi:
    \if_case:w #2 \exp_stop_f:
      \__fp_case_use:nw
        { \__fp_division_by_zero_o:Nnw \c_minus_inf_fp { ln } }
    \or:
    \else:
      \__fp_case_return_same_o:w
    \fi:
    \__fp_ln_npos_o:w \s__fp \__fp_chk:w #2#3#4\__fp_sep:
  }
\cs_new:Npn \__fp_ln_npos_o:w \s__fp \__fp_chk:w 10#1#2#3\__fp_sep:
  { %^^A todo: ln(1) should be "exact zero", not "underflow"
    \exp_after:wN \__fp_sanitize:Nw
    \int_value:w % for the overall sign
      \if_int_compare:w #1 < \c_one_int
        2
      \else:
        0
      \fi:
      \exp_after:wN \exp_stop_f:
```

```
      \int_value:w \__fp_int_eval:w % for the exponent
        \__fp_ln_significand:NNNNnnnN #2#3
        \__fp_ln_exponent:wn {#1}
  }
\cs_new:Npn \__fp_ln_significand:NNNNnnnN #1#2#3#4
  {
    \exp_after:wN \__fp_ln_x_ii:wnnnn
    \int_value:w
      \if_case:w #1 \exp_stop_f:
      \or:
        \if_int_compare:w #2 < 4 \exp_stop_f:
          \__fp_int_eval:w 10 - #2
        \else:
          6
        \fi:
      \or: 4
      \or: 3
      \or: 2
      \or: 2
      \or: 2
      \else: 1
      \fi:
    \__fp_sep: { #1 #2 #3 #4 }
  }
\cs_new:Npn \__fp_ln_x_ii:wnnnn #1\__fp_sep: #2#3#4#5
  {
    \exp_after:wN \__fp_ln_div_after:Nw
    \cs:w c__fp_ln_ \__fp_int_to_roman:w #1 _fixed_tl \exp_after:wN \cs_end:
    \int_value:w
      \exp_after:wN \__fp_ln_x_iv:wnnnnnnnn
      \int_value:w \__fp_int_eval:w
        \exp_after:wN \__fp_ln_x_iii_var:NNNNNw
        \int_value:w \__fp_int_eval:w 9999 9990 + #1*#2#3 +
          \exp_after:wN \__fp_ln_x_iii:NNNNNNw
          \int_value:w \__fp_int_eval:w 10 0000 0000 + #1*#4#5 \__fp_sep:
    {20000} {0000} {0000} {0000}
  } %^^A todo: reoptimize (a generalization attempt failed).
\cs_new:Npn \__fp_ln_x_iii:NNNNNNw #1#2 #3#4#5#6 #7\__fp_sep:
  { #1#2\__fp_sep: {#3#4#5#6} {#7} }
\cs_new:Npn \__fp_ln_x_iii_var:NNNNNw #1 #2#3#4#5 #6\__fp_sep:
  {
    #1#2#3#4#5 + 1 \__fp_sep:
    {#1#2#3#4#5} {#6}
  }
\cs_new:Npn \__fp_ln_x_iv:wnnnnnnnn #1\__fp_sep: #2#3#4#5 #6#7#8#9
  {
    \exp_after:wN \__fp_div_significand_pack:NNN
    \int_value:w \__fp_int_eval:w
    \__fp_ln_div_i:w #1 \__fp_sep:
      #6 #7 \__fp_sep: {#8} {#9}
      {#2} {#3} {#4} {#5}
      { \exp_after:wN \__fp_ln_div_ii:wwn \int_value:w #1 }
      { \exp_after:wN \__fp_ln_div_ii:wwn \int_value:w #1 }
      { \exp_after:wN \__fp_ln_div_ii:wwn \int_value:w #1 }
      { \exp_after:wN \__fp_ln_div_ii:wwn \int_value:w #1 }
      { \exp_after:wN \__fp_ln_div_vi:wwn \int_value:w #1 }
  }
\cs_new:Npn \__fp_ln_div_i:w #1\__fp_sep:
  {
    \exp_after:wN \__fp_div_significand_calc:wwnnnnnnn
    \int_value:w \__fp_int_eval:w 999999 + 2 0000 0000 / #1 \__fp_sep: % Q1
  }
\cs_new:Npn \__fp_ln_div_ii:wwn
    #1\__fp_sep: #2\__fp_sep:#3 % y\__fp_sep: B1\__fp_sep:B2 <- for k=1
  {
    \exp_after:wN \__fp_div_significand_pack:NNN
    \int_value:w \__fp_int_eval:w
      \exp_after:wN \__fp_div_significand_calc:wwnnnnnnn
      \int_value:w \__fp_int_eval:w 999999 + #2 #3 / #1 \__fp_sep: % Q2
      #2 #3 \__fp_sep:
  }
\cs_new:Npn \__fp_ln_div_vi:wwn
    #1\__fp_sep: #2\__fp_sep:#3#4#5 #6#7#8#9 %y\__fp_sep:F1\__fp_sep:F2F3F4x1x2x3x4
  {
    \exp_after:wN \__fp_div_significand_pack:NNN
    \int_value:w \__fp_int_eval:w 1000000 + #2 #3 / #1 \__fp_sep: % Q6
  }
\cs_new:Npn \__fp_ln_div_after:Nw #1#2\__fp_sep:
  {
    \if_meaning:w 0 #2
      \exp_after:wN \__fp_ln_t_small:Nw
    \else:
      \exp_after:wN \__fp_ln_t_large:NNw
      \exp_after:wN -
    \fi:
    #1
  }
\cs_new:Npn \__fp_ln_t_small:Nw
    #1 #2\__fp_sep: #3\__fp_sep: #4\__fp_sep: #5\__fp_sep: #6\__fp_sep: #7\__fp_sep:
  {
    \exp_after:wN \__fp_ln_t_large:NNw
    \exp_after:wN + % <sign>
    \exp_after:wN #1
    \int_value:w \__fp_int_eval:w 9999 - #2 \exp_after:wN \__fp_sep:
    \int_value:w \__fp_int_eval:w 9999 - #3 \exp_after:wN \__fp_sep:
    \int_value:w \__fp_int_eval:w 9999 - #4 \exp_after:wN \__fp_sep:
    \int_value:w \__fp_int_eval:w 9999 - #5 \exp_after:wN \__fp_sep:
    \int_value:w \__fp_int_eval:w 9999 - #6 \exp_after:wN \__fp_sep:
    \int_value:w \__fp_int_eval:w 1 0000 - #7 \__fp_sep:
  }
\cs_new:Npn \__fp_ln_t_large:NNw
    #1 #2
    #3\__fp_sep: #4\__fp_sep: #5\__fp_sep: #6\__fp_sep: #7\__fp_sep: #8\__fp_sep:
  {
    \exp_after:wN \__fp_ln_square_t_after:w
    \int_value:w \__fp_int_eval:w 9999 0000 + #3*#3
      \exp_after:wN \__fp_ln_square_t_pack:NNNNNw
      \int_value:w \__fp_int_eval:w 9999 0000 + 2*#3*#4
        \exp_after:wN \__fp_ln_square_t_pack:NNNNNw
        \int_value:w \__fp_int_eval:w 9999 0000 + 2*#3*#5 + #4*#4
          \exp_after:wN \__fp_ln_square_t_pack:NNNNNw
          \int_value:w \__fp_int_eval:w 9999 0000 + 2*#3*#6 + 2*#4*#5
            \exp_after:wN \__fp_ln_square_t_pack:NNNNNw
            \int_value:w \__fp_int_eval:w
              1 0000 0000 + 2*#3*#7 + 2*#4*#6 + #5*#5
              + (2*#3*#8 + 2*#4*#7 + 2*#5*#6) / 1 0000
              % \__fp_sep: \__fp_sep: \__fp_sep:
    \exp_after:wN \__fp_ln_twice_t_after:w
    \int_value:w \__fp_int_eval:w -1 + 2*#3
      \exp_after:wN \__fp_ln_twice_t_pack:Nw
      \int_value:w \__fp_int_eval:w 9999 + 2*#4
        \exp_after:wN \__fp_ln_twice_t_pack:Nw
        \int_value:w \__fp_int_eval:w 9999 + 2*#5
          \exp_after:wN \__fp_ln_twice_t_pack:Nw
          \int_value:w \__fp_int_eval:w 9999 + 2*#6
            \exp_after:wN \__fp_ln_twice_t_pack:Nw
            \int_value:w \__fp_int_eval:w 9999 + 2*#7
              \exp_after:wN \__fp_ln_twice_t_pack:Nw
              \int_value:w \__fp_int_eval:w 10000 + 2*#8 \__fp_sep: \__fp_sep:
    { \__fp_ln_c:NwNw #1 }
    #2
  }
\cs_new:Npn \__fp_ln_twice_t_pack:Nw #1 #2\__fp_sep: { + #1 \__fp_sep: {#2} }
\cs_new:Npn \__fp_ln_twice_t_after:w #1\__fp_sep:
  { \__fp_sep:\__fp_sep:\__fp_sep: {#1} }
\cs_new:Npn \__fp_ln_square_t_pack:NNNNNw #1 #2#3#4#5 #6\__fp_sep:
  { + #1#2#3#4#5 \__fp_sep: {#6} }
\cs_new:Npn \__fp_ln_square_t_after:w 1 0 #1#2#3 #4\__fp_sep:
  { \__fp_ln_Taylor:wwNw {0#1#2#3} {#4} }
\cs_new:Npn \__fp_ln_Taylor:wwNw
  {
    \__fp_ln_Taylor_loop:www
      21 \__fp_sep: {0000}{0000}{0000}{0000}{0000}{0000} \__fp_sep:
  }
\cs_new:Npn \__fp_ln_Taylor_loop:www #1\__fp_sep: #2\__fp_sep: #3\__fp_sep:
  {
    \if_int_compare:w #1 = \c_one_int
      \__fp_ln_Taylor_break:w
    \fi:
    \exp_after:wN \__fp_fixed_div_int:wwN \c__fp_one_fixed_tl #1\__fp_sep:
    \__fp_fixed_add:wwn #2\__fp_sep:
    \__fp_fixed_mul:wwn #3\__fp_sep:
    {
      \exp_after:wN \__fp_ln_Taylor_loop:www
      \int_value:w \__fp_int_eval:w #1 - 2 \__fp_sep:
    }
    #3\__fp_sep:
  }
\cs_new:Npn \__fp_ln_Taylor_break:w
    \fi: #1 \__fp_fixed_add:wwn #2#3\__fp_sep: #4 \__fp_sep:\__fp_sep:
  {
    \fi:
    \exp_after:wN \__fp_fixed_mul:wwn
    \exp_after:wN { \int_value:w \__fp_int_eval:w 10000 + #2 } #3\__fp_sep:
  }
\cs_new:Npn \__fp_ln_c:NwNw #1 #2\__fp_sep: #3
  {
    \if_meaning:w + #1
      \exp_after:wN \exp_after:wN \exp_after:wN \__fp_fixed_sub:wwn
    \else:
      \exp_after:wN \exp_after:wN \exp_after:wN \__fp_fixed_add:wwn
    \fi:
    #3 #2 \__fp_sep:
  }
\cs_new:Npn \__fp_ln_exponent:wn #1\__fp_sep: #2
  {
    \if_case:w #2 \exp_stop_f:
      0 \__fp_case_return:nw { \__fp_fixed_to_float_o:Nw 2 }
    \or:
      \exp_after:wN \__fp_ln_exponent_one:ww \int_value:w
    \else:
      \if_int_compare:w #2 > \c_zero_int
        \exp_after:wN \__fp_ln_exponent_small:NNww
        \exp_after:wN 0
        \exp_after:wN \__fp_fixed_sub:wwn \int_value:w
      \else:
        \exp_after:wN \__fp_ln_exponent_small:NNww
        \exp_after:wN 2
        \exp_after:wN \__fp_fixed_add:wwn \int_value:w -
      \fi:
    \fi:
    #2\__fp_sep: #1\__fp_sep:
  }
\cs_new:Npn \__fp_ln_exponent_one:ww 1\__fp_sep: #1\__fp_sep:
  {
    0
    \exp_after:wN \__fp_fixed_sub:wwn \c__fp_ln_x_fixed_tl #1\__fp_sep:
    \__fp_fixed_to_float_o:wN 0
  }
\cs_new:Npn \__fp_ln_exponent_small:NNww #1#2#3\__fp_sep: #4#5#6#7#8#9\__fp_sep:
  {
    4
    \exp_after:wN \__fp_fixed_mul:wwn
      \c__fp_ln_x_fixed_tl
      {#3}{0000}{0000}{0000}{0000}{0000} \__fp_sep:
    #2
      {0000}{#4}{#5}{#6}{#7}{#8}\__fp_sep:
    \__fp_fixed_to_float_o:wN #1
  }
\cs_new:Npn \__fp_exp_o:w #1 \s__fp \__fp_chk:w #2#3#4\__fp_sep: @
  {
    \if_case:w #2 \exp_stop_f:
      \__fp_case_return_o:Nw \c_one_fp
    \or:
      \exp_after:wN \__fp_exp_normal_o:w
    \or:
      \if_meaning:w 0 #3
        \exp_after:wN \__fp_case_return_o:Nw
        \exp_after:wN \c_inf_fp
      \else:
        \exp_after:wN \__fp_case_return_o:Nw
        \exp_after:wN \c_zero_fp
      \fi:
    \or:
      \__fp_case_return_same_o:w
    \fi:
    \s__fp \__fp_chk:w #2#3#4\__fp_sep:
  }
\cs_new:Npn \__fp_exp_normal_o:w \s__fp \__fp_chk:w 1#1
  {
    \if_meaning:w 0 #1
      \__fp_exp_pos_o:NNwnw + \__fp_fixed_to_float_o:wN
    \else:
      \__fp_exp_pos_o:NNwnw - \__fp_fixed_inv_to_float_o:wN
    \fi:
  }
\cs_new:Npn \__fp_exp_pos_o:NNwnw #1#2#3 \fi: #4#5\__fp_sep:
  {
    \fi:
    \if_int_compare:w #4 > \c__fp_max_exp_exponent_int
      \token_if_eq_charcode:NNTF + #1
        { \__fp_exp_overflow:NN \__fp_overflow:w \c_inf_fp }
        { \__fp_exp_overflow:NN \__fp_underflow:w \c_zero_fp }
      \exp:w
    \else:
      \exp_after:wN \__fp_sanitize:Nw
      \exp_after:wN 0
      \int_value:w #1 \__fp_int_eval:w
        \if_int_compare:w #4 < \c_zero_int
          \exp_after:wN \use_i:nn
        \else:
          \exp_after:wN \use_ii:nn
        \fi:
        {
          0
          \__fp_decimate:nNnnnn { - #4 }
            \__fp_exp_Taylor:Nnnwn
        }
        {
          \__fp_decimate:nNnnnn { \c__fp_prec_int - #4 }
            \__fp_exp_pos_large:NnnNwn
        }
        #5
        {#4}
        #1 #2 0
        \exp:w
    \fi:
    \exp_after:wN \exp_end:
  }
\cs_new:Npn \__fp_exp_overflow:NN #1#2
  {
    \exp_after:wN \exp_after:wN
    \exp_after:wN #1
    \exp_after:wN #2
  }
\cs_new:Npn \__fp_exp_Taylor:Nnnwn #1#2#3 #4\__fp_sep: #5 #6
```

```
  {
    #6
    \__fp_pack_twice_four:wNNNNNNNN
    \__fp_pack_twice_four:wNNNNNNNN
    \__fp_pack_twice_four:wNNNNNNNN
    \__fp_exp_Taylor_ii:ww
    \__fp_sep: #2#3#4 0000 0000 \__fp_sep:
  }
\cs_new:Npn \__fp_exp_Taylor_ii:ww #1\__fp_sep: #2\__fp_sep:
  { \__fp_exp_Taylor_loop:www 10 \__fp_sep: #1 \__fp_sep: #1 \__fp_sep: \s__fp_stop }
\cs_new:Npn \__fp_exp_Taylor_loop:www #1\__fp_sep: #2\__fp_sep: #3\__fp_sep:
  {
    \if_int_compare:w #1 = \c_one_int
      \exp_after:wN \__fp_exp_Taylor_break:Nww
    \fi:
    \__fp_fixed_div_int:wwN #3 \__fp_sep: #1 \__fp_sep:
    \__fp_fixed_add_one:wN
    \__fp_fixed_mul:wwn #2 \__fp_sep:
    {
      \exp_after:wN \__fp_exp_Taylor_loop:www
      \int_value:w \__fp_int_eval:w #1 - 1 \__fp_sep:
      #2 \__fp_sep:
    }
  }
\cs_new:Npn \__fp_exp_Taylor_break:Nww #1 #2\__fp_sep: #3 \s__fp_stop
  { \__fp_fixed_add_one:wN #2 \__fp_sep: }
\intarray_const_from_clist:Nn \c__fp_exp_intarray
  {
         1 , 1 1105 1709 , 1 1807 5647 , 1 6248 1171 ,
         1 , 1 1221 4027 , 1 5816 0169 , 1 8339 2107 ,
         1 , 1 1349 8588 , 1 0757 6003 , 1 1039 8374 ,
         1 , 1 1491 8246 , 1 9764 1270 , 1 3178 2485 ,
         1 , 1 1648 7212 , 1 7070 0128 , 1 1468 4865 ,
         1 , 1 1822 1188 , 1 0039 0508 , 1 9748 7537 ,
         1 , 1 2013 7527 , 1 0747 0476 , 1 5216 2455 ,
         1 , 1 2225 5409 , 1 2849 2467 , 1 6045 7954 ,
         1 , 1 2459 6031 , 1 1115 6949 , 1 6638 0013 ,
         1 , 1 2718 2818 , 1 2845 9045 , 1 2353 6029 ,
         1 , 1 7389 0560 , 1 9893 0650 , 1 2272 3043 ,
         2 , 1 2008 5536 , 1 9231 8766 , 1 7740 9285 ,
         2 , 1 5459 8150 , 1 0331 4423 , 1 9078 1103 ,
         3 , 1 1484 1315 , 1 9102 5766 , 1 0342 1116 ,
         3 , 1 4034 2879 , 1 3492 7351 , 1 2260 8387 ,
         4 , 1 1096 6331 , 1 5842 8458 , 1 5992 6372 ,
         4 , 1 2980 9579 , 1 8704 1728 , 1 2747 4359 ,
         4 , 1 8103 0839 , 1 2757 5384 , 1 0077 1000 ,
         5 , 1 2202 6465 , 1 7948 0671 , 1 6516 9579 ,
         9 , 1 4851 6519 , 1 5409 7902 , 1 7796 9107 ,
        14 , 1 1068 6474 , 1 5815 2446 , 1 2146 9905 ,
        18 , 1 2353 8526 , 1 6837 0199 , 1 8540 7900 ,
        22 , 1 5184 7055 , 1 2858 7072 , 1 4640 8745 ,
        27 , 1 1142 0073 , 1 8981 5684 , 1 2836 6296 ,
        31 , 1 2515 4386 , 1 7091 9167 , 1 0062 6578 ,
        35 , 1 5540 6223 , 1 8439 3510 , 1 0525 7117 ,
        40 , 1 1220 4032 , 1 9431 7840 , 1 8020 0271 ,
        44 , 1 2688 1171 , 1 4181 6135 , 1 4484 1263 ,
        87 , 1 7225 9737 , 1 6812 5749 , 1 2581 7748 ,
       131 , 1 1942 4263 , 1 9524 1255 , 1 9365 8421 ,
       174 , 1 5221 4696 , 1 8976 4143 , 1 9505 8876 ,
       218 , 1 1403 5922 , 1 1785 2837 , 1 4107 3977 ,
       261 , 1 3773 0203 , 1 0092 9939 , 1 8234 0143 ,
       305 , 1 1014 2320 , 1 5473 5004 , 1 5094 5533 ,
       348 , 1 2726 3745 , 1 7211 2566 , 1 5673 6478 ,
       391 , 1 7328 8142 , 1 2230 7421 , 1 7051 8866 ,
       435 , 1 1970 0711 , 1 1401 7046 , 1 9938 8888 ,
       869 , 1 3881 1801 , 1 9428 4368 , 1 5764 8232 ,
      1303 , 1 7646 2009 , 1 8905 4704 , 1 8893 1073 ,
      1738 , 1 1506 3559 , 1 7005 0524 , 1 9009 7592 ,
      2172 , 1 2967 6283 , 1 8402 3667 , 1 0689 6630 ,
      2606 , 1 5846 4389 , 1 5650 2114 , 1 7278 5046 ,
      3041 , 1 1151 7900 , 1 5080 6878 , 1 2914 4154 ,
      3475 , 1 2269 1083 , 1 0850 6857 , 1 8724 4002 ,
      3909 , 1 4470 3047 , 1 3316 5442 , 1 6408 6591 ,
      4343 , 1 8806 8182 , 1 2566 2921 , 1 5872 6150 ,
      8686 , 1 7756 0047 , 1 2598 6861 , 1 0458 3204 ,
     13029 , 1 6830 5723 , 1 7791 4884 , 1 1932 7351 ,
     17372 , 1 6015 5609 , 1 3095 3052 , 1 3494 7574 ,
     21715 , 1 5297 7951 , 1 6443 0315 , 1 3251 3576 ,
     26058 , 1 4665 6719 , 1 0099 3379 , 1 5527 2929 ,
     30401 , 1 4108 9724 , 1 3326 3186 , 1 5271 5665 ,
     34744 , 1 3618 6973 , 1 3140 0875 , 1 3856 4102 ,
     39087 , 1 3186 9209 , 1 6113 3900 , 1 6705 9685 ,
  }
\cs_new:Npn \__fp_exp_pos_large:NnnNwn #1#2#3 #4#5\__fp_sep: #6
  {
    \exp_after:wN \exp_after:wN \exp_after:wN \__fp_exp_large:NwN
    \exp_after:wN \exp_after:wN \exp_after:wN #6
    \exp_after:wN \c__fp_one_fixed_tl
    \int_value:w #3 #4 \exp_stop_f:
    #5 00000 \__fp_sep:
  }
\cs_new:Npn \__fp_exp_large:NwN #1#2\__fp_sep: #3
```

```
  {
    \if_case:w #3 ~
      \exp_after:wN \__fp_fixed_continue:wn
    \else:
      \exp_after:wN \__fp_exp_intarray:w
      \int_value:w \__fp_int_eval:w 36 * #1 + 4 * #3 \exp_after:wN \__fp_sep:
    \fi:
    #2\__fp_sep:
    {
      \if_meaning:w 0 #1
        \exp_after:wN \__fp_exp_large_after:wwn
      \else:
        \exp_after:wN \__fp_exp_large:NwN
        \int_value:w \__fp_int_eval:w #1 - 1 \exp_after:wN \scan_stop:
      \fi:
    }
  }
\cs_new:Npn \__fp_exp_intarray:w #1 \__fp_sep:
  {
    +
    \__kernel_intarray_item:Nn \c__fp_exp_intarray
      { \__fp_int_eval:w #1 - 3 \scan_stop: }
    \exp_after:wN \use_i:nnn
    \exp_after:wN \__fp_fixed_mul:wwn
    \int_value:w 0
    \exp_after:wN \__fp_exp_intarray_aux:w
    \int_value:w \__kernel_intarray_item:Nn
                   \c__fp_exp_intarray { \__fp_int_eval:w #1 - 2 }
    \exp_after:wN \__fp_exp_intarray_aux:w
    \int_value:w \__kernel_intarray_item:Nn
                   \c__fp_exp_intarray { \__fp_int_eval:w #1 - 1 }
    \exp_after:wN \__fp_exp_intarray_aux:w
    \int_value:w \__kernel_intarray_item:Nn
                   \c__fp_exp_intarray {#1} \__fp_sep: \__fp_sep:
  }
\cs_new:Npn \__fp_exp_intarray_aux:w 1 #1#2#3#4#5 \__fp_sep:
  { \__fp_sep: {#1#2#3#4} {#5} }
\cs_new:Npn \__fp_exp_large_after:wwn #1\__fp_sep: #2\__fp_sep: #3
  {
    \__fp_exp_Taylor:Nnnwn ? { } { } 0 #2\__fp_sep: {} #3
    \__fp_fixed_mul:wwn #1\__fp_sep:
  }
\cs_new:cpn { __fp_ \iow_char:N \^ _o:ww }
    \s__fp \__fp_chk:w #1#2#3\__fp_sep: \s__fp \__fp_chk:w #4#5#6\__fp_sep:
  {
    \if_meaning:w 0 #4
```

```
      \__fp_case_return_o:Nw \c_one_fp
    \fi:
    \if_case:w #2 \exp_stop_f:
      \exp_after:wN \use_i:nn
    \or:
      \__fp_case_return_o:Nw \c_nan_fp
    \else:
      \exp_after:wN \__fp_pow_neg:www
      \exp:w \exp_end_continue_f:w \exp_after:wN \use:nn
    \fi:
    {
      \if_meaning:w 1 #1
        \exp_after:wN \__fp_pow_normal_o:ww
      \else:
        \exp_after:wN \__fp_pow_zero_or_inf:ww
      \fi:
      \s__fp \__fp_chk:w #1#2#3\__fp_sep:
    }
    { \s__fp \__fp_chk:w #4#5#6\__fp_sep: \s__fp \__fp_chk:w #1#2#3\__fp_sep: }
    \s__fp \__fp_chk:w #4#5#6\__fp_sep:
  }
\cs_new:Npn \__fp_pow_zero_or_inf:ww
    \s__fp \__fp_chk:w #1#2\__fp_sep: \s__fp \__fp_chk:w #3#4
  {
    \if_meaning:w 1 #4
      \__fp_case_return_same_o:w
    \fi:
    \if_meaning:w #1 #4
      \__fp_case_return_o:Nw \c_zero_fp
    \fi:
    \if_meaning:w 2 #1
      \__fp_case_return_o:Nw \c_inf_fp
    \fi:
    \if_meaning:w 2 #3
      \__fp_case_return_o:Nw \c_inf_fp
    \else:
      \__fp_case_use:nw
        {
          \__fp_division_by_zero_o:NNww \c_inf_fp ^
            \s__fp \__fp_chk:w #1 #2 \__fp_sep:
        }
    \fi:
    \s__fp \__fp_chk:w #3#4
  }
\cs_new:Npn \__fp_pow_normal_o:ww
    \s__fp \__fp_chk:w 1 #1#2#3\__fp_sep: \s__fp \__fp_chk:w #4#5
  {
    \if:w 0 \__fp_str_if_eq:nn { #2 #3 } { 1 {1000} {0000} {0000} {0000} }
      \if_int_compare:w #4 #1 = 32 \exp_stop_f:
        \exp_after:wN \__fp_case_return_ii_o:ww
      \fi:
      \__fp_case_return_o:Nww \c_one_fp
    \fi:
    \if_case:w #4 \exp_stop_f:
    \or:
      \exp_after:wN \__fp_pow_npos_o:Nww
      \exp_after:wN #5
    \or:
      \if_meaning:w 2 #5 \exp_after:wN \reverse_if:N \fi:
      \if_int_compare:w #2 > \c_zero_int
        \exp_after:wN \__fp_case_return_o:Nww
        \exp_after:wN \c_inf_fp
      \else:
        \exp_after:wN \__fp_case_return_o:Nww
        \exp_after:wN \c_zero_fp
      \fi:
    \or:
      \__fp_case_return_ii_o:ww
    \fi:
    \s__fp \__fp_chk:w 1 #1 {#2} #3 \__fp_sep:
    \s__fp \__fp_chk:w #4 #5
  }
\cs_new:Npn \__fp_pow_npos_o:Nww #1 \s__fp \__fp_chk:w 1#2#3
  {
    \exp_after:wN \__fp_sanitize:Nw
    \exp_after:wN 0
    \int_value:w
      \if:w #1 \if_int_compare:w #3 > \c_zero_int 0 \else: 2 \fi:
        \exp_after:wN \__fp_pow_npos_aux:NNnww
        \exp_after:wN +
        \exp_after:wN \__fp_fixed_to_float_o:wN
      \else:
        \exp_after:wN \__fp_pow_npos_aux:NNnww
        \exp_after:wN -
        \exp_after:wN \__fp_fixed_inv_to_float_o:wN
      \fi:
      {#3}
  }
\cs_new:Npn \__fp_pow_npos_aux:NNnww
    #1#2#3#4#5\__fp_sep: \s__fp \__fp_chk:w 1#6#7#8\__fp_sep:
  {
    #1
    \__fp_int_eval:w
      \__fp_ln_significand:NNNNnnnN #4#5
      \__fp_pow_exponent:wnN {#3}
      \__fp_fixed_mul:wwn #8 {0000}{0000} \__fp_sep:
      \__fp_pow_B:wwN #7\__fp_sep:
      #1 #2 0 % fixed_to_float_o:wN
    }
\cs_new:Npn \__fp_pow_exponent:wnN #1\__fp_sep: #2
    {
    \if_int_compare:w #2 > \c_zero_int
      \exp_after:wN \__fp_pow_exponent:Nwnnnnnw % n\ln(10) - (-\ln(x))
      \exp_after:wN +
    \else:
      \exp_after:wN \__fp_pow_exponent:Nwnnnnnw % -(|n|\ln(10) + (-\ln(x)))
      \exp_after:wN -
    \fi:
    #2\__fp_sep: #1\__fp_sep:
    }
\cs_new:Npn \__fp_pow_exponent:Nwnnnnnw #1#2\__fp_sep: #3#4#5#6#7#8\__fp_sep:
  { %^^A todo: use that in ln.
    \exp_after:wN \__fp_fixed_mul_after:wwn
    \int_value:w \__fp_int_eval:w \c__fp_leading_shift_int
      \exp_after:wN \__fp_pack:NNNNNw
      \int_value:w \__fp_int_eval:w \c__fp_middle_shift_int
       #1#2*23025 - #1 #3
        \exp_after:wN \__fp_pack:NNNNNw
        \int_value:w \__fp_int_eval:w \c__fp_middle_shift_int
         #1 #2*8509 - #1 #4
          \exp_after:wN \__fp_pack:NNNNNw
          \int_value:w \__fp_int_eval:w \c__fp_middle_shift_int
           #1 #2*2994 - #1 #5
            \exp_after:wN \__fp_pack:NNNNNw
            \int_value:w \__fp_int_eval:w \c__fp_middle_shift_int
             #1 #2*0456 - #1 #6
              \exp_after:wN \__fp_pack:NNNNNw
              \int_value:w \__fp_int_eval:w \c__fp_trailing_shift_int
               #1 #2*8401 - #1 #7
                #1 ( #2*7991 - #8 ) / 1 0000 \__fp_sep: \__fp_sep:
  }
\cs_new:Npn \__fp_pow_B:wwN #1#2#3#4#5#6\__fp_sep: #7\__fp_sep:
  {
    \if_int_compare:w #7 < \c_zero_int
      \exp_after:wN \__fp_pow_C_neg:w \int_value:w -
    \else:
      \if_int_compare:w #7 < 22 \exp_stop_f:
        \exp_after:wN \__fp_pow_C_pos:w \int_value:w
```

```
      \else:
        \exp_after:wN \__fp_pow_C_overflow:w \int_value:w
      \fi:
    \fi:
    #7 \exp_after:wN \__fp_sep:
    \int_value:w \__fp_int_eval:w 10 0000 + #1 \__fp_int_eval_end:
    #2#3#4#5#6 0000 0000 0000 0000 0000 0000 \__fp_sep: %^^A todo: how many 0?
  }
\cs_new:Npn \__fp_pow_C_overflow:w #1\__fp_sep: #2\__fp_sep: #3
  {
    + 2 * \c__fp_max_exponent_int
    \exp_after:wN \__fp_fixed_continue:wn \c__fp_one_fixed_tl
  }
\cs_new:Npn \__fp_pow_C_neg:w #1 \__fp_sep: 1
  {
    \exp_after:wN \exp_after:wN \exp_after:wN \__fp_pow_C_pack:w
    \prg_replicate:nn {#1} {0}
  }
\cs_new:Npn \__fp_pow_C_pos:w #1\__fp_sep: 1
  { \__fp_pow_C_pos_loop:wN #1\__fp_sep: }
\cs_new:Npn \__fp_pow_C_pos_loop:wN #1\__fp_sep: #2
  {
    \if_meaning:w 0 #1
      \exp_after:wN \__fp_pow_C_pack:w
      \exp_after:wN #2
    \else:
      \if_meaning:w 0 #2
        \exp_after:wN \__fp_pow_C_pos_loop:wN \int_value:w
      \else:
        \exp_after:wN \__fp_pow_C_overflow:w \int_value:w
      \fi:
      \__fp_int_eval:w #1 - 1 \exp_after:wN \__fp_sep:
    \fi:
  }
\cs_new:Npn \__fp_pow_C_pack:w
  {
    \exp_after:wN \__fp_exp_large:NwN
    \exp_after:wN 5
    \c__fp_one_fixed_tl
  }
\cs_new:Npn \__fp_pow_neg:www
    \s__fp \__fp_chk:w #1#2\__fp_sep: #3\__fp_sep: #4\__fp_sep:
  {
    \if_case:w \__fp_pow_neg_case:w #4 \__fp_sep:
      \exp_after:wN \__fp_pow_neg_aux:wNN
    \or:
      \if_int_compare:w \__fp_int_eval:w #1 / 2 = \c_one_int
        \__fp_invalid_operation_o:Nww ^ #3\__fp_sep: #4\__fp_sep:
        \exp:w \exp_end_continue_f:w
        \exp_after:wN \exp_after:wN
        \exp_after:wN \__fp_use_none_until_s:w
      \fi:
    \fi:
    \__fp_exp_after_o:w
    \s__fp \__fp_chk:w #1#2\__fp_sep:
  }
\cs_new:Npn \__fp_pow_neg_aux:wNN #1 \s__fp \__fp_chk:w #2#3
  {
    \exp_after:wN \__fp_exp_after_o:w
    \exp_after:wN \s__fp
    \exp_after:wN \__fp_chk:w
    \exp_after:wN #2
    \int_value:w \__fp_int_eval:w 2 - #3 \__fp_int_eval_end:
  }
\cs_new:Npn \__fp_pow_neg_case:w \s__fp \__fp_chk:w #1#2#3\__fp_sep:
  {
    \if_case:w #1 \exp_stop_f:
           -1
    \or:   \__fp_pow_neg_case_aux:nnnnn #3
    \or:   -1
    \else: 1
    \fi:
    \exp_stop_f:
  }
\cs_new:Npn \__fp_pow_neg_case_aux:nnnnn #1#2#3#4#5
  {
    \if_int_compare:w #1 > \c__fp_prec_int
      -1
    \else:
      \__fp_decimate:nNnnnn { \c__fp_prec_int - #1 }
        \__fp_pow_neg_case_aux:Nnnw
        {#2} {#3} {#4} {#5}
    \fi:
  }
\cs_new:Npn \__fp_pow_neg_case_aux:Nnnw #1#2#3#4 \__fp_sep:
  {
    \if_meaning:w 0 #1
      \if_int_odd:w #3 \exp_stop_f:
        0
      \else:
        -1
      \fi:
```

```
    \else:
      1
    \fi:
  }
\int_const:Nn \c__fp_fact_max_arg_int { 3248 }
\cs_new:Npn \__fp_fact_o:w #1 \s__fp \__fp_chk:w #2#3#4\__fp_sep: @
  {
    \if_case:w #2 \exp_stop_f:
      \__fp_case_return_o:Nw \c_one_fp
    \or:
    \or:
      \if_meaning:w 0 #3
        \exp_after:wN \__fp_case_return_same_o:w
      \fi:
    \or:
      \__fp_case_return_same_o:w
    \fi:
    \if_meaning:w 2 #3
      \__fp_case_use:nw { \__fp_invalid_operation_o:fw { fact } }
    \fi:
    \__fp_fact_pos_o:w
    \s__fp \__fp_chk:w #2 #3 #4 \__fp_sep:
  }
\cs_new:Npn \__fp_fact_pos_o:w #1\__fp_sep:
  {
    \__fp_small_int:wTF #1\__fp_sep:
      { \__fp_fact_int_o:n }
      { \__fp_invalid_operation_o:fw { fact } #1\__fp_sep: }
  }
\cs_new:Npn \__fp_fact_int_o:n #1
  {
    \if_int_compare:w #1 > \c__fp_fact_max_arg_int
      \__fp_case_return:nw
        {
          \exp_after:wN \exp_after:wN \exp_after:wN \__fp_overflow:w
          \exp_after:wN \c_inf_fp
        }
    \fi:
    \exp_after:wN \__fp_sanitize:Nw
    \exp_after:wN 0
    \int_value:w \__fp_int_eval:w
    \__fp_fact_loop_o:w #1 . 4 , { 1 } { } { } { } { } { } \__fp_sep:
  }
\cs_new:Npn \__fp_fact_loop_o:w #1 . #2 \__fp_sep:
  {
    \if_int_compare:w #1 < 12 \exp_stop_f:
      \__fp_fact_small_o:w #1
    \fi:
    \exp_after:wN \__fp_ep_mul:wwwwn
    \exp_after:wN 4 \exp_after:wN ,
    \exp_after:wN { \int_value:w \__fp_int_eval:w #1 * (#1 - 1) }
    { } { } { } { } { } \__fp_sep:
    #2 \__fp_sep:
    {
      \exp_after:wN \__fp_fact_loop_o:w
      \int_value:w \__fp_int_eval:w #1 - 2 .
    }
  }
\cs_new:Npn \__fp_fact_small_o:w #1 \fi: #2 \__fp_sep: #3 \__fp_sep: #4
  {
    \fi:
    \exp_after:wN \__fp_ep_mul:wwwwn
    \exp_after:wN 4 \exp_after:wN ,
    \exp_after:wN
      {
        \int_value:w
        \if_case:w #1 \exp_stop_f:
        1 \or: 1 \or: 2 \or: 6 \or: 24 \or: 120 \or: 720 \or: 5040
        \or: 40320 \or: 362880 \or: 3628800 \or: 39916800
        \fi:
      } { } { } { } { } { } \__fp_sep:
    #3 \__fp_sep:
    \__fp_ep_to_float_o:wwN 0
  }
%% File: l3fp-trig.dtx
\tl_map_inline:nn
  {
    {acos} {acsc} {asec} {asin}
    {cos} {cot} {csc} {sec} {sin} {tan}
  }
  {
    \cs_new:cpe { __fp_parse_word_#1:N }
      {
        \exp_not:N \__fp_parse_unary_function:NNN
        \exp_not:c { __fp_#1_o:w }
        \exp_not:N \use_i:nn
      }
    \cs_new:cpe { __fp_parse_word_#1d:N }
      {
        \exp_not:N \__fp_parse_unary_function:NNN
        \exp_not:c { __fp_#1_o:w }
        \exp_not:N \use_ii:nn
      }
  }
\cs_new:Npn \__fp_parse_word_acot:N
  { \__fp_parse_function:NNN \__fp_acot_o:Nw \use_i:nn }
\cs_new:Npn \__fp_parse_word_acotd:N
  { \__fp_parse_function:NNN \__fp_acot_o:Nw \use_ii:nn }
\cs_new:Npn \__fp_parse_word_atan:N
  { \__fp_parse_function:NNN \__fp_atan_o:Nw \use_i:nn }
\cs_new:Npn \__fp_parse_word_atand:N
  { \__fp_parse_function:NNN \__fp_atan_o:Nw \use_ii:nn }
\cs_new:Npn \__fp_sin_o:w #1 \s__fp \__fp_chk:w #2#3#4\__fp_sep: @
  {
    \if_case:w #2 \exp_stop_f:
           \__fp_case_return_same_o:w
    \or:   \__fp_case_use:nw
             {
               \__fp_trig:NNNNNwn #1 \__fp_sin_series_o:NNwwww
                 \__fp_ep_to_float_o:wwN #3 0
             }
    \or:   \__fp_case_use:nw
             { \__fp_invalid_operation_o:fw { #1 { sin } { sind } } }
    \else: \__fp_case_return_same_o:w
    \fi:
    \s__fp \__fp_chk:w #2 #3 #4\__fp_sep:
  }
\cs_new:Npn \__fp_cos_o:w #1 \s__fp \__fp_chk:w #2#3\__fp_sep: @
  {
    \if_case:w #2 \exp_stop_f:
           \__fp_case_return_o:Nw \c_one_fp
    \or:   \__fp_case_use:nw
             {
               \__fp_trig:NNNNNwn #1 \__fp_sin_series_o:NNwwww
                 \__fp_ep_to_float_o:wwN 0 2
             }
    \or:   \__fp_case_use:nw
             { \__fp_invalid_operation_o:fw { #1 { cos } { cosd } } }
    \else: \__fp_case_return_same_o:w
    \fi:
    \s__fp \__fp_chk:w #2 #3\__fp_sep:
  }
\cs_new:Npn \__fp_csc_o:w #1 \s__fp \__fp_chk:w #2#3#4\__fp_sep: @
  {
    \if_case:w #2 \exp_stop_f:
           \__fp_cot_zero_o:Nfw #3 { #1 { csc } { cscd } }
    \or:   \__fp_case_use:nw
             {
               \__fp_trig:NNNNNwn #1 \__fp_sin_series_o:NNwwww
                 \__fp_ep_inv_to_float_o:wwN #3 0
             }
    \or:   \__fp_case_use:nw
             { \__fp_invalid_operation_o:fw { #1 { csc } { cscd } } }
    \else: \__fp_case_return_same_o:w
    \fi:
    \s__fp \__fp_chk:w #2 #3 #4\__fp_sep:
  }
\cs_new:Npn \__fp_sec_o:w #1 \s__fp \__fp_chk:w #2#3\__fp_sep: @
  {
    \if_case:w #2 \exp_stop_f:
           \__fp_case_return_o:Nw \c_one_fp
    \or:   \__fp_case_use:nw
             {
               \__fp_trig:NNNNNwn #1 \__fp_sin_series_o:NNwwww
                 \__fp_ep_inv_to_float_o:wwN 0 2
             }
    \or:   \__fp_case_use:nw
             { \__fp_invalid_operation_o:fw { #1 { sec } { secd } } }
    \else: \__fp_case_return_same_o:w
    \fi:
    \s__fp \__fp_chk:w #2 #3\__fp_sep:
  }
\cs_new:Npn \__fp_tan_o:w #1 \s__fp \__fp_chk:w #2#3#4\__fp_sep: @
  {
    \if_case:w #2 \exp_stop_f:
           \__fp_case_return_same_o:w
    \or:   \__fp_case_use:nw
             {
               \__fp_trig:NNNNNwn #1
                 \__fp_tan_series_o:NNwwww 0 #3 1
             }
    \or:   \__fp_case_use:nw
             { \__fp_invalid_operation_o:fw { #1 { tan } { tand } } }
    \else: \__fp_case_return_same_o:w
    \fi:
    \s__fp \__fp_chk:w #2 #3 #4\__fp_sep:
  }
\cs_new:Npn \__fp_cot_o:w #1 \s__fp \__fp_chk:w #2#3#4\__fp_sep: @
  {
    \if_case:w #2 \exp_stop_f:
           \__fp_cot_zero_o:Nfw #3 { #1 { cot } { cotd } }
    \or:   \__fp_case_use:nw
             {
               \__fp_trig:NNNNNwn #1
```

```
                 \__fp_tan_series_o:NNwwww 2 #3 3
             }
    \or:   \__fp_case_use:nw
             { \__fp_invalid_operation_o:fw { #1 { cot } { cotd } } }
    \else: \__fp_case_return_same_o:w
    \fi:
    \s__fp \__fp_chk:w #2 #3 #4\__fp_sep:
  }
\cs_new:Npn \__fp_cot_zero_o:Nfw #1#2#3 \fi:
  {
    \fi:
    \token_if_eq_meaning:NNTF 0 #1
      { \exp_args:NNf \__fp_division_by_zero_o:Nnw \c_inf_fp }
      { \exp_args:NNf \__fp_division_by_zero_o:Nnw \c_minus_inf_fp }
    {#2}
  }
\cs_new:Npn \__fp_trig:NNNNNwn #1#2#3#4#5 \s__fp \__fp_chk:w 1#6#7#8\__fp_sep:
  {
    \exp_after:wN #2
    \exp_after:wN #3
    \exp_after:wN #4
    \int_value:w \__fp_int_eval:w #5
      \exp_after:wN \exp_after:wN \exp_after:wN \exp_after:wN
      \if_int_compare:w #7 > #1 0 1 \exp_stop_f:
        #1 \__fp_trig_large:ww \__fp_trigd_large:ww
      \else:
        #1 \__fp_trig_small:ww \__fp_trigd_small:ww
      \fi:
    #7,#8{0000}{0000}\__fp_sep:
  }
\cs_new:Npn \__fp_trig_small:ww #1,#2\__fp_sep:
  { \__fp_ep_to_fixed:wwn #1,#2\__fp_sep: . #1,#2\__fp_sep: }
\cs_new:Npn \__fp_trigd_small:ww #1,#2\__fp_sep:
  {
    \__fp_ep_mul_raw:wwwwN
      -1,{1745}{3292}{5199}{4329}{5769}{2369}\__fp_sep: #1,#2\__fp_sep:
    \__fp_trig_small:ww
  }
\cs_new:Npn \__fp_trigd_large:ww #1, #2#3#4#5#6#7\__fp_sep:
  {
    \exp_after:wN \__fp_pack_eight:wNNNNNNNN
    \exp_after:wN \__fp_pack_eight:wNNNNNNNN
    \exp_after:wN \__fp_pack_twice_four:wNNNNNNNN
    \exp_after:wN \__fp_pack_twice_four:wNNNNNNNN
    \exp_after:wN \__fp_trigd_large_auxi:nnnnwNNNN
    \exp_after:wN \__fp_sep:
```

```
    \exp:w \exp_end_continue_f:w
    \prg_replicate:nn { \int_max:nn { 22 - #1 } { 0 } } { 0 }
    #2#3#4#5#6#7 0000 0000 0000 !
  }
\cs_new:Npn \__fp_trigd_large_auxi:nnnnwNNNN #1#2#3#4#5\__fp_sep: #6#7#8#9
  {
    \exp_after:wN \__fp_trigd_large_auxii:wNw
    \int_value:w \__fp_int_eval:w #1 + #2
      - (#1 + #2 - 4) / 9 * 9 \__fp_int_eval_end:
    #3\__fp_sep:
    #4\__fp_sep: #5{#6#7#8#9}\__fp_sep:
  }
\cs_new:Npn \__fp_trigd_large_auxii:wNw #1\__fp_sep: #2#3\__fp_sep:
  {
    + (#1#2 - 4) / 9 * 2
    \exp_after:wN \__fp_trigd_large_auxiii:www
    \int_value:w \__fp_int_eval:w #1#2
      - (#1#2 - 4) / 9 * 9 \__fp_int_eval_end: #3 \__fp_sep:
  }
\cs_new:Npn \__fp_trigd_large_auxiii:www #1\__fp_sep: #2\__fp_sep: #3!
  {
    \if_int_compare:w #1 < 4500 \exp_stop_f:
      \exp_after:wN \__fp_use_i_until_s:nw
      \exp_after:wN \__fp_fixed_continue:wn
    \else:
      + 1
    \fi:
    \__fp_fixed_sub:wwn {9000}{0000}{0000}{0000}{0000}{0000}\__fp_sep:
      {#1}#2{0000}{0000}\__fp_sep:
    { \__fp_trigd_small:ww 2, }
  }
\intarray_const_from_clist:Nn \c__fp_trig_intarray
  {
    100000000, 100000000, 115915494, 130918953, 135768883, 176337251,
    143620344, 159645740, 145644874, 176673440, 158896797, 163422653,
    150901138, 102766253, 108595607, 128427267, 157958036, 189291184,
    161145786, 152877967, 141073169, 198392292, 139966937, 140907757,
    130777463, 196925307, 168871739, 128962173, 197661693, 136239024,
    117236290, 111832380, 111422269, 197557159, 140461890, 108690267,
    139561204, 189410936, 193784408, 155287230, 199946443, 140024867,
    123477394, 159610898, 132309678, 130749061, 166986462, 180469944,
    186521878, 181574786, 156696424, 110389958, 174139348, 160998386,
    180991999, 162442875, 158517117, 188584311, 117518767, 116054654,
    175369880, 109739460, 136475933, 137680593, 102494496, 163530532,
    171567755, 103220324, 177781639, 171660229, 146748119, 159816584,
    106060168, 103035998, 113391198, 174988327, 186654435, 127975507,
100162406, 177564388, 184957131, 108801221, 199376147, 168137776,
147378906, 133068046, 145797848, 117613124, 127314069, 196077502,
145002977, 159857089, 105690279, 167851315, 125210016, 131774602,
109248116, 106240561, 145620314, 164840892, 148459191, 143521157,
154075562, 100871526, 160680221, 171591407, 157474582, 172259774,
162853998, 175155329, 139081398, 117724093, 158254797, 107332871,
190406999, 175907657, 170784934, 170393589, 182808717, 134256403,
166895116, 162545705, 194332763, 112686500, 126122717, 197115321,
112599504, 138667945, 103762556, 108363171, 116952597, 158128224,
194162333, 143145106, 112353687, 185631136, 136692167, 114206974,
169601292, 150578336, 105311960, 185945098, 139556718, 170995474,
165104316, 123815517, 158083944, 129799709, 199505254, 138756612,
194458833, 106846050, 178529151, 151410404, 189298850, 163881607,
176196993, 107341038, 199957869, 118905980, 193737772, 106187543,
122271893, 101366255, 126123878, 103875388, 181106814, 106765434,
108282785, 126933426, 179955607, 107903860, 160352738, 199624512,
159957492, 176297023, 159409558, 143011648, 129641185, 157771240,
157544494, 157021789, 176979240, 194903272, 194770216, 164960356,
153181535, 144003840, 168987471, 176915887, 163190966, 150696440,
147769706, 187683656, 177810477, 197954503, 153395758, 130188183,
186879377, 166124814, 195305996, 155802190, 183598751, 103512712,
190432315, 180498719, 168687775, 194656634, 162210342, 104440855,
149785037, 192738694, 129353661, 193778292, 187359378, 143470323,
102371458, 137923557, 111863634, 119294601, 183182291, 196416500,
187830793, 131353497, 179099745, 186492902, 167450609, 189368909,
145883050, 133703053, 180547312, 132158094, 131976760, 132283131,
141898097, 149822438, 133517435, 169898475, 101039500, 168388003,
197867235, 199608024, 100273901, 108749548, 154787923, 156826113,
199489032, 168997427, 108349611, 149208289, 103776784, 174303550,
145684560, 183671479, 130845672, 133270354, 185392556, 120208683,
193240995, 162211753, 131839402, 109707935, 170774965, 149880868,
160663609, 168661967, 103747454, 121028312, 119251846, 122483499,
111611495, 166556037, 196967613, 199312829, 196077608, 127799010,
107830360, 102338272, 198790854, 102387615, 157445430, 192601191,
100543379, 198389046, 154921248, 129516070, 172853005, 122721023,
160175233, 113173179, 175931105, 103281551, 109373913, 163964530,
157926071, 180083617, 195487672, 146459804, 173977292, 144810920,
109371257, 186918332, 189588628, 139904358, 168666639, 175673445,
114095036, 137327191, 174311388, 106638307, 125923027, 159734506,
105482127, 178037065, 133778303, 121709877, 134966568, 149080032,
169885067, 141791464, 168350828, 116168533, 114336160, 173099514,
198531198, 119733758, 144420984, 116559541, 152250643, 139431286,
144403838, 183561508, 179771645, 101706470, 167518774, 156059160,
187168578, 157939226, 123475633, 117111329, 198655941, 159689071,
198506887, 144230057, 151919770, 156900382, 118392562, 120338742,
135362568, 108354156, 151729710, 188117217, 195936832, 156488518,
174997487, 108553116, 159830610, 113921445, 144601614, 188452770,
125114110, 170248521, 173974510, 138667364, 103872860, 109967489,
131735618, 112071174, 104788993, 168886556, 192307848, 150230570,
157144063, 163863202, 136852010, 174100574, 185922811, 115721968,
100397824, 175953001, 166958522, 112303464, 118773650, 143546764,
164565659, 171901123, 108476709, 193097085, 191283646, 166919177,
169387914, 133315566, 150669813, 121641521, 100895711, 172862384,
126070678, 145176011, 113450800, 169947684, 122356989, 162488051,
157759809, 153397080, 185475059, 175362656, 149034394, 145420581,
178864356, 183042000, 131509559, 147434392, 152544850, 167491429,
108647514, 142303321, 133245695, 111634945, 167753939, 142403609,
105438335, 152829243, 142203494, 184366151, 146632286, 102477666,
166049531, 140657343, 157553014, 109082798, 180914786, 169343492,
127376026, 134997829, 195701816, 119643212, 133140475, 176289748,
140828911, 174097478, 126378991, 181699939, 148749771, 151989818,
172666294, 160183053, 195832752, 109236350, 168538892, 128468247,
125997252, 183007668, 156937583, 165972291, 198244297, 147406163,
181831139, 158306744, 134851692, 185973832, 137392662, 140243450,
119978099, 140402189, 161348342, 173613676, 144991382, 171541660,
163424829, 136374185, 106122610, 186132119, 198633462, 184709941,
183994274, 129559156, 128333990, 148038211, 175011612, 111667205,
119125793, 103552929, 124113440, 131161341, 112495318, 138592695,
184904438, 146807849, 109739828, 108855297, 104515305, 139914009,
188698840, 188365483, 166522246, 168624087, 125401404, 100911787,
142122045, 123075334, 173972538, 114940388, 141905868, 142311594,
163227443, 139066125, 116239310, 162831953, 123883392, 113153455,
163815117, 152035108, 174595582, 101123754, 135976815, 153401874,
107394340, 136339780, 138817210, 104531691, 182951948, 179591767,
139541778, 179243527, 161740724, 160593916, 102732282, 187946819,
136491289, 149714953, 143255272, 135916592, 198072479, 198580612,
169007332, 118844526, 179433504, 155801952, 149256630, 162048766,
116134365, 133992028, 175452085, 155344144, 109905129, 182727454,
165911813, 122232840, 151166615, 165070983, 175574337, 129548631,
120411217, 116380915, 160616116, 157320000, 183306114, 160618128,
103262586, 195951602, 146321661, 138576614, 180471993, 127077713,
116441201, 159496011, 106328305, 120759583, 148503050, 179095584,
198298218, 167402898, 138551383, 123957020, 180763975, 150429225,
198476470, 171016426, 197438450, 143091658, 164528360, 132493360,
143546572, 137557916, 113663241, 120457809, 196971566, 134022158,
180545794, 131328278, 100552461, 132088901, 187421210, 192448910,
141005215, 149680971, 113720754, 100571096, 134066431, 135745439,
191597694, 135788920, 179342561, 177830222, 137011486, 142492523,
192487287, 113132021, 176673607, 156645598, 127260957, 141566023,
143787436, 129132109, 174858971, 150713073, 191040726, 143541417,
197057222, 165479803, 181512759, 157912400, 125344680, 148220261,
173422990, 101020483, 106246303, 137964746, 178190501, 181183037,
151538028, 179523433, 141955021, 135689770, 191290561, 143178787,
192086205, 174499925, 178975690, 118492103, 124206471, 138519113,
188147564, 102097605, 154895793, 178514140, 141453051, 151583964,
128232654, 106020603, 131189158, 165702720, 186250269, 191639375,
115278873, 160608114, 155694842, 110322407, 177272742, 116513642,
134366992, 171634030, 194053074, 180652685, 109301658, 192136921,
141431293, 171341061, 157153714, 106203978, 147618426, 150297807,
186062669, 169960809, 118422347, 163350477, 146719017, 145045144,
161663828, 146208240, 186735951, 102371302, 190444377, 194085350,
134454426, 133413062, 163074595, 113830310, 122931469, 134466832,
185176632, 182415152, 110179422, 164439571, 181217170, 121756492,
119644493, 196532222, 118765848, 182445119, 109401340, 150443213,
198586286, 121083179, 139396084, 143898019, 114787389, 177233102,
186310131, 148695521, 126205182, 178063494, 157118662, 177825659,
188310053, 151552316, 165984394, 109022180, 163144545, 121212978,
197344714, 188741258, 126822386, 102360271, 109981191, 152056882,
134723983, 158013366, 106837863, 128867928, 161973236, 172536066,
185216856, 132011948, 197807339, 158419190, 166595838, 167852941,
124187182, 117279875, 106103946, 106481958, 157456200, 160892122,
184163943, 173846549, 158993202, 184812364, 133466119, 170732430,
195458590, 173361878, 162906318, 150165106, 126757685, 112163575,
188696307, 145199922, 100107766, 176830946, 198149756, 122682434,
179367131, 108412102, 119520899, 148191244, 140487511, 171059184,
141399078, 189455775, 118462161, 190415309, 134543802, 180893862,
180732375, 178615267, 179711433, 123241969, 185780563, 176301808,
184386640, 160717536, 183213626, 129671224, 126094285, 140110963,
121826276, 151201170, 122552929, 128965559, 146082049, 138409069,
107606920, 103954646, 119164002, 115673360, 117909631, 187289199,
186343410, 186903200, 157966371, 103128612, 135698881, 176403642,
152540837, 109810814, 183519031, 121318624, 172281810, 150845123,
169019064, 166322359, 138872454, 163073727, 128087898, 130041018,
194859136, 173742589, 141812405, 167291912, 138003306, 134499821,
196315803, 186381054, 124578934, 150084553, 128031351, 118843410,
107373060, 159565443, 173624887, 171292628, 198074235, 139074061,
178690578, 144431052, 174262641, 176783005, 182214864, 162289361,
192966929, 192033046, 169332843, 181580535, 164864073, 118444059,
195496893, 153773183, 167266131, 130108623, 158802128, 180432893,
144562140, 147978945, 142337360, 158506327, 104399819, 132635916,
168734194, 136567839, 101281912, 120281622, 195003330, 112236091,
185875592, 101959081, 122415367, 194990954, 148881099, 175891989,
108115811, 163538891, 163394029, 123722049, 184837522, 142362091,
100834097, 156679171, 100841679, 157022331, 178971071, 102928884,
189701309, 195339954, 124415335, 106062584, 139214524, 133864640,
134324406, 157317477, 155340540, 144810061, 177612569, 108474646,
114329765, 143900008, 138265211, 145210162, 136643111, 197987319,
102751191, 144121361, 169620456, 193602633, 161023559, 162140467,
102901215, 167964187, 135746835, 187317233, 110047459, 163339773,
124770449, 118885134, 141536376, 100915375, 164267438, 145016622,
113937193, 106748706, 128815954, 164819775, 119220771, 102367432,
189062690, 170911791, 194127762, 112245117, 123546771, 115640433,
135772061, 166615646, 174474627, 130562291, 133320309, 153340551,
138417181, 194605321, 150142632, 180008795, 151813296, 175497284,
167018836, 157425342, 150169942, 131069156, 134310662, 160434122,
105213831, 158797111, 150754540, 163290657, 102484886, 148697402,
187203725, 198692811, 149360627, 140384233, 128749423, 132178578,
177507355, 171857043, 178737969, 134023369, 102911446, 196144864,
197697194, 134527467, 144296030, 189437192, 154052665, 188907106,
162062575, 150993037, 199766583, 167936112, 181374511, 104971506,
115378374, 135795558, 167972129, 135876446, 130937572, 103221320,
124605656, 161129971, 131027586, 191128460, 143251843, 143269155,
129284585, 173495971, 150425653, 199302112, 118494723, 121323805,
116549802, 190991967, 168151180, 122483192, 151273721, 199792134,
133106764, 121874844, 126215985, 112167639, 167793529, 182985195,
185453921, 106957880, 158685312, 132775454, 133229161, 198905318,
190537253, 191582222, 192325972, 178133427, 181825606, 148823337,
160719681, 101448145, 131983362, 137910767, 112550175, 128826351,
183649210, 135725874, 110356573, 189469487, 154446940, 118175923,
106093708, 128146501, 185742532, 149692127, 164624247, 183221076,
154737505, 168198834, 156410354, 158027261, 125228550, 131543250,
139591848, 191898263, 104987591, 115406321, 103542638, 190012837,
142615518, 178773183, 175862355, 117537850, 169565995, 170028011,
158412588, 170150030, 117025916, 174630208, 142412449, 112839238,
105257725, 114737141, 123102301, 172563968, 130555358, 132628403,
183638157, 168682846, 143304568, 105994018, 170010719, 152092970,
117799058, 132164175, 179868116, 158654714, 177489647, 116547948,
183121404, 131836079, 184431405, 157311793, 149677763, 173989893,
102277656, 107058530, 140837477, 152640947, 143507039, 152145247,
101683884, 107090870, 161471944, 137225650, 128231458, 172995869,
173831689, 171268519, 139042297, 111072135, 107569780, 137262545,
181410950, 138270388, 198736451, 162848201, 180468288, 120582913,
153390138, 135649144, 130040157, 106509887, 192671541, 174507066,
186888783, 143805558, 135011967, 145862340, 180595327, 124727843,
182925939, 157715840, 136885940, 198993925, 152416883, 178793572,
179679516, 154076673, 192703125, 164187609, 162190243, 104699348,
159891990, 160012977, 174692145, 132970421, 167781726, 115178506,
153008552, 155999794, 102099694, 155431545, 127458567, 104403686,
168042864, 184045128, 181182309, 179349696, 127218364, 192935516,
120298724, 169583299, 148193297, 183358034, 159023227, 105261254,
121144370, 184359584, 194433836, 138388317, 175184116, 108817112,
151279233, 137457721, 193398208, 119005406, 132929377, 175306906,
160741530, 149976826, 147124407, 176881724, 186734216, 185881509,
191334220, 175930947, 117385515, 193408089, 157124410, 163472089,
    131949128, 180783576, 131158294, 100549708, 191802336, 165960770,
    170927599, 101052702, 181508688, 197828549, 143403726, 142729262,
    110348701, 139928688, 153550062, 106151434, 130786653, 196085995,
    100587149, 139141652, 106530207, 100852656, 124074703, 166073660,
    153338052, 163766757, 120188394, 197277047, 122215363, 138511354,
    183463624, 161985542, 159938719, 133367482, 104220974, 149956672,
    170250544, 164232439, 157506869, 159133019, 137469191, 142980999,
    134242305, 150172665, 121209241, 145596259, 160554427, 159095199,
    168243130, 184279693, 171132070, 121049823, 123819574, 171759855,
    119501864, 163094029, 175943631, 194450091, 191506160, 149228764,
    132319212, 197034460, 193584259, 126727638, 168143633, 109856853,
    127860243, 132141052, 133076065, 188414958, 158718197, 107124299,
    159592267, 181172796, 144388537, 196763139, 127431422, 179531145,
    100064922, 112650013, 132686230, 121550837,
  }
\cs_new:Npn \__fp_trig_large:ww #1, #2#3#4#5#6\__fp_sep:
  {
    \exp_after:wN \__fp_trig_large_auxi:w
    \int_value:w \__fp_int_eval:w (#1 - 4) / 8 \exp_after:wN ,
    \int_value:w #1 , \__fp_sep:
    {#2}{#3}{#4}{#5} \__fp_sep:
  }
\cs_new:Npn \__fp_trig_large_auxi:w #1, #2,
  {
    \exp_after:wN \exp_after:wN
    \exp_after:wN \__fp_trig_large_auxii:w
    \cs:w
      use_none:n \prg_replicate:nn { #2 - #1 * 8 } { n }
      \exp_after:wN
    \cs_end:
    \int_value:w
    \__kernel_intarray_item:Nn \c__fp_trig_intarray
      { \__fp_int_eval:w #1 + 1 \scan_stop: }
    \exp_after:wN \__fp_trig_large_auxiii:w \int_value:w
    \__kernel_intarray_item:Nn \c__fp_trig_intarray
      { \__fp_int_eval:w #1 + 2 \scan_stop: }
    \exp_after:wN \__fp_trig_large_auxiii:w \int_value:w
    \__kernel_intarray_item:Nn \c__fp_trig_intarray
      { \__fp_int_eval:w #1 + 3 \scan_stop: }
    \exp_after:wN \__fp_trig_large_auxiii:w \int_value:w
    \__kernel_intarray_item:Nn \c__fp_trig_intarray
      { \__fp_int_eval:w #1 + 4 \scan_stop: }
    \exp_after:wN \__fp_trig_large_auxiii:w \int_value:w
    \__kernel_intarray_item:Nn \c__fp_trig_intarray
      { \__fp_int_eval:w #1 + 5 \scan_stop: }
    \exp_after:wN \__fp_trig_large_auxiii:w \int_value:w
    \__kernel_intarray_item:Nn \c__fp_trig_intarray
      { \__fp_int_eval:w #1 + 6 \scan_stop: }
    \exp_after:wN \__fp_trig_large_auxiii:w \int_value:w
    \__kernel_intarray_item:Nn \c__fp_trig_intarray
      { \__fp_int_eval:w #1 + 7 \scan_stop: }
    \exp_after:wN \__fp_trig_large_auxiii:w \int_value:w
    \__kernel_intarray_item:Nn \c__fp_trig_intarray
      { \__fp_int_eval:w #1 + 8 \scan_stop: }
    \exp_after:wN \__fp_trig_large_auxiii:w \int_value:w
    \__kernel_intarray_item:Nn \c__fp_trig_intarray
      { \__fp_int_eval:w #1 + 9 \scan_stop: }
    \exp_stop_f:
  }
\cs_new:Npn \__fp_trig_large_auxii:w
  {
    \__fp_pack_twice_four:wNNNNNNNN \__fp_pack_twice_four:wNNNNNNNN
    \__fp_pack_twice_four:wNNNNNNNN \__fp_pack_twice_four:wNNNNNNNN
    \__fp_pack_twice_four:wNNNNNNNN \__fp_pack_twice_four:wNNNNNNNN
    \__fp_pack_twice_four:wNNNNNNNN \__fp_pack_twice_four:wNNNNNNNN
    \__fp_trig_large_auxv:www \__fp_sep:
  }
\cs_new:Npn \__fp_trig_large_auxiii:w 1 { \exp_stop_f: }
\cs_new:Npn \__fp_trig_large_auxv:www #1\__fp_sep: #2\__fp_sep: #3\__fp_sep:
  {
    \exp_after:wN \__fp_use_i_until_s:nw
    \exp_after:wN \__fp_trig_large_auxvii:w
    \int_value:w \__fp_int_eval:w \c__fp_leading_shift_int
      \prg_replicate:nn { 13 }
        { \__fp_trig_large_auxvi:wnnnnnnnn }
      + \c__fp_trailing_shift_int - \c__fp_middle_shift_int
      \__fp_use_i_until_s:nw
      \__fp_sep: #3 #1 \__fp_sep: \__fp_sep:
  }
\cs_new:Npn \__fp_trig_large_auxvi:wnnnnnnnn #1\__fp_sep: #2#3#4#5#6#7#8#9
  {
    \exp_after:wN \__fp_trig_large_pack:NNNNNw
    \int_value:w \__fp_int_eval:w \c__fp_middle_shift_int
      + #2*#9 + #3*#8 + #4*#7 + #5*#6
      #1\__fp_sep: {#2}{#3}{#4}{#5} {#7}{#8}{#9}
  }
\cs_new:Npn \__fp_trig_large_pack:NNNNNw #1#2#3#4#5#6\__fp_sep:
  { + #1#2#3#4#5 \__fp_sep: #6 }
\cs_new:Npn \__fp_trig_large_auxvii:w #1#2#3
  {
    \exp_after:wN \__fp_trig_large_auxviii:ww
    \int_value:w \__fp_int_eval:w (#1#2#3 - 62) / 125 \__fp_sep:
```

```
    #1#2#3
  }
\cs_new:Npn \__fp_trig_large_auxviii:ww #1\__fp_sep:
  {
    + #1
    \if_int_odd:w #1 \exp_stop_f:
      \exp_after:wN \__fp_trig_large_auxix:Nw
      \exp_after:wN -
    \else:
      \exp_after:wN \__fp_trig_large_auxix:Nw
      \exp_after:wN +
    \fi:
  }
\cs_new:Npn \__fp_trig_large_auxix:Nw
  {
    \exp_after:wN \__fp_use_i_until_s:nw
    \exp_after:wN \__fp_trig_large_auxxi:w
    \int_value:w \__fp_int_eval:w \c__fp_leading_shift_int
      \prg_replicate:nn { 13 }
        { \__fp_trig_large_auxx:wNNNNN }
      + \c__fp_trailing_shift_int - \c__fp_middle_shift_int
      \__fp_sep:
  }
\cs_new:Npn \__fp_trig_large_auxx:wNNNNN #1\__fp_sep: #2 #3#4#5#6
  {
    \exp_after:wN \__fp_trig_large_pack:NNNNNw
    \int_value:w \__fp_int_eval:w \c__fp_middle_shift_int
      #2 8 * #3#4#5#6
      #1\__fp_sep: #2
  }
\cs_new:Npn \__fp_trig_large_auxxi:w #1\__fp_sep:
  {
    \exp_after:wN \__fp_ep_mul_raw:wwwwN
    \int_value:w \__fp_int_eval:w 0 \__fp_ep_to_ep_loop:N #1 \__fp_sep: \__fp_sep: !
    0,{7853}{9816}{3397}{4483}{0961}{5661}\__fp_sep:
    \__fp_trig_small:ww
  }
\cs_new:Npn \__fp_sin_series_o:NNwwww #1#2#3. #4\__fp_sep:
  {
    \__fp_fixed_mul:wwn #4\__fp_sep: #4\__fp_sep:
    {
      \exp_after:wN \__fp_sin_series_aux_o:NNnwww
      \exp_after:wN #1
      \int_value:w
        \if_int_odd:w \__fp_int_eval:w (#3 + 2) / 4 \__fp_int_eval_end:
          #2
        \else:
          \if_meaning:w #2 0 2 \else: 0 \fi:
        \fi:
      {#3}
    }
  }
\cs_new:Npn \__fp_sin_series_aux_o:NNnwww #1#2#3 #4\__fp_sep: #5,#6\__fp_sep:
  {
    \if_int_odd:w \__fp_int_eval:w #3 / 2 \__fp_int_eval_end:
      \exp_after:wN \use_i:nn
    \else:
      \exp_after:wN \use_ii:nn
    \fi:
    { % 1/18!
      \__fp_fixed_mul_sub_back:wwwn {0000}{0000}{0000}{0001}{5619}{2070}\__fp_sep:
                                  #4\__fp_sep:
                                  {0000}{0000}{0000}{0477}{9477}{3324}\__fp_sep:
      \__fp_fixed_mul_sub_back:wwwn #4\__fp_sep:
                                  {0000}{0000}{0011}{4707}{4559}{7730}\__fp_sep:
      \__fp_fixed_mul_sub_back:wwwn #4\__fp_sep:
                                  {0000}{0000}{2087}{6756}{9878}{6810}\__fp_sep:
      \__fp_fixed_mul_sub_back:wwwn #4\__fp_sep:
                                  {0000}{0027}{5573}{1922}{3985}{8907}\__fp_sep:
      \__fp_fixed_mul_sub_back:wwwn #4\__fp_sep:
                                  {0000}{2480}{1587}{3015}{8730}{1587}\__fp_sep:
      \__fp_fixed_mul_sub_back:wwwn #4\__fp_sep:
                                  {0013}{8888}{8888}{8888}{8888}{8889}\__fp_sep:
      \__fp_fixed_mul_sub_back:wwwn #4\__fp_sep:
                                  {0416}{6666}{6666}{6666}{6666}{6667}\__fp_sep:
      \__fp_fixed_mul_sub_back:wwwn #4\__fp_sep:
                                  {5000}{0000}{0000}{0000}{0000}{0000}\__fp_sep:
      \__fp_fixed_mul_sub_back:wwwn#4\__fp_sep:
                                  {10000}{0000}{0000}{0000}{0000}{0000}\__fp_sep:
      { \__fp_fixed_continue:wn 0, }
    }
    { % 1/17!
      \__fp_fixed_mul_sub_back:wwwn {0000}{0000}{0000}{0028}{1145}{7254}\__fp_sep:
                                  #4\__fp_sep:
                                  {0000}{0000}{0000}{7647}{1637}{3182}\__fp_sep:
      \__fp_fixed_mul_sub_back:wwwn #4\__fp_sep:
                                  {0000}{0000}{0160}{5904}{3836}{8216}\__fp_sep:
      \__fp_fixed_mul_sub_back:wwwn #4\__fp_sep:
                                  {0000}{0002}{5052}{1083}{8544}{1719}\__fp_sep:
      \__fp_fixed_mul_sub_back:wwwn #4\__fp_sep:
                                  {0000}{0275}{5731}{9223}{9858}{9065}\__fp_sep:
      \__fp_fixed_mul_sub_back:wwwn #4\__fp_sep:
```

```
                                  {0001}{9841}{2698}{4126}{9841}{2698}\__fp_sep:
      \__fp_fixed_mul_sub_back:wwwn #4\__fp_sep:
                                  {0083}{3333}{3333}{3333}{3333}{3333}\__fp_sep:
      \__fp_fixed_mul_sub_back:wwwn #4\__fp_sep:
                                  {1666}{6666}{6666}{6666}{6666}{6667}\__fp_sep:
      \__fp_fixed_mul_sub_back:wwwn#4\__fp_sep:
                                  {10000}{0000}{0000}{0000}{0000}{0000}\__fp_sep:
      { \__fp_ep_mul:wwwwn 0, } #5,#6\__fp_sep:
    }
    {
      \exp_after:wN \__fp_sanitize:Nw
      \exp_after:wN #2
      \int_value:w \__fp_int_eval:w #1
    }
    #2
  }
\cs_new:Npn \__fp_tan_series_o:NNwwww #1#2#3. #4\__fp_sep:
  {
    \__fp_fixed_mul:wwn #4\__fp_sep: #4\__fp_sep:
    {
      \exp_after:wN \__fp_tan_series_aux_o:Nnwww
      \int_value:w
        \if_int_odd:w \__fp_int_eval:w #3 / 2 \__fp_int_eval_end:
          \exp_after:wN \reverse_if:N
        \fi:
        \if_meaning:w #1#2 2 \else: 0 \fi:
      {#3}
    }
  }
\cs_new:Npn \__fp_tan_series_aux_o:Nnwww #1 #2 #3\__fp_sep: #4,#5\__fp_sep:
  {
    \__fp_fixed_mul_sub_back:wwwn {0000}{0000}{1527}{3493}{0856}{7059}\__fp_sep:
                                #3\__fp_sep:
                                {0000}{0159}{6080}{0274}{5257}{6472}\__fp_sep:
    \__fp_fixed_mul_sub_back:wwwn #3\__fp_sep:
                                {0002}{4571}{2320}{0157}{2558}{8481}\__fp_sep:
    \__fp_fixed_mul_sub_back:wwwn #3\__fp_sep:
                                {0115}{5830}{7533}{5397}{3168}{2147}\__fp_sep:
    \__fp_fixed_mul_sub_back:wwwn #3\__fp_sep:
                                {1929}{8245}{6140}{3508}{7719}{2982}\__fp_sep:
    \__fp_fixed_mul_sub_back:wwwn #3\__fp_sep:
                                {10000}{0000}{0000}{0000}{0000}{0000}\__fp_sep:
    { \__fp_ep_mul:wwwwn 0, } #4,#5\__fp_sep:
    {
      \__fp_fixed_mul_sub_back:wwwn {0000}{0007}{0258}{0681}{9408}{4706}\__fp_sep:
                                  #3\__fp_sep:
```

```
                                  {0000}{2343}{7175}{1399}{6151}{7670}\__fp_sep:
      \__fp_fixed_mul_sub_back:wwwn #3\__fp_sep:
                                  {0019}{2638}{4588}{9232}{8861}{3691}\__fp_sep:
      \__fp_fixed_mul_sub_back:wwwn #3\__fp_sep:
                                  {0536}{6357}{0691}{4344}{6852}{4252}\__fp_sep:
      \__fp_fixed_mul_sub_back:wwwn #3\__fp_sep:
                                  {5263}{1578}{9473}{6842}{1052}{6315}\__fp_sep:
      \__fp_fixed_mul_sub_back:wwwn#3\__fp_sep:
                                  {10000}{0000}{0000}{0000}{0000}{0000}\__fp_sep:
      {
        \reverse_if:N \if_int_odd:w
            \__fp_int_eval:w (#2 - 1) / 2 \__fp_int_eval_end:
          \exp_after:wN \__fp_reverse_args:Nww
        \fi:
        \__fp_ep_div:wwwwn 0,
      }
    }
    {
      \exp_after:wN \__fp_sanitize:Nw
      \exp_after:wN #1
      \int_value:w \__fp_int_eval:w \__fp_ep_to_float_o:wwN
    }
    #1
  }
\cs_new:Npn \__fp_atan_o:Nw #1
  {
    \__fp_parse_function_one_two:nnw
      { #1 { atan } { atand } }
      { \__fp_atan_default:w \__fp_atanii_o:Nww #1 }
  }
\cs_new:Npn \__fp_acot_o:Nw #1
  {
    \__fp_parse_function_one_two:nnw
      { #1 { acot } { acotd } }
      { \__fp_atan_default:w \__fp_acotii_o:Nww #1 }
  }
\cs_new:Npe \__fp_atan_default:w #1#2#3 @ { #1 #2 #3 \c_one_fp @ }
\cs_new:Npn \__fp_atanii_o:Nww
    #1 \s__fp \__fp_chk:w #2#3#4\__fp_sep: \s__fp \__fp_chk:w #5 #6 @
  {
    \if_meaning:w 3 #2 \__fp_case_return_i_o:ww \fi:
    \if_meaning:w 3 #5 \__fp_case_return_ii_o:ww \fi:
    \if_case:w
      \if_meaning:w #2 #5
        \if_meaning:w 1 #2 10 \else: 0 \fi:
      \else:
        \if_int_compare:w #2 > #5 \exp_stop_f: 1 \else: 2 \fi:
      \fi:
      \exp_stop_f:
         \__fp_case_return:nw { \__fp_atan_inf_o:NNNw #1 #3 2 }
    \or: \__fp_case_return:nw { \__fp_atan_inf_o:NNNw #1 #3 4 }
    \or: \__fp_case_return:nw { \__fp_atan_inf_o:NNNw #1 #3 0 }
    \fi:
    \__fp_atan_normal_o:NNnwNnw #1
    \s__fp \__fp_chk:w #2#3#4\__fp_sep:
    \s__fp \__fp_chk:w #5 #6
  }
\cs_new:Npn \__fp_acotii_o:Nww #1#2\__fp_sep: #3\__fp_sep:
  { \__fp_atanii_o:Nww #1#3\__fp_sep: #2\__fp_sep: }
\cs_new:Npn \__fp_atan_inf_o:NNNw #1#2#3 \s__fp \__fp_chk:w #4#5#6\__fp_sep:
  {
    \exp_after:wN \__fp_atan_combine_o:NwwwwwN
    \exp_after:wN #2
    \int_value:w \__fp_int_eval:w
      \if_meaning:w 2 #5 7 - \fi: #3 \exp_after:wN \__fp_sep:
    \c__fp_one_fixed_tl
    {0000}{0000}{0000}{0000}{0000}{0000}\__fp_sep:
    0,{0000}{0000}{0000}{0000}{0000}{0000}\__fp_sep: #1
  }
\cs_new_protected:Npn \__fp_atan_normal_o:NNnwNnw
    #1 \s__fp \__fp_chk:w 1#2#3#4\__fp_sep: \s__fp \__fp_chk:w 1#5#6#7\__fp_sep:
  {
    \__fp_atan_test_o:NwwNwwN
      #2 #3, #4{0000}{0000}\__fp_sep:
      #5 #6, #7{0000}{0000}\__fp_sep: #1
  }
\cs_new:Npn \__fp_atan_test_o:NwwNwwN #1#2,#3\__fp_sep: #4#5,#6\__fp_sep:
  {
    \exp_after:wN \__fp_atan_combine_o:NwwwwwN
    \exp_after:wN #1
    \int_value:w \__fp_int_eval:w
      \if_meaning:w 2 #4
        7 - \__fp_int_eval:w
      \fi:
      \if_int_compare:w
          \__fp_ep_compare:wwww #2,#3\__fp_sep: #5,#6\__fp_sep: > \c_zero_int
        3 -
        \exp_after:wN \__fp_reverse_args:Nww
      \fi:
      \__fp_atan_div:wnwwnw #2,#3\__fp_sep: #5,#6\__fp_sep:
  }
\cs_new:Npn \__fp_atan_div:wnwwnw #1,#2#3\__fp_sep: #4,#5#6\__fp_sep:
  {
    \if_int_compare:w
      \__fp_int_eval:w 41421 * #5 < #2 000
        \if_case:w \__fp_int_eval:w #4 - #1 \__fp_int_eval_end:
          00 \or: 0 \fi:
      \exp_stop_f:
      \exp_after:wN \__fp_atan_near:wwwn
    \fi:
    0
    \__fp_ep_div:wwwwn #1,{#2}#3\__fp_sep: #4,{#5}#6\__fp_sep:
    \__fp_atan_auxi:ww
  }
\cs_new:Npn \__fp_atan_near:wwwn
    0 \__fp_ep_div:wwwwn #1,#2\__fp_sep: #3,
  {
    1
    \__fp_ep_to_fixed:wwn #1 - #3, #2\__fp_sep:
    \__fp_atan_near_aux:wwn
  }
\cs_new:Npn \__fp_atan_near_aux:wwn #1\__fp_sep: #2\__fp_sep:
  {
    \__fp_fixed_add:wwn #1\__fp_sep: #2\__fp_sep:
    { \__fp_fixed_sub:wwn #2\__fp_sep: #1\__fp_sep: { \__fp_ep_div:wwwwn 0, } 0, }
  }
\cs_new:Npn \__fp_atan_auxi:ww #1,#2\__fp_sep:
  { \__fp_ep_to_fixed:wwn #1,#2\__fp_sep: \__fp_atan_auxii:w #1,#2\__fp_sep: }
\cs_new:Npn \__fp_atan_auxii:w #1\__fp_sep:
  {
    \__fp_fixed_mul:wwn #1\__fp_sep: #1\__fp_sep:
    {
      \__fp_atan_Taylor_loop:www 39 \__fp_sep:
        {0000}{0000}{0000}{0000}{0000}{0000} \__fp_sep:
    }
    ! #1\__fp_sep:
  }
\cs_new:Npn \__fp_atan_Taylor_loop:www #1\__fp_sep: #2\__fp_sep: #3\__fp_sep:
  {
    \if_int_compare:w #1 = - \c_one_int
      \__fp_atan_Taylor_break:w
    \fi:
    \exp_after:wN \__fp_fixed_div_int:wwN \c__fp_one_fixed_tl #1\__fp_sep:
    \__fp_rrot:www \__fp_fixed_mul_sub_back:wwwn #2\__fp_sep: #3\__fp_sep:
    {
      \exp_after:wN \__fp_atan_Taylor_loop:www
      \int_value:w \__fp_int_eval:w #1 - 2 \__fp_sep:
    }
    #3\__fp_sep:
  }
\cs_new:Npn \__fp_atan_Taylor_break:w
    \fi: #1 \__fp_fixed_mul_sub_back:wwwn #2\__fp_sep: #3 !
  { \fi: \__fp_sep: #2 \__fp_sep: }
\cs_new:Npn \__fp_atan_combine_o:NwwwwwN
    #1 #2\__fp_sep: #3\__fp_sep: #4\__fp_sep: #5,#6\__fp_sep: #7
  {
    \exp_after:wN \__fp_sanitize:Nw
    \exp_after:wN #1
    \int_value:w \__fp_int_eval:w
      \if_meaning:w 0 #2
        \exp_after:wN \use_i:nn
      \else:
        \exp_after:wN \use_ii:nn
      \fi:
      { #5 \__fp_fixed_mul:wwn #3\__fp_sep: #6\__fp_sep: }
      {
        \__fp_fixed_mul:wwn #3\__fp_sep: #4\__fp_sep:
        {
          \exp_after:wN \__fp_atan_combine_aux:ww
          \int_value:w \__fp_int_eval:w #2 / 2 \__fp_sep: #2\__fp_sep:
        }
      }
      { #7 \__fp_fixed_to_float_o:wN \__fp_fixed_to_float_rad_o:wN }
      #1
  }
\cs_new:Npn \__fp_atan_combine_aux:ww #1\__fp_sep: #2\__fp_sep:
  {
    \__fp_fixed_mul_short:wwn
      {7853}{9816}{3397}{4483}{0961}{5661}\__fp_sep:
      {#1}{0000}{0000}\__fp_sep:
    {
      \if_int_odd:w #2 \exp_stop_f:
        \exp_after:wN \__fp_fixed_sub:wwn
      \else:
        \exp_after:wN \__fp_fixed_add:wwn
      \fi:
    }
  }
\cs_new:Npn \__fp_asin_o:w #1 \s__fp \__fp_chk:w #2#3\__fp_sep: @
  {
    \if_case:w #2 \exp_stop_f:
      \__fp_case_return_same_o:w
    \or:
      \__fp_case_use:nw
```

```
        { \__fp_asin_normal_o:NfwNnnnnw #1 { #1 { asin } { asind } } }
    \or:
      \__fp_case_use:nw
        { \__fp_invalid_operation_o:fw { #1 { asin } { asind } } }
    \else:
      \__fp_case_return_same_o:w
    \fi:
    \s__fp \__fp_chk:w #2 #3\__fp_sep:
  }
\cs_new:Npn \__fp_acos_o:w #1 \s__fp \__fp_chk:w #2#3\__fp_sep: @
  {
    \if_case:w #2 \exp_stop_f:
      \__fp_case_use:nw { \__fp_atan_inf_o:NNNw #1 0 4 }
    \or:
      \__fp_case_use:nw
        {
          \__fp_asin_normal_o:NfwNnnnnw #1 { #1 { acos } { acosd } }
            \__fp_reverse_args:Nww
        }
    \or:
      \__fp_case_use:nw
        { \__fp_invalid_operation_o:fw { #1 { acos } { acosd } } }
    \else:
      \__fp_case_return_same_o:w
    \fi:
    \s__fp \__fp_chk:w #2 #3\__fp_sep:
  }
\cs_new:Npn \__fp_asin_normal_o:NfwNnnnnw
    #1#2#3 \s__fp \__fp_chk:w 1#4#5#6#7#8#9\__fp_sep:
  {
    \if_int_compare:w #5 < \c_one_int
      \exp_after:wN \__fp_use_none_until_s:w
    \fi:
    \if_int_compare:w \__fp_int_eval:w #5 + #6#7 + #8#9 = 1000 0001 ~
      \exp_after:wN \__fp_use_none_until_s:w
    \fi:
    \__fp_use_i:ww
    \__fp_invalid_operation_o:fw {#2}
      \s__fp \__fp_chk:w 1#4{#5}{#6}{#7}{#8}{#9}\__fp_sep:
    \__fp_asin_auxi_o:NnNww
      #1 {#3} #4 #5,{#6}{#7}{#8}{#9}{0000}{0000}\__fp_sep:
  }
\cs_new:Npn \__fp_asin_auxi_o:NnNww #1#2#3#4,#5\__fp_sep:
  {
    \__fp_ep_to_fixed:wwn #4,#5\__fp_sep:
    \__fp_asin_isqrt:wn
    \__fp_ep_mul:wwwwn #4,#5\__fp_sep:
    \__fp_ep_to_ep:wwN
    \__fp_fixed_continue:wn
    { #2 \__fp_atan_test_o:NwwNwwN #3 }
    0 1,{1000}{0000}{0000}{0000}{0000}{0000}\__fp_sep: #1
  }
\cs_new:Npn \__fp_asin_isqrt:wn #1\__fp_sep:
  {
    \exp_after:wN \__fp_fixed_sub:wwn \c__fp_one_fixed_tl #1\__fp_sep:
    {
      \__fp_fixed_add_one:wN #1\__fp_sep:
      \__fp_fixed_continue:wn { \__fp_ep_mul:wwwwn 0, } 0,
    }
    \__fp_ep_isqrt:wwn
  }
\cs_new:Npn \__fp_acsc_o:w #1 \s__fp \__fp_chk:w #2#3#4\__fp_sep: @
  {
    \if_case:w \if_meaning:w 2 #2 #3 \fi: #2 \exp_stop_f:
           \__fp_case_use:nw
             { \__fp_invalid_operation_o:fw { #1 { acsc } { acscd } } }
    \or:   \__fp_case_use:nw
             { \__fp_acsc_normal_o:NfwNnw #1 { #1 { acsc } { acscd } } }
    \or:   \__fp_case_return_o:Nw \c_zero_fp
    \or:   \__fp_case_return_same_o:w
    \else: \__fp_case_return_o:Nw \c_minus_zero_fp
    \fi:
    \s__fp \__fp_chk:w #2 #3 #4\__fp_sep:
  }
\cs_new:Npn \__fp_asec_o:w #1 \s__fp \__fp_chk:w #2#3\__fp_sep: @
  {
    \if_case:w #2 \exp_stop_f:
      \__fp_case_use:nw
        { \__fp_invalid_operation_o:fw { #1 { asec } { asecd } } }
    \or:
      \__fp_case_use:nw
        {
          \__fp_acsc_normal_o:NfwNnw #1 { #1 { asec } { asecd } }
            \__fp_reverse_args:Nww
        }
    \or:   \__fp_case_use:nw { \__fp_atan_inf_o:NNNw #1 0 4 }
    \else: \__fp_case_return_same_o:w
    \fi:
    \s__fp \__fp_chk:w #2 #3\__fp_sep:
  }
\cs_new:Npn \__fp_acsc_normal_o:NfwNnw #1#2#3 \s__fp \__fp_chk:w 1#4#5#6\__fp_sep:
  {
    \int_compare:nNnTF {#5} < 1
      {
        \__fp_invalid_operation_o:fw {#2}
          \s__fp \__fp_chk:w 1#4{#5}#6\__fp_sep:
      }
      {
        \__fp_ep_div:wwwwn
          1,{1000}{0000}{0000}{0000}{0000}{0000}\__fp_sep:
          #5,#6{0000}{0000}\__fp_sep:
        { \__fp_asin_auxi_o:NnNww #1 {#3} #4 }
      }
  }
%% File: l3fp-convert.dtx
\cs_new:Npn \__fp_tuple_convert:Nw #1 \s__fp_tuple \__fp_tuple_chk:w #2 \__fp_sep:
  {
    \int_case:nnF { \__fp_array_count:n {#2} }
      {
        { 0 } { ( ) }
        { 1 } { \__fp_tuple_convert_end:w @ { #1 #2 , } }
      }
      {
        \__fp_tuple_convert_loop:nNw { } #1
          #2 { ? \__fp_tuple_convert_end:w } \__fp_sep:
          @ { \use_none:nn }
      }
  }
\cs_new:Npn \__fp_tuple_convert_loop:nNw #1#2#3#4\__fp_sep: #5 @ #6
  {
    \use_none:n #3
    \exp_args:Nf \__fp_tuple_convert_loop:nNw { #2 #3#4 \__fp_sep: } #2 #5
      @ { #6 , ~ #1 }
  }
\cs_new:Npn \__fp_tuple_convert_end:w #1 @ #2
  { \exp_after:wN ( \exp:w \exp_end_continue_f:w #2 ) }
\cs_new:Npn \__fp_trim_zeros:w #1 \__fp_sep:
  {
    \__fp_trim_zeros_loop:w #1 \__fp_sep:
      \__fp_trim_zeros_loop:w 0\__fp_sep:
      \__fp_trim_zeros_dot:w .\__fp_sep:
      \s__fp_stop
  }
\cs_new:Npn \__fp_trim_zeros_loop:w #1 0\__fp_sep: #2 { #2 #1 \__fp_sep: #2 }
\cs_new:Npn \__fp_trim_zeros_dot:w #1 .\__fp_sep:
  { \__fp_trim_zeros_end:w #1 \__fp_sep: }
\cs_new:Npn \__fp_trim_zeros_end:w #1 \__fp_sep: #2 \s__fp_stop { #1 }
\cs_new:Npn \fp_to_scientific:N #1
```

```
  { \exp_after:wN \__fp_to_scientific_dispatch:w #1 }
\cs_generate_variant:Nn \fp_to_scientific:N { c }
\cs_new:Npn \fp_to_scientific:n
  {
    \exp_after:wN \__fp_to_scientific_dispatch:w
    \exp:w \exp_end_continue_f:w \__fp_parse:n
  }
\cs_new:Npn \__fp_to_scientific_dispatch:w #1
  {
    \__fp_change_func_type:NNN
      #1 \__fp_to_scientific:w \__fp_to_scientific_recover:w
    #1
  }
\cs_new:Npn \__fp_to_scientific_recover:w #1 #2 \__fp_sep:
  {
    \__fp_error:nffn { unknown-type } { \tl_to_str:n { #2 \__fp_sep: } } { } { }
    nan
  }
\cs_new:Npn \__fp_tuple_to_scientific:w
  { \__fp_tuple_convert:Nw \__fp_to_scientific_dispatch:w }
\cs_new:Npn \__fp_to_scientific:w \s__fp \__fp_chk:w #1#2
  {
    \if_meaning:w 2 #2 \exp_after:wN - \exp:w \exp_end_continue_f:w \fi:
    \if_case:w #1 \exp_stop_f:
         \__fp_case_return:nw { 0.000000000000000e0 }
    \or: \exp_after:wN \__fp_to_scientific_normal:wnnnnn
    \or:
      \__fp_case_use:nw
        {
          \__fp_invalid_operation:nnw
            { \fp_to_scientific:N \c__fp_overflowing_fp }
            { fp_to_scientific }
        }
    \or:
      \__fp_case_use:nw
        {
          \__fp_invalid_operation:nnw
            { \fp_to_scientific:N \c_zero_fp }
            { fp_to_scientific }
        }
    \fi:
    \s__fp \__fp_chk:w #1 #2
  }
\cs_new:Npn \__fp_to_scientific_normal:wnnnnn
  \s__fp \__fp_chk:w 1 #1 #2 #3#4#5#6 \__fp_sep:
  {
    \exp_after:wN \__fp_to_scientific_normal:wNw
    \exp_after:wN e
    \int_value:w \__fp_int_eval:w #2 - 1
    \__fp_sep: #3 #4 #5 #6 \__fp_sep:
  }
\cs_new:Npn \__fp_to_scientific_normal:wNw #1 \__fp_sep: #2#3\__fp_sep:
  { #2.#3 #1 }
\cs_new:Npn \fp_to_decimal:N #1
  { \exp_after:wN \__fp_to_decimal_dispatch:w #1 }
\cs_generate_variant:Nn \fp_to_decimal:N { c }
\cs_new:Npn \fp_to_decimal:n
  {
    \exp_after:wN \__fp_to_decimal_dispatch:w
    \exp:w \exp_end_continue_f:w \__fp_parse:n
  }
\cs_new:Npn \__fp_to_decimal_dispatch:w #1
  {
    \__fp_change_func_type:NNN
      #1 \__fp_to_decimal:w \__fp_to_decimal_recover:w
    #1
  }
\cs_new:Npn \__fp_to_decimal_recover:w #1 #2 \__fp_sep:
  {
    \__fp_error:nffn { unknown-type } { \tl_to_str:n { #2 \__fp_sep: } } { } { }
    nan
  }
\cs_new:Npn \__fp_tuple_to_decimal:w
  { \__fp_tuple_convert:Nw \__fp_to_decimal_dispatch:w }
\cs_new:Npn \__fp_to_decimal:w \s__fp \__fp_chk:w #1#2
  {
    \if_meaning:w 2 #2 \exp_after:wN - \exp:w \exp_end_continue_f:w \fi:
    \if_case:w #1 \exp_stop_f:
         \__fp_case_return:nw { 0 }
    \or: \exp_after:wN \__fp_to_decimal_normal:wnnnnn
    \or:
      \__fp_case_use:nw
        {
          \__fp_invalid_operation:nnw
            { \fp_to_decimal:N \c__fp_overflowing_fp }
            { fp_to_decimal }
        }
    \or:
      \__fp_case_use:nw
        {
          \__fp_invalid_operation:nnw
            { 0 }
            { fp_to_decimal }
        }
    \fi:
    \s__fp \__fp_chk:w #1 #2
  }
\cs_new:Npn \__fp_to_decimal_normal:wnnnnn
    \s__fp \__fp_chk:w 1 #1 #2 #3#4#5#6 \__fp_sep:
  {
    \int_compare:nNnTF {#2} > 0
      {
        \int_compare:nNnTF {#2} < \c__fp_prec_int
          {
            \__fp_decimate:nNnnnn { \c__fp_prec_int - #2 }
              \__fp_to_decimal_large:Nnnw
          }
          {
            \exp_after:wN \exp_after:wN
            \exp_after:wN \__fp_to_decimal_huge:wnnnn
            \prg_replicate:nn { #2 - \c__fp_prec_int } { 0 } \__fp_sep:
          }
        {#3} {#4} {#5} {#6}
      }
      {
        \exp_after:wN \__fp_trim_zeros:w
        \exp_after:wN 0
        \exp_after:wN .
        \exp:w \exp_end_continue_f:w \prg_replicate:nn { - #2 } { 0 }
        #3#4#5#6 \__fp_sep:
      }
  }
\cs_new:Npn \__fp_to_decimal_large:Nnnw #1#2#3#4\__fp_sep:
  {
    \exp_after:wN \__fp_trim_zeros:w \int_value:w
      \if_int_compare:w #2 > \c_zero_int
        #2
      \fi:
      \exp_stop_f:
      #3.#4 \__fp_sep:
  }
\cs_new:Npn \__fp_to_decimal_huge:wnnnn #1\__fp_sep: #2#3#4#5 { #2#3#4#5 #1 }
\cs_new:Npn \fp_to_tl:N #1 { \exp_after:wN \__fp_to_tl_dispatch:w #1 }
\cs_generate_variant:Nn \fp_to_tl:N { c }
\cs_new:Npn \fp_to_tl:n
  {
    \exp_after:wN \__fp_to_tl_dispatch:w
    \exp:w \exp_end_continue_f:w \__fp_parse:n
  }
\cs_new:Npn \__fp_to_tl_dispatch:w #1
  { \__fp_change_func_type:NNN #1 \__fp_to_tl:w \__fp_to_tl_recover:w #1 }
\cs_new:Npn \__fp_to_tl_recover:w #1 #2 \__fp_sep:
  {
    \__fp_error:nffn { unknown-type } { \tl_to_str:n { #2 \__fp_sep: } } { } { }
    nan
  }
\cs_new:Npn \__fp_tuple_to_tl:w
  { \__fp_tuple_convert:Nw \__fp_to_tl_dispatch:w }
\cs_new:Npn \__fp_to_tl:w \s__fp \__fp_chk:w #1#2
  {
    \if_meaning:w 2 #2 \exp_after:wN - \exp:w \exp_end_continue_f:w \fi:
    \if_case:w #1 \exp_stop_f:
           \__fp_case_return:nw { 0 }
    \or:   \exp_after:wN \__fp_to_tl_normal:nnnnn
    \or:   \__fp_case_return:nw { inf }
    \else: \__fp_case_return:nw { nan }
    \fi:
  }
\cs_new:Npn \__fp_to_tl_normal:nnnnn #1
  {
    \int_compare:nTF
      { -2 <= #1 <= \c__fp_prec_int }
      { \__fp_to_decimal_normal:wnnnnn }
      { \__fp_to_tl_scientific:wnnnnn }
    \s__fp \__fp_chk:w 1 0 {#1}
  }
\cs_new:Npn \__fp_to_tl_scientific:wnnnnn
  \s__fp \__fp_chk:w 1 #1 #2 #3#4#5#6 \__fp_sep:
  {
    \exp_after:wN \__fp_to_tl_scientific:wNw
    \exp_after:wN e
    \int_value:w \__fp_int_eval:w #2 - 1
    \__fp_sep: #3 #4 #5 #6 \__fp_sep:
  }
\cs_new:Npn \__fp_to_tl_scientific:wNw #1 \__fp_sep: #2#3\__fp_sep:
  { \__fp_trim_zeros:w #2.#3 \__fp_sep: #1 }
\cs_new:Npn \fp_to_dim:N #1
  { \exp_after:wN \__fp_to_dim_dispatch:w #1 }
\cs_generate_variant:Nn \fp_to_dim:N { c }
\cs_new:Npn \fp_to_dim:n
  {
    \exp_after:wN \__fp_to_dim_dispatch:w
    \exp:w \exp_end_continue_f:w \__fp_parse:n
  }
```

```
\cs_new:Npn \__fp_to_dim_dispatch:w #1#2 \__fp_sep:
  {
    \__fp_change_func_type:NNN #1 \__fp_to_dim:w \__fp_to_dim_recover:w
    #1 #2 \__fp_sep:
  }
\cs_new:Npn \__fp_to_dim_recover:w #1
  { \__fp_invalid_operation:nnw { 0pt } { fp_to_dim } }
\cs_new:Npn \__fp_to_dim:w #1 \__fp_sep: { \__fp_to_decimal:w #1 \__fp_sep: pt }
\cs_new:Npn \fp_to_int:N #1 { \exp_after:wN \__fp_to_int_dispatch:w #1 }
\cs_generate_variant:Nn \fp_to_int:N { c }
\cs_new:Npn \fp_to_int:n
  {
    \exp_after:wN \__fp_to_int_dispatch:w
    \exp:w \exp_end_continue_f:w \__fp_parse:n
  }
\cs_new:Npn \__fp_to_int_dispatch:w #1#2 \__fp_sep:
  {
    \__fp_change_func_type:NNN #1 \__fp_to_int:w \__fp_to_int_recover:w
    #1 #2 \__fp_sep:
  }
\cs_new:Npn \__fp_to_int_recover:w #1
  { \__fp_invalid_operation:nnw { 0 } { fp_to_int } }
\cs_new:Npn \__fp_to_int:w #1\__fp_sep:
  {
    \exp_after:wN \__fp_to_decimal:w \exp:w \exp_end_continue_f:w
    \__fp_round:Nwn \__fp_round_to_nearest:NNN #1\__fp_sep: { 0 }
  }
\cs_new:Npn \dim_to_fp:n #1
  {
    \exp_after:wN \__fp_from_dim_test:ww
    \exp_after:wN 0
    \exp_after:wN ,
    \int_value:w \tex_glueexpr:D #1 \__fp_sep:
  }
\cs_new:Npn \__fp_from_dim_test:ww #1, #2
  {
    \if_meaning:w 0 #2
      \__fp_case_return:nw { \exp_after:wN \c_zero_fp }
    \else:
      \exp_after:wN \__fp_from_dim:wNw
      \int_value:w \__fp_int_eval:w #1 - 4
        \if_meaning:w - #2
          \exp_after:wN , \exp_after:wN 2 \int_value:w
        \else:
          \exp_after:wN , \exp_after:wN 0 \int_value:w #2
        \fi:
```

```
    \fi:
  }
\cs_new:Npn \__fp_from_dim:wNw #1,#2#3\__fp_sep:
  {
    \__fp_pack_twice_four:wNNNNNNNN \__fp_from_dim:wNNnnnnnn \__fp_sep:
    #3 000 0000 00 {10}987654321\__fp_sep: #2 {#1}
  }
\cs_new:Npn \__fp_from_dim:wNNnnnnnn #1\__fp_sep: #2#3#4#5#6#7#8#9
  { \__fp_from_dim:wnnnnwNn #1 {#2#300} {0000} \__fp_sep: }
\cs_new:Npn \__fp_from_dim:wnnnnwNn #1\__fp_sep: #2#3#4#5#6\__fp_sep: #7#8
  {
    \__fp_mul_npos_o:Nww #7
      \s__fp \__fp_chk:w 1 #7 {#5} #1 \__fp_sep:
      \s__fp \__fp_chk:w 1 0 {#8} {1525} {8789} {0625} {0000} \__fp_sep:
      \prg_do_nothing:
  }
\cs_new_eq:NN \fp_use:N \fp_to_decimal:N
\cs_generate_variant:Nn \fp_use:N { c }
\cs_new_eq:NN \fp_eval:n \fp_to_decimal:n
\cs_new:Npn \fp_sign:n #1
  { \fp_to_decimal:n { sign \__fp_parse:n {#1} } }
\cs_new:Npn \fp_abs:n #1
  { \fp_to_decimal:n { abs \__fp_parse:n {#1} } }
\cs_new:Npn \fp_max:nn #1#2
  { \fp_to_decimal:n { max ( \__fp_parse:n {#1} , \__fp_parse:n {#2} ) } }
\cs_new:Npn \fp_min:nn #1#2
  { \fp_to_decimal:n { min ( \__fp_parse:n {#1} , \__fp_parse:n {#2} ) } }
\cs_new:Npn \__fp_array_to_clist:n #1
  {
    \tl_if_empty:nF {#1}
      {
        \exp_last_unbraced:Ne \use_ii:nn
          {
            \__fp_array_to_clist_loop:Nw #1 { ? \prg_break: } \__fp_sep:
            \prg_break_point:
          }
      }
  }
\cs_new:Npn \__fp_array_to_clist_loop:Nw #1#2\__fp_sep:
  {
    \use_none:n #1
    , ~
    \exp_not:f { \__fp_to_tl_dispatch:w #1 #2 \__fp_sep: }
    \__fp_array_to_clist_loop:Nw
  }
%% File: l3fp-random.dtx
```

```
\cs_new:Npn \__fp_parse_word_rand:N
  { \__fp_parse_function:NNN \__fp_rand_o:Nw ? }
\cs_new:Npn \__fp_parse_word_randint:N
  { \__fp_parse_function:NNN \__fp_randint_o:Nw ? }
\int_const:Nn \c__kernel_randint_max_int { 131071 }
\cs_new:Npn \__kernel_randint:n #1
  {
    (#1 * \tex_uniformdeviate:D 16384
    + \tex_uniformdeviate:D #1 + 8192 ) / 16384
  }
\cs_new:Npn \__fp_rand_myriads:n #1
  { \__fp_rand_myriads_loop:w #1 \prg_break: X \prg_break_point: \__fp_sep: }
\cs_new:Npn \__fp_rand_myriads_loop:w #1 X
  {
    #1
    \exp_after:wN \__fp_rand_myriads_get:w
    \int_value:w \__fp_int_eval:w 9999 +
      \__kernel_randint:n { 10000 }
    \__fp_rand_myriads_loop:w
  }
\cs_new:Npn \__fp_rand_myriads_get:w 1 #1 \__fp_sep: { \__fp_sep: {#1} }
\cs_new:Npn \__fp_rand_o:Nw ? #1 @
  {
    \tl_if_empty:nTF {#1}
      {
        \exp_after:wN \__fp_rand_o:w
        \exp:w \exp_end_continue_f:w
        \__fp_rand_myriads:n { XXXX } { 0000 } { 0000 } \__fp_sep: 0
      }
      {
        \__fp_error_num_args:ffff { rand } { 0 } { 0 }
          { \__fp_array_count:n {#1} }
        \exp_after:wN \c_nan_fp
      }
  }
\cs_new:Npn \__fp_rand_o:w \__fp_sep:
  {
    \exp_after:wN \__fp_sanitize:Nw
    \exp_after:wN 0
    \int_value:w \__fp_int_eval:w \c_zero_int
      \__fp_fixed_to_float_o:wN
  }
\cs_new:Npn \__fp_randint_o:Nw ?
  {
    \__fp_parse_function_one_two:nnw
      { randint }
      { \__fp_randint_default:w \__fp_randint_o:w }
  }
\cs_new:Npn \__fp_randint_default:w #1 { \exp_after:wN #1 \c_one_fp }
\cs_new:Npn \__fp_randint_badarg:w \s__fp \__fp_chk:w #1#2#3\__fp_sep:
  {
    \__fp_int:wTF \s__fp \__fp_chk:w #1#2#3\__fp_sep:
      {
        \if_meaning:w 1 #1
          \if_int_compare:w
              \__fp_use_i_until_s:nw #3 \__fp_sep: > \c__fp_prec_int
            \c_one_int
          \fi:
        \fi:
      }
      { \c_one_int }
  }
\cs_new:Npn \__fp_randint_o:w #1\__fp_sep: #2\__fp_sep: @
  {
    \if_case:w
        \__fp_randint_badarg:w #1\__fp_sep:
        \__fp_randint_badarg:w #2\__fp_sep:
        \if:w 1 \__fp_compare_back:ww #2\__fp_sep: #1\__fp_sep: \c_one_int \fi:
        \c_zero_int
      \__fp_randint_auxi_o:ww #1\__fp_sep: #2\__fp_sep:
    \or:
      \__fp_invalid_operation_tl_o:ff
        { randint } { \__fp_array_to_clist:n { #1\__fp_sep: #2\__fp_sep: } }
      \exp:w
    \fi:
    \exp_after:wN \exp_end:
  }
\cs_new:Npn \__fp_randint_auxi_o:ww #1 \__fp_sep: #2 \__fp_sep: #3 \exp_end:
  {
    \fi:
    \__fp_randint_auxii:wn #2 \__fp_sep:
    { \__fp_randint_auxii:wn #1 \__fp_sep: \__fp_randint_auxiii_o:ww }
  }
\cs_new:Npn \__fp_randint_auxii:wn \s__fp \__fp_chk:w #1#2#3#4 \__fp_sep:
  {
    \if_meaning:w 0 #1
      \exp_after:wN \use_i:nn
    \else:
      \exp_after:wN \use_ii:nn
    \fi:
    { \exp_after:wN \__fp_fixed_continue:wn \c__fp_one_fixed_tl }
    {
      \exp_after:wN \__fp_ep_to_fixed:wwn
      \int_value:w \__fp_int_eval:w
        #3 - \c__fp_prec_int , #4 {0000} {0000} \__fp_sep:
      {
        \if_meaning:w 0 #2
          \exp_after:wN \use_i:nnnn
          \exp_after:wN \__fp_fixed_add_one:wN
        \fi:
        \exp_after:wN \__fp_fixed_sub:wwn \c__fp_one_fixed_tl
      }
      \__fp_fixed_continue:wn
    }
  }
\cs_new:Npn \__fp_randint_auxiii_o:ww #1 \__fp_sep: #2 \__fp_sep:
  {
    \__fp_fixed_add:wwn #2 \__fp_sep:
      {0000} {0000} {0000} {0001} {0000} {0000} \__fp_sep:
    \__fp_fixed_sub:wwn #1 \__fp_sep:
    {
      \exp_after:wN \use_i:nn
      \exp_after:wN \__fp_fixed_mul_add:wwwn
      \exp:w \exp_end_continue_f:w \__fp_rand_myriads:n { XXXXXX } \__fp_sep:
    }
    #1 \__fp_sep:
    \__fp_randint_auxiv_o:ww
    #2 \__fp_sep:
    \__fp_randint_auxv_o:w #1 \__fp_sep: @
  }
\cs_new:Npn \__fp_randint_auxiv_o:ww #1#2#3#4#5 \__fp_sep: #6#7#8#9
  {
    \if_int_compare:w
      \if_int_compare:w #1#2 > #6#7 \exp_stop_f: 1 \else:
        \if_int_compare:w #1#2 < #6#7 \exp_stop_f: - \fi: \fi:
      #3#4 > #8#9 \exp_stop_f:
      \__fp_use_i_until_s:nw
    \fi:
    \__fp_randint_auxv_o:w {#1}{#2}{#3}{#4}#5
  }
\cs_new:Npn \__fp_randint_auxv_o:w #1#2#3#4#5 \__fp_sep: #6 @
  {
    \exp_after:wN \__fp_sanitize:Nw
    \int_value:w
    \if_int_compare:w #1 < 10000 \exp_stop_f:
      2
    \else:
      0
      \exp_after:wN \exp_after:wN
      \exp_after:wN \__fp_reverse_args:Nww
    \fi:
    \exp_after:wN \__fp_fixed_sub:wwn \c__fp_one_fixed_tl
    {#1} {#2} {#3} {#4} {0000} {0000} \__fp_sep:
    {
      \exp_after:wN \exp_stop_f:
      \int_value:w \__fp_int_eval:w \c__fp_prec_int
        \__fp_fixed_to_float_o:wN
    }
    0
    \exp:w \exp_after:wN \exp_end:
  }
\cs_new:Npn \int_rand:nn #1#2
  {
    \int_eval:n
      {
        \exp_after:wN \__fp_randint:ww
        \int_value:w \int_eval:n {#1} \exp_after:wN \__fp_sep:
        \int_value:w \int_eval:n {#2} \__fp_sep:
      }
  }
\cs_new:Npn \__fp_randint:ww #1\__fp_sep: #2\__fp_sep:
  {
    \if_int_compare:w #1 > #2 \exp_stop_f:
      \msg_expandable_error:nnnn
        { kernel } { randint-backward-range } {#1} {#2}
      \__fp_randint:ww #2\__fp_sep: #1\__fp_sep:
    \else:
      \if_int_compare:w \__fp_int_eval:w #2
          \if_int_compare:w #1 > \c_zero_int
            - #1 < \__fp_int_eval:w
          \else:
            < \__fp_int_eval:w #1 +
          \fi:
          \c__kernel_randint_max_int
          \__fp_int_eval_end:
        \__kernel_randint:n
          { \__fp_int_eval:w #2 - #1 + 1 \__fp_int_eval_end: }
        - 1 + #1
      \else:
        \__kernel_randint:nn {#1} {#2}
      \fi:
    \fi:
  }
\cs_new:Npn \__kernel_randint:nn #1#2
```

```
  {
    #1
    \exp_after:wN \__fp_randint_wide_aux:w
    \int_value:w
      \exp_after:wN \__fp_randint_split_o:Nw
      \tex_uniformdeviate:D 268435456 \__fp_sep:
    \int_value:w
      \exp_after:wN \__fp_randint_split_o:Nw
      \tex_uniformdeviate:D 268435456 \__fp_sep:
    \int_value:w
      \exp_after:wN \__fp_randint_split_o:Nw
      \int_value:w \__fp_int_eval:w 131072 +
        \exp_after:wN \__fp_randint_split_o:Nw
        \int_value:w
          \__kernel_int_add:nnn {#2} { -#1 } { -\c_max_int } \__fp_sep:
    .
  }
\cs_new:Npn \__fp_randint_split_o:Nw #1#2 \__fp_sep:
  {
    \if_meaning:w 0 #1
      0 \exp_after:wN \__fp_sep: \int_value:w 0
    \else:
      \exp_after:wN \__fp_randint_split_aux:w
      \int_value:w \__fp_int_eval:w (#1#2 - 8192) / 16384 \__fp_sep:
      + #1#2
    \fi:
    \exp_after:wN \__fp_sep:
  }
\cs_new:Npn \__fp_randint_split_aux:w #1 \__fp_sep:
  {
    #1 \exp_after:wN \__fp_sep:
    \int_value:w \__fp_int_eval:w - #1 * 16384
  }
\cs_new:Npn \__fp_randint_wide_aux:w
    #1\__fp_sep:#2\__fp_sep: #3\__fp_sep:#4\__fp_sep:
    #5\__fp_sep:#6\__fp_sep:#7\__fp_sep: .
  {
    \exp_after:wN \__fp_randint_wide_auxii:w
    \int_value:w \__fp_int_eval:w #5 * #3 + #6 * #1 +
      (#5 * #4 + #6 * #3 + #7 * #1 +
        (#5 * #2 +           #7 * #3 +
          (16384 * #6 + #7) * (16384 * #4 + #2) / 268435456) / 16384
      ) / 16384 \exp_after:wN \__fp_sep:
    \int_value:w \__fp_int_eval:w (#5 + #6) * 16384 + #7 \__fp_sep:
    #1 \__fp_sep: #5 \__fp_sep:
  }
```

```
\cs_new:Npn \__fp_randint_wide_auxii:w
    #1\__fp_sep: #2\__fp_sep: #3\__fp_sep: #4\__fp_sep:
  {
    \if_int_odd:w 0
        \if_int_compare:w #1 = #2 \else: \exp_stop_f: \fi:
        \if_int_compare:w #4 = \c_zero_int 1 \fi:
        \if_int_compare:w #3 = 16383 ~ 1 \fi:
        \exp_stop_f:
      \exp_after:wN \prg_break:
    \fi:
    \if_int_compare:w #4 < 8 \exp_stop_f:
      + #4 * #3 * 16384
    \else:
      + 8 * #3 * 16384 + (#4 - 8) * #3 * 16384
    \fi:
    + #1
    \prg_break_point:
  }
\cs_new:Npn \int_rand:n #1
  {
    \int_eval:n
      { \exp_args:Nf \__fp_randint:n { \int_eval:n {#1} } }
  }
\cs_new:Npn \__fp_randint:n #1
  {
    \if_int_compare:w #1 < \c_one_int
      \msg_expandable_error:nnnn
        { kernel } { randint-backward-range } { 1 } {#1}
      \__fp_randint:ww #1\__fp_sep: 1\__fp_sep:
    \else:
      \if_int_compare:w #1 > \c__kernel_randint_max_int
        \__kernel_randint:nn { 1 } {#1}
      \else:
        \__kernel_randint:n {#1}
      \fi:
    \fi:
  }
%% File l3fp-types.dtx (C) Copyright 2012-2015,2017,2018,2020,2021,2023 The LaTeX Project
\cs_new:Npe \__fp_types_cs_to_op:N #1
  {
    \exp_not:N \exp_after:wN \exp_not:N \__fp_types_cs_to_op_auxi:wwwn
      \exp_not:N \token_to_str:N #1 \s__fp_mark
      \exp_not:N \__fp_use_i_delimit_by_s_stop:nw
      \tl_to_str:n { __fp_ _o:w } \s__fp_mark
        { \exp_not:N \__fp_use_i_delimit_by_s_stop:nw ? }
      \s__fp_stop
  }
\use:e
  {
    \cs_new:Npn \exp_not:N \__fp_types_cs_to_op_auxi:wwwn
      #1 \tl_to_str:n { __fp_ } #2
      \tl_to_str:n { _o:w } #3 \s__fp_mark #4 { #4 {#2} }
  }
\cs_new:Npn \__fp_types_unary:NNw #1
  {
    \exp_args:Nf \__fp_types_unary_auxi:nNw
      { \__fp_types_cs_to_op:N #1 }
  }
\cs_new:Npn \__fp_types_unary_auxi:nNw #1#2#3
  {
    \exp_after:wN \__fp_types_unary_auxii:NnNw
    \cs:w __fp_#1 \__fp_type_from_scan:N #3 _o:w \cs_end:
    {#1}
    #2#3
  }
\cs_new:Npn \__fp_types_unary_auxii:NnNw #1#2#3
  {
    \token_if_eq_meaning:NNTF \scan_stop: #1
      { \__fp_invalid_operation_o:nw {#2} }
      { #1 #3 }
  }
\cs_new:Npn \__fp_types_binary:Nww #1
  {
    \exp_last_unbraced:Nf \__fp_types_binary_auxi:Nww
      { \__fp_types_cs_to_op:N #1 }
  }
\cs_new:Npn \__fp_types_binary_auxi:Nww #1#2#3\__fp_sep: #4#5\__fp_sep: @
  {
    \exp_after:wN \__fp_types_binary_auxii:NNww
    \cs:w
      __fp
      \__fp_type_from_scan:N #2
      _#1
      \__fp_type_from_scan:N #4
      _o:ww
    \cs_end:
    #1 #2#3\__fp_sep: #4#5\__fp_sep:
  }
\cs_new:Npn \__fp_types_binary_auxii:NNww #1#2
  {
    \token_if_eq_meaning:NNTF \scan_stop: #1
      { \__fp_invalid_operation_o:Nww #2 }
      {#1}
  }
%% File l3fp-symbolic.dtx (C) Copyright 2012-2025 The LaTeX Project
\fp_new:N \l__fp_symbolic_fp
\scan_new:N \s__fp_symbolic
\cs_new_protected:Npn \__fp_symbolic_chk:w #1,#2#3\__fp_sep:
  {
    \msg_error:nne { fp } { misused-fp }
      {
        \__fp_to_tl_dispatch:w
          \s__fp_symbolic \__fp_symbolic_chk:w #1,{#2}\__fp_sep:
      }
  }
\cs_new:Npn \__fp_if_has_symbolic:nTF #1
  {
    \__fp_if_has_symbolic_aux:w
      #1             \s__fp_mark \use_i:nn
      \s__fp_symbolic \s__fp_mark \use_ii:nn
    \s__fp_stop
  }
\cs_new:Npn \__fp_if_has_symbolic_aux:w
    #1 \s__fp_symbolic #2 \s__fp_mark #3#4 \s__fp_stop { #3 }
\cs_new:Npn \__fp_exp_after_symbolic_f:nw
    #1 \s__fp_symbolic \__fp_symbolic_chk:w #2, #3#4\__fp_sep:
  {
    \exp_after:wN \__fp_exp_after_symbolic_aux:w
    \exp:w
    \__fp_exp_after_symbolic_loop:N #2
      { , \exp:w \use_none:nn }
    \exp_after:wN \exp_end: \exp_after:wN
      {
        \exp:w \exp_end_continue_f:w
        \__fp_exp_after_array_f:w #3 \s__fp_expr_stop
        \exp_after:wN
      }
    \exp_after:wN \__fp_sep:
    \exp:w \exp_end_continue_f:w #1
  }
\cs_new:Npn \__fp_exp_after_symbolic_aux:w #1, #2\__fp_sep:
  {
    \__fp_if_has_symbolic:nTF {#2}
      { \s__fp_symbolic \__fp_symbolic_chk:w #1, {#2} \__fp_sep: }
      { #1 #2 @ \prg_do_nothing: }
  }
\cs_new:Npn \__fp_exp_after_symbolic_loop:N #1
  {
    \exp_after:wN \exp_end:
    \exp_after:wN #1
    \exp:w
    \__fp_exp_after_symbolic_loop:N
  }
\cs_new:Npn \__fp_symbolic_binary_o:Nww #1 #2\__fp_sep: #3\__fp_sep:
  {
    \__fp_exp_after_symbolic_f:nw { \exp_after:wN \exp_stop_f: }
      \s__fp_symbolic \__fp_symbolic_chk:w
      \__fp_types_binary:Nww #1 , { #2\__fp_sep: #3\__fp_sep: } \__fp_sep:
  }
\cs_set_protected:Npn \__fp_tmp:w #1#2
  {
    \cs_new:cpn
      { __fp_symbolic_#2_symbolic_o:ww }
      { \__fp_symbolic_binary_o:Nww #1 }
    \cs_new_eq:cc
      { __fp_symbolic_#2         _o:ww }
      { __fp_symbolic_#2_symbolic_o:ww }
    \cs_new_eq:cc
      { __fp         _#2_symbolic_o:ww }
      { __fp_symbolic_#2_symbolic_o:ww }
  }
\tl_map_inline:nn { + - * / ^ & | } % |
  { \exp_args:Nc \__fp_tmp:w { __fp_#1_o:ww } {#1} }
\cs_new:Npn \__fp_symbolic_unary_o:NNw #1#2#3\__fp_sep: @
  {
    \__fp_exp_after_symbolic_f:nw { \exp_after:wN \exp_stop_f: }
      \s__fp_symbolic \__fp_symbolic_chk:w
      \__fp_types_unary:NNw #1#2 , { #3\__fp_sep: } \__fp_sep:
  }
\tl_map_inline:nn
  {
    {acos} {acsc} {asec} {asin} {cos} {cot} {csc} {exp} {fact} {ln}
    {not} {sec} {set_sign} {sin} {sign} {sqrt} {tan}
  }
  {
    \cs_new:cpe { __fp_symbolic_#1_o:w }
      {
        \exp_not:N \__fp_symbolic_unary_o:NNw
        \exp_not:c { __fp_#1_o:w }
      }
  }
\cs_set_protected:Npn \__fp_tmp:w #1#2#3
  {
    \cs_new:cpn { __fp_symbolic_to_#1:w }
```

```
      {
        \exp_after:wN \__fp_symbolic_convert:wnnN
        \exp:w \exp_end_continue_f:w
        \__fp_exp_after_symbolic_f:nw { { #2 } { fp_to_#1 } #3 }
      }
  }
\__fp_tmp:w { decimal    } { 0   } \__fp_to_decimal_dispatch:w
\__fp_tmp:w { int        } { 0   } \__fp_to_int_dispatch:w
\__fp_tmp:w { scientific } { nan } \__fp_to_scientific_dispatch:w
\cs_new:Npn \__fp_symbolic_convert:wnnN #1#2\__fp_sep: #3#4#5
  {
    \str_if_eq:nnTF {#1} { \s__fp_symbolic }
      { \__fp_invalid_operation:nnw {#3} {#4} #1#2\__fp_sep: }
      { #5 #1#2\__fp_sep: }
  }
\cs_new:Npn \__fp_symbolic_cs_arg_to_fn:NN #1
  {
    \exp_args:Nf \__fp_symbolic_op_arg_to_fn:nN
      { \__fp_types_cs_to_op:N #1 }
  }
\cs_new:Npn \__fp_symbolic_op_arg_to_fn:nN #1#2
  {
    \str_case:nnF { #1 #2 }
      {
        { not ? } { ! }
        { set_sign 0 } { abs }
        { set_sign 2 } { - }
      }
      {
        \token_if_eq_meaning:NNTF #2 \use_ii:nn
          { #1 d } {#1}
      }
  }
\cs_new:Npn \__fp_symbolic_to_tl:w
    \s__fp_symbolic \__fp_symbolic_chk:w #1#2, #3#4\__fp_sep:
  {
    \str_case:nnTF {#1}
      {
        { \__fp_types_unary:NNw } { \__fp_symbolic_unary_to_tl:NNw }
        { \__fp_types_binary:Nww } { \__fp_symbolic_binary_to_tl:Nww }
        { \__fp_function_o:w } { \__fp_symbolic_function_to_tl:Nw }
      }
      { #2, #3 @ }
      { \tl_to_str:n {#2} }
  }
\cs_new:Npn \__fp_symbolic_unary_to_tl:NNw #1#2 , #3 @
```

```
  {
    \use:e
      {
        \__fp_symbolic_cs_arg_to_fn:NN #1#2
        ( \__fp_to_tl_dispatch:w #3 )
      }
  }
\cs_new:Npn \__fp_symbolic_binary_to_tl:Nww #1, #2\__fp_sep: #3\__fp_sep: @
  {
    \use:e
      {
        ( \__fp_to_tl_dispatch:w #2\__fp_sep: )
        \__fp_types_cs_to_op:N #1
        ( \__fp_to_tl_dispatch:w #3\__fp_sep: )
      }
  }
\cs_new:Npn \__fp_symbolic_function_to_tl:Nw #1, #2@
  {
    \use:e
      {
        \__fp_types_cs_to_op:N #1
        ( \__fp_array_to_clist:n {#2} )
      }
  }
\prg_new_protected_conditional:Npnn
    \__fp_id_if_invalid:n #1 { T , F , TF }
  {
    \tl_if_empty:nTF {#1}
      { \prg_return_true: }
      {
        \tl_if_in:nnTF { #1 } { ~ }
          { \prg_return_true: }
          {
            \__fp_id_if_invalid_aux:N #1
              { ? \prg_break:n \prg_return_false: }
            \prg_break_point:
          }
      }
  }
\cs_new:Npn \__fp_id_if_invalid_aux:N #1
  {
    \use_none:n #1
    \int_compare:nF { `a <= `#1 <= `z }
      {
        \int_compare:nF { `A <= `#1 <= `Z }
          { \prg_break:n \prg_return_true: }
      }
    \__fp_id_if_invalid_aux:N
  }
\cs_new:Npn \__fp_variable_o:w #1 @ #2
  {
    \fp_if_exist:cTF { l__fp_variable_#1_fp }
      {
        \exp_last_unbraced:Nf \__fp_exp_after_array_f:w
          { \use:c { l__fp_variable_#1_fp } } \s__fp_expr_stop
        \exp_after:wN \exp_stop_f: #2
      }
      {
        \token_if_eq_meaning:NNTF #2 \prg_do_nothing:
          {
            \s__fp_symbolic \__fp_symbolic_chk:w
              \__fp_variable_o:w #1 , { } \__fp_sep:
          }
          {
            \exp_after:wN \s__fp_symbolic
            \exp_after:wN \__fp_symbolic_chk:w
            \exp_after:wN \__fp_variable_o:w
            \exp:w
            \__fp_exp_after_symbolic_loop:N #1
              { , \exp:w \use_none:nn }
            \exp_after:wN \exp_end:
            \exp_after:wN { \exp_after:wN } \exp_after:wN \__fp_sep:
            #2
          }
      }
  }
\cs_new_protected:Npn \__fp_variable_set_parsing:Nn #1#2
  {
    \cs_set:Npn \__fp_tmp:w
      {
        \__fp_exp_after_symbolic_f:nw { \__fp_parse_infix:NN }
        \s__fp_symbolic \__fp_symbolic_chk:w
          \__fp_variable_o:w #2 , { } \__fp_sep:
      }
    \exp_args:NNc \__fp_variable_set_parsing_aux:NNn #1
      { __fp_parse_word_#2:N } {#2}
  }
\cs_new_protected:Npn \__fp_variable_set_parsing_aux:NNn #1#2#3
  {
    \cs_if_eq:NNF #2 \__fp_tmp:w
      {
        \cs_if_exist:NTF #2
```

```
          {
            \msg_warning:nnnn
              { fp } { id-used-elsewhere } {#3} { variable }
            #1 #2 \__fp_tmp:w
          }
          {
            \cs_new_eq:NN #2 \scan_stop: % to declare the function
            #1 #2 \__fp_tmp:w
          }
      }
  }
\cs_new_protected:Npn \fp_clear_variable:n #1
  {
    \exp_args:No \__fp_clear_variable:n { \tl_to_str:n {#1} }
  }
\cs_new_protected:Npn \__fp_clear_variable:n #1
  {
    \__fp_id_if_invalid:nTF {#1}
      { \msg_error:nnn { fp } { id-invalid } {#1} }
      { \__fp_clear_variable_aux:n {#1} }
  }
\cs_new_protected:Npn \__fp_clear_variable_aux:n #1
  {
    \cs_set_eq:cN { l__fp_variable_#1_fp } \tex_undefined:D
    \__fp_variable_set_parsing:Nn \cs_set_eq:NN {#1}
  }
\cs_new_protected:Npn \fp_new_variable:n #1
  {
    \exp_args:No \__fp_new_variable:n { \tl_to_str:n {#1} }
  }
\cs_new_protected:Npn \__fp_new_variable:n #1
  {
    \__fp_id_if_invalid:nTF {#1}
      { \msg_error:nnn { fp } { id-invalid } {#1} }
      {
        \cs_if_exist:cT { __fp_parse_word_#1:N }
          {
            \msg_error:nnn
              { fp } { id-already-defined } {#1}
            \cs_undefine:c { __fp_parse_word_#1:N }
            \cs_set_eq:cN { l__fp_variable_#1_fp } \tex_undefined:D
          }
      \__fp_variable_set_parsing:Nn \cs_gset_eq:NN {#1}
      }
  }
\flag_new:N \l__fp_symbolic_flag
```

```
\cs_new_protected:Npn \fp_set_variable:nn #1
  {
    \exp_args:No \__fp_set_variable:nn { \tl_to_str:n {#1} }
  }
\cs_new_protected:Npn \__fp_set_variable:nn #1#2
  {
    \__fp_id_if_invalid:nTF {#1}
      { \msg_error:nnn { fp } { id-invalid } {#1} }
      {
        \cs_if_exist:cF { __fp_parse_word_#1:N }
          {
            \msg_error:nnn {fp} { id-undefined } {#1}
            \__fp_variable_set_parsing:Nn \cs_set_eq:NN {#1}
          }
        \fp_set:Nn \l__fp_symbolic_fp {#2}
        \cs_set_nopar:cpn { l__fp_variable_#1_fp }
          { \flag_ensure_raised:N \l__fp_symbolic_flag \c_nan_fp }
        \flag_clear:N \l__fp_symbolic_flag
        \fp_set:cn { l__fp_variable_#1_fp } { \l__fp_symbolic_fp }
        \flag_if_raised:NT \l__fp_symbolic_flag
          {
            \msg_error:nneee { fp } { id-loop }
              { #1 }
              { \tl_to_str:n {#2} }
              { \fp_to_tl:N \l__fp_symbolic_fp }
          }
      }
  }
\msg_new:nnnn { fp } { id-invalid }
  { Floating~point~identifier~'#1'~invalid. }
  {
    LaTeX~has~been~asked~to~create~a~new~floating~point~identifier~'#1'~
    but~this~may~only~contain~ASCII~letters.
  }
\msg_new:nnnn { fp } { id-already-defined }
  { Floating~point~identifier~'#1'~already~defined. }
  {
    LaTeX~has~been~asked~to~create~a~new~floating~point~identifier~'#1'~
    but~this~name~has~already~been~used~elsewhere.
  }
\msg_new:nnnn { fp } { id-undefined }
  { Floating~point~identifier~'#1'~is~undefined. }
  {
    LaTeX~has~been~asked~to~set~a~floating~point~identifier~'#1'~
    but~this~name~has~not~been~declared.
  }
```

```
\msg_new:nnnn { fp } { id-used-elsewhere }
  { Floating~point~identifier~'#1'~already~used~for~something~else. }
  {
    LaTeX~has~been~asked~to~create~a~new~floating~point~identifier~'#1'~
    but~this~name~is~used,~and~is~not~a~user-defined~#2.
  }
\msg_new:nnnn { fp } { id-loop }
  { Variable~'#1'~used~in~the~definition~of~'#1'. }
  {
    LaTeX~has~been~asked~to~set~the~floating~point~identifier~'#1'~
    to~the~expression~'#2'.~Evaluating~this~expression~yields~'#3',~
    which~contains~'#1'~itself.
  }
%% File l3fp-functions.dtx (C) Copyright 2012-2018,2020,2021,2023 The LaTeX Project
\cs_new_protected:Npn \fp_new_function:n #1
  { \exp_args:No \__fp_new_function:n { \tl_to_str:n {#1} } }
\cs_new_protected:Npn \__fp_new_function:n #1
  {
    \__fp_id_if_invalid:nTF {#1}
      { \msg_error:nnn { fp } { id-invalid } {#1} }
      {
        \cs_if_exist:cT { __fp_parse_word_#1:N }
          {
            \msg_error:nnn
              { fp } { id-already-defined } {#1}
            \cs_undefine:c { __fp_parse_word_#1:N }
            \cs_undefine:c { __fp_#1_o:w }
          }
        \__fp_function_set_parsing:Nn \cs_gset_eq:NN {#1}
      }
  }
\cs_new_protected:Npn \__fp_function_set_parsing:Nn #1#2
  {
    \exp_args:NNc \__fp_function_set_parsing_aux:NNn #1
      { __fp_parse_word_#2:N } {#2}
  }
\cs_new_protected:Npn \__fp_function_set_parsing_aux:NNn #1#2#3
  {
    \cs_set:Npe \__fp_tmp:w
      {
        \exp_not:N \__fp_parse_function:NNN
        \exp_not:N \__fp_function_o:w
        \exp_not:c { __fp_#3_o:w }
      }
    \cs_if_eq:NNF #2 \__fp_tmp:w
      {
        \cs_if_exist:NTF #2
          {
            \msg_warning:nnnn
              { fp } { id-used-elsewhere } {#3} { function }
            #1 #2 \__fp_tmp:w
          }
          {
            \cs_new_eq:NN #2 \scan_stop: % to declare the function
            #1 #2 \__fp_tmp:w
          }
      }
  }
\cs_new:Npn \__fp_function_o:w #1#2 @
  {
    \cs_if_exist:NTF #1
      { #1 #2 @ }
      {
        \exp_after:wN \s__fp_symbolic
        \exp_after:wN \__fp_symbolic_chk:w
        \exp_after:wN \__fp_function_o:w
        \exp_after:wN #1
        \exp_after:wN ,
        \exp_after:wN {
          \exp:w \exp_end_continue_f:w
          \__fp_exp_after_array_f:w #2 \s__fp_expr_stop
          \exp_after:wN
        }
        \exp_after:wN \__fp_sep:
      }
  }
\int_new:N \l__fp_function_arg_int
\cs_new_protected:Npn \fp_set_function:nnn #1
  {
    \exp_args:NNo \__fp_set_function:Nnnn \cs_set_eq:cN
      { \tl_to_str:n {#1} }
  }
\cs_new_protected:Npn \__fp_set_function:Nnnn #1#2#3#4
  {
    \__fp_id_if_invalid:nTF {#2}
      { \msg_error:nnn { fp } { id-invalid } {#2} }
      {
        \cs_if_exist:cF { __fp_parse_word_#2:N }
          {
            \msg_error:nnn {fp} { id-undefined } {#2}
            \__fp_function_set_parsing:Nn \cs_set_eq:NN {#2}
          }
        \group_begin:
          \int_zero:N \l__fp_function_arg_int
          \exp_args:No \clist_map_inline:nn { \tl_to_str:n {#3} }
            {
              \int_incr:N \l__fp_function_arg_int
              \exp_args:Ne \__fp_clear_variable_aux:n
                {
                  \c_underscore_str \tex_romannumeral:D \l__fp_function_arg_int
                }
              \fp_clear_variable:n {##1}
              \cs_set_nopar:cpe { l__fp_variable_##1_fp }
                {
                  \exp_not:N \s__fp_symbolic
                  \exp_not:N \__fp_symbolic_chk:w
                  \exp_not:N \__fp_function_arg_o:w
                  \int_use:N \l__fp_function_arg_int
                  ########1 , { } \__fp_sep:
                }
            }
          \cs_set:Npn \__fp_function_arg_o:w ##1 @
            {
              \exp_after:wN \s__fp_symbolic
              \exp_after:wN \__fp_symbolic_chk:w
              \exp_after:wN \__fp_function_arg_o:w
              \tex_romannumeral:D
              \__fp_exp_after_symbolic_loop:N ##1
                { , \tex_romannumeral:D \use_none:nn }
              \exp_after:wN \c_zero_int
              \exp_after:wN { \exp_after:wN } \exp_after:wN \__fp_sep:
            }
          \fp_set:Nn \l__fp_symbolic_fp {#4}
          \use:e
            {
              \exp_not:n { \cs_gset:Npn \__fp_tmp:w ##1 }
                { \exp_not:o { \l__fp_symbolic_fp } }
            }
          \use:e
            {
              \exp_not:n { \cs_gset:Npn \__fp_tmp:w ##1 @ }
                {
                  \exp_not:N \__fp_exp_after_symbolic_f:nw
                  \exp_not:n { { \exp_after:wN \exp_stop_f: } }
                  \exp_not:o { \__fp_tmp:w { . , {##1} } }
                }
            }
        \group_end:
        #1 { __fp_#2_o:w } \__fp_tmp:w
      }
  }
\cs_new:Npn \__fp_function_arg_o:w #1. #2
  {
    \if_meaning:w @ #2
      \exp_after:wN \__fp_function_arg_few:w
    \fi:
    \if_int_compare:w #1 = \c_one_int
      \exp_after:wN \__fp_function_arg_get:w
    \fi:
    \__fp_use_i_until_s:nw
      {
        \exp_after:wN \__fp_function_arg_o:w
        \int_value:w \int_eval:n { #1 - 1 } .
      }
      #2
  }
\cs_new:Npn \__fp_function_arg_few:w #1 @ { \exp_after:wN \c_nan_fp }
\cs_new:Npn \__fp_function_arg_get:w #1#2#3\__fp_sep: #4 @
  {
    \__fp_exp_after_array_f:w #3\__fp_sep: \s__fp_expr_stop
    \exp_after:wN \exp_stop_f:
  }
\cs_new_protected:Npn \fp_clear_function:n #1
  { \exp_args:No \__fp_clear_function:n { \tl_to_str:n {#1} } }
\cs_new_protected:Npn \__fp_clear_function:n #1
  {
    \__fp_id_if_invalid:nTF {#1}
      { \msg_error:nnn { fp } { id-invalid } {#1} }
      {
        \cs_set_eq:cN { __fp_#1_o:w } \tex_undefine:D
        \__fp_function_set_parsing:Nn \cs_set_eq:NN {#1}
      }
  }
%% File: l3fparray.dtx
\int_new:N \g__fp_array_int
\int_new:N \l__fp_array_loop_int
\cs_new_protected:Npn \fparray_new:Nn #1#2
  {
    \tl_new:N #1
    \prg_replicate:nn { 3 }
      {
        \int_gincr:N \g__fp_array_int
        \exp_args:NNc \tl_gput_right:Nn #1
          { g__fp_array_ \__fp_int_to_roman:w \g__fp_array_int _intarray }
      }
    \exp_last_unbraced:Nfo \__fp_array_new:nNNNN
      { \int_eval:n {#2} } #1 #1
  }
\cs_generate_variant:Nn \fparray_new:Nn { c }
\cs_new_protected:Npn \__fp_array_new:nNNNN #1#2#3#4#5
  {
    \int_compare:nNnTF {#1} < 0
      {
        \msg_error:nnn { kernel } { negative-array-size } {#1}
        \cs_undefine:N #1
        \int_gsub:Nn \g__fp_array_int { 3 }
      }
      {
        \intarray_new:Nn #2 {#1}
        \intarray_new:Nn #3 {#1}
        \intarray_new:Nn #4 {#1}
      }
  }
\cs_new:Npn \fparray_count:N #1
  {
    \exp_after:wN \use_i:nnn
    \exp_after:wN \intarray_count:N #1
  }
\cs_generate_variant:Nn \fparray_count:N { c }
\cs_new:Npn \__fp_array_bounds:NNnTF #1#2#3#4#5
  {
    \if_int_compare:w 1 > #3 \exp_stop_f:
      \__fp_array_bounds_error:NNn #1 #2 {#3}
      #5
    \else:
      \if_int_compare:w #3 > \fparray_count:N #2 \exp_stop_f:
        \__fp_array_bounds_error:NNn #1 #2 {#3}
        #5
      \else:
        #4
      \fi:
    \fi:
  }
\cs_new:Npn \__fp_array_bounds_error:NNn #1#2#3
  {
    #1 { kernel } { out-of-bounds }
      { \token_to_str:N #2 } {#3} { \fparray_count:N #2 }
  }
\cs_new_protected:Npn \fparray_gset:Nnn #1#2#3
  {
    \exp_after:wN \exp_after:wN
    \exp_after:wN \__fp_array_gset:NNNNww
    \exp_after:wN #1
    \exp_after:wN #1
    \int_value:w \int_eval:n {#2} \exp_after:wN \__fp_sep:
    \exp:w \exp_end_continue_f:w \__fp_parse:n {#3}
  }
\cs_generate_variant:Nn \fparray_gset:Nnn { c }
\cs_new_protected:Npn \__fp_array_gset:NNNNww #1#2#3#4#5 \__fp_sep: #6 \__fp_sep:
  {
    \__fp_array_bounds:NNnTF \msg_error:nneee #4 {#5}
      {
        \exp_after:wN \__fp_change_func_type:NNN
          \__fp_use_i_until_s:nw #6 \__fp_sep:
          \__fp_array_gset:w
          \__fp_array_gset_recover:Nw
        #6 \__fp_sep: {#5} #1 #2 #3
      }
      { }
  }
\cs_new_protected:Npn \__fp_array_gset_recover:Nw #1#2 \__fp_sep:
  {
    \__fp_error:nffn { unknown-type } { \tl_to_str:n { #2 \__fp_sep: } } { } { }
    \exp_after:wN #1 \c_nan_fp
  }
\cs_new_protected:Npn \__fp_array_gset:w \s__fp \__fp_chk:w #1#2
  {
    \if_case:w #1 \exp_stop_f:
         \__fp_case_return:nw { \__fp_array_gset_special:nnNNN {#2} }
    \or: \exp_after:wN \__fp_array_gset_normal:w
    \or: \__fp_case_return:nw { \__fp_array_gset_special:nnNNN { #2 3 } }
    \or: \__fp_case_return:nw { \__fp_array_gset_special:nnNNN { 1 } }
    \fi:
    \s__fp \__fp_chk:w #1 #2
  }
\cs_new_protected:Npn \__fp_array_gset_normal:w
  \s__fp \__fp_chk:w 1 #1 #2 #3#4#5 \__fp_sep: #6#7#8#9
  {
    \__kernel_intarray_gset:Nnn #7 {#6} {#2}
    \__kernel_intarray_gset:Nnn #8 {#6}
      { \if_meaning:w 2 #1 3 \else: 1 \fi: #3#4 }
    \__kernel_intarray_gset:Nnn #9 {#6} { 1 \use:nn #5 }
  }
\cs_new_protected:Npn \__fp_array_gset_special:nnNNN #1#2#3#4#5
  {
    \__kernel_intarray_gset:Nnn #3 {#2} {#1}
    \__kernel_intarray_gset:Nnn #4 {#2} {0}
    \__kernel_intarray_gset:Nnn #5 {#2} {0}
  }
\cs_new_protected:Npn \fparray_gzero:N #1
  {
    \int_zero:N \l__fp_array_loop_int
    \prg_replicate:nn { \fparray_count:N #1 }
      {
        \int_incr:N \l__fp_array_loop_int
        \exp_after:wN \__fp_array_gset_special:nnNNN
        \exp_after:wN 0
        \exp_after:wN \l__fp_array_loop_int
        #1
      }
  }
\cs_generate_variant:Nn \fparray_gzero:N { c }
\cs_new:Npn \fparray_item:Nn #1#2
  {
    \exp_after:wN \__fp_array_item:NwN
    \exp_after:wN #1
    \int_value:w \int_eval:n {#2} \__fp_sep:
    \__fp_to_decimal:w
  }
\cs_generate_variant:Nn \fparray_item:Nn { c }
\cs_new:Npn \fparray_item_to_tl:Nn #1#2
  {
    \exp_after:wN \__fp_array_item:NwN
    \exp_after:wN #1
    \int_value:w \int_eval:n {#2} \__fp_sep:
    \__fp_to_tl:w
  }
\cs_generate_variant:Nn \fparray_item_to_tl:Nn { c }
\cs_new:Npn \__fp_array_item:NwN #1#2 \__fp_sep: #3
  {
    \__fp_array_bounds:NNnTF \msg_expandable_error:nnfff #1 {#2}
      { \exp_after:wN \__fp_array_item:NNNnN #1 {#2} #3 }
      { \exp_after:wN #3 \c_nan_fp }
  }
\cs_new:Npn \__fp_array_item:NNNnN #1#2#3#4
  {
    \exp_after:wN \__fp_array_item:N
    \int_value:w \__kernel_intarray_item:Nn #2 {#4} \exp_after:wN \__fp_sep:
    \int_value:w \__kernel_intarray_item:Nn #3 {#4} \exp_after:wN \__fp_sep:
    \int_value:w \__kernel_intarray_item:Nn #1 {#4} \__fp_sep:
  }
\cs_new:Npn \__fp_array_item:N #1
```

```
  {
    \if_meaning:w 0 #1 \exp_after:wN \__fp_array_item_special:w \fi:
    \__fp_array_item:w #1
  }
\cs_new:Npn \__fp_array_item:w #1 #2#3#4#5 #6 \__fp_sep: 1 #7 \__fp_sep:
  {
    \exp_after:wN \__fp_array_item_normal:w
    \int_value:w \if_meaning:w #1 1 0 \else: 2 \fi: \exp_stop_f:
    #7 \__fp_sep: {#2#3#4#5} {#6} \__fp_sep:
  }
\cs_new:Npn \__fp_array_item_special:w #1 \__fp_sep: #2 \__fp_sep: #3 \__fp_sep: #4
  {
    \exp_after:wN #4
    \exp:w \exp_end_continue_f:w
    \if_case:w #3 \exp_stop_f:
         \exp_after:wN \c_zero_fp
    \or: \exp_after:wN \c_nan_fp
    \or: \exp_after:wN \c_minus_zero_fp
    \or: \exp_after:wN \c_inf_fp
    \else: \exp_after:wN \c_minus_inf_fp
    \fi:
  }
\cs_new:Npn \__fp_array_item_normal:w
    #1 #2#3#4#5 #6 \__fp_sep: #7 \__fp_sep: #8 \__fp_sep: #9
  { #9 \s__fp \__fp_chk:w 1 #1 {#8} #7 {#2#3#4#5} {#6} \__fp_sep: }
\prg_new_eq_conditional:NNn \fparray_if_exist:N \cs_if_exist:N
  { TF , T , F , p }
\prg_new_eq_conditional:NNn \fparray_if_exist:c \cs_if_exist:c
  { TF , T , F , p }
%% File: l3bitset.dtx
\cs_if_exist:NT \@expl@finalise@setup@@
  {
    \tl_gput_right:Nn \@expl@finalise@setup@@
      { \declare@file@substitution { l3bitset.sty } { null.tex } }
  }
\cs_new_protected:Npn \bitset_new:N #1
  {
    \__kernel_chk_if_free_cs:N #1
    \cs_gset_eq:NN #1 \c_zero_str
    \prop_new:c { g__bitset_ \cs_to_str:N #1 _name_prop }
  }
\cs_new_protected:Npn \bitset_new:Nn #1 #2
  {
    \__kernel_chk_if_free_cs:N #1
    \cs_gset_eq:NN #1 \c_zero_str
    \prop_new:c { g__bitset_ \cs_to_str:N #1 _name_prop }
    \prop_gset_from_keyval:cn
      { g__bitset_ \cs_to_str:N #1 _name_prop }
      {#2}
  }
\cs_generate_variant:Nn \bitset_new:N { c }
\cs_generate_variant:Nn \bitset_new:Nn { c }
\cs_new_protected:Npn \bitset_addto_named_index:Nn #1#2
  {
    \prop_gput_from_keyval:cn
      { g__bitset_ \cs_to_str:N #1 _name_prop } { #2 }
  }
\prg_new_eq_conditional:NNn
  \bitset_if_exist:N \str_if_exist:N { p , T , F , TF }
\prg_new_eq_conditional:NNn
  \bitset_if_exist:c \str_if_exist:c { p , T , F , TF }
\cs_new_protected:Npn \__bitset_set_true:Nn #1#2
  { \__bitset_set:NNnN \str_set:Ne #1 {#2} 1 }
\cs_new_protected:Npn \__bitset_gset_true:Nn #1#2
  { \__bitset_set:NNnN \str_gset:Ne #1 {#2} 1 }
\cs_new_protected:Npn \__bitset_set_false:Nn #1#2
  { \__bitset_set:NNnN \str_set:Ne #1 {#2} 0 }
\cs_new_protected:Npn \__bitset_gset_false:Nn #1#2
  { \__bitset_set:NNnN \str_gset:Ne #1 {#2} 0 }
\cs_new_protected:Npn \__bitset_set:NNnN #1#2#3#4
  {
    \int_compare:nNnT {#3} > { 0 }
      {
        \int_compare:nNnTF { \str_count:N #2 } < {#3}
          {
            #1 #2
              {
                #4
                \prg_replicate:nn { #3 - \str_count:N #2 - 1 } { 0 }
                #2
              }
          }
          {
            #1 #2
              {
                \str_range:Nnn #2 { 1 } { -1 - (#3) }
                #4
                \str_range:Nnn #2 { 1 - (#3) } { -1 }
              }
          }
      }
  }
```

```
\int_new:N \l__bitset_internal_int
\prg_new_protected_conditional:Npnn \__bitset_test_digits:n #1 { TF }
  {
    \tex_afterassignment:D \__bitset_test_digits:w
    \l__bitset_internal_int = 0 \tl_trim_spaces_apply:nN {#1} \tl_to_str:n
    \__bitset_test_digits_end:
    \use_i:nnn \if_false:
    \__bitset_test_digits_end:
    \if_int_compare:w \c_zero_int < \l__bitset_internal_int
      \prg_return_true:
    \else:
      \prg_return_false:
    \fi:
  }
\cs_new_eq:NN \__bitset_test_digits_end: \exp_stop_f:
\cs_new_protected:Npn \__bitset_test_digits:w #1 \__bitset_test_digits_end: { }
\cs_new_protected:Npn \bitset_set_true:Nn #1#2
  { \__bitset_set:NNn \__bitset_set_true:Nn #1 {#2} }
\cs_new_protected:Npn \bitset_gset_true:Nn #1#2
  { \__bitset_set:NNn \__bitset_gset_true:Nn #1 {#2} }
\cs_new_protected:Npn \bitset_set_false:Nn #1#2
  { \__bitset_set:NNn \__bitset_set_false:Nn #1 {#2} }
\cs_new_protected:Npn \bitset_gset_false:Nn #1#2
  { \__bitset_set:NNn \__bitset_gset_false:Nn #1 {#2} }
\cs_new_protected:Npn \__bitset_set:NNn #1#2#3
  {
    \prop_if_in:cnTF { g__bitset_ \cs_to_str:N #2 _name_prop } {#3}
      {
        #1 #2
          {
            \prop_item:cn  { g__bitset_ \cs_to_str:N #2 _name_prop } {#3}
          }
      }
      {
        \__bitset_test_digits:nTF {#3}
          {
            #1 #2 {#3}
            \prop_gput:cnn { g__bitset_ \cs_to_str:N #2 _name_prop } {#3} {#3}
          }
          {
            \msg_warning:nnee { bitset } { unknown-name }
              { \token_to_str:N #2 }
              { \tl_to_str:n {#3} }
          }
      }
  }
```

```
\cs_generate_variant:Nn \bitset_set_true:Nn  { c }
\cs_generate_variant:Nn \bitset_gset_true:Nn { c }
\cs_generate_variant:Nn \bitset_set_false:Nn  { c }
\cs_generate_variant:Nn \bitset_gset_false:Nn { c }
\cs_new_protected:Npn \bitset_clear:N #1
  {
    \str_set_eq:NN #1 \c_zero_str
  }
\cs_new_protected:Npn \bitset_gclear:N #1
  {
    \str_gset_eq:NN #1 \c_zero_str
  }
\cs_generate_variant:Nn \bitset_clear:N { c }
\cs_generate_variant:Nn \bitset_gclear:N { c }
\cs_new:Npn \bitset_to_arabic:N #1
  {
    \int_compare:nNnTF { \str_count:N #1 } < { 32 }
      { \exp_args:No \int_from_bin:n {#1} }
      {
        \exp_after:wN \__bitset_to_int:nN \exp_after:wN 0
        #1 \q_recursion_tail \q_recursion_stop
      }
  }
\cs_new:Npn \__bitset_to_int:nN #1#2
  {
    \quark_if_recursion_tail_stop_do:Nn #2 {#1}
    \exp_args:Nf \__bitset_to_int:nN { \fp_eval:n { #1 * 2 + #2 } }
  }
\cs_new:Npn \bitset_to_bin:N #1
  {
    #1
  }
\cs_generate_variant:Nn \bitset_to_arabic:N  { c }
\cs_generate_variant:Nn \bitset_to_bin:N { c }
\cs_new_eq:NN \bitset_use:N \tl_use:N
\cs_generate_variant:Nn \bitset_use:N { c }
\cs_new:Npn \bitset_item:Nn #1#2
  {
    \prop_if_in:cnTF { g__bitset_ \cs_to_str:N #1 _name_prop } {#2}
      {
        \int_eval:n
          {
            \str_item:Nn #1
              { 0 - ( \prop_item:cn { g__bitset_ \cs_to_str:N #1 _name_prop } {#2} ) }
            +0
          }
      }
      {
        0
      }
  }
\cs_generate_variant:Nn \bitset_item:Nn { c }
\cs_new_protected:Npn   \bitset_show:N { \__bitset_show:NN \msg_show:nneeee }
\cs_generate_variant:Nn \bitset_show:N { c }
\cs_new_protected:Npn   \bitset_log:N  { \__bitset_show:NN \msg_log:nneeee }
\cs_generate_variant:Nn \bitset_log:N  { c }
\cs_new_protected:Npn \__bitset_show:NN #1#2
  {
    \__kernel_chk_defined:NT #2
      {
        #1 { bitset } { show }
          { \token_to_str:N #2 }
          { \bitset_to_bin:N #2 }
          { \bitset_to_arabic:N #2 }
          { }
      }
  }
\cs_new_protected:Npn \bitset_show_named_index:N
  { \__bitset_show_named_index:NN \msg_show:nneeee }
\cs_generate_variant:Nn \bitset_show_named_index:N { c }
\cs_new_protected:Npn \bitset_log_named_index:N
  { \__bitset_show_named_index:NN \msg_log:nneeee }
\cs_generate_variant:Nn \bitset_log_named_index:N { c }
\cs_new_protected:Npn \__bitset_show_named_index:NN #1#2
  {
    \__kernel_chk_defined:NT #2
      {
        #1 { bitset } { show-names }
          { \token_to_str:N #2 }
          { \prop_map_function:cN { g__bitset_ \cs_to_str:N #2 _name_prop } \msg_show_item:nn }
          { } { }
      }
  }
\msg_new:nnn { bitset } { show }
  {
    The~bitset~#1~has~the~representation: \\
    >~binary:~#2  \\
    >~arabic:~#3 .
  }
\msg_new:nnn { bitset } { show-names }
  {
    The~bitset~#1~
    \tl_if_empty:nTF {#2}
      { knows~no~names~yet \\>~ . }
      { knows~the~name/index~pairs~(without~outer~braces): #2 . }
  }
\msg_new:nnn { bitset } { unknown-name }
  { The~name~'#2'~is~unknown~for~bitset~\tl_to_str:n {#1} }
\prop_gput:Nnn \g_msg_module_name_prop { bitset } { LaTeX }
\prop_gput:Nnn \g_msg_module_type_prop { bitset } { }
%% File: l3cctab.dtx
\seq_new:N \g__cctab_stack_seq
\seq_new:N \g__cctab_unused_seq
\seq_new:N \g__cctab_group_seq
\int_new:N  \g__cctab_allocate_int
\tl_new:N \l__cctab_internal_a_tl
\tl_new:N \l__cctab_internal_b_tl
\prop_new:N \g__cctab_endlinechar_prop
\sys_if_engine_luatex:TF
  {
    \cs_new_protected:Npn \cctab_new:N #1
      {
        \__kernel_chk_if_free_cs:N #1
        \__cctab_new:N #1
      }
    \cs_new_protected:Npn \__cctab_new:N #1
      {
        \newcatcodetable #1
        \tex_initcatcodetable:D #1
      }
  }
  {
    \cs_new_protected:Npn \__cctab_new:N #1
      {
        \debug_suspend:
        \intarray_new:Nn #1 { 257 }
        \debug_resume:
      }
    \cs_new_protected:Npn \__cctab_gstore:Nnn #1#2#3
      { \intarray_gset:Nnn #1 { #2 + 1 } {#3} }
    \cs_new_protected:Npn \cctab_new:N #1
      {
        \__kernel_chk_if_free_cs:N #1
        \__cctab_new:N #1
        \int_step_inline:nn { 256 }
          { \__kernel_intarray_gset:Nnn #1 {##1} { 12 } }
        \__kernel_intarray_gset:Nnn #1 { 257 } { 13 }
        \__cctab_gstore:Nnn #1 { 0 } { 9 }
        \__cctab_gstore:Nnn #1 { 13 } { 5 }
        \__cctab_gstore:Nnn #1 { 32 } { 10 }
        \__cctab_gstore:Nnn #1 { 37 } { 14 }
        \int_step_inline:nnn { 65 } { 90 }
          { \__cctab_gstore:Nnn #1 {##1} { 11 } }
        \__cctab_gstore:Nnn #1 { 92 } { 0 }
        \int_step_inline:nnn { 97 } { 122 }
          { \__cctab_gstore:Nnn #1 {##1} { 11 } }
        \__cctab_gstore:Nnn #1 { 127 } { 15 }
      }
  }
\cs_generate_variant:Nn \cctab_new:N { c }
\sys_if_engine_luatex:TF
  {
    \cs_new_protected:Npn \__cctab_gset:n #1
      { \exp_args:Nf \__cctab_gset_aux:n { \int_eval:n {#1} } }
    \cs_new_protected:Npn \__cctab_gset_aux:n #1
      {
        \tex_savecatcodetable:D #1 \scan_stop:
        \int_compare:nNnTF { \tex_endlinechar:D } = { 13 }
          { \prop_gremove:Nn \g__cctab_endlinechar_prop {#1} }
          {
            \prop_gput:NnV \g__cctab_endlinechar_prop {#1}
              \tex_endlinechar:D
          }
      }
  }
  {
    \cs_new_protected:Npn \__cctab_gset:n #1
      {
        \int_step_inline:nn { 256 }
          {
            \__kernel_intarray_gset:Nnn #1 {##1}
              { \char_value_catcode:n { ##1 - 1 } }
          }
        \__kernel_intarray_gset:Nnn #1 { 257 }
          { \tex_endlinechar:D }
      }
  }
\cs_new_protected:Npn \cctab_gset:Nn #1#2
  {
    \__cctab_chk_if_valid:NT #1
      {
        \group_begin:
          \cctab_select:N \c_initex_cctab
          #2 \scan_stop:
          \__cctab_gset:n {#1}
        \group_end:
      }
  }
\cs_generate_variant:Nn \cctab_gset:Nn { c }
\cs_new_protected:Npn \cctab_gsave_current:N #1
  {
    \__cctab_chk_if_valid:NT #1
      { \__cctab_gset:n {#1} }
  }
\cs_generate_variant:Nn \cctab_gsave_current:N { c }
\sys_if_engine_luatex:T
  {
    \__cctab_new:N \g__cctab_internal_cctab
    \cs_new:Npn \__cctab_internal_cctab_name:
      {
        g__cctab_internal
        \tex_romannumeral:D \tex_currentgrouplevel:D
        _cctab
      }
  }
\cs_new_protected:Npn \cctab_select:N #1
  { \__cctab_chk_if_valid:NT #1 { \__cctab_select:N #1 } }
\cs_generate_variant:Nn \cctab_select:N { c }
\sys_if_engine_luatex:TF
  {
    \cs_new_protected:Npn \__cctab_select:N #1
      {
        \tex_catcodetable:D #1
        \prop_get:NVNTF \g__cctab_endlinechar_prop #1 \l__cctab_internal_a_tl
          { \int_set:Nn \tex_endlinechar:D { \l__cctab_internal_a_tl } }
          { \int_set:Nn \tex_endlinechar:D { 13 } }
        \cs_if_exist:cF { \__cctab_internal_cctab_name: }
          { \exp_args:Nc \__cctab_new:N { \__cctab_internal_cctab_name: } }
        \exp_args:Nc \tex_savecatcodetable:D { \__cctab_internal_cctab_name: }
        \exp_args:Nc \tex_catcodetable:D { \__cctab_internal_cctab_name: }
      }
  }
  {
    \cs_new_protected:Npn \__cctab_select:N #1
      {
        \int_step_inline:nn { 256 }
          {
            \char_set_catcode:nn { ##1 - 1 }
              { \__kernel_intarray_item:Nn #1 {##1} }
          }
        \int_set:Nn \tex_endlinechar:D
          { \__kernel_intarray_item:Nn #1 { 257 } }
      }
  }
\sys_if_engine_luatex:TF
  {
    \cs_new_protected:Npn \__cctab_begin_aux:
      {
        \__cctab_new:N \g__cctab_next_cctab
        \tl_set:NV \l__cctab_internal_a_tl \g__cctab_next_cctab
        \cs_undefine:N \g__cctab_next_cctab
      }
  }
  {
    \cs_new_protected:Npn \__cctab_begin_aux:
      {
        \int_gincr:N \g__cctab_allocate_int
        \exp_args:Nc \__cctab_new:N
          { g__cctab_ \int_use:N \g__cctab_allocate_int _cctab }
        \exp_args:NNc \tl_set:Nn \l__cctab_internal_a_tl
          { g__cctab_ \int_use:N \g__cctab_allocate_int _cctab }
      }
  }
\cs_new_protected:Npn \cctab_begin:N #1
  {
    \__cctab_chk_if_valid:NT #1
      {
        \seq_gpop:NNF \g__cctab_unused_seq \l__cctab_internal_a_tl
          { \__cctab_begin_aux: }
        \__cctab_chk_group_begin:e
          { \__cctab_nesting_number:N \l__cctab_internal_a_tl }
        \seq_gpush:NV \g__cctab_stack_seq \l__cctab_internal_a_tl
        \exp_args:NV \__cctab_gset:n \l__cctab_internal_a_tl
        \__cctab_select:N #1
      }
  }
\cs_generate_variant:Nn \cctab_begin:N { c }
\cs_new_protected:Npn \cctab_end:
  {
    \seq_gpop:NNTF \g__cctab_stack_seq \l__cctab_internal_a_tl
      {
        \seq_gpush:NV \g__cctab_unused_seq \l__cctab_internal_a_tl
        \exp_args:Ne \__cctab_chk_group_end:n
          { \__cctab_nesting_number:N \l__cctab_internal_a_tl }
        \__cctab_select:N \l__cctab_internal_a_tl
      }
      { \msg_error:nn { cctab } { extra-end } }
  }
\cs_new_protected:Npn \__cctab_chk_group_begin:n #1
  {
    \seq_gpush:Ne \g__cctab_group_seq
      { \int_use:N \tex_currentgrouplevel:D }
    \cs_set_eq:cN { __cctab_group_ #1 _chk: } \prg_do_nothing:
  }
\cs_generate_variant:Nn \__cctab_chk_group_begin:n { e }
\cs_new_protected:Npn \__cctab_chk_group_end:n #1
  {
    \seq_gpop:NN \g__cctab_group_seq \l__cctab_internal_b_tl
    \bool_lazy_and:nnF
      {
        \int_compare_p:nNn
          { \tex_currentgrouplevel:D } = { \l__cctab_internal_b_tl }
      }
      { \cs_if_exist_p:c { __cctab_group_ #1 _chk: } }
      {
        \msg_error:nne { cctab } { group-mismatch }
          {
            \int_sign:n
              { \tex_currentgrouplevel:D - \l__cctab_internal_b_tl }
          }
      }
    \cs_undefine:c { __cctab_group_ #1 _chk: }
  }
\sys_if_engine_luatex:TF
  { \cs_new:Npn \__cctab_nesting_number:N #1 {#1} }
  {
    \cs_new:Npn \__cctab_nesting_number:N #1
      {
        \exp_after:wN \exp_after:wN \exp_after:wN \__cctab_nesting_number:w
          \exp_after:wN \token_to_str:N #1
      }
    \use:e
      {
        \cs_new:Npn \exp_not:N \__cctab_nesting_number:w
          #1 \tl_to_str:n { g__cctab_ } #2 \tl_to_str:n { _cctab } {#2}
      }
  }
\cs_if_exist:NT \hook_gput_code:nnn
  {
    \hook_gput_code:nnn { enddocument/end } { cctab }
      {
        \seq_if_empty:NF \g__cctab_stack_seq
          { \msg_error:nn { cctab } { missing-end } }
      }
  }
\cs_new:Npn \cctab_item:Nn #1#2
  { \exp_args:Nf \__cctab_item:nN { \int_eval:n {#2} } #1 }
\sys_if_engine_luatex:TF
  {
    \cs_new:Npn \__cctab_item:nN #1#2
      { \lua_now:e { tex.print(-2, tex.getcatcode(\int_use:N #2, #1)) } }
  }
  {
    \cs_new:Npn \__cctab_item:nN #1#2
      {
        \int_compare:nNnTF {#1} < { 256 }
          { \intarray_item:Nn #2 { #1 + 1 } }
          { \char_value_catcode:n {#1} }
      }
  }
\cs_generate_variant:Nn \cctab_item:Nn { c }
\prg_new_eq_conditional:NNn \cctab_if_exist:N \cs_if_exist:N
  { TF , T , F , p }
\prg_new_eq_conditional:NNn \cctab_if_exist:c \cs_if_exist:c
  { TF , T , F , p }
\prg_new_protected_conditional:Npnn \__cctab_chk_if_valid:N #1
  { TF , T , F }
  {
    \cctab_if_exist:NTF #1
      {
        \__cctab_chk_if_valid_aux:NTF #1
          { \prg_return_true: }
          {
            \msg_error:nne { cctab } { invalid-cctab }
              { \token_to_str:N #1 }
            \prg_return_false:
          }
      }
      {
        \msg_error:nne { kernel } { command-not-defined }
          { \token_to_str:N #1 }
        \prg_return_false:
      }
  }
\sys_if_engine_luatex:TF
  {
    \cs_new_protected:Npn \__cctab_chk_if_valid_aux:NTF #1
```

```
      {
        \int_compare:nNnTF {#1-1} < { \e@alloc@ccodetable@count }
      }
    \cs_if_exist:NT \c_syst_catcodes_n
      {
        \cs_gset_protected:Npn \__cctab_chk_if_valid_aux:NTF #1
          {
            \int_compare:nTF { #1 <= \c_syst_catcodes_n }
          }
      }
  }
  {
    \cs_new_protected:Npn \__cctab_chk_if_valid_aux:NTF #1
      {
        \exp_args:Nf \str_if_in:nnTF
          { \cs_meaning:N #1 }
          { select~font~cmr10~at~ }
      }
  }
\cs_new_protected:Npn \cctab_const:Nn #1#2
  {
    \__kernel_chk_if_free_cs:N #1
    \__cctab_new:N #1
    \group_begin:
      \cctab_select:N \c_initex_cctab
      #2 \scan_stop:
      \__cctab_gset:n {#1}
    \group_end:
  }
\cs_generate_variant:Nn \cctab_const:Nn { c }
\cctab_new:N \c_initex_cctab
\cctab_const:Nn \c_other_cctab
  {
    \cctab_select:N \c_initex_cctab
    \int_set:Nn \tex_endlinechar:D     { -1 }
    \int_step_inline:nnn { 0 } { 127 }
      { \char_set_catcode_other:n {#1} }
  }
\cctab_const:Nn \c_str_cctab
  {
    \cctab_select:N \c_other_cctab
    \char_set_catcode_space:n { 32 }
  }
\cs_if_exist:NTF \@expl@finalise@setup@@
  { \tl_gput_right:Nn \@expl@finalise@setup@@ }
  { \use:n }
  {
    \__cctab_new:N \c_code_cctab
    \group_begin:
      \int_set:Nn \tex_endlinechar:D { 32 }
      \char_set_catcode_invalid:n { 0 }
      \sys_if_engine_opentype:TF
        { \int_step_function:nN { 31 } \char_set_catcode_invalid:n }
        { \int_step_function:nN { 31 } \char_set_catcode_active:n }
      \int_step_function:nnN { 33 } { 64 } \char_set_catcode_other:n
      \int_step_function:nnN { 65 } { 90 } \char_set_catcode_letter:n
      \int_step_function:nnN { 91 } { 96 } \char_set_catcode_other:n
      \int_step_function:nnN { 97 } { 122 } \char_set_catcode_letter:n
      \char_set_catcode_ignore:n           { 9 }   % tab
      \char_set_catcode_other:n            { 10 }  % lf
      \char_set_catcode_active:n           { 12 }  % ff
      \char_set_catcode_end_line:n         { 13 }  % cr
      \char_set_catcode_ignore:n           { 32 }  % space
      \char_set_catcode_parameter:n        { 35 }  % hash
      \char_set_catcode_math_toggle:n      { 36 }  % dollar
      \char_set_catcode_comment:n          { 37 }  % percent
      \char_set_catcode_alignment:n        { 38 }  % ampersand
      \char_set_catcode_letter:n           { 58 }  % colon
      \char_set_catcode_escape:n           { 92 }  % backslash
      \char_set_catcode_math_superscript:n { 94 }  % circumflex
      \char_set_catcode_letter:n           { 95 }  % underscore
      \char_set_catcode_group_begin:n      { 123 } % left brace
      \char_set_catcode_other:n            { 124 } % pipe
      \char_set_catcode_group_end:n        { 125 } % right brace
      \char_set_catcode_space:n            { 126 } % tilde
      \char_set_catcode_invalid:n          { 127 } % ^^?
      \sys_if_engine_opentype:F
        { \int_step_function:nnN { 128 } { 255 } \char_set_catcode_active:n }
      \__cctab_gset:n { \c_code_cctab }
    \group_end:
    \cctab_const:Nn \c_document_cctab
      {
        \cctab_select:N \c_code_cctab
        \int_set:Nn \tex_endlinechar:D { 13 }
        \char_set_catcode_space:n          { 9 }
        \char_set_catcode_space:n          { 32 }
        \char_set_catcode_other:n          { 58 }
        \char_set_catcode_math_subscript:n { 95 }
        \char_set_catcode_active:n         { 126 }
      }
  }
\cctab_new:N \g_tmpa_cctab
```

```
\cctab_new:N \g_tmpb_cctab
\msg_new:nnnn { cctab } { stack-full }
  { The~category~code~table~stack~is~exhausted. }
  {
    LaTeX~has~been~asked~to~switch~to~a~new~category~code~table,~
    but~there~is~no~more~space~to~do~this!
  }
\msg_new:nnnn { cctab } { extra-end }
  { Extra~\iow_char:N\\cctab_end:~ignored~\msg_line_context:. }
  {
    LaTeX~came~across~a~\iow_char:N\\cctab_end:~without~a~matching~
    \iow_char:N\\cctab_begin:N.~This~command~will~be~ignored.
  }
\msg_new:nnnn { cctab } { missing-end }
  { Missing~\iow_char:N\\cctab_end:~before~end~of~TeX~run. }
  {
    LaTeX~came~across~more~\iow_char:N\\cctab_begin:N~than~
    \iow_char:N\\cctab_end:.
  }
\msg_new:nnnn { cctab } { invalid-cctab }
  { Invalid~\iow_char:N\\catcode~table. }
  {
    You~can~only~switch~to~a~\iow_char:N\\catcode~table~that~is~
    initialized~using~\iow_char:N\\cctab_new:N~or~
    \iow_char:N\\cctab_const:Nn.
  }
\msg_new:nnnn { cctab } { group-mismatch }
  {
    \iow_char:N\\cctab_end:~occurred~in~a~
    \int_case:nn {#1}
      {
        { 0 } { different~group }
        { 1 } { higher~group~level }
        { -1 } { lower~group~level }
      } ~than~
    the~matching~\iow_char:N\\cctab_begin:N.
  }
  {
    Catcode~tables~and~groups~must~be~properly~nested,~but~
    you~tried~to~interleave~them.~LaTeX~will~try~to~proceed,~
    but~results~may~be~unexpected.
  }
\prop_gput:Nnn \g_msg_module_name_prop { cctab } { LaTeX }
\prop_gput:Nnn \g_msg_module_type_prop { cctab } { }
%% File l3sort.dtx
\cs_new_eq:NN \__sort_sep: \__kernel_int_sep:
```

```
\seq_new:N \g__sort_internal_seq
\tl_new:N \g__sort_internal_tl
\int_new:N \l__sort_length_int
\int_new:N \l__sort_min_int
\int_new:N \l__sort_top_int
\int_new:N \l__sort_max_int
\int_new:N \l__sort_true_max_int
\int_new:N \l__sort_block_int
\int_new:N \l__sort_begin_int
\int_new:N \l__sort_end_int
\int_new:N \l__sort_A_int
\int_new:N \l__sort_B_int
\int_new:N \l__sort_C_int
\scan_new:N \s__sort_mark
\scan_new:N \s__sort_stop
\cs_new_protected:Npn \__sort_shrink_range:
  {
    \int_set:Nn \l__sort_A_int
      { \l__sort_true_max_int - \l__sort_min_int + 1 }
    \int_set:Nn \l__sort_block_int { \c_max_register_int / 2 }
    \__sort_shrink_range_loop:
    \int_set:Nn \l__sort_max_int
      {
        \int_compare:nNnTF
          { \l__sort_block_int * 3 / 2 } > \l__sort_A_int
          {
            \l__sort_min_int
            + ( \l__sort_A_int - 1 ) / 2
            + \l__sort_block_int / 4
            - 1
          }
          { \l__sort_true_max_int - \l__sort_block_int / 2 }
      }
  }
\cs_new_protected:Npn \__sort_shrink_range_loop:
  {
    \if_int_compare:w \l__sort_A_int < \l__sort_block_int
      \tex_divide:D \l__sort_block_int 2 \exp_stop_f:
      \exp_after:wN \__sort_shrink_range_loop:
    \fi:
  }
\cs_new_protected:Npn \__sort_compute_range:
  {
    \int_set:Nn \l__sort_min_int { \tex_count:D 15 + 1 }
    \int_set:Nn \l__sort_true_max_int { \c_max_register_int + 1 }
    \__sort_shrink_range:
```

```
    \if_meaning:w \loctoks \tex_undefined:D \else:
      \if_meaning:w \loctoks \scan_stop: \else:
        \__sort_redefine_compute_range:
        \__sort_compute_range:
      \fi:
    \fi:
  }
\cs_new_protected:Npn \__sort_redefine_compute_range:
  {
    \cs_if_exist:cTF { ver@elocalloc.sty }
      {
        \cs_gset_protected:Npn \__sort_compute_range:
          {
            \int_set:Nn \l__sort_min_int { \tex_count:D 265 }
            \int_set_eq:NN \l__sort_true_max_int \e@alloc@top
            \__sort_shrink_range:
          }
      }
      {
        \cs_gset_protected:Npn \__sort_compute_range:
          {
            \int_set:Nn \l__sort_min_int { \tex_count:D 265 }
            \int_set:Nn \l__sort_true_max_int { \tex_count:D 275 }
            \__sort_shrink_range:
          }
      }
  }
\cs_if_exist:NT \loctoks { \__sort_redefine_compute_range: }
\tl_map_inline:nn { \lastallocatedtoks \c_syst_last_allocated_toks }
  {
    \cs_if_exist:NT #1
      {
        \cs_gset_protected:Npn \__sort_compute_range:
          {
            \int_set:Nn \l__sort_min_int { #1 + 1 }
            \int_set:Nn \l__sort_true_max_int { \c_max_register_int + 1 }
            \__sort_shrink_range:
          }
      }
  }
\cs_new_protected:Npn \__sort_main:NNNn #1#2#3#4
  {
    \__sort_disable_toksdef:
    \__sort_compute_range:
    \int_set_eq:NN \l__sort_top_int \l__sort_min_int
    #1 #3
```

```
      {
        \if_int_compare:w \l__sort_top_int = \l__sort_max_int
          \__sort_too_long_error:NNw #2 #3
        \fi:
        \tex_toks:D \l__sort_top_int {##1}
        \int_incr:N \l__sort_top_int
      }
    \int_set:Nn \l__sort_length_int
      { \l__sort_top_int - \l__sort_min_int }
    \cs_set:Npn \__sort_compare:nn ##1 ##2 {#4}
    \int_set:Nn \l__sort_block_int { 1 }
    \__sort_level:
  }
\cs_new_protected:Npn \tl_sort:Nn { \__sort_tl:NNn \tl_set_eq:NN }
\cs_generate_variant:Nn \tl_sort:Nn { c }
\cs_new_protected:Npn \tl_gsort:Nn { \__sort_tl:NNn \tl_gset_eq:NN }
\cs_generate_variant:Nn \tl_gsort:Nn { c }
\cs_new_protected:Npn \__sort_tl:NNn #1#2#3
  {
    \group_begin:
      \__sort_main:NNNn \tl_map_inline:Nn \tl_map_break:n #2 {#3}
      \__kernel_tl_gset:Nx \g__sort_internal_tl
        { \__sort_tl_toks:w \l__sort_min_int \__sort_sep: }
    \group_end:
    #1 #2 \g__sort_internal_tl
    \tl_gclear:N \g__sort_internal_tl
    \prg_break_point:
  }
\cs_new:Npn \__sort_tl_toks:w #1 \__sort_sep:
  {
    \if_int_compare:w #1 < \l__sort_top_int
      { \tex_the:D \tex_toks:D #1 }
      \exp_after:wN \__sort_tl_toks:w
      \int_value:w \int_eval:n { #1 + 1 } \exp_after:wN \__sort_sep:
    \fi:
  }
\cs_new_protected:Npn \seq_sort:Nn
  { \__sort_seq:NNNNn \seq_map_inline:Nn \seq_map_break:n \seq_set_eq:NN }
\cs_generate_variant:Nn \seq_sort:Nn { c }
\cs_new_protected:Npn \seq_gsort:Nn
  { \__sort_seq:NNNNn \seq_map_inline:Nn \seq_map_break:n \seq_gset_eq:NN }
\cs_generate_variant:Nn \seq_gsort:Nn { c }
\cs_new_protected:Npn \clist_sort:Nn
  {
    \__sort_seq:NNNNn \clist_map_inline:Nn \clist_map_break:n
      \clist_set_from_seq:NN
  }
\cs_generate_variant:Nn \clist_sort:Nn { c }
\cs_new_protected:Npn \clist_gsort:Nn
  {
    \__sort_seq:NNNNn \clist_map_inline:Nn \clist_map_break:n
      \clist_gset_from_seq:NN
  }
\cs_generate_variant:Nn \clist_gsort:Nn { c }
\cs_new_protected:Npn \__sort_seq:NNNNn #1#2#3#4#5
  {
    \group_begin:
      \__sort_main:NNNn #1 #2 #4 {#5}
      \seq_gclear:N \g__sort_internal_seq
      \int_step_inline:nnn
        \l__sort_min_int { \l__sort_top_int - 1 }
        {
          \seq_gput_right:Ne \g__sort_internal_seq
            { \tex_the:D \tex_toks:D ##1 }
        }
    \group_end:
    #3 #4 \g__sort_internal_seq
    \seq_gclear:N \g__sort_internal_seq
    \prg_break_point:
  }
\cs_new_protected:Npn \__sort_level:
  {
    \if_int_compare:w \l__sort_block_int < \l__sort_length_int
      \l__sort_end_int \l__sort_min_int
      \__sort_merge_blocks:
      \tex_advance:D \l__sort_block_int \l__sort_block_int
      \exp_after:wN \__sort_level:
    \fi:
  }
\cs_new_protected:Npn \__sort_merge_blocks:
  {
    \l__sort_begin_int \l__sort_end_int
    \tex_advance:D \l__sort_end_int \l__sort_block_int
    \if_int_compare:w \l__sort_end_int < \l__sort_top_int
      \l__sort_A_int \l__sort_end_int
      \tex_advance:D \l__sort_end_int \l__sort_block_int
      \if_int_compare:w \l__sort_end_int > \l__sort_top_int
        \l__sort_end_int \l__sort_top_int
      \fi:
      \l__sort_B_int \l__sort_A_int
      \l__sort_C_int \l__sort_top_int
      \__sort_copy_block:
```

```
      \int_decr:N \l__sort_A_int
      \int_decr:N \l__sort_B_int
      \int_decr:N \l__sort_C_int
      \exp_after:wN \__sort_merge_blocks_aux:
      \exp_after:wN \__sort_merge_blocks:
    \fi:
  }
\cs_new_protected:Npn \__sort_copy_block:
  {
    \tex_toks:D \l__sort_C_int \tex_toks:D \l__sort_B_int
    \int_incr:N \l__sort_C_int
    \int_incr:N \l__sort_B_int
    \if_int_compare:w \l__sort_B_int = \l__sort_end_int
      \use_i:nn
    \fi:
    \__sort_copy_block:
  }
\cs_new_protected:Npn \__sort_merge_blocks_aux:
  {
    \exp_after:wN \__sort_compare:nn \exp_after:wN
      { \tex_the:D \tex_toks:D \exp_after:wN \l__sort_A_int \exp_after:wN }
      \exp_after:wN { \tex_the:D \tex_toks:D \l__sort_C_int }
    \prg_do_nothing:
    \__sort_return_mark:w
    \__sort_return_mark:w
    \s__sort_mark
    \__sort_return_none_error:
  }
\cs_new_protected:Npn \sort_return_same:
    #1 \__sort_return_mark:w #2 \s__sort_mark
  {
    #1
    #2
    \__sort_return_two_error:
    \__sort_return_mark:w
    \s__sort_mark
    \__sort_return_same:w
  }
\cs_new_protected:Npn \sort_return_swapped:
    #1 \__sort_return_mark:w #2 \s__sort_mark
  {
    #1
    #2
    \__sort_return_two_error:
    \__sort_return_mark:w
    \s__sort_mark
    \__sort_return_swapped:w
  }
\cs_new_protected:Npn \__sort_return_mark:w #1 \s__sort_mark { }
\cs_new_protected:Npn \__sort_return_none_error:
  {
    \msg_error:nnee { sort } { return-none }
      { \tex_the:D \tex_toks:D \l__sort_A_int }
      { \tex_the:D \tex_toks:D \l__sort_C_int }
    \__sort_return_same:w \__sort_return_none_error:
  }
\cs_new_protected:Npn \__sort_return_two_error:
  {
    \msg_error:nnee { sort } { return-two }
      { \tex_the:D \tex_toks:D \l__sort_A_int }
      { \tex_the:D \tex_toks:D \l__sort_C_int }
  }
\cs_new_protected:Npn \__sort_return_same:w #1 \__sort_return_none_error:
  {
    \tex_toks:D \l__sort_B_int \tex_toks:D \l__sort_C_int
    \int_decr:N \l__sort_B_int
    \int_decr:N \l__sort_C_int
    \if_int_compare:w \l__sort_C_int < \l__sort_top_int
      \use_i:nn
    \fi:
    \__sort_merge_blocks_aux:
  }
\cs_new_protected:Npn \__sort_return_swapped:w #1 \__sort_return_none_error:
  {
    \tex_toks:D \l__sort_B_int \tex_toks:D \l__sort_A_int
    \int_decr:N \l__sort_B_int
    \int_decr:N \l__sort_A_int
    \if_int_compare:w \l__sort_A_int < \l__sort_begin_int
      \__sort_merge_blocks_end: \use_i:nn
    \fi:
    \__sort_merge_blocks_aux:
  }
\cs_new_protected:Npn \__sort_merge_blocks_end:
  {
    \tex_toks:D \l__sort_B_int \tex_toks:D \l__sort_C_int
    \int_decr:N \l__sort_B_int
    \int_decr:N \l__sort_C_int
    \if_int_compare:w \l__sort_B_int < \l__sort_begin_int
      \use_i:nn
    \fi:
    \__sort_merge_blocks_end:
  }
```

```
\cs_new:Npn \tl_sort:nN #1#2
  {
    \exp_not:f
      {
        \tl_if_blank:nF {#1}
          {
            \__sort_quick_prepare:Nnnn #2 { } { }
              #1
              { \prg_break_point: \__sort_quick_prepare_end:NNNnw }
            \s__sort_stop
          }
      }
  }
\cs_new:Npn \__sort_quick_prepare:Nnnn #1#2#3#4
  {
    \prg_break: #4 \prg_break_point:
    \__sort_quick_prepare:Nnnn #1 { #2 #3 } { #1 {#4} }
  }
\cs_new:Npn \__sort_quick_prepare_end:NNNnw #1#2#3#4#5 \s__sort_stop
  {
    \__sort_quick_split:NnNn #4 \__sort_quick_end:nnTFNn { }
    \s__sort_mark { \__sort_quick_cleanup:w \exp_stop_f: }
    \s__sort_mark \s__sort_stop
  }
\cs_new:Npn \__sort_quick_cleanup:w #1 \s__sort_mark \s__sort_stop {#1}
\cs_new:Npn \__sort_quick_split:NnNn #1#2#3#4
  {
    #3 {#2} {#4} \__sort_quick_only_ii:NnnnnNn
      \__sort_quick_only_i:NnnnnNn
      \__sort_quick_single_end:nnnwnw
      { #3 {#4} } { } { } {#2}
  }
\cs_new:Npn \__sort_quick_only_i:NnnnnNn #1#2#3#4#5#6#7
  {
    #6 {#5} {#7} \__sort_quick_split_ii:NnnnnNn
      \__sort_quick_only_i:NnnnnNn
      \__sort_quick_only_i_end:nnnwnw
      { #6 {#7} } { #3 #2 } { } {#5}
  }
\cs_new:Npn \__sort_quick_only_ii:NnnnnNn #1#2#3#4#5#6#7
  {
    #6 {#5} {#7} \__sort_quick_only_ii:NnnnnNn
      \__sort_quick_split_i:NnnnnNn
      \__sort_quick_only_ii_end:nnnwnw
      { #6 {#7} } { } { #4 #2 } {#5}
  }
```

```
\cs_new:Npn \__sort_quick_split_i:NnnnnNn #1#2#3#4#5#6#7
  {
    #6 {#5} {#7} \__sort_quick_split_ii:NnnnnNn
      \__sort_quick_split_i:NnnnnNn
      \__sort_quick_split_end:nnnwnw
      { #6 {#7} } { #3 #2 } {#4} {#5}
  }
\cs_new:Npn \__sort_quick_split_ii:NnnnnNn #1#2#3#4#5#6#7
  {
    #6 {#5} {#7} \__sort_quick_split_ii:NnnnnNn
      \__sort_quick_split_i:NnnnnNn
      \__sort_quick_split_end:nnnwnw
      { #6 {#7} } {#3} { #4 #2 } {#5}
  }
\cs_new:Npn \__sort_quick_end:nnTFNn #1#2#3#4#5#6 {#5}
\cs_new:Npn \__sort_quick_single_end:nnnwnw #1#2#3#4 \s__sort_mark #5#6 \s__sort_stop
  { #5 {#3} #6 \s__sort_stop }
\cs_new:Npn \__sort_quick_only_i_end:nnnwnw #1#2#3#4 \s__sort_mark #5#6 \s__sort_stop
  {
    \__sort_quick_split:NnNn #1
      \__sort_quick_end:nnTFNn { } \s__sort_mark {#5}
    {#3}
    #6 \s__sort_stop
  }
\cs_new:Npn \__sort_quick_only_ii_end:nnnwnw #1#2#3#4 \s__sort_mark #5#6 \s__sort_stop
  {
    \__sort_quick_split:NnNn #2
      \__sort_quick_end:nnTFNn { } \s__sort_mark { #5 {#3} }
    #6 \s__sort_stop
  }
\cs_new:Npn \__sort_quick_split_end:nnnwnw #1#2#3#4 \s__sort_mark #5#6 \s__sort_stop
  {
    \__sort_quick_split:NnNn #2 \__sort_quick_end:nnTFNn { } \s__sort_mark
      {
        \__sort_quick_split:NnNn #1
          \__sort_quick_end:nnTFNn { } \s__sort_mark {#5}
        {#3}
      }
    #6 \s__sort_stop
  }
\cs_new_protected:Npn \__sort_error:
  {
    \cs_set_eq:NN \__sort_merge_blocks_aux: \prg_do_nothing:
    \cs_set_eq:NN \__sort_merge_blocks: \prg_do_nothing:
    \cs_set_protected:Npn \__sort_level: { \group_end: \prg_break: }
  }
```

```
\cs_new_protected:Npn \__sort_disable_toksdef:
  { \cs_set_eq:NN \toksdef \__sort_disabled_toksdef:n }
\cs_new_protected:Npn \__sort_disabled_toksdef:n #1
  {
    \msg_error:nne { sort } { toksdef }
      { \token_to_str:N #1 }
    \__sort_error:
    \tex_toksdef:D #1
  }
\msg_new:nnnn { sort } { toksdef }
  { Allocation~of~\iow_char:N\\toks~registers~impossible~while~sorting. }
  {
    The~comparison~code~used~for~sorting~a~list~has~attempted~to~
    define~#1~as~a~new~\iow_char:N\\toks~register~using~
    \iow_char:N\\newtoks~
    or~a~similar~command.~The~list~will~not~be~sorted.
  }
\cs_new_protected:Npn \__sort_too_long_error:NNw #1#2 \fi:
  {
    \fi:
    \msg_error:nneee { sort } { too-large }
      { \token_to_str:N #2 }
      { \int_eval:n { \l__sort_true_max_int - \l__sort_min_int } }
      { \int_eval:n { \l__sort_top_int - \l__sort_min_int } }
    #1 \__sort_error:
  }
\msg_new:nnnn { sort } { too-large }
  { The~list~#1~is~too~long~to~be~sorted~by~TeX. }
  {
    TeX~has~#2~toks~registers~still~available:~
    this~only~allows~to~sort~with~up~to~#3~
    items.~The~list~will~not~be~sorted.
  }
\msg_new:nnnn { sort } { return-none }
  { The~comparison~code~did~not~return. }
  {
    When~sorting~a~list,~the~code~to~compare~items~#1~and~#2~
    did~not~call~
    \iow_char:N\\sort_return_same: ~nor~
    \iow_char:N\\sort_return_swapped: .~
    Exactly~one~of~these~should~be~called.
  }
\msg_new:nnnn { sort } { return-two }
  { The~comparison~code~returned~multiple~times. }
  {
    When~sorting~a~list,~the~code~to~compare~items~#1~and~#2~called~
    \iow_char:N\\sort_return_same: ~or~
    \iow_char:N\\sort_return_swapped: ~multiple~times.~
    Exactly~one~of~these~should~be~called.
  }
\prop_gput:Nnn \g_msg_module_name_prop { sort } { LaTeX }
\prop_gput:Nnn \g_msg_module_type_prop { sort } { }
%% File: l3str-convert.dtx
\cs_new_protected:Npn \__str_tmp:w { }
\tl_new:N \l__str_internal_tl
\tl_new:N \g__str_result_tl
\int_const:Nn \c__str_replacement_char_int { "FFFD }
\int_const:Nn \c__str_max_byte_int { 255 }
\scan_new:N \s__str
\quark_new:N \q__str_nil
\prop_new:N \g__str_alias_prop
\prop_gput:Nnn \g__str_alias_prop { latin1 } { iso88591 }
\prop_gput:Nnn \g__str_alias_prop { latin2 } { iso88592 }
\prop_gput:Nnn \g__str_alias_prop { latin3 } { iso88593 }
\prop_gput:Nnn \g__str_alias_prop { latin4 } { iso88594 }
\prop_gput:Nnn \g__str_alias_prop { latin5 } { iso88599 }
\prop_gput:Nnn \g__str_alias_prop { latin6 } { iso885910 }
\prop_gput:Nnn \g__str_alias_prop { latin7 } { iso885913 }
\prop_gput:Nnn \g__str_alias_prop { latin8 } { iso885914 }
\prop_gput:Nnn \g__str_alias_prop { latin9 } { iso885915 }
\prop_gput:Nnn \g__str_alias_prop { latin10 } { iso885916 }
\prop_gput:Nnn \g__str_alias_prop { utf16le } { utf16 }
\prop_gput:Nnn \g__str_alias_prop { utf16be } { utf16 }
\prop_gput:Nnn \g__str_alias_prop { utf32le } { utf32 }
\prop_gput:Nnn \g__str_alias_prop { utf32be } { utf32 }
\prop_gput:Nnn \g__str_alias_prop { hexadecimal } { hex }
\bool_lazy_any:nTF
  {
    \sys_if_engine_luatex_p:
    \sys_if_engine_xetex_p:
  }
  {
    \prop_gput:Nnn \g__str_alias_prop { default } {  }
  }
  {
    \prop_gput:Nnn \g__str_alias_prop { default } { utf8 }
  }
\bool_new:N \g__str_error_bool
\flag_new:N \l__str_byte_flag
\flag_new:N \l__str_error_flag
\prg_new_conditional:Npnn \__str_if_contains_char:Nn #1#2 { T , TF }
  {
    \exp_after:wN \__str_if_contains_char_aux:nn \exp_after:wN {#1} {#2}
      { \prg_break:n { ? \fi: } }
    \prg_break_point:
    \prg_return_false:
  }
\cs_new:Npn \__str_if_contains_char_aux:nn #1#2
  { \__str_if_contains_char_auxi:nN {#2} #1 }
\prg_new_conditional:Npnn \__str_if_contains_char:nn #1#2 { TF }
  {
    \__str_if_contains_char_auxi:nN {#2} #1 { \prg_break:n { ? \fi: } }
    \prg_break_point:
    \prg_return_false:
  }
\cs_new:Npn \__str_if_contains_char_auxi:nN #1#2
  {
    \if_charcode:w #1 #2
      \exp_after:wN \__str_if_contains_char_true:
    \fi:
    \__str_if_contains_char_auxi:nN {#1}
  }
\cs_new:Npn \__str_if_contains_char_true:
  { \prg_break:n { \prg_return_true: \use_none:n } }
\prg_new_conditional:Npnn \__str_octal_use:N #1 { TF }
  {
    \if_int_compare:w 1 < '1 \token_to_str:N #1 \exp_stop_f:
      #1 \prg_return_true:
    \else:
      \prg_return_false:
    \fi:
  }
\prg_new_conditional:Npnn \__str_hexadecimal_use:N #1 { TF }
  {
    \if_int_compare:w 1 < "1 \token_to_str:N #1 \exp_stop_f:
      #1 \prg_return_true:
    \else:
      \if_case:w \int_eval:n { \exp_after:wN ` \token_to_str:N #1 - `a }
           A
      \or: B
      \or: C
      \or: D
      \or: E
      \or: F
      \else:
        \prg_return_false:
        \exp_after:wN \use_none:n
      \fi:
```

```
      \prg_return_true:
    \fi:
  }
\group_begin:
  \__kernel_tl_set:Nx \l__str_internal_tl { \tl_to_str:n { 0123456789ABCDEF } }
  \tl_map_inline:Nn \l__str_internal_tl
    {
      \tl_map_inline:Nn \l__str_internal_tl
        {
          \tl_const:ce { c__str_byte_ \int_eval:n {"#1##1} _tl }
            { \char_generate:nn { "#1##1 } { 12 } #1 ##1 }
        }
    }
\group_end:
\tl_const:cn { c__str_byte_-1_tl } { { } \use_none:n { } }
\cs_new:Npn \__str_output_byte:n #1
  { \__str_output_byte:w #1 \__str_output_end: }
\cs_new:Npn \__str_output_byte:w
  {
    \exp_after:wN \exp_after:wN
    \exp_after:wN \use_i:nnn
    \cs:w c__str_byte_ \int_eval:w
  }
\cs_new:Npn \__str_output_hexadecimal:n #1
  {
    \exp_after:wN \exp_after:wN
    \exp_after:wN \use_none:n
    \cs:w c__str_byte_ \int_eval:n {#1} _tl \cs_end:
  }
\cs_new:Npn \__str_output_end:
  { \scan_stop: _tl \cs_end: }
\cs_new:Npn \__str_output_byte_pair_be:n #1
  {
    \exp_args:Nf \__str_output_byte_pair:nnN
      { \int_div_truncate:nn { #1 } { "100 } } {#1} \use:nn
  }
\cs_new:Npn \__str_output_byte_pair_le:n #1
  {
    \exp_args:Nf \__str_output_byte_pair:nnN
      { \int_div_truncate:nn { #1 } { "100 } } {#1} \use_ii_i:nn
  }
\cs_new:Npn \__str_output_byte_pair:nnN #1#2#3
  {
    #3
      { \__str_output_byte:n { #1 } }
      { \__str_output_byte:n { #2 - #1 * "100 } }
  }
\cs_new_protected:Npn \__str_convert_gmap:N #1
  {
    \__kernel_tl_gset:Nx \g__str_result_tl
      {
        \exp_after:wN \__str_convert_gmap_loop:NN
        \exp_after:wN #1
          \g__str_result_tl { ? \prg_break: }
        \prg_break_point:
      }
  }
\cs_new:Npn \__str_convert_gmap_loop:NN #1#2
  {
    \use_none:n #2
    #1#2
    \__str_convert_gmap_loop:NN #1
  }
\cs_new_protected:Npn \__str_convert_gmap_internal:N #1
  {
    \__kernel_tl_gset:Nx \g__str_result_tl
      {
        \exp_after:wN \__str_convert_gmap_internal_loop:Nww
        \exp_after:wN #1
          \g__str_result_tl \s__str \s__str_stop \prg_break: \s__str
        \prg_break_point:
      }
  }
\cs_new:Npn \__str_convert_gmap_internal_loop:Nww #1 #2 \s__str #3 \s__str
  {
    \__str_use_none_delimit_by_s_stop:w #3 \s__str_stop
    #1 {#3}
    \__str_convert_gmap_internal_loop:Nww #1
  }
\cs_new_protected:Npn \__str_if_flag_error:Nne #1
  {
    \flag_if_raised:NTF #1
      { \msg_error:nne { str } }
      { \use_none:nn }
  }
\cs_new_protected:Npn \__str_if_flag_no_error:Nne #1#2#3
  { \flag_if_raised:NT #1 { \bool_gset_true:N \g__str_error_bool } }
\cs_new:Npn \__str_if_flag_times:NT #1#2
  { \flag_if_raised:NT #1 { #2~(x \flag_height:N #1 ) } }
\cs_new_protected:Npn \str_set_convert:Nnnn
  { \__str_convert:nNNnnn { } \tl_set_eq:NN }
\cs_new_protected:Npn \str_gset_convert:Nnnn
```

```
  { \__str_convert:nNNnnn { } \tl_gset_eq:NN }
\prg_new_protected_conditional:Npnn
    \str_set_convert:Nnnn #1#2#3#4 { T , F , TF }
  {
    \bool_gset_false:N \g__str_error_bool
    \__str_convert:nNNnnn
      { \cs_set_eq:NN \__str_if_flag_error:Nne \__str_if_flag_no_error:Nne }
      \tl_set_eq:NN #1 {#2} {#3} {#4}
    \bool_if:NTF \g__str_error_bool \prg_return_false: \prg_return_true:
  }
\prg_new_protected_conditional:Npnn
    \str_gset_convert:Nnnn #1#2#3#4 { T , F , TF }
  {
    \bool_gset_false:N \g__str_error_bool
    \__str_convert:nNNnnn
      { \cs_set_eq:NN \__str_if_flag_error:Nne \__str_if_flag_no_error:Nne }
      \tl_gset_eq:NN #1 {#2} {#3} {#4}
    \bool_if:NTF \g__str_error_bool \prg_return_false: \prg_return_true:
  }
\cs_new_protected:Npn \__str_convert:nNNnnn #1#2#3#4#5#6
  {
    \group_begin:
      #1
      \__kernel_tl_gset:Nx \g__str_result_tl { \__kernel_str_to_other_fast:n {#4} }
      \exp_after:wN \__str_convert:wwwnn
        \tl_to_str:n {#5} /// \s__str_stop
        { decode } { unescape }
        \prg_do_nothing:
        \__str_convert_decode_:
      \exp_after:wN \__str_convert:wwwnn
        \tl_to_str:n {#6} /// \s__str_stop
        { encode } { escape }
        \use_ii_i:nn
        \__str_convert_encode_:
        \__kernel_tl_gset:Nx \g__str_result_tl
          { \tl_to_str:V \g__str_result_tl }
    \group_end:
    #2 #3 \g__str_result_tl
  }
\cs_new_protected:Npn \__str_convert:wwwnn
    #1 / #2 // #3 \s__str_stop #4#5
  {
    \__str_convert:nnn {enc} {#4} {#1}
    \__str_convert:nnn {esc} {#5} {#2}
    \exp_args:Ncc \__str_convert:NNnNN
      { __str_convert_#4_#1: } { __str_convert_#5_#2: } {#2}
  }
\cs_new_protected:Npn \__str_convert:NNnNN #1#2#3#4#5
  {
    \if_meaning:w #1 #5
      \tl_if_empty:nF {#3}
        { \msg_error:nne { str } { native-escaping } {#3} }
      #1
    \else:
      #4 #2 #1
    \fi:
  }
\cs_new_protected:Npn \__str_convert:nnn #1#2#3
  {
    \cs_if_exist:cF { __str_convert_#2_#3: }
      {
        \exp_args:Ne \__str_convert:nnnn
          { \__str_convert_lowercase_alphanum:n {#3} }
          {#1} {#2} {#3}
      }
  }
\cs_new_protected:Npn \__str_convert:nnnn #1#2#3#4
  {
    \cs_if_exist:cF { __str_convert_#3_#1: }
      {
        \prop_get:NnNF \g__str_alias_prop {#1} \l__str_internal_tl
          { \tl_set:Nn \l__str_internal_tl {#1} }
        \cs_if_exist:cF { __str_convert_#3_ \l__str_internal_tl : }
          {
            \file_if_exist:nTF { l3str-#2- \l__str_internal_tl .def }
              {
                \group_begin:
                  \cctab_select:N \c_code_cctab
                  \file_input:n { l3str-#2- \l__str_internal_tl .def }
                \group_end:
              }
              {
                \tl_clear:N \l__str_internal_tl
                \msg_error:nnee { str } { unknown-#2 } {#4} {#1}
              }
          }
        \cs_if_exist:cF { __str_convert_#3_#1: }
          {
            \cs_gset_eq:cc { __str_convert_#3_#1: }
              { __str_convert_#3_ \l__str_internal_tl : }
          }
      }
    \cs_gset_eq:cc { __str_convert_#3_#4: } { __str_convert_#3_#1: }
  }
\cs_new:Npn \__str_convert_lowercase_alphanum:n #1
  {
    \exp_after:wN \__str_convert_lowercase_alphanum_loop:N
      \tl_to_str:n {#1} { ? \prg_break: }
    \prg_break_point:
  }
\cs_new:Npn \__str_convert_lowercase_alphanum_loop:N #1
  {
    \use_none:n #1
    \if_int_compare:w `#1 > `Z \exp_stop_f:
      \if_int_compare:w `#1 > `z \exp_stop_f: \else:
        \if_int_compare:w `#1 < `a \exp_stop_f: \else:
          #1
        \fi:
      \fi:
    \else:
      \if_int_compare:w `#1 < `A \exp_stop_f:
        \if_int_compare:w 1 < 1#1 \exp_stop_f:
          #1
        \fi:
      \else:
        \__str_output_byte:n { `#1 + `a - `A }
      \fi:
    \fi:
    \__str_convert_lowercase_alphanum_loop:N
  }
\bool_lazy_any:nTF
  {
    \sys_if_engine_luatex_p:
    \sys_if_engine_xetex_p:
  }
  {
    \cs_new:Npn \__str_filter_bytes:n #1
      {
        \__str_filter_bytes_aux:N #1
          { ? \prg_break: }
        \prg_break_point:
      }
    \cs_new:Npn \__str_filter_bytes_aux:N #1
      {
        \use_none:n #1
        \if_int_compare:w `#1 < 256 \exp_stop_f:
          #1
        \else:
          \flag_raise:N \l__str_byte_flag
        \fi:
        \__str_filter_bytes_aux:N
      }
  }
  { \cs_new_eq:NN \__str_filter_bytes:n \use:n }
\bool_lazy_any:nTF
  {
    \sys_if_engine_luatex_p:
    \sys_if_engine_xetex_p:
  }
  {
    \cs_new_protected:Npn \__str_convert_unescape_:
      {
        \flag_clear:N \l__str_byte_flag
        \__kernel_tl_gset:Nx \g__str_result_tl
          { \exp_args:No \__str_filter_bytes:n \g__str_result_tl }
        \__str_if_flag_error:Nne \l__str_byte_flag { non-byte } { bytes }
      }
  }
  { \cs_new_protected:Npn \__str_convert_unescape_: { } }
\cs_new_eq:NN \__str_convert_unescape_bytes: \__str_convert_unescape_:
\cs_new_protected:Npn \__str_convert_escape_: { }
\cs_new_eq:NN \__str_convert_escape_bytes: \__str_convert_escape_:
\cs_new_protected:Npn \__str_convert_decode_:
  { \__str_convert_gmap:N \__str_decode_native_char:N }
\cs_new:Npn \__str_decode_native_char:N #1
  { #1 \s__str \int_value:w `#1 \s__str }
\bool_lazy_any:nTF
  {
    \sys_if_engine_luatex_p:
    \sys_if_engine_xetex_p:
  }
  {
    \cs_new_protected:Npn \__str_convert_encode_:
      { \__str_convert_gmap_internal:N \__str_encode_native_char:n }
    \cs_new:Npn \__str_encode_native_char:n #1
      { \char_generate:nn {#1} {12} }
  }
  {
    \cs_new_protected:Npn \__str_convert_encode_:
      {
        \flag_clear:N \l__str_error_flag
        \__str_convert_gmap_internal:N \__str_encode_native_char:n
        \__str_if_flag_error:Nne \l__str_error_flag
          { native-overflow } { }
      }
    \cs_new:Npn \__str_encode_native_char:n #1
      {
        \if_int_compare:w #1 > \c__str_max_byte_int
          \flag_raise:N \l__str_error_flag
          ?
        \else:
          \char_generate:nn {#1} {12}
        \fi:
      }
    \msg_new:nnnn { str } { native-overflow }
      { Character~code~too~large~for~this~engine. }
      {
        This~engine~only~support~8-bit~characters:~
        valid~character~codes~are~in~the~range~[0,255].~
        To~manipulate~arbitrary~Unicode,~use~LuaTeX~or~XeTeX.
      }
  }
\cs_new_protected:Npn \__str_convert_decode_clist:
  {
    \clist_gset:No \g__str_result_tl \g__str_result_tl
    \__kernel_tl_gset:Nx \g__str_result_tl
      {
        \exp_args:No \clist_map_function:nN
          \g__str_result_tl \__str_decode_clist_char:n
      }
  }
\cs_new:Npn \__str_decode_clist_char:n #1
  { #1 \s__str \int_eval:n {#1} \s__str }
\cs_new_protected:Npn \__str_convert_encode_clist:
  {
    \__str_convert_gmap_internal:N \__str_encode_clist_char:n
    \__kernel_tl_gset:Nx \g__str_result_tl { \tl_tail:N \g__str_result_tl }
  }
\cs_new:Npn \__str_encode_clist_char:n #1 { , #1 }
\cs_new_protected:Npn \__str_declare_eight_bit_encoding:nnnn #1
  {
    \tl_set:Nn \l__str_internal_tl {#1}
    \cs_new_protected:cpn { __str_convert_decode_#1: }
      { \__str_convert_decode_eight_bit:n {#1} }
    \cs_new_protected:cpn { __str_convert_encode_#1: }
      { \__str_convert_encode_eight_bit:n {#1} }
    \exp_args:Ncc \__str_declare_eight_bit_aux:NNnnn
      { g__str_decode_#1_intarray } { g__str_encode_#1_intarray }
  }
\cs_new_protected:Npn \__str_declare_eight_bit_aux:NNnnn #1#2#3#4#5
```

```
  {
    \intarray_new:Nn #1 { 256 }
    \int_step_inline:nnn { 0 } { 255 }
      { \intarray_gset:Nnn #1 { 1 + ##1 } {##1} }
    \__str_declare_eight_bit_loop:Nnn #1
      #4 { \s__str_stop \prg_break: } { }
    \prg_break_point:
    \__str_declare_eight_bit_loop:Nn #1
      #5 { \s__str_stop \prg_break: }
    \prg_break_point:
    \intarray_new:Nn #2 {#3}
    \int_step_inline:nnn { 0 } { 255 }
      {
        \int_compare:nNnF { \intarray_item:Nn #1 { 1 + ##1 } } = { -1 }
          {
            \intarray_gset:Nnn #2
              {
                1 +
                \int_mod:nn { \intarray_item:Nn #1 { 1 + ##1 } }
                  { \intarray_count:N #2 }
              }
              {##1}
          }
      }
  }
\cs_new_protected:Npn \__str_declare_eight_bit_loop:Nnn #1#2#3
  {
    \__str_use_none_delimit_by_s_stop:w #2 \s__str_stop
    \intarray_gset:Nnn #1 { 1 + "#2 } { "#3 }
    \__str_declare_eight_bit_loop:Nnn #1
  }
\cs_new_protected:Npn \__str_declare_eight_bit_loop:Nn #1#2
  {
    \__str_use_none_delimit_by_s_stop:w #2 \s__str_stop
    \intarray_gset:Nnn #1 { 1 + "#2 } { -1 }
    \__str_declare_eight_bit_loop:Nn #1
  }
\cs_new_protected:Npn \__str_convert_decode_eight_bit:n #1
  {
    \cs_set:Npe \__str_tmp:w
      {
        \exp_not:N \__str_decode_eight_bit_aux:Nn
        \exp_not:c { g__str_decode_#1_intarray }
      }
    \flag_clear:N \l__str_error_flag
    \__str_convert_gmap:N \__str_tmp:w
    \__str_if_flag_error:Nne \l__str_error_flag { decode-8-bit } {#1}
  }
\cs_new:Npn \__str_decode_eight_bit_aux:Nn #1#2
  {
    #2 \s__str
    \exp_args:Nf \__str_decode_eight_bit_aux:n
      { \intarray_item:Nn #1 { 1 + `#2 } }
    \s__str
  }
\cs_new:Npn \__str_decode_eight_bit_aux:n #1
  {
    \if_int_compare:w #1 < \c_zero_int
      \flag_raise:N \l__str_error_flag
      \int_value:w \c__str_replacement_char_int
    \else:
      #1
    \fi:
  }
\int_new:N \l__str_modulo_int
\cs_new_protected:Npn \__str_convert_encode_eight_bit:n #1
  {
    \cs_set:Npe \__str_tmp:w
      {
        \exp_not:N \__str_encode_eight_bit_aux:NNn
        \exp_not:c { g__str_encode_#1_intarray }
        \exp_not:c { g__str_decode_#1_intarray }
      }
    \flag_clear:N \l__str_error_flag
    \__str_convert_gmap_internal:N \__str_tmp:w
    \__str_if_flag_error:Nne \l__str_error_flag { encode-8-bit } {#1}
  }
\cs_new:Npn \__str_encode_eight_bit_aux:NNn #1#2#3
  {
    \exp_args:Nf \__str_encode_eight_bit_aux:nnN
      {
        \intarray_item:Nn #1
          { 1 + \int_mod:nn {#3} { \intarray_count:N #1 } }
      }
      {#3}
      #2
  }
\cs_new:Npn \__str_encode_eight_bit_aux:nnN #1#2#3
  {
    \int_compare:nNnTF { \intarray_item:Nn #3 { 1 + #1 } } = {#2}
      { \__str_output_byte:n {#1} }
      { \flag_raise:N \l__str_error_flag }
  }
\msg_new:nnn { str } { unknown-esc }
  { Escaping~scheme~'#1'~(filtered:~'#2')~unknown. }
\msg_new:nnn { str } { unknown-enc }
  { Encoding~scheme~'#1'~(filtered:~'#2')~unknown. }
\msg_new:nnnn { str } { native-escaping }
  { The~'native'~encoding~scheme~does~not~support~any~escaping. }
  {
    Since~native~strings~do~not~consist~in~bytes,~
    none~of~the~escaping~methods~make~sense.~
    The~specified~escaping,~'#1',~will~be~ignored.
  }
\msg_new:nnn { str } { file-not-found }
  { File~'l3str-#1.def'~not~found. }
\bool_lazy_any:nT
  {
    \sys_if_engine_luatex_p:
    \sys_if_engine_xetex_p:
  }
  {
    \msg_new:nnnn { str } { non-byte }
      { String~invalid~in~escaping~'#1':~it~may~only~contain~bytes. }
      {
        Some~characters~in~the~string~you~asked~to~convert~are~not~
        8-bit~characters.~Perhaps~the~string~is~a~'native'~Unicode~string?~
        If~it~is,~try~using\\
        \\
        \iow_indent:n
          {
            \iow_char:N\\str_set_convert:Nnnn \\
            \␣\␣<str~var>~\{~<string>~\}~\{~native~\}~\{~<target~encoding>~\}
          }
      }
  }
\msg_new:nnnn { str } { decode-8-bit }
  { Invalid~string~in~encoding~'#1'. }
  {
    LaTeX~came~across~a~byte~which~is~not~defined~to~represent~
    any~character~in~the~encoding~'#1'.
  }
\msg_new:nnnn { str } { encode-8-bit }
  { Unicode~string~cannot~be~converted~to~encoding~'#1'. }
  {
    The~encoding~'#1'~only~contains~a~subset~of~all~Unicode~characters.~
    LaTeX~was~asked~to~convert~a~string~to~that~encoding,~but~that~
    string~contains~a~character~that~'#1'~does~not~support.
  }
\cs_new_protected:Npn \__str_convert_unescape_hex:
  {
    \group_begin:
      \flag_clear:N \l__str_error_flag
      \int_set:Nn \tex_escapechar:D { 92 }
      \__kernel_tl_gset:Nx \g__str_result_tl
        {
          \__str_output_byte:w "
            \exp_last_unbraced:Nf \__str_unescape_hex_auxi:N
              { \tl_to_str:N \g__str_result_tl }
            0 { ? 0 - 1 \prg_break: }
            \prg_break_point:
          \__str_output_end:
        }
      \__str_if_flag_error:Nne \l__str_error_flag { unescape-hex } { }
    \group_end:
  }
\cs_new:Npn \__str_unescape_hex_auxi:N #1
  {
    \use_none:n #1
    \__str_hexadecimal_use:NTF #1
      { \__str_unescape_hex_auxii:N }
      {
        \flag_raise:N \l__str_error_flag
        \__str_unescape_hex_auxi:N
      }
  }
\cs_new:Npn \__str_unescape_hex_auxii:N #1
  {
    \use_none:n #1
    \__str_hexadecimal_use:NTF #1
      {
        \__str_output_end:
        \__str_output_byte:w " \__str_unescape_hex_auxi:N
      }
      {
        \flag_raise:N \l__str_error_flag
        \__str_unescape_hex_auxii:N
      }
  }
\msg_new:nnnn { str } { unescape-hex }
  { String~invalid~in~escaping~'hex':~only~hexadecimal~digits~allowed. }
  {
    Some~characters~in~the~string~you~asked~to~convert~are~not~
    hexadecimal~digits~(0-9,~A-F,~a-f)~nor~spaces.
  }
\cs_set_protected:Npn \__str_tmp:w #1#2#3
  {
    \cs_new_protected:cpn { __str_convert_unescape_#2: }
      {
        \group_begin:
          \flag_clear:N \l__str_byte_flag
          \flag_clear:N \l__str_error_flag
          \int_set:Nn \tex_escapechar:D { 92 }
          \__kernel_tl_gset:Nx \g__str_result_tl
            {
              \exp_after:wN #3 \g__str_result_tl
                #1 ? { ? \prg_break: }
              \prg_break_point:
            }
          \__str_if_flag_error:Nne \l__str_byte_flag { non-byte } { #2 }
          \__str_if_flag_error:Nne \l__str_error_flag { unescape-#2 } { }
        \group_end:
      }
    \cs_new:Npn #3 ##1#1##2##3
      {
        \__str_filter_bytes:n {##1}
        \use_none:n ##3
        \__str_output_byte:w "
          \__str_hexadecimal_use:NTF ##2
            {
              \__str_hexadecimal_use:NTF ##3
                { }
                {
                  \flag_raise:N \l__str_error_flag
                  * 0 + `#1 \use_i:nn
                }
            }
            {
              \flag_raise:N \l__str_error_flag
              0 + `#1 \use_i:nn
            }
        \__str_output_end:
        \use_i:nnn #3 ##2##3
      }
    \msg_new:nnnn { str } { unescape-#2 }
      { String~invalid~in~escaping~'#2'. }
      {
        LaTeX~came~across~the~escape~character~'#1'~not~followed~by~
        two~hexadecimal~digits.~This~is~invalid~in~the~escaping~'#2'.
      }
  }
\exp_after:wN \__str_tmp:w \c_hash_str { name }
  \__str_unescape_name_loop:wNN
\exp_after:wN \__str_tmp:w \c_percent_str { url }
  \__str_unescape_url_loop:wNN
\group_begin:
  \char_set_catcode_other:N \^^J
  \char_set_catcode_other:N \^^M
  \cs_set_protected:Npn \__str_tmp:w #1
    {
      \cs_new_protected:Npn \__str_convert_unescape_string:
        {
          \group_begin:
            \flag_clear:N \l__str_byte_flag
            \flag_clear:N \l__str_error_flag
            \int_set:Nn \tex_escapechar:D { 92 }
            \__kernel_tl_gset:Nx \g__str_result_tl
              {
                \exp_after:wN \__str_unescape_string_newlines:wN
                  \g__str_result_tl \prg_break: ^^M ?
                \prg_break_point:
              }
            \__kernel_tl_gset:Nx \g__str_result_tl
              {
                \exp_after:wN \__str_unescape_string_loop:wNNN
                  \g__str_result_tl #1 ?? { ? \prg_break: }
                \prg_break_point:
              }
            \__str_if_flag_error:Nne \l__str_byte_flag { non-byte } { string }
            \__str_if_flag_error:Nne \l__str_error_flag { unescape-string } { }
          \group_end:
        }
    }
  \exp_args:No \__str_tmp:w { \c_backslash_str }
  \exp_last_unbraced:NNNNo
    \cs_new:Npn \__str_unescape_string_loop:wNNN #1 \c_backslash_str #2#3#4
        {
          \__str_filter_bytes:n {#1}
          \use_none:n #4
          \__str_output_byte:w '
            \__str_octal_use:NTF #2
              {
                \__str_octal_use:NTF #3
                  {
                    \__str_octal_use:NTF #4
                      {
```

```
                      \if_int_compare:w #2 > 3 \exp_stop_f:
                        - 256
                      \fi:
                      \__str_unescape_string_repeat:NNNNNN
                    }
                    { \__str_unescape_string_repeat:NNNNNN ? }
                }
                { \__str_unescape_string_repeat:NNNNNN ?? }
            }
            {
              \str_case_e:nnF {#2}
                {
                  { \c_backslash_str } { 134 }
                  { ( } { 50 }
                  { ) } { 51 }
                  { r } { 15 }
                  { f } { 14 }
                  { n } { 12 }
                  { t } { 11 }
                  { b } { 10 }
                  { ^^J } { 0 - 1 }
                }
                {
                  \flag_raise:N \l__str_error_flag
                  0 - 1 \use_i:nn
                }
            }
        \__str_output_end:
        \use_i:nn \__str_unescape_string_loop:wNNN #2#3#4
      }
\cs_new:Npn \__str_unescape_string_repeat:NNNNNN #1#2#3#4#5#6
  { \__str_output_end: \__str_unescape_string_loop:wNNN }
\cs_new:Npn \__str_unescape_string_newlines:wN #1 ^^M #2
  {
    #1
    \if_charcode:w ^^J #2 \else: ^^J \fi:
    \__str_unescape_string_newlines:wN #2
  }
\msg_new:nnnn { str } { unescape-string }
  { String~invalid~in~escaping~'string'. }
  {
    LaTeX~came~across~an~escape~character~'\c_backslash_str'~
    not~followed~by~any~of:~'n',~'r',~'t',~'b',~'f',~'(',~')',~
    '\c_backslash_str',~one~to~three~octal~digits,~or~the~end~
    of~a~line.
  }
```

```
\group_end:
\cs_new_protected:Npn \__str_convert_escape_hex:
  { \__str_convert_gmap:N \__str_escape_hex_char:N }
\cs_new:Npn \__str_escape_hex_char:N #1
  { \__str_output_hexadecimal:n { `#1 } }
\str_const:Nn \c__str_escape_name_not_str { ! " $ & ' } %$
\str_const:Nn \c__str_escape_name_str { {}/<>[] }
\cs_new_protected:Npn \__str_convert_escape_name:
  { \__str_convert_gmap:N \__str_escape_name_char:n }
\cs_new:Npn \__str_escape_name_char:n #1
  {
    \__str_if_escape_name:nTF {#1} {#1}
      { \c_hash_str \__str_output_hexadecimal:n {`#1} }
  }
\prg_new_conditional:Npnn \__str_if_escape_name:n #1 { TF }
  {
    \if_int_compare:w `#1 < "2A \exp_stop_f:
      \__str_if_contains_char:NnTF \c__str_escape_name_not_str {#1}
        \prg_return_true: \prg_return_false:
    \else:
      \if_int_compare:w `#1 > "7E \exp_stop_f:
        \prg_return_false:
      \else:
        \__str_if_contains_char:NnTF \c__str_escape_name_str {#1}
          \prg_return_false: \prg_return_true:
      \fi:
    \fi:
  }
\str_const:Ne \c__str_escape_string_str
  { \c_backslash_str ( ) }
\cs_new_protected:Npn \__str_convert_escape_string:
  { \__str_convert_gmap:N \__str_escape_string_char:N }
\cs_new:Npn \__str_escape_string_char:N #1
  {
    \__str_if_escape_string:NTF #1
      {
        \__str_if_contains_char:NnT
          \c__str_escape_string_str {#1}
          { \c_backslash_str }
        #1
      }
      {
        \c_backslash_str
        \int_div_truncate:nn {`#1} {64}
        \int_mod:nn { \int_div_truncate:nn {`#1} { 8 } } { 8 }
        \int_mod:nn {`#1} { 8 }
      }
  }
\prg_new_conditional:Npnn \__str_if_escape_string:N #1 { TF }
  {
    \if_int_compare:w `#1 < "27 \exp_stop_f:
      \prg_return_false:
    \else:
      \if_int_compare:w `#1 > "7A \exp_stop_f:
        \prg_return_false:
      \else:
        \prg_return_true:
      \fi:
    \fi:
  }
\cs_new_protected:Npn \__str_convert_escape_url:
  { \__str_convert_gmap:N \__str_escape_url_char:n }
\cs_new:Npn \__str_escape_url_char:n #1
  {
    \__str_if_escape_url:nTF {#1} {#1}
      { \c_percent_str \__str_output_hexadecimal:n { `#1 } }
  }
\prg_new_conditional:Npnn \__str_if_escape_url:n #1 { TF }
  {
    \if_int_compare:w `#1 < "30 \exp_stop_f:
      \__str_if_contains_char:nnTF { "-. } {#1}
        \prg_return_true: \prg_return_false:
    \else:
      \if_int_compare:w `#1 > "7E \exp_stop_f:
        \prg_return_false:
      \else:
        \__str_if_contains_char:nnTF { : ; = ? @ [ ] } {#1}
          \prg_return_false: \prg_return_true:
      \fi:
    \fi:
  }
\cs_new_protected:cpn { __str_convert_encode_utf8: }
  { \__str_convert_gmap_internal:N \__str_encode_utf_viii_char:n }
\cs_new:Npn \__str_encode_utf_viii_char:n #1
  {
    \__str_encode_utf_viii_loop:wwnnw #1 \__str_sep: - 1 + 0 * \__str_sep:
      { 128 } {       0 }
      {  32 } {     192 }
      {  16 } {     224 }
      {   8 } {     240 }
    \s__str_stop
  }
```

```
\cs_new:Npn \__str_encode_utf_viii_loop:wwnnw
  #1 \__str_sep: #2 \__str_sep: #3#4 #5 \s__str_stop
  {
    \if_int_compare:w #1 < #3 \exp_stop_f:
      \__str_output_byte:n { #1 + #4 }
      \exp_after:wN \__str_use_none_delimit_by_s_stop:w
    \fi:
    \exp_after:wN \__str_encode_utf_viii_loop:wwnnw
      \int_value:w \int_div_truncate:nn {#1} {64} \__str_sep: #1 \__str_sep:
      #5 \s__str_stop
    \__str_output_byte:n { #2 - 64 * ( #1 - 2 ) }
  }
\flag_clear_new:N \l__str_missing_flag
\flag_clear_new:N \l__str_extra_flag
\flag_clear_new:N \l__str_overlong_flag
\flag_clear_new:N \l__str_overflow_flag
\msg_new:nnnn { str } { utf8-decode }
  {
    Invalid~UTF-8~string:
    \exp_last_unbraced:Nf \use_none:n
      {
        \__str_if_flag_times:NT \l__str_missing_flag  { ,~missing~continuation~byte }
        \__str_if_flag_times:NT \l__str_extra_flag    { ,~extra~continuation~byte }
        \__str_if_flag_times:NT \l__str_overlong_flag { ,~overlong~form }
        \__str_if_flag_times:NT \l__str_overflow_flag { ,~code~point~too~large }
      }
    .
  }
  {
    In~the~UTF-8~encoding,~each~Unicode~character~consists~in~
    1~to~4~bytes,~with~the~following~bit~pattern: \\
    \iow_indent:n
      {
        Code~point~\␣\␣\␣\␣<~128:~0xxxxxxx \\
        Code~point~\␣\␣\␣ <~2048:~110xxxxx~10xxxxxx \\
        Code~point~\␣\␣  <~65536:~1110xxxx~10xxxxxx~10xxxxxx \\
        Code~point~    <~1114112:~11110xxx~10xxxxxx~10xxxxxx~10xxxxxx \\
      }
    Bytes~of~the~form~10xxxxxx~are~called~continuation~bytes.
    \flag_if_raised:NT \l__str_missing_flag
      {
        \\\\
        A~leading~byte~(in~the~range~[192,255])~was~not~followed~by~
        the~appropriate~number~of~continuation~bytes.
      }
    \flag_if_raised:NT \l__str_extra_flag
```

```
      {
        \\\\
        LaTeX~came~across~a~continuation~byte~when~it~was~not~expected.
      }
    \flag_if_raised:NT \l__str_overlong_flag
      {
        \\\\
        Every~Unicode~code~point~must~be~expressed~in~the~shortest~
        possible~form.~For~instance,~'0xC0'~'0x83'~is~not~a~valid~
        representation~for~the~code~point~3.
      }
    \flag_if_raised:NT \l__str_overflow_flag
      {
        \\\\
        Unicode~limits~code~points~to~the~range~[0,1114111].
      }
  }
\prop_gput:Nnn \g_msg_module_name_prop { str } { LaTeX }
\prop_gput:Nnn \g_msg_module_type_prop { str } { }
\cs_new_protected:cpn { __str_convert_decode_utf8: }
  {
    \flag_clear:N \l__str_error_flag
    \flag_clear:N \l__str_missing_flag
    \flag_clear:N \l__str_extra_flag
    \flag_clear:N \l__str_overlong_flag
    \flag_clear:N \l__str_overflow_flag
    \__kernel_tl_gset:Nx \g__str_result_tl
      {
        \exp_after:wN \__str_decode_utf_viii_start:N \g__str_result_tl
          { \prg_break: \__str_decode_utf_viii_end: }
        \prg_break_point:
      }
    \__str_if_flag_error:Nne \l__str_error_flag { utf8-decode } { }
  }
\cs_new:Npn \__str_decode_utf_viii_start:N #1
  {
    #1
    \if_int_compare:w `#1 < "C0 \exp_stop_f:
      \s__str
      \if_int_compare:w `#1 < "80 \exp_stop_f:
        \int_value:w `#1
      \else:
        \flag_raise:N \l__str_extra_flag
        \flag_raise:N \l__str_error_flag
        \int_use:N \c__str_replacement_char_int
      \fi:
```

```
    \else:
      \exp_after:wN \__str_decode_utf_viii_continuation:wwN
      \int_value:w \int_eval:n { `#1 - "C0 } \exp_after:wN
    \fi:
    \s__str
    \__str_use_none_delimit_by_s_stop:w {"80} {"800} {"10000} {"110000} \s__str_stop
    \__str_decode_utf_viii_start:N
  }
\cs_new:Npn \__str_decode_utf_viii_continuation:wwN
    #1 \s__str #2 \__str_decode_utf_viii_start:N #3
  {
    \use_none:n #3
    \if_int_compare:w `#3 <
          \if_int_compare:w `#3 < "80 \exp_stop_f: - \fi:
          "C0 \exp_stop_f:
      #3
      \exp_after:wN \__str_decode_utf_viii_aux:wNnnwN
      \int_value:w \int_eval:n { #1 * "40 + `#3 - "80 } \exp_after:wN
    \else:
      \s__str
      \flag_raise:N \l__str_missing_flag
      \flag_raise:N \l__str_error_flag
      \int_use:N \c__str_replacement_char_int
    \fi:
    \s__str
    #2
    \__str_decode_utf_viii_start:N #3
  }
\cs_new:Npn \__str_decode_utf_viii_aux:wNnnwN
    #1 \s__str #2#3#4 #5 \__str_decode_utf_viii_start:N #6
  {
    \if_int_compare:w #1 < #4 \exp_stop_f:
      \s__str
      \if_int_compare:w #1 < #3 \exp_stop_f:
        \flag_raise:N \l__str_overlong_flag
        \flag_raise:N \l__str_error_flag
        \int_use:N \c__str_replacement_char_int
      \else:
        #1
      \fi:
    \else:
      \if_meaning:w \s__str_stop #5
        \__str_decode_utf_viii_overflow:w #1
      \fi:
      \exp_after:wN \__str_decode_utf_viii_continuation:wwN
      \int_value:w \int_eval:n { #1 - #4 } \exp_after:wN
    \fi:
    \s__str
    #2 {#4} #5
    \__str_decode_utf_viii_start:N
  }
\cs_new:Npn \__str_decode_utf_viii_overflow:w #1 \fi: #2 \fi:
  {
    \fi: \fi:
    \flag_raise:N \l__str_overflow_flag
    \flag_raise:N \l__str_error_flag
    \int_use:N \c__str_replacement_char_int
  }
\cs_new:Npn \__str_decode_utf_viii_end:
  {
    \s__str
    \flag_raise:N \l__str_missing_flag
    \flag_raise:N \l__str_error_flag
    \int_use:N \c__str_replacement_char_int \s__str
    \prg_break:
  }
\group_begin:
  \char_set_catcode_other:N \^^fe
  \char_set_catcode_other:N \^^ff
  \cs_new_protected:cpn { __str_convert_encode_utf16: }
    {
      \__str_encode_utf_xvi_aux:N \__str_output_byte_pair_be:n
      \tl_gput_left:Ne \g__str_result_tl { ^^fe ^^ff }
    }
  \cs_new_protected:cpn { __str_convert_encode_utf16be: }
    { \__str_encode_utf_xvi_aux:N \__str_output_byte_pair_be:n }
  \cs_new_protected:cpn { __str_convert_encode_utf16le: }
    { \__str_encode_utf_xvi_aux:N \__str_output_byte_pair_le:n }
  \cs_new_protected:Npn \__str_encode_utf_xvi_aux:N #1
    {
      \flag_clear:N \l__str_error_flag
      \cs_set_eq:NN \__str_tmp:w #1
      \__str_convert_gmap_internal:N \__str_encode_utf_xvi_char:n
      \__str_if_flag_error:Nne \l__str_error_flag { utf16-encode } { }
    }
  \cs_new:Npn \__str_encode_utf_xvi_char:n #1
    {
      \if_int_compare:w #1 < "D800 \exp_stop_f:
        \__str_tmp:w {#1}
      \else:
        \if_int_compare:w #1 < "10000 \exp_stop_f:
          \if_int_compare:w #1 < "E000 \exp_stop_f:
```

```
          \flag_raise:N \l__str_error_flag
          \__str_tmp:w { \c__str_replacement_char_int }
        \else:
          \__str_tmp:w {#1}
        \fi:
      \else:
        \exp_args:Nf \__str_tmp:w { \int_div_truncate:nn {#1} {"400} + "D7C0 }
        \exp_args:Nf \__str_tmp:w { \int_mod:nn {#1} {"400} + "DC00 }
      \fi:
    \fi:
  }
\flag_clear_new:N \l__str_missing_flag
\flag_clear_new:N \l__str_extra_flag
\flag_clear_new:N \l__str_end_flag
\msg_new:nnnn { str } { utf16-encode }
  { Unicode~string~cannot~be~expressed~in~UTF-16:~surrogate. }
  {
    Surrogate~code~points~(in~the~range~[U+D800,~U+DFFF])~
    can~be~expressed~in~the~UTF-8~and~UTF-32~encodings,~
    but~not~in~the~UTF-16~encoding.
  }
\msg_new:nnnn { str } { utf16-decode }
  {
    Invalid~UTF-16~string:
    \exp_last_unbraced:Nf \use_none:n
      {
        \__str_if_flag_times:NT \l__str_missing_flag  { ,~missing~trail~surrogate }
        \__str_if_flag_times:NT \l__str_extra_flag    { ,~extra~trail~surrogate }
        \__str_if_flag_times:NT \l__str_end_flag      { ,~odd~number~of~bytes }
      }
    .
  }
  {
    In~the~UTF-16~encoding,~each~Unicode~character~is~encoded~as~
    2~or~4~bytes: \\
    \iow_indent:n
      {
        Code~point~in~[U+0000,~U+D7FF]:~two~bytes \\
        Code~point~in~[U+D800,~U+DFFF]:~illegal \\
        Code~point~in~[U+E000,~U+FFFF]:~two~bytes \\
        Code~point~in~[U+10000,~U+10FFFF]:~
          a~lead~surrogate~and~a~trail~surrogate \\
      }
    Lead~surrogates~are~pairs~of~bytes~in~the~range~[0xD800,~0xDBFF],~
    and~trail~surrogates~are~in~the~range~[0xDC00,~0xDFFF].
    \flag_if_raised:NT \l__str_missing_flag
```

```
      {
        \\\\
        A~lead~surrogate~was~not~followed~by~a~trail~surrogate.
      }
    \flag_if_raised:NT \l__str_extra_flag
      {
        \\\\
        LaTeX~came~across~a~trail~surrogate~when~it~was~not~expected.
      }
    \flag_if_raised:NT \l__str_end_flag
      {
        \\\\
        The~string~contained~an~odd~number~of~bytes.~This~is~invalid:~
        the~basic~code~unit~for~UTF-16~is~16~bits~(2~bytes).
      }
  }
\cs_new_protected:cpn { __str_convert_decode_utf16be: }
  { \__str_decode_utf_xvi:Nw 1 \g__str_result_tl \s__str_stop }
\cs_new_protected:cpn { __str_convert_decode_utf16le: }
  { \__str_decode_utf_xvi:Nw 2 \g__str_result_tl \s__str_stop }
\cs_new_protected:cpn { __str_convert_decode_utf16: }
  {
    \exp_after:wN \__str_decode_utf_xvi_bom:NN
      \g__str_result_tl \s__str_stop \s__str_stop \s__str_stop
  }
\cs_new_protected:Npn \__str_decode_utf_xvi_bom:NN #1#2
  {
    \str_if_eq:nnTF { #1#2 } { ^^ff ^^fe }
      { \__str_decode_utf_xvi:Nw 2 }
      {
        \str_if_eq:nnTF { #1#2 } { ^^fe ^^ff }
          { \__str_decode_utf_xvi:Nw 1 }
          { \__str_decode_utf_xvi:Nw 1 #1#2 }
      }
  }
\cs_new_protected:Npn \__str_decode_utf_xvi:Nw #1#2 \s__str_stop
  {
    \flag_clear:N \l__str_error_flag
    \flag_clear:N \l__str_missing_flag
    \flag_clear:N \l__str_extra_flag
    \flag_clear:N \l__str_end_flag
    \cs_set:Npn \__str_tmp:w ##1 ##2 { ` ## #1 }
    \__kernel_tl_gset:Nx \g__str_result_tl
      {
        \exp_after:wN \__str_decode_utf_xvi_pair:NN
          #2 \q__str_nil \q__str_nil
        \prg_break_point:
      }
    \__str_if_flag_error:Nne \l__str_error_flag { utf16-decode } { }
  }
\cs_new:Npn \__str_decode_utf_xvi_pair:NN #1#2
  {
    \if_meaning:w \q__str_nil #2
      \__str_decode_utf_xvi_pair_end:Nw #1
    \fi:
    \if_case:w
      \int_eval:n { ( \__str_tmp:w #1#2 - "D6 ) / 4 } \scan_stop:
    \or: \exp_after:wN \__str_decode_utf_xvi_quad:NNwNN
    \or: \exp_after:wN \__str_decode_utf_xvi_extra:NNw
    \fi:
    #1#2 \s__str
    \int_eval:n { "100 * \__str_tmp:w #1#2 + \__str_tmp:w #2#1 } \s__str
    \__str_decode_utf_xvi_pair:NN
  }
\cs_new:Npn \__str_decode_utf_xvi_quad:NNwNN
    #1#2 #3 \__str_decode_utf_xvi_pair:NN #4#5
  {
    \if_meaning:w \q__str_nil #5
      \__str_decode_utf_xvi_error:nNN { missing } #1#2
      \__str_decode_utf_xvi_pair_end:Nw #4
    \fi:
    \if_int_compare:w
        \if_int_compare:w \__str_tmp:w #4#5 < "DC \exp_stop_f:
          0 = 1
        \else:
          \__str_tmp:w #4#5 < "E0
        \fi:
        \exp_stop_f:
      #1 #2 #4 #5 \s__str
      \int_eval:n
        {
          ( "100 * \__str_tmp:w #1#2 + \__str_tmp:w #2#1 - "D7F7 ) * "400
          + "100 * \__str_tmp:w #4#5 + \__str_tmp:w #5#4
        }
      \s__str
      \exp_after:wN \use_i:nnn
    \else:
      \__str_decode_utf_xvi_error:nNN { missing } #1#2
    \fi:
    \__str_decode_utf_xvi_pair:NN #4#5
  }
\cs_new:Npn \__str_decode_utf_xvi_pair_end:Nw #1 \fi:
    {
      \fi:
      \if_meaning:w \q__str_nil #1
      \else:
        \__str_decode_utf_xvi_error:nNN { end } #1 \prg_do_nothing:
      \fi:
      \prg_break:
    }
  \cs_new:Npn \__str_decode_utf_xvi_extra:NNw #1#2 \s__str #3 \s__str
    { \__str_decode_utf_xvi_error:nNN { extra } #1#2 }
  \cs_new:Npn \__str_decode_utf_xvi_error:nNN #1#2#3
    {
      \flag_raise:N \l__str_error_flag
      \flag_raise:c { l__str_#1_flag }
      #2 #3 \s__str
      \int_use:N \c__str_replacement_char_int \s__str
    }
\group_end:
\group_begin:
  \char_set_catcode_other:N \^^00
  \char_set_catcode_other:N \^^fe
  \char_set_catcode_other:N \^^ff
  \cs_new_protected:cpn { __str_convert_encode_utf32: }
    {
      \__str_convert_gmap_internal:N \__str_encode_utf_xxxii_be:n
      \tl_gput_left:Ne \g__str_result_tl { ^^00 ^^00 ^^fe ^^ff }
    }
  \cs_new_protected:cpn { __str_convert_encode_utf32be: }
    { \__str_convert_gmap_internal:N \__str_encode_utf_xxxii_be:n }
  \cs_new_protected:cpn { __str_convert_encode_utf32le: }
    { \__str_convert_gmap_internal:N \__str_encode_utf_xxxii_le:n }
  \cs_new:Npn \__str_encode_utf_xxxii_be:n #1
    {
      \exp_args:Nf \__str_encode_utf_xxxii_be_aux:nn
        { \int_div_truncate:nn {#1} { "100 } } {#1}
    }
  \cs_new:Npn \__str_encode_utf_xxxii_be_aux:nn #1#2
    {
      ^^00
      \__str_output_byte_pair_be:n {#1}
      \__str_output_byte:n { #2 - #1 * "100 }
    }
  \cs_new:Npn \__str_encode_utf_xxxii_le:n #1
    {
      \exp_args:Nf \__str_encode_utf_xxxii_le_aux:nn
        { \int_div_truncate:nn {#1} { "100 } } {#1}
    }
  \cs_new:Npn \__str_encode_utf_xxxii_le_aux:nn #1#2
    {
      \__str_output_byte:n { #2 - #1 * "100 }
      \__str_output_byte_pair_le:n {#1}
      ^^00
    }
  \flag_clear_new:N \l__str_overflow_flag
  \flag_clear_new:N \l__str_end_flag
  \msg_new:nnnn { str } { utf32-decode }
    {
      Invalid~UTF-32~string:
      \exp_last_unbraced:Nf \use_none:n
        {
          \__str_if_flag_times:NT \l__str_overflow_flag { ,~code~point~too~large }
          \__str_if_flag_times:NT \l__str_end_flag      { ,~truncated~string }
        }
      .
    }
    {
      In~the~UTF-32~encoding,~every~Unicode~character~
      (in~the~range~[U+0000,~U+10FFFF])~is~encoded~as~4~bytes.
      \flag_if_raised:NT \l__str_overflow_flag
        {
          \\\\
          LaTeX~came~across~a~code~point~larger~than~1114111,~
          the~maximum~code~point~defined~by~Unicode.~
          Perhaps~the~string~was~not~encoded~in~the~UTF-32~encoding?
        }
      \flag_if_raised:NT \l__str_end_flag
        {
          \\\\
          The~length~of~the~string~is~not~a~multiple~of~4.~
          Perhaps~the~string~was~truncated?
        }
    }
  \cs_new_protected:cpn { __str_convert_decode_utf32be: }
    { \__str_decode_utf_xxxii:Nw 1 \g__str_result_tl \s__str_stop }
  \cs_new_protected:cpn { __str_convert_decode_utf32le: }
    { \__str_decode_utf_xxxii:Nw 2 \g__str_result_tl \s__str_stop }
  \cs_new_protected:cpn { __str_convert_decode_utf32: }
    {
      \exp_after:wN \__str_decode_utf_xxxii_bom:NNNN \g__str_result_tl
        \s__str_stop \s__str_stop \s__str_stop \s__str_stop \s__str_stop
    }
  \cs_new_protected:Npn \__str_decode_utf_xxxii_bom:NNNN #1#2#3#4
```

```
  {
    \str_if_eq:nnTF { #1#2#3#4 } { ^^ff ^^fe ^^00 ^^00 }
      { \__str_decode_utf_xxxii:Nw 2 }
      {
        \str_if_eq:nnTF { #1#2#3#4 } { ^^00 ^^00 ^^fe ^^ff }
          { \__str_decode_utf_xxxii:Nw 1 }
          { \__str_decode_utf_xxxii:Nw 1 #1#2#3#4 }
      }
  }
\cs_new_protected:Npn \__str_decode_utf_xxxii:Nw #1#2 \s__str_stop
  {
    \flag_clear:N \l__str_overflow_flag
    \flag_clear:N \l__str_end_flag
    \flag_clear:N \l__str_error_flag
    \cs_set:Npn \__str_tmp:w ##1 ##2 { ` ## #1 }
    \__kernel_tl_gset:Nx \g__str_result_tl
      {
        \exp_after:wN \__str_decode_utf_xxxii_loop:NNNN
          #2 \s__str_stop \s__str_stop \s__str_stop \s__str_stop
        \prg_break_point:
      }
    \__str_if_flag_error:Nne \l__str_error_flag { utf32-decode } { }
  }
\cs_new:Npn \__str_decode_utf_xxxii_loop:NNNN #1#2#3#4
  {
    \if_meaning:w \s__str_stop #4
      \exp_after:wN \__str_decode_utf_xxxii_end:w
    \fi:
    #1#2#3#4 \s__str
    \if_int_compare:w \__str_tmp:w #1#4 > \c_zero_int
      \flag_raise:N \l__str_overflow_flag
      \flag_raise:N \l__str_error_flag
      \int_use:N \c__str_replacement_char_int
    \else:
      \if_int_compare:w \__str_tmp:w #2#3 > 16 \exp_stop_f:
        \flag_raise:N \l__str_overflow_flag
        \flag_raise:N \l__str_error_flag
        \int_use:N \c__str_replacement_char_int
      \else:
        \int_eval:n
          { \__str_tmp:w #2#3*"10000 + \__str_tmp:w #3#2*"100 + \__str_tmp:w #4#1 }
      \fi:
    \fi:
    \s__str
    \__str_decode_utf_xxxii_loop:NNNN
  }
  \cs_new:Npn \__str_decode_utf_xxxii_end:w #1 \s__str_stop
    {
      \tl_if_empty:nF {#1}
        {
          \flag_raise:N \l__str_end_flag
          \flag_raise:N \l__str_error_flag
          #1 \s__str
          \int_use:N \c__str_replacement_char_int \s__str
        }
      \prg_break:
    }
\group_end:
\cs_new:Npn \str_convert_pdfname:n #1
  {
    \exp_args:Ne \tl_to_str:n
      { \str_map_function:nN {#1} \__str_convert_pdfname:n }
  }
\sys_if_engine_opentype:TF
  {
    \cs_new:Npn \__str_convert_pdfname:n #1
      {
        \int_compare:nNnTF { `#1 } > { "7F }
          { \__str_convert_pdfname_bytes:n {#1} }
          { \__str_escape_name_char:n {#1} }
      }
    \cs_new:Npn \__str_convert_pdfname_bytes:n #1
      {
        \exp_args:Ne \__str_convert_pdfname_bytes_aux:n
          { \__kernel_codepoint_to_bytes:n {`#1} }
      }
    \cs_new:Npn \__str_convert_pdfname_bytes_aux:n #1
      { \__str_convert_pdfname_bytes_aux:nnnn #1 }
    \cs_new:Npe \__str_convert_pdfname_bytes_aux:nnnn #1#2#3#4
      {
        \c_hash_str \exp_not:N \__str_output_hexadecimal:n {#1}
        \c_hash_str \exp_not:N \__str_output_hexadecimal:n {#2}
        \exp_not:N \tl_if_blank:nF {#3}
          {
            \c_hash_str \exp_not:N \__str_output_hexadecimal:n {#3}
            \exp_not:N \tl_if_blank:nF {#4}
              {
                \c_hash_str \exp_not:N \__str_output_hexadecimal:n {#4}
              }
          }
      }
  }
  { \cs_new_eq:NN \__str_convert_pdfname:n \__str_escape_name_char:n }
%% File: l3tl-analysis.dtx
\scan_new:N \s__tl
\cs_new_eq:NN \l__tl_analysis_token ?
\cs_new_eq:NN \l__tl_analysis_char_token ?
\tl_new:N \l__tl_peek_code_tl
\group_begin:
\char_set_active_eq:NN \_ \scan_stop:
\tl_const:Ne \c__tl_peek_catcodes_tl
  {
    \char_generate:nn { 32 } { 3 }   3
    \char_generate:nn { 32 } { 4 }   4
    \char_generate:nn { 32 } { 7 }   7
    \char_generate:nn { 32 } { 8 }   8
    \c_space_tl                     \token_to_str:N A
    \char_generate:nn { 32 } { 11 } \token_to_str:N B
    \char_generate:nn { 32 } { 12 } \token_to_str:N C
    \char_generate:nn { 32 } { 13 } \token_to_str:N D
  }
\group_end:
\int_new:N \l__tl_analysis_normal_int
\int_new:N \l__tl_analysis_index_int
\int_new:N \l__tl_analysis_nesting_int
\int_new:N \l__tl_analysis_type_int
\tl_new:N \g__tl_analysis_result_tl
\cs_new:Npn \__tl_analysis_extract_charcode:
  {
    \exp_after:wN \__tl_analysis_extract_charcode_aux:w
      \token_to_meaning:N \l__tl_analysis_token
  }
\cs_new:Npn \__tl_analysis_extract_charcode_aux:w #1 ~ #2 ~ { ` }
\bool_lazy_and:nnT
  { \cs_if_exist_p:N \tex_luatexversion:D }
  { \int_compare_p:nNn { \int_div_truncate:nn { \tex_luatexversion:D } { 100 } } > 1 }
  {
    \cs_gset:Npn \__tl_analysis_extract_charcode_aux:w #1 + #2 ~ ' #3 ' {"#2}
  }
\cs_new:Npe \__tl_analysis_cs_space_count:NN #1 #2
  {
    \exp_not:N \exp_after:wN #1
    \exp_not:N \int_value:w \exp_not:N \int_eval:w 0
      \exp_not:N \exp_after:wN \exp_not:N \__tl_analysis_cs_space_count:w
        \exp_not:N \token_to_str:N #2
        \exp_not:N \fi: \exp_not:N \__tl_analysis_cs_space_count_end:w
          \exp_not:N \__tl_sep: \c_space_tl !
  }
```

```
\cs_new:Npn \__tl_analysis_cs_space_count:w #1 ~
  {
    \if_false: #1 #1 \fi:
    + 1
    \__tl_analysis_cs_space_count:w
  }
\cs_new:Npn \__tl_analysis_cs_space_count_end:w \__tl_sep: #1 \fi: #2 !
  {
    \exp_after:wN \__tl_sep: \int_value:w
      \str_count_ignore_spaces:n {#1} \__tl_sep:
  }
\cs_new_protected:Npn \__tl_analysis:n #1
  {
    \group_begin:
      \group_align_safe_begin:
        \__tl_analysis_a:n {#1}
        \__tl_analysis_b:n {#1}
      \group_align_safe_end:
    \group_end:
  }
\group_begin:
  \char_set_catcode_active:N \^^@
  \cs_new_protected:Npn \__tl_analysis_disable:n #1
    {
      \tex_lccode:D 0 = #1 \exp_stop_f:
      \tex_lowercase:D { \tex_let:D ^^@ } \tex_undefined:D
    }
  \bool_lazy_or:nnT
    { \sys_if_engine_ptex_p: }
    { \sys_if_engine_uptex_p: }
    {
      \cs_gset_protected:Npn \__tl_analysis_disable:n #1
        {
          \if_int_compare:w 256 > #1 \exp_stop_f:
            \tex_lccode:D 0 = #1 \exp_stop_f:
            \tex_lowercase:D { \tex_let:D ^^@ } \tex_undefined:D
          \fi:
        }
    }
\group_end:
\group_begin:
  \char_set_catcode_active:N \^^@
  \cs_new_protected:Npn \__tl_analysis_disable_char:N #1
    {
      \tex_lccode:D `#1 = 32 \exp_stop_f:
      \tex_lowercase:D { \if_meaning:w #1 } \tex_undefined:D
```

```
        \tex_let:D #1 \tex_undefined:D
      \fi:
    }
  \bool_lazy_or:nnT
    { \sys_if_engine_ptex_p: }
    { \sys_if_engine_uptex_p: }
    {
      \cs_gset_protected:Npn \__tl_analysis_disable_char:N #1
        {
          \if_int_compare:w 256 > `#1 \exp_stop_f:
            \tex_lccode:D `#1 = 32 \exp_stop_f:
            \tex_lowercase:D { \if_meaning:w #1 } \tex_undefined:D
              \tex_let:D #1 \tex_undefined:D
            \fi:
          \fi:
        }
    }
\group_end:
\cs_new_protected:Npn \__tl_analysis_a:n #1
  {
    \__tl_analysis_disable:n { 32 }
    \int_set:Nn \tex_escapechar:D { 92 }
    \int_zero:N \l__tl_analysis_normal_int
    \int_zero:N \l__tl_analysis_index_int
    \int_zero:N \l__tl_analysis_nesting_int
    \if_false: { \fi: \__tl_analysis_a_loop:w #1 }
    \int_decr:N \l__tl_analysis_index_int
  }
\cs_new_protected:Npn \__tl_analysis_a_loop:w
  { \tex_futurelet:D \l__tl_analysis_token \__tl_analysis_a_type:w }
\cs_new_protected:Npn \__tl_analysis_a_type:w
  {
    \l__tl_analysis_type_int =
      \if_meaning:w \l__tl_analysis_token \c_space_token
        0
      \else:
        \if_catcode:w \exp_not:N \l__tl_analysis_token \c_group_begin_token
          1
        \else:
          \if_catcode:w \exp_not:N \l__tl_analysis_token \c_group_end_token
            - 1
          \else:
            2
          \fi:
        \fi:
      \fi:
```

```
      \exp_stop_f:
    \if_case:w \l__tl_analysis_type_int
         \exp_after:wN \__tl_analysis_a_space:w
    \or: \exp_after:wN \__tl_analysis_a_bgroup:w
    \or: \exp_after:wN \__tl_analysis_a_safe:N
    \else: \exp_after:wN \__tl_analysis_a_egroup:w
    \fi:
  }
\cs_new_protected:Npn \__tl_analysis_a_space:w
  {
    \tex_afterassignment:D \__tl_analysis_a_space_test:w
    \exp_after:wN \cs_set_eq:NN
    \exp_after:wN \l__tl_analysis_char_token
    \token_to_str:N
  }
\cs_new_protected:Npn \__tl_analysis_a_space_test:w
  {
    \if_meaning:w \l__tl_analysis_char_token \c_space_token
      \tex_toks:D \l__tl_analysis_index_int { \exp_not:n { ~ } }
      \__tl_analysis_a_store:
    \else:
      \int_incr:N \l__tl_analysis_normal_int
    \fi:
    \__tl_analysis_a_loop:w
  }
\group_begin:
  \char_set_catcode_group_begin:N \^^@ % {
  \cs_new_protected:Npn \__tl_analysis_a_bgroup:w
    { \__tl_analysis_a_group:nw { \exp_after:wN ^^@ \if_false: } \fi: } }
  \char_set_catcode_group_end:N \^^@
  \cs_new_protected:Npn \__tl_analysis_a_egroup:w
    { \__tl_analysis_a_group:nw { \if_false: { \fi: ^^@ } } % }
\group_end:
\cs_new_protected:Npn \__tl_analysis_a_group:nw #1
  {
    \tex_lccode:D 0 = \__tl_analysis_extract_charcode: \scan_stop:
    \tex_lowercase:D { \tex_toks:D \l__tl_analysis_index_int {#1} }
    \if_int_compare:w \tex_lccode:D 0 = \tex_escapechar:D
      \int_set:Nn \tex_escapechar:D { 139 - \tex_escapechar:D }
    \fi:
    \__tl_analysis_disable:n { \tex_lccode:D 0 }
    \tex_futurelet:D \l__tl_analysis_token \__tl_analysis_a_group_aux:w
  }
\cs_new_protected:Npn \__tl_analysis_a_group_aux:w
  {
    \if_meaning:w \l__tl_analysis_token \tex_undefined:D
```

```
      \exp_after:wN \__tl_analysis_a_safe:N
    \else:
      \exp_after:wN \__tl_analysis_a_group_auxii:w
    \fi:
  }
\cs_new_protected:Npn \__tl_analysis_a_group_auxii:w
  {
    \tex_afterassignment:D \__tl_analysis_a_group_test:w
    \exp_after:wN \cs_set_eq:NN
    \exp_after:wN \l__tl_analysis_char_token
    \token_to_str:N
  }
\cs_new_protected:Npn \__tl_analysis_a_group_test:w
  {
    \if_charcode:w \l__tl_analysis_token \l__tl_analysis_char_token
      \__tl_analysis_a_store:
    \else:
      \int_incr:N \l__tl_analysis_normal_int
    \fi:
    \__tl_analysis_a_loop:w
  }
\cs_new_protected:Npn \__tl_analysis_a_store:
  {
    \tex_advance:D \l__tl_analysis_nesting_int \l__tl_analysis_type_int
    \if_int_compare:w \tex_lccode:D 0 = `\␣\exp_stop_f:
      \tex_advance:D \l__tl_analysis_type_int \l__tl_analysis_type_int
    \fi:
    \tex_skip:D \l__tl_analysis_index_int
      = \l__tl_analysis_normal_int sp
         plus \l__tl_analysis_type_int sp \scan_stop:
    \int_incr:N \l__tl_analysis_index_int
    \int_zero:N \l__tl_analysis_normal_int
    \if_int_compare:w \l__tl_analysis_nesting_int = - \c_one_int
      \cs_set_eq:NN \__tl_analysis_a_loop:w \scan_stop:
    \fi:
  }
\cs_new_protected:Npn \__tl_analysis_a_safe:N #1
  {
    \if_charcode:w
        \scan_stop:
        \exp_after:wN \use_none:n \token_to_str:N #1 \prg_do_nothing:
        \scan_stop:
      \exp_after:wN \use_i:nn
    \else:
      \exp_after:wN \use_ii:nn
    \fi:
      {
        \__tl_analysis_disable_char:N #1
        \int_incr:N \l__tl_analysis_normal_int
      }
      { \__tl_analysis_cs_space_count:NN \__tl_analysis_a_cs:ww #1 }
    \__tl_analysis_a_loop:w
  }
\cs_new_protected:Npn \__tl_analysis_a_cs:ww #1 \__tl_sep: #2 \__tl_sep:
  {
    \if_int_compare:w #1 > \c_zero_int
      \tex_skip:D \l__tl_analysis_index_int
        = \int_eval:n { \l__tl_analysis_normal_int + 1 } sp \exp_stop_f:
      \tex_advance:D \l__tl_analysis_index_int #1 \exp_stop_f:
    \else:
      \tex_advance:D
    \fi:
    \l__tl_analysis_normal_int #2 \exp_stop_f:
  }
\cs_new_protected:Npn \__tl_analysis_b:n #1
  {
    \__kernel_tl_gset:Nx \g__tl_analysis_result_tl
      {
        \__tl_analysis_b_loop:w 0 \__tl_sep: #1
        \prg_break_point:
      }
  }
\cs_new:Npn \__tl_analysis_b_loop:w #1 \__tl_sep:
  {
    \exp_after:wN \__tl_analysis_b_normals:ww
      \int_value:w \tex_skip:D #1 \__tl_sep: #1 \__tl_sep:
  }
\cs_new:Npn \__tl_analysis_b_normals:ww #1 \__tl_sep:
  {
    \if_int_compare:w #1 = \c_zero_int
      \__tl_analysis_b_special:w
    \fi:
    \__tl_analysis_b_normal:wwN #1 \__tl_sep:
  }
\cs_new:Npn \__tl_analysis_b_normal:wwN #1 \__tl_sep: #2 \__tl_sep: #3
  {
    \exp_not:n { \exp_not:n { #3 } } \s__tl
    \if_charcode:w
        \scan_stop:
        \exp_after:wN \use_none:n \token_to_str:N #3 \prg_do_nothing:
        \scan_stop:
      \exp_after:wN \__tl_analysis_b_char:Nn
      \exp_after:wN \__tl_analysis_b_char_aux:nww
    \else:
      \exp_after:wN \__tl_analysis_b_cs:Nww
    \fi:
    #3 #1 \__tl_sep: #2 \__tl_sep:
  }
\cs_new:Npe \__tl_analysis_b_char:Nn #1#2
  {
    \exp_not:N \if_meaning:w #2 \exp_not:N \tex_undefined:D
      \token_to_str:N D \exp_not:N \else:
    \exp_not:N \if_catcode:w #2 \c_catcode_other_token
      \token_to_str:N C \exp_not:N \else:
    \exp_not:N \if_catcode:w #2 \c_catcode_letter_token
      \token_to_str:N B \exp_not:N \else:
    \exp_not:N \if_catcode:w #2 \c_math_toggle_token      3
      \exp_not:N \else:
    \exp_not:N \if_catcode:w #2 \c_alignment_token        4
      \exp_not:N \else:
    \exp_not:N \if_catcode:w #2 \c_math_superscript_token 7
      \exp_not:N \else:
    \exp_not:N \if_catcode:w #2 \c_math_subscript_token   8
      \exp_not:N \else:
    \exp_not:N \if_catcode:w #2 \c_space_token
      \token_to_str:N A \exp_not:N \else:
      6
    \exp_not:n { \fi: \fi: \fi: \fi: \fi: \fi: \fi: \fi: }
    #1 {#2}
  }
\cs_new:Npn \__tl_analysis_b_char_aux:nww #1
  {
    \int_value:w `#1 \s__tl
    \exp_after:wN \__tl_analysis_b_normals:ww
      \int_value:w \int_eval:w - 1 +
  }
\cs_new:Npn \__tl_analysis_b_cs:Nww #1
  {
    0 -1 \s__tl
    \__tl_analysis_cs_space_count:NN \__tl_analysis_b_cs_test:ww #1
  }
\cs_new:Npn \__tl_analysis_b_cs_test:ww
  #1 \__tl_sep: #2 \__tl_sep: #3 \__tl_sep: #4 \__tl_sep:
  {
    \exp_after:wN \__tl_analysis_b_normals:ww
    \int_value:w \int_eval:w
    \if_int_compare:w #1 = \c_zero_int
      #3
    \else:
      \tex_skip:D \int_eval:n { #4 + #1 } \exp_stop_f:
    \fi:
    - #2
    \exp_after:wN \__tl_sep:
    \int_value:w \int_eval:n { #4 + #1 } \__tl_sep:
  }
\group_begin:
  \char_set_catcode_other:N A
  \cs_new:Npn \__tl_analysis_b_special:w
      \fi: \__tl_analysis_b_normal:wwN 0 \__tl_sep: #1 \__tl_sep:
    {
      \fi:
      \if_int_compare:w #1 = \l__tl_analysis_index_int
        \exp_after:wN \prg_break:
      \fi:
      \tex_the:D \tex_toks:D #1 \s__tl
      \if_case:w \tex_gluestretch:D \tex_skip:D #1 \exp_stop_f:
             \token_to_str:N A
      \or:   1
      \or:   1
      \else: 2
      \fi:
      \if_int_odd:w \tex_gluestretch:D \tex_skip:D #1 \exp_stop_f:
        \exp_after:wN \__tl_analysis_b_special_char:wN \int_value:w
      \else:
        \exp_after:wN \__tl_analysis_b_special_space:w \int_value:w
      \fi:
      \int_eval:n { 1 + #1 } \exp_after:wN \__tl_sep:
      \token_to_str:N
    }
\group_end:
\cs_new:Npn \__tl_analysis_b_special_char:wN #1 \__tl_sep: #2
  {
    \int_value:w `#2 \s__tl
    \__tl_analysis_b_loop:w #1 \__tl_sep:
  }
\use:e
  {
    \cs_new:Npn \exp_not:N \__tl_analysis_b_special_space:w
      #1 \exp_not:N \__tl_sep: \c_space_tl
  }
  {
    32 \s__tl
    \__tl_analysis_b_loop:w #1 \__tl_sep:
  }
```

```
\cs_new_protected:Npn \tl_analysis_map_inline:Nn #1
  { \exp_args:No \tl_analysis_map_inline:nn #1 }
\cs_new_protected:Npn \tl_analysis_map_inline:nn #1
  {
    \__tl_analysis:n {#1}
    \int_gincr:N \g__kernel_prg_map_int
    \exp_args:Nc \__tl_analysis_map:Nn
      { __tl_analysis_map_inline_ \int_use:N \g__kernel_prg_map_int :wNw }
  }
\cs_new_protected:Npn \__tl_analysis_map:Nn #1#2
  {
    \cs_gset_protected:Npn #1 ##1##2##3 {#2}
    \exp_after:wN \__tl_analysis_map:NwNw \exp_after:wN #1
      \g__tl_analysis_result_tl
      \s__tl { ? \tl_map_break: } \s__tl
    \prg_break_point:Nn \tl_map_break:
      { \int_gdecr:N \g__kernel_prg_map_int }
  }
\cs_new_protected:Npn \__tl_analysis_map:NwNw #1 #2 \s__tl #3 #4 \s__tl
  {
    \use_none:n #3
    #1 {#2} {#4} {#3}
    \__tl_analysis_map:NwNw #1
  }
\cs_new_protected:Npn \tl_analysis_show:N
  { \__tl_analysis_show:NNnnN \msg_show:nneeee \tl_show:N {} {} }
\cs_new_protected:Npn \tl_analysis_log:N
  {
    \__tl_analysis_show:NNnnN \msg_log:nneeee \tl_log:N
      { \iow_newline: >~ . } { . }
  }
\cs_new_protected:Npn \__tl_analysis_show:NNnnN #1#2#3#4#5
  {
    \tl_if_exist:NTF #5
      {
        \exp_args:No \__tl_analysis:n {#5}
        #1 { tl } { show-analysis }
          { \token_to_str:N #5 } { \__tl_analysis_show: } {#3} {#4}
      }
      { #2 #3 }
  }
\cs_new_protected:Npn \tl_analysis_show:n
  { \__tl_analysis_show:Nnnn \msg_show:nneeee {} {} }
\cs_new_protected:Npn \tl_analysis_log:n
  { \__tl_analysis_show:Nnnn \msg_log:nneeee { \iow_newline: >~ . } { . } }
\cs_new_protected:Npn \__tl_analysis_show:Nnnn #1#2#3#4
```

```
  {
    \__tl_analysis:n {#4}
    #1 { tl } { show-analysis } { } { \__tl_analysis_show: } {#2} {#3}
  }
\cs_new:Npn \__tl_analysis_show:
  {
    \exp_after:wN \__tl_analysis_show_loop:wNw \g__tl_analysis_result_tl
    \s__tl { ? \prg_break: } \s__tl
    \prg_break_point:
  }
\cs_new:Npn \__tl_analysis_show_loop:wNw #1 \s__tl #2 #3 \s__tl
  {
    \use_none:n #2
    \iow_newline: > \use:nn { ~ } { ~ }
    \if_int_compare:w "#2 = \c_zero_int
      \exp_after:wN \__tl_analysis_show_cs:n
    \else:
      \if_int_compare:w "#2 = 13 \exp_stop_f:
        \exp_after:wN \exp_after:wN
        \exp_after:wN \__tl_analysis_show_active:n
      \else:
        \exp_after:wN \exp_after:wN
        \exp_after:wN \__tl_analysis_show_normal:n
      \fi:
    \fi:
    {#1}
    \__tl_analysis_show_loop:wNw
  }
\cs_new:Npn \__tl_analysis_show_normal:n #1
  {
    \exp_after:wN \token_to_str:N #1 ~
    ( \exp_after:wN \token_to_meaning:N #1 )
  }
\cs_new:Npn \__tl_analysis_show_value:N #1
  {
    \token_if_expandable:NF #1
      {
        \token_if_chardef:NTF       #1 \prg_break: { }
        \token_if_mathchardef:NTF   #1 \prg_break: { }
        \token_if_dim_register:NTF  #1 \prg_break: { }
        \token_if_int_register:NTF  #1 \prg_break: { }
        \token_if_skip_register:NTF #1 \prg_break: { }
        \token_if_toks_register:NTF #1 \prg_break: { }
        \use_none:nnn
        \prg_break_point:
        \use:n { \exp_after:wN = \tex_the:D #1 }
      }
  }
\cs_new:Npn \__tl_analysis_show_cs:n #1
  { \exp_args:No \__tl_analysis_show_long:nn {#1} { control~sequence= } }
\cs_new:Npn \__tl_analysis_show_active:n #1
  { \exp_args:No \__tl_analysis_show_long:nn {#1} { active~character= } }
\cs_new:Npn \__tl_analysis_show_long:nn #1
  {
    \__tl_analysis_show_long_aux:oofn
      { \token_to_str:N #1 }
      { \token_to_meaning:N #1 }
      { \__tl_analysis_show_value:N #1 }
  }
\cs_new:Npn \__tl_analysis_show_long_aux:nnnn #1#2#3#4
  {
    \int_compare:nNnTF
      { \str_count:n { #1 ~ ( #4 #2 #3 ) } }
      > { \l_iow_line_count_int - 3 }
      {
        \str_range:nnn { #1 ~ ( #4 #2 #3 ) } { 1 }
          {
            \l_iow_line_count_int - 3
            - \str_count:N \c__tl_analysis_show_etc_str
          }
        \c__tl_analysis_show_etc_str
      }
      { #1 ~ ( #4 #2 #3 ) }
  }
\cs_generate_variant:Nn \__tl_analysis_show_long_aux:nnnn { oof }
\cs_new:Npn \peek_analysis_map_break:
  { \prg_map_break:Nn \peek_analysis_map_break: { } }
\cs_new:Npn \peek_analysis_map_break:n
  { \prg_map_break:Nn \peek_analysis_map_break: }
\int_new:N \l__tl_peek_charcode_int
\cs_new:Npn \__tl_analysis_char_arg:Nw
  {
    \if_meaning:w \l__tl_analysis_token \c_space_token
      \exp_after:wN \__tl_analysis_char_arg_aux:Nw
    \fi:
  }
\cs_new:Npn \__tl_analysis_char_arg_aux:Nw #1 ~ { #1 { ~ } }
\cs_new_protected:Npn \peek_analysis_map_inline:n #1
  {
    \group_align_safe_begin:
    \int_gincr:N \g__kernel_prg_map_int
    \cs_set_protected:cpn
```

```
      { __tl_analysis_map_ \int_use:N \g__kernel_prg_map_int :nnN }
      ##1##2##3
      {
        \group_end:
        #1
        \__tl_peek_analysis_loop:NNn
          \prg_break_point:Nn \peek_analysis_map_break:
            {
              \int_gdecr:N \g__kernel_prg_map_int
              \group_align_safe_end:
            }
      }
    \__tl_peek_analysis_loop:NNn ? ? ?
  }
\cs_new_protected:Npn \__tl_peek_analysis_loop:NNn #1#2#3
  {
    \group_begin:
    \tl_set:Ne \l__tl_peek_code_tl
      {
        \exp_not:c
          { __tl_analysis_map_ \int_use:N \g__kernel_prg_map_int :nnN }
      }
    \int_set:Nn \tex_escapechar:D { `\\ }
    \peek_after:Nw \__tl_peek_analysis_test:
  }
\cs_new_protected:Npn \__tl_peek_analysis_test:
  {
    \if_case:w
      \if_catcode:w \exp_not:N \l_peek_token {   \c_max_int \fi:
      \if_catcode:w \exp_not:N \l_peek_token }   \c_max_int \fi:
      \if_meaning:w \l_peek_token \c_space_token \c_max_int \fi:
      \exp_after:wN \if_meaning:w \exp_not:N \l_peek_token \l_peek_token
        \c_one_int
      \fi:
      \c_zero_int
      \exp_after:wN \exp_after:wN
      \exp_after:wN \__tl_peek_analysis_exp:N
      \exp_after:wN \exp_not:N
    \or:
      \exp_after:wN \__tl_peek_analysis_nonexp:N
    \else:
      \exp_after:wN \__tl_peek_analysis_special:
    \fi:
  }
\cs_new_protected:Npn \__tl_peek_analysis_exp:N #1
  {
    \cs_set_nopar:Npe \l__tl_peek_code_tl
      {
        \tex_let:D \exp_not:N #1 \l__tl_peek_code_tl
        \l__tl_peek_code_tl
        {
          \exp_not:n { \__kernel_exp_not:w \exp_after:wN }
            { \exp_not:N \exp_not:N \exp_not:N #1 }
        }
        \exp_after:wN \__tl_peek_analysis_exp_aux:Nw
        \token_to_str:N #1 { } \s__tl
      }
    \l__tl_peek_code_tl
  }
\cs_new:Npe \__tl_peek_analysis_exp_aux:Nw #1#2 \s__tl
  {
    \exp_not:N \if_meaning:w \scan_stop: #2 \scan_stop:
      { \exp_not:N \int_value:w `#1 ~ } \token_to_str:N D
    \exp_not:N \else:
      { -1 } 0
    \exp_not:N \fi:
  }
\cs_new_protected:Npn \__tl_peek_analysis_nonexp:N #1
  {
    \if_charcode:w
        \scan_stop:
        \exp_after:wN \use_none:n \token_to_str:N #1 \prg_do_nothing:
        \scan_stop:
      \exp_after:wN \__tl_peek_analysis_char:N
    \else:
      \exp_after:wN \__tl_peek_analysis_cs:N
    \fi:
    #1
  }
\cs_new_protected:Npn \__tl_peek_analysis_cs:N #1
  { \l__tl_peek_code_tl { \exp_not:n {#1} } { -1 } 0 }
\group_begin:
\char_set_active_eq:NN \_\scan_stop:
\cs_new_protected:Npe \__tl_peek_analysis_char:N #1
  {
    \cs_set_eq:NN
      \char_generate:nn { 32 } { 13 }
      \exp_not:N \tex_undefined:D
    \tex_lccode:D `#1 = 32 \exp_stop_f:
    \tex_lowercase:D
      {
        \tl_put_right:Ne \exp_not:N \l__tl_peek_code_tl
```

```
          { \exp_not:n { \__tl_analysis_b_char:Nn \use_none:n } {#1} }
      }
    \exp_not:n
      {
        \exp_after:wN \__tl_peek_analysis_char:w
        \int_value:w
      }
      `#1
    \exp_not:n { \exp_after:wN \s__tl \l__tl_peek_code_tl }
    #1
  }
\group_end:
\cs_new_protected:Npn \__tl_peek_analysis_char:w #1 \s__tl #2#3#4
  {
    \if_charcode:w 6 #3
    \else:
      \if_charcode:w D #3
      \else:
        \exp_args:NNNo
      \fi:
    \fi:
    #2 { \exp_not:n {#4} } {#1} #3
  }
\cs_new_protected:Npn \__tl_peek_analysis_special:
  {
    \tex_let:D \l__tl_analysis_token = ~ \l_peek_token
    \int_set:Nn \l__tl_peek_charcode_int
      { \__tl_analysis_extract_charcode: }
    \if_int_compare:w \l__tl_peek_charcode_int = \tex_escapechar:D
      \int_set:Nn \tex_escapechar:D { `\/ }
    \fi:
    \char_set_active_eq:nN { \l__tl_peek_charcode_int } \scan_stop:
    \char_set_active_eq:nN { \tex_escapechar:D } \scan_stop:
    \cs_set_eq:cN { } \scan_stop:
    \tex_futurelet:D \l__tl_analysis_token
    \__tl_peek_analysis_retest:
  }
\cs_new_protected:Npn \__tl_peek_analysis_retest:
  {
    \if_meaning:w \l__tl_analysis_token \scan_stop:
      \exp_after:wN \__tl_peek_analysis_nonexp:N
    \else:
      \exp_after:wN \__tl_peek_analysis_str:
    \fi:
  }
\cs_new_protected:Npn \__tl_peek_analysis_str:
  {
    \exp_after:wN \tex_futurelet:D
    \exp_after:wN \l__tl_analysis_token
    \exp_after:wN \__tl_peek_analysis_str:w
    \token_to_str:N
  }
\cs_new_protected:Npn \__tl_peek_analysis_str:w
  { \__tl_analysis_char_arg:Nw \__tl_peek_analysis_str:n }
\cs_new_protected:Npn \__tl_peek_analysis_str:n #1
  {
    \int_case:nnF { `#1 }
      {
        { \l__tl_peek_charcode_int }
          { \__tl_peek_analysis_explicit:n {#1} }
        { \tex_escapechar:D } { \__tl_peek_analysis_escape: }
      }
      { \__tl_peek_analysis_active_str:n {#1} }
  }
\cs_new_protected:Npn \__tl_peek_analysis_active_str:n #1
  {
    \tl_put_right:Ne \l__tl_peek_code_tl
      {
        { \char_generate:nn { `#1 } { 13 } }
        { \int_value:w `#1 }
        \token_to_str:N D
      }
    \l__tl_peek_code_tl
  }
\cs_new_protected:Npn \__tl_peek_analysis_explicit:n #1
  {
    \tl_put_right:Ne \l__tl_peek_code_tl
      {
        \if_meaning:w \l_peek_token \c_space_token
          { ~ } { 32 } \token_to_str:N A
        \else:
          \if_catcode:w \l_peek_token \c_group_begin_token
            {
              \exp_not:N \exp_after:wN
              \char_generate:nn { `#1 } { 1 }
              \exp_not:N \if_false:
              \if_false: { \fi: }
              \exp_not:N \fi:
            }
            { \int_value:w `#1 }
            1
          \else:
            {
              \exp_not:N \if_false:
              { \if_false: } \fi:
              \exp_not:N \fi:
              \char_generate:nn { `#1 } { 2 }
            }
            { \int_value:w `#1 }
            2
          \fi:
        \fi:
      }
    \l__tl_peek_code_tl
  }
\cs_new_protected:Npn \__tl_peek_analysis_escape:
  {
    \tl_clear:N \l__tl_internal_a_tl
    \tex_futurelet:D \l__tl_analysis_token
      \__tl_peek_analysis_collect:w
  }
\cs_new_protected:Npn \__tl_peek_analysis_collect:w
  { \__tl_analysis_char_arg:Nw \__tl_peek_analysis_collect:n }
\cs_new_protected:Npn \__tl_peek_analysis_collect:n #1
  {
    \tl_put_right:Nn \l__tl_internal_a_tl {#1}
    \__tl_peek_analysis_collect_loop:
  }
\cs_new_protected:Npn \__tl_peek_analysis_collect_loop:
  {
    \exp_after:wN \if_meaning:w
      \cs:w
      \if_cs_exist:w \l__tl_internal_a_tl \cs_end:
        \l__tl_internal_a_tl
      \else:
        c_one % anything short
      \fi:
      \cs_end:
      \l_peek_token
      \__tl_peek_analysis_collect_end:NNNN
    \fi:
    \tex_futurelet:D \l__tl_analysis_token
      \__tl_peek_analysis_collect:w
  }
\cs_new_protected:Npn \__tl_peek_analysis_collect_end:NNNN #1#2#3#4
  {
    #1
    \tl_put_right:Ne \l__tl_peek_code_tl
```

```
      {
        { \exp_not:N \exp_not:n { \exp_not:c { \l__tl_internal_a_tl } } }
        { -1 }
        0
      }
    \l__tl_peek_code_tl
  }
\tl_const:Ne \c__tl_analysis_show_etc_str % (
  { \token_to_str:N \ETC.) }
\msg_new:nnn { tl } { show-analysis }
  {
    The~token~list~ \tl_if_empty:nF {#1} { #1 ~ }
    \tl_if_empty:nTF {#2}
      { is~empty #3 }
      { contains~the~tokens: #2 #4 }
  }
%% File: l3benchmark.dtx
\fp_new:N \g_benchmark_duration_target_fp
\fp_gset:Nn \g_benchmark_duration_target_fp { 1 }
\int_new:N \g__benchmark_nesting_int
\cs_new_protected:Npn \__benchmark_raw:nN #1
  {
    \int_gincr:N \g__benchmark_nesting_int
    \exp_args:Nc \__benchmark_raw_aux:N
      { g__benchmark_ \int_use:N \g__benchmark_nesting_int _int }
    \__benchmark_raw_aux:
    #1
    \__benchmark_raw_end:N
  }
\cs_new_protected:Npn \__benchmark_raw_aux:N #1
  {
    \int_gzero_new:N #1
    \cs_gset_protected:Npn \__benchmark_raw_aux: { \int_gset:Nn #1 { \sys_timer: } }
  }
\cs_new_protected:Npn \__benchmark_raw_end:N #1
  {
    \int_gset:Nn #1
      {
        \sys_timer: -
        \int_use:c { g__benchmark_ \int_use:N \g__benchmark_nesting_int _int }
      }
    \int_gdecr:N \g__benchmark_nesting_int
  }
\cs_new_eq:NN \__benchmark_tmp:w ?
\cs_new_protected:Npn \__benchmark_raw_replicate:nnN #1
  {
    \int_compare:nNnTF {#1} > { 500000 }
      { \__benchmark_raw_replicate_large:nnN {#1} }
      { \__benchmark_raw_replicate_small:nnN {#1} }
  }
\cs_new_protected:Npn \__benchmark_raw_replicate_large:nnN #1#2
  {
    \cs_set:Npe \__benchmark_tmp:w ##1 { \prg_replicate:nn { 5000 } {##1} }
    \exp_args:Nno \__benchmark_raw_replicate:nnN { #1 / 5000 }
      { \__benchmark_tmp:w {#2} }
  }
\cs_new_protected:Npn \__benchmark_raw_replicate_small:nnN #1#2
  {
    \cs_set:Npe \__benchmark_tmp:w ##1##2 { \prg_replicate:nn {#1} {##1} }
    \__benchmark_raw:nN { \__benchmark_tmp:w {#2} { } } \g__benchmark_time_int
    \exp_args:No \__benchmark_raw_replicate_aux:nnN
      { \int_use:N \g__benchmark_time_int } {#2}
  }
\cs_new_protected:Npn \__benchmark_raw_replicate_aux:nnN #1#2#3
  {
    \__benchmark_raw:nN { \__benchmark_tmp:w { } {#2} } \g__benchmark_time_int
    \int_gset:Nn #3 { #1 - \g__benchmark_time_int }
    \cs_set_eq:NN \__benchmark_tmp:w \prg_do_nothing:
  }
\fp_new:N \g_benchmark_time_fp
\fp_new:N \g_benchmark_ops_fp
\int_new:N \g__benchmark_duration_int
\int_new:N \g__benchmark_time_int
\int_new:N \g__benchmark_time_a_int
\int_new:N \g__benchmark_time_b_int
\int_new:N \g__benchmark_time_c_int
\int_new:N \g__benchmark_time_d_int
\int_new:N \g__benchmark_repeat_int
\tl_new:N \g__benchmark_code_tl
\cs_new_protected:Npn \benchmark_once:n #1
  {
    \benchmark_once_silent:n {#1}
    \__benchmark_display:
  }
\cs_new_protected:Npn \benchmark_once_silent:n #1
  {
    \__benchmark_measure_op:
    \__benchmark_raw:nN {#1} \g__benchmark_time_int
    \fp_gset:Nn \g_benchmark_time_fp { \g__benchmark_time_int / 65536 }
    \fp_gset:Nn \g_benchmark_ops_fp { \g_benchmark_time_fp / \g__benchmark_one_op_fp }
  }
\cs_new_protected:Npn \benchmark:n #1
```

```
  {
    \benchmark_silent:n {#1}
    \__benchmark_display:
  }
\cs_new_protected:Npn \benchmark_silent:n #1
  {
    \__benchmark_measure_op:
    \tl_gset:Nn \g__benchmark_code_tl {#1}
    \__benchmark_aux:
    \fp_gset:Nn \g_benchmark_ops_fp { \g_benchmark_time_fp / \g__benchmark_one_op_fp }
  }
\cs_new_protected:Npn \__benchmark_aux:
  {
    \int_gset:Nn \g__benchmark_repeat_int { 1 }
    \__benchmark_raw:nN { \g__benchmark_code_tl } \g__benchmark_time_int
    \int_compare:nNnF \g__benchmark_time_int < { \g__benchmark_duration_int / 2 }
      { \prg_break: }
    \bool_until_do:nn
      {
        \int_compare_p:nNn \g__benchmark_time_int > { \g__benchmark_duration_int / 32 }
        || \int_compare_p:nNn \g__benchmark_repeat_int > { \c_max_int / 4 }
      }
      {
        \int_gset:Nn \g__benchmark_repeat_int { 4 * \g__benchmark_repeat_int }
        \__benchmark_run:N \g__benchmark_time_int
      }
    \int_gset:Nn \g__benchmark_repeat_int
      {
        \fp_to_int:n
          {
            max ( 1 , min ( \c_max_int ,
            \g__benchmark_duration_int * \g__benchmark_repeat_int /
            \int_eval:n { 4 * \g__benchmark_time_int } ) )
          }
      }
    \int_compare:nNnTF \g__benchmark_repeat_int = 1
      { \int_gset_eq:NN \g__benchmark_time_a_int \g__benchmark_time_int }
      { \__benchmark_run:N \g__benchmark_time_a_int }
    \__benchmark_run:N \g__benchmark_time_b_int
    \__benchmark_run:N \g__benchmark_time_c_int
    \__benchmark_run:N \g__benchmark_time_d_int
    \int_gset:Nn \g__benchmark_time_int
      {
        \int_min:nn
          { \int_min:nn \g__benchmark_time_a_int \g__benchmark_time_b_int }
          { \int_min:nn \g__benchmark_time_c_int \g__benchmark_time_d_int }
      }
    \prg_break_point:
    \int_compare:nNnT \g__benchmark_time_int < 3 { \int_gzero:N \g__benchmark_time_int }
    \fp_gset:Nn \g_benchmark_time_fp
      { \g__benchmark_time_int / \g__benchmark_repeat_int / 65536 }
  }
\cs_new_protected:Npn \__benchmark_run:N
  { \exp_args:NNo \__benchmark_raw_replicate:nnN \g__benchmark_repeat_int { \g__benchmark_code_tl } }
\fp_new:N \g__benchmark_one_op_fp
\cs_new_protected:Npn \__benchmark_measure_op:
  {
    \int_gset:Nn \g__benchmark_duration_int
      { \fp_to_int:n { 65536 * \g_benchmark_duration_target_fp } / 4 }
    \tl_gset:Nn \g__benchmark_code_tl
      { \int_gadd:Nn \g__benchmark_duration_int { 0 } }
    \__benchmark_aux:
    \fp_gset:Nn \g__benchmark_one_op_fp { max(\g_benchmark_time_fp, 1e-8) }
    \int_gset:Nn \g__benchmark_duration_int
      { \fp_to_int:n { 65536 * \g_benchmark_duration_target_fp } }
  }
\cs_new:Npn \__benchmark_fp_to_tl:N #1
  {
    \fp_compare:nTF { abs(#1) < 1000 }
      { \fp_to_tl:n { round(#1, 2 - logb(#1)) } }
      {
        \exp_args:Nf \__benchmark_fp_to_tl_aux:nN
          { \fp_to_int:n { logb(#1) } } #1
      }
  }
\cs_new:Npn \__benchmark_fp_to_tl_aux:nN #1#2
  { \fp_to_tl:n { round(#2 * 1e-#1, 2) } e#1 }
\cs_new_protected:Npn \__benchmark_display:
  {
    \iow_term:e
      {
        \__benchmark_fp_to_tl:N \g_benchmark_time_fp \c_space_tl seconds \c_space_tl
        ( \__benchmark_fp_to_tl:N \g_benchmark_ops_fp \c_space_tl ops)
      }
  }
\int_new:N \g__benchmark_tictoc_int
\seq_new:N \g__benchmark_tictoc_seq
\tl_new:N \l__benchmark_tictoc_pop_tl
\cs_new:Npn \__benchmark_tictoc_prefix:
  {
    (l3benchmark) \c_space_tl
    + \prg_replicate:nn { \g__benchmark_tictoc_int } { -+ } \c_space_tl
  }
\cs_new_protected:Npn \benchmark_tic:
  {
    \iow_term:e { \__benchmark_tictoc_prefix: TIC }
    \exp_args:NNf \seq_gput_right:Nn \g__benchmark_tictoc_seq { \sys_timer: }
    \int_gincr:N \g__benchmark_tictoc_int
  }
\cs_new:Npn \benchmark_toc:
  {
    \seq_gpop_right:NNTF \g__benchmark_tictoc_seq \l__benchmark_tictoc_pop_tl
      { \__benchmark_toc: }
      { \msg_error:nn { benchmark } { toc-first } }
  }
\cs_new_protected:Npn \__benchmark_toc:
  {
    \int_gdecr:N \g__benchmark_tictoc_int
    \fp_gset:Nn \g_benchmark_time_fp
      { ( \sys_timer: - \l__benchmark_tictoc_pop_tl ) / 65536 }
    \iow_term:e
      {
        \__benchmark_tictoc_prefix:
        TOC: \c_space_tl
        \__benchmark_fp_to_tl:N \g_benchmark_time_fp \c_space_tl s
      }
  }
\msg_new:nnn { benchmark } { toc-first }
  {
    \token_to_str:N \benchmark_toc: \c_space_tl without~
    \token_to_str:N \benchmark_tic: \c_space_tl !
  }
%% File: l3regex.dtx
\cs_new_eq:NN \__regex_int_eval:w \tex_numexpr:D
\cs_new_eq:NN \__regex_sep: \__kernel_int_sep:
\cs_new_protected:Npn \__regex_standard_escapechar:
  { \int_set:Nn \tex_escapechar:D { `\\ } }
\cs_new:Npn \__regex_toks_use:w { \tex_the:D \tex_toks:D }
\cs_new_protected:Npn \__regex_toks_clear:N #1
  { \tex_toks:D #1 = { } }
\cs_new_eq:NN \__regex_toks_set:Nn \tex_toks:D
\cs_new_protected:Npn \__regex_toks_set:No #1
  { \tex_toks:D #1 = \exp_after:wN }
\cs_new_protected:Npn \__regex_toks_memcpy:NNn #1#2#3
  {
    \prg_replicate:nn {#3}
      {
        \tex_toks:D #1 = \tex_toks:D #2
        \int_incr:N #1
        \int_incr:N #2
      }
  }
\cs_if_exist:NTF \tex_etokspre:D
  { \cs_new_eq:NN \__regex_toks_put_left:Ne \tex_etokspre:D }
  {
    \cs_new_protected:Npn \__regex_toks_put_left:Ne #1#2
      { \tex_toks:D #1 = \tex_expanded:D {{ #2 \tex_the:D \tex_toks:D #1 }} }
  }
\cs_if_exist:NTF \tex_etoksapp:D
  { \cs_new_eq:NN \__regex_toks_put_right:Ne \tex_etoksapp:D }
  {
    \cs_new_protected:Npn \__regex_toks_put_right:Ne #1#2
      { \tex_toks:D #1 = \tex_expanded:D {{ \tex_the:D \tex_toks:D #1 #2 }} }
  }
\cs_if_exist:NTF \tex_toksapp:D
  { \cs_new_eq:NN \__regex_toks_put_right:Nn \tex_toksapp:D }
  {
    \cs_new_protected:Npn \__regex_toks_put_right:Nn #1#2
      { \tex_toks:D #1 = \exp_after:wN { \tex_the:D \tex_toks:D #1 #2 } }
  }
\cs_new:Npn \__regex_curr_cs_to_str:
  {
    \exp_after:wN \exp_after:wN \exp_after:wN \cs_to_str:N
    \l__regex_curr_token_tl
  }
\cs_new:Npn \__regex_intarray_item:NnF #1#2
  { \exp_args:No \__regex_intarray_item_aux:nNF { \tex_the:D \__regex_int_eval:w #2 } #1 }
\cs_new:Npn \__regex_intarray_item_aux:nNF #1#2
  {
    \if_int_compare:w #1 > \c_zero_int
      \exp_after:wN \use_ii:nnn
    \fi:
    \use_ii:nn { \__kernel_intarray_item:Nn #2 {#1} }
  }
\cs_new:Npn \__regex_maplike_break:
  { \prg_map_break:Nn \__regex_maplike_break: { } }
\cs_new:Npn \__regex_tl_odd_items:n #1 { \__regex_tl_even_items:n { ? #1 } }
\cs_new:Npn \__regex_tl_even_items:n #1
  {
    \__regex_tl_even_items_loop:nn #1 \q__regex_nil \q__regex_nil
    \prg_break_point:
  }
```

```
\cs_new:Npn \__regex_tl_even_items_loop:nn #1#2
  {
    \__regex_use_none_delimit_by_q_nil:w #2 \prg_break: \q__regex_nil
    { \exp_not:n {#2} }
    \__regex_tl_even_items_loop:nn
  }
\cs_new:Npn \__regex_tmp:w { }
\tl_new:N   \l__regex_internal_a_tl
\tl_new:N   \l__regex_internal_b_tl
\int_new:N  \l__regex_internal_a_int
\int_new:N  \l__regex_internal_b_int
\int_new:N  \l__regex_internal_c_int
\bool_new:N \l__regex_internal_bool
\seq_new:N  \l__regex_internal_seq
\tl_new:N   \g__regex_internal_tl
\tl_new:N \l__regex_build_tl
\tl_const:Nn \c__regex_no_match_regex
  {
    \__regex_branch:n
      { \__regex_class:NnnnN \c_true_bool { } { 1 } { 0 } \c_true_bool }
  }
\int_new:N \l__regex_balance_int
\int_const:Nn \c__regex_ascii_min_int { 0 }
\int_const:Nn \c__regex_ascii_max_control_int { 31 }
\int_const:Nn \c__regex_ascii_max_int { 127 }
\int_const:Nn \c__regex_ascii_lower_int { `a - `A }
\quark_new:N \q__regex_recursion_stop
\quark_new:N \q__regex_nil
\cs_new:Npn \__regex_use_none_delimit_by_q_recursion_stop:w
  #1 \q__regex_recursion_stop { }
\cs_new:Npn \__regex_use_i_delimit_by_q_recursion_stop:nw
  #1 #2 \q__regex_recursion_stop {#1}
\cs_new:Npn \__regex_use_none_delimit_by_q_nil:w #1 \q__regex_nil { }
\__kernel_quark_new_conditional:Nn \__regex_quark_if_nil:N { F }
\cs_new_protected:Npn \__regex_break_true:w
  #1 \__regex_break_point:TF #2 #3 {#2}
\cs_new_protected:Npn \__regex_break_point:TF #1 #2 { #2 }
\cs_new_protected:Npn \__regex_item_reverse:n #1
  {
    #1
    \__regex_break_point:TF { } \__regex_break_true:w
  }
\cs_new_protected:Npn \__regex_item_caseful_equal:n #1
  {
    \if_int_compare:w #1 = \l__regex_curr_char_int
      \exp_after:wN \__regex_break_true:w
    \fi:
  }
\cs_new_protected:Npn \__regex_item_caseful_range:nn #1 #2
  {
    \reverse_if:N \if_int_compare:w #1 > \l__regex_curr_char_int
      \reverse_if:N \if_int_compare:w #2 < \l__regex_curr_char_int
        \exp_after:wN \exp_after:wN \exp_after:wN \__regex_break_true:w
      \fi:
    \fi:
  }
\cs_new_protected:Npn \__regex_item_caseless_equal:n #1
  {
    \if_int_compare:w #1 = \l__regex_curr_char_int
      \exp_after:wN \__regex_break_true:w
    \fi:
    \__regex_maybe_compute_ccc:
    \if_int_compare:w #1 = \l__regex_case_changed_char_int
      \exp_after:wN \__regex_break_true:w
    \fi:
  }
\cs_new_protected:Npn \__regex_item_caseless_range:nn #1 #2
  {
    \reverse_if:N \if_int_compare:w #1 > \l__regex_curr_char_int
      \reverse_if:N \if_int_compare:w #2 < \l__regex_curr_char_int
        \exp_after:wN \exp_after:wN \exp_after:wN \__regex_break_true:w
      \fi:
    \fi:
    \__regex_maybe_compute_ccc:
    \reverse_if:N \if_int_compare:w #1 > \l__regex_case_changed_char_int
      \reverse_if:N \if_int_compare:w #2 < \l__regex_case_changed_char_int
        \exp_after:wN \exp_after:wN \exp_after:wN \__regex_break_true:w
      \fi:
    \fi:
  }
\cs_new_protected:Npn \__regex_compute_case_changed_char:
  {
    \int_set_eq:NN \l__regex_case_changed_char_int \l__regex_curr_char_int
    \if_int_compare:w \l__regex_curr_char_int > `Z \exp_stop_f:
      \if_int_compare:w \l__regex_curr_char_int > `z \exp_stop_f: \else:
        \if_int_compare:w \l__regex_curr_char_int < `a \exp_stop_f: \else:
          \int_sub:Nn \l__regex_case_changed_char_int \c__regex_ascii_lower_int
        \fi:
      \fi:
    \else:
      \if_int_compare:w \l__regex_curr_char_int < `A \exp_stop_f: \else:
        \int_add:Nn \l__regex_case_changed_char_int \c__regex_ascii_lower_int
      \fi:
    \fi:
    \cs_set_eq:NN \__regex_maybe_compute_ccc: \prg_do_nothing:
  }
\cs_new_eq:NN \__regex_maybe_compute_ccc: \__regex_compute_case_changed_char:
\cs_new_eq:NN \__regex_item_equal:n ?
\cs_new_eq:NN \__regex_item_range:nn ?
\cs_new_protected:Npn \__regex_item_catcode:
  {
    "
    \if_case:w \l__regex_curr_catcode_int
         1       \or: 4       \or: 10      \or: 40
    \or: 100     \or:         \or: 1000    \or: 4000
    \or: 10000   \or:         \or: 100000  \or: 400000
    \or: 1000000 \or: 4000000 \else: 1*0
    \fi:
  }
\prg_new_protected_conditional:Npnn \__regex_item_catcode:n #1 { T }
  {
    \if_int_odd:w \__regex_int_eval:w #1 / \__regex_item_catcode: \scan_stop:
      \prg_return_true:
    \else:
      \prg_return_false:
    \fi:
  }
\cs_new_protected:Npn \__regex_item_catcode_reverse:nT #1#2
  { \__regex_item_catcode:nT {#1} { \__regex_item_reverse:n {#2} } }
\cs_new_protected:Npn \__regex_item_exact:nn #1#2
  {
    \if_int_compare:w #1 = \l__regex_curr_catcode_int
      \if_int_compare:w #2 = \l__regex_curr_char_int
        \exp_after:wN \exp_after:wN \exp_after:wN \__regex_break_true:w
      \fi:
    \fi:
  }
\cs_new_protected:Npn \__regex_item_exact_cs:n #1
  {
    \int_compare:nNnTF \l__regex_curr_catcode_int = \c_zero_int
      {
        \__kernel_tl_set:Nx \l__regex_internal_a_tl
          { \scan_stop: \__regex_curr_cs_to_str: \scan_stop: }
        \tl_if_in:noTF { \scan_stop: #1 \scan_stop: }
          \l__regex_internal_a_tl
          { \__regex_break_true:w } { }
      }
      { }
  }
\cs_new_protected:Npn \__regex_item_cs:n #1
  {
    \int_compare:nNnT \l__regex_curr_catcode_int = \c_zero_int
      {
        \group_begin:
          \__regex_single_match:
          \__regex_disable_submatches:
          \__regex_build_for_cs:n {#1}
          \bool_set_eq:NN \l__regex_saved_success_bool
            \g__regex_success_bool
          \exp_args:Ne \__regex_match_cs:n { \__regex_curr_cs_to_str: }
          \if_meaning:w \c_true_bool \g__regex_success_bool
            \group_insert_after:N \__regex_break_true:w
          \fi:
          \bool_gset_eq:NN \g__regex_success_bool
            \l__regex_saved_success_bool
        \group_end:
      }
  }
\cs_new_protected:Npn \__regex_prop_d:
  { \__regex_item_caseful_range:nn { `0 } { `9 } }
\cs_new_protected:Npn \__regex_prop_h:
  {
    \__regex_item_caseful_equal:n { `\␣}
    \__regex_item_caseful_equal:n { `\^^I }
  }
\cs_new_protected:Npn \__regex_prop_s:
  {
    \__regex_item_caseful_equal:n { `\␣}
    \__regex_item_caseful_equal:n { `\^^I }
    \__regex_item_caseful_equal:n { `\^^J }
    \__regex_item_caseful_equal:n { `\^^L }
    \__regex_item_caseful_equal:n { `\^^M }
  }
\cs_new_protected:Npn \__regex_prop_v:
  { \__regex_item_caseful_range:nn { `\^^J } { `\^^M } } % lf, vtab, ff, cr
\cs_new_protected:Npn \__regex_prop_w:
  {
    \__regex_item_caseful_range:nn { `a } { `z }
    \__regex_item_caseful_range:nn { `A } { `Z }
    \__regex_item_caseful_range:nn { `0 } { `9 }
    \__regex_item_caseful_equal:n { `_ }
  }
\cs_new_protected:Npn \__regex_prop_N:
  {
    \__regex_item_reverse:n
      { \__regex_item_caseful_equal:n { `\^^J } }
  }
\cs_new_protected:Npn \__regex_posix_alnum:
  { \__regex_posix_alpha: \__regex_posix_digit: }
\cs_new_protected:Npn \__regex_posix_alpha:
  { \__regex_posix_lower: \__regex_posix_upper: }
\cs_new_protected:Npn \__regex_posix_ascii:
  {
    \__regex_item_caseful_range:nn
      \c__regex_ascii_min_int
      \c__regex_ascii_max_int
  }
\cs_new_eq:NN \__regex_posix_blank: \__regex_prop_h:
\cs_new_protected:Npn \__regex_posix_cntrl:
  {
    \__regex_item_caseful_range:nn
      \c__regex_ascii_min_int
      \c__regex_ascii_max_control_int
    \__regex_item_caseful_equal:n \c__regex_ascii_max_int
  }
\cs_new_eq:NN \__regex_posix_digit: \__regex_prop_d:
\cs_new_protected:Npn \__regex_posix_graph:
  { \__regex_item_caseful_range:nn { `! } { `\~ } }
\cs_new_protected:Npn \__regex_posix_lower:
  { \__regex_item_caseful_range:nn { `a } { `z } }
\cs_new_protected:Npn \__regex_posix_print:
  { \__regex_item_caseful_range:nn { `\␣ } { `\~ } }
\cs_new_protected:Npn \__regex_posix_punct:
  {
    \__regex_item_caseful_range:nn { `! } { `/ }
    \__regex_item_caseful_range:nn { `: } { `@ }
    \__regex_item_caseful_range:nn { `[ } { `` }
    \__regex_item_caseful_range:nn { `\{ } { `\~ }
  }
\cs_new_protected:Npn \__regex_posix_space:
  {
    \__regex_item_caseful_equal:n { `\␣ }
    \__regex_item_caseful_range:nn { `\^^I } { `\^^M }
  }
\cs_new_protected:Npn \__regex_posix_upper:
  { \__regex_item_caseful_range:nn { `A } { `Z } }
\cs_new_eq:NN \__regex_posix_word: \__regex_prop_w:
\cs_new_protected:Npn \__regex_posix_xdigit:
  {
    \__regex_posix_digit:
```

```
    \__regex_item_caseful_range:nn { `A } { `F }
    \__regex_item_caseful_range:nn { `a } { `f }
  }
\cs_new_protected:Npn \__regex_escape_use:nnnn #1#2#3#4
  {
    \group_begin:
      \tl_clear:N \l__regex_internal_a_tl
      \cs_set:Npn \__regex_escape_unescaped:N ##1 { #1 }
      \cs_set:Npn \__regex_escape_escaped:N ##1 { #2 }
      \cs_set:Npn \__regex_escape_raw:N ##1 { #3 }
      \__regex_standard_escapechar:
      \__kernel_tl_gset:Nx \g__regex_internal_tl
        { \__kernel_str_to_other_fast:n {#4} }
      \tl_put_right:Ne \l__regex_internal_a_tl
        {
          \exp_after:wN \__regex_escape_loop:N \g__regex_internal_tl
          \scan_stop: \prg_break_point:
        }
      \exp_after:wN
    \group_end:
    \l__regex_internal_a_tl
  }
\cs_new:Npn \__regex_escape_loop:N #1
  {
    \cs_if_exist_use:cF { __regex_escape_\token_to_str:N #1:w }
      { \__regex_escape_unescaped:N #1 }
    \__regex_escape_loop:N
  }
\cs_new:cpn { __regex_escape_ \c_backslash_str :w }
    \__regex_escape_loop:N #1
  {
    \cs_if_exist_use:cF { __regex_escape_/\token_to_str:N #1:w }
      { \__regex_escape_escaped:N #1 }
    \__regex_escape_loop:N
  }
\cs_new_eq:NN \__regex_escape_unescaped:N ?
\cs_new_eq:NN \__regex_escape_escaped:N   ?
\cs_new_eq:NN \__regex_escape_raw:N       ?
\cs_new_eq:cN { __regex_escape_ \iow_char:N\\scan_stop: :w } \prg_break:
\cs_new:cpn { __regex_escape_/ \iow_char:N\\scan_stop: :w }
  {
    \msg_expandable_error:nn { regex } { trailing-backslash }
    \prg_break:
  }
\cs_new:cpn { __regex_escape_~:w } { }
\cs_new:cpe { __regex_escape_/a:w }
```

```
  { \exp_not:N \__regex_escape_raw:N \iow_char:N \^^G }
\cs_new:cpe { __regex_escape_/t:w }
  { \exp_not:N \__regex_escape_raw:N \iow_char:N \^^I }
\cs_new:cpe { __regex_escape_/n:w }
  { \exp_not:N \__regex_escape_raw:N \iow_char:N \^^J }
\cs_new:cpe { __regex_escape_/f:w }
  { \exp_not:N \__regex_escape_raw:N \iow_char:N \^^L }
\cs_new:cpe { __regex_escape_/r:w }
  { \exp_not:N \__regex_escape_raw:N \iow_char:N \^^M }
\cs_new:cpe { __regex_escape_/e:w }
  { \exp_not:N \__regex_escape_raw:N \iow_char:N \^^[ }
\cs_new:cpn { __regex_escape_/x:w } \__regex_escape_loop:N
  {
    \exp_after:wN \__regex_escape_x_end:w
    \int_value:w "0 \__regex_escape_x_test:N
  }
\cs_new:Npn \__regex_escape_x_end:w #1 \__regex_sep:
  {
    \int_compare:nNnTF {#1} > \c_max_char_int
      {
        \msg_expandable_error:nnff { regex } { x-overflow }
          {#1} { \int_to_Hex:n {#1} }
      }
      {
        \exp_last_unbraced:Nf \__regex_escape_raw:N
          { \char_generate:nn {#1} { 12 } }
      }
  }
\cs_new:Npn \__regex_escape_x_test:N #1
  {
    \if_meaning:w \scan_stop: #1
      \exp_after:wN \use_i:nnn \exp_after:wN \__regex_sep:
    \fi:
    \use:n
      {
        \if_charcode:w \c_space_token #1
          \exp_after:wN \__regex_escape_x_test:N
        \else:
          \exp_after:wN \__regex_escape_x_testii:N
          \exp_after:wN #1
        \fi:
      }
  }
\cs_new:Npn \__regex_escape_x_testii:N #1
  {
    \if_charcode:w \c_left_brace_str #1
```

```
      \exp_after:wN \__regex_escape_x_loop:N
    \else:
      \__regex_hexadecimal_use:NTF #1
        { \exp_after:wN \__regex_escape_x:N }
        { \__regex_sep: \exp_after:wN \__regex_escape_loop:N \exp_after:wN #1 }
    \fi:
  }
\cs_new:Npn \__regex_escape_x:N #1
  {
    \if_meaning:w \scan_stop: #1
      \exp_after:wN \use_i:nnn \exp_after:wN \__regex_sep:
    \fi:
    \use:n
      {
        \__regex_hexadecimal_use:NTF #1
          { \__regex_sep: \__regex_escape_loop:N }
          { \__regex_sep: \__regex_escape_loop:N #1 }
      }
  }
\cs_new:Npn \__regex_escape_x_loop:N #1
  {
    \if_meaning:w \scan_stop: #1
      \exp_after:wN \use_ii:nnn
    \fi:
    \use_ii:nn
      { \__regex_sep: \__regex_escape_x_loop_error:n { } {#1} }
      {
        \__regex_hexadecimal_use:NTF #1
          { \__regex_escape_x_loop:N }
          {
            \token_if_eq_charcode:NNTF \c_space_token #1
              { \__regex_escape_x_loop:N }
              {
                \__regex_sep:
                \exp_after:wN
                \token_if_eq_charcode:NNTF \c_right_brace_str #1
                  { \__regex_escape_loop:N }
                  { \__regex_escape_x_loop_error:n {#1} }
              }
          }
      }
  }
\cs_new:Npn \__regex_escape_x_loop_error:n #1
  {
    \msg_expandable_error:nnn { regex } { x-missing-rbrace } {#1}
    \__regex_escape_loop:N #1
  }
\cs_new:Npn \__regex_hexadecimal_use:NTF #1
  {
    \if_int_compare:w \c_one_int < "1 \token_to_str:N #1 \exp_stop_f:
      #1
    \else:
      \if_case:w
        \__regex_int_eval:w \exp_after:wN ` \token_to_str:N #1 - `a \scan_stop:
           A
      \or: B
      \or: C
      \or: D
      \or: E
      \or: F
      \else:
        \exp_after:wN \exp_after:wN \exp_after:wN \use_iii:nnn
      \fi:
    \fi:
    \use_i:nn
  }
\prg_new_conditional:Npnn \__regex_char_if_special:N #1 { TF }
  {
    \if:w
        T
        \if_int_compare:w `#1 > `Z \exp_stop_f:
          \if_int_compare:w `#1 > `z \exp_stop_f:
            \if_int_compare:w `#1 < \c__regex_ascii_max_int
              \else: F \fi:
          \else:
            \if_int_compare:w `#1 < `a \exp_stop_f:
              \else: F \fi:
          \fi:
        \else:
          \if_int_compare:w `#1 > `9 \exp_stop_f:
            \if_int_compare:w `#1 < `A \exp_stop_f:
              \else: F \fi:
          \else:
            \if_int_compare:w `#1 < `0 \exp_stop_f:
              \if_int_compare:w `#1 < `\␣\exp_stop_f:
                F \fi:
            \else: F \fi:
          \fi:
        \fi:
        T
      \prg_return_true:
    \else:
```

```
   \prg_return_false:
  \fi:
 }
\prg_new_conditional:Npnn \__regex_char_if_alphanumeric:N #1 { TF }
 {
  \if:w
    T
    \if_int_compare:w `#1 > `Z \exp_stop_f:
     \if_int_compare:w `#1 > `z \exp_stop_f:
      F
     \else:
      \if_int_compare:w `#1 < `a \exp_stop_f:
       F \fi:
     \fi:
    \else:
     \if_int_compare:w `#1 > `9 \exp_stop_f:
      \if_int_compare:w `#1 < `A \exp_stop_f:
       F \fi:
     \else:
      \if_int_compare:w `#1 < `0 \exp_stop_f:
       F \fi:
     \fi:
    \fi:
    T
   \prg_return_true:
  \else:
   \prg_return_false:
  \fi:
 }
\int_new:N \l__regex_group_level_int
\int_new:N \l__regex_mode_int
\int_const:Nn \c__regex_cs_in_class_mode_int { -6 }
\int_const:Nn \c__regex_cs_mode_int { -2 }
\int_const:Nn \c__regex_outer_mode_int { 0 }
\int_const:Nn \c__regex_catcode_mode_int { 2 }
\int_const:Nn \c__regex_class_mode_int { 3 }
\int_const:Nn \c__regex_catcode_in_class_mode_int { 6 }
\int_new:N \l__regex_catcodes_int
\int_new:N \l__regex_default_catcodes_int
\bool_new:N \l__regex_catcodes_bool
\int_const:Nn \c__regex_catcode_C_int { "1 }
\int_const:Nn \c__regex_catcode_B_int { "4 }
\int_const:Nn \c__regex_catcode_E_int { "10 }
\int_const:Nn \c__regex_catcode_M_int { "40 }
\int_const:Nn \c__regex_catcode_T_int { "100 }
\int_const:Nn \c__regex_catcode_P_int { "1000 }
```

```
\int_const:Nn \c__regex_catcode_U_int { "4000 }
\int_const:Nn \c__regex_catcode_D_int { "10000 }
\int_const:Nn \c__regex_catcode_S_int { "100000 }
\int_const:Nn \c__regex_catcode_L_int { "400000 }
\int_const:Nn \c__regex_catcode_O_int { "1000000 }
\int_const:Nn \c__regex_catcode_A_int { "4000000 }
\int_const:Nn \c__regex_all_catcodes_int { "5515155 }
\cs_new_eq:NN \l__regex_internal_regex \c__regex_no_match_regex
\seq_new:N \l__regex_show_prefix_seq
\int_new:N \l__regex_show_lines_int
\cs_new:Npn \__regex_two_if_eq:NNNNTF #1#2#3#4
  {
    \if_meaning:w #1 #3
      \if:w #2 #4
        \exp_after:wN \exp_after:wN \exp_after:wN \use_ii:nnn
      \fi:
    \fi:
    \use_ii:nn
  }
\cs_new_protected:Npn \__regex_get_digits:NTFw #1#2#3#4#5
  {
    \__regex_if_raw_digit:NNTF #4 #5
      { #1 = #5 \__regex_get_digits_loop:nw {#2} }
      { #3 #4 #5 }
  }
\cs_new:Npn \__regex_get_digits_loop:nw #1#2#3
  {
    \__regex_if_raw_digit:NNTF #2 #3
      { #3 \__regex_get_digits_loop:nw {#1} }
      { \scan_stop: #1 #2 #3 }
  }
\cs_new:Npn \__regex_if_raw_digit:NNTF #1#2
  {
    \if_meaning:w \__regex_compile_raw:N #1
      \if_int_compare:w \c_one_int < 1 #2 \exp_stop_f:
        \exp_after:wN \exp_after:wN \exp_after:wN \use_ii:nnn
      \fi:
    \fi:
    \use_ii:nn
  }
\prg_new_conditional:Npnn \__regex_if_in_class: { TF }
  {
    \if_int_odd:w \l__regex_mode_int
      \prg_return_true:
    \else:
      \prg_return_false:
```

```
    \fi:
  }
\cs_new:Npn \__regex_if_in_cs:TF
  {
    \if_int_odd:w \l__regex_mode_int
    \else:
      \if_int_compare:w \l__regex_mode_int < \c__regex_outer_mode_int
        \exp_after:wN \exp_after:wN \exp_after:wN \use_ii:nnn
      \fi:
    \fi:
    \use_ii:nn
  }
\cs_new:Npn \__regex_if_in_class_or_catcode:TF
  {
    \if_int_odd:w \l__regex_mode_int
    \else:
      \if_int_compare:w \l__regex_mode_int > \c__regex_outer_mode_int
      \else:
        \exp_after:wN \exp_after:wN \exp_after:wN \use_iii:nnn
      \fi:
    \fi:
    \use_i:nn
  }
\prg_new_conditional:Npnn \__regex_if_within_catcode: { TF }
  {
    \if_int_compare:w \l__regex_mode_int > \c__regex_outer_mode_int
      \prg_return_true:
    \else:
      \prg_return_false:
    \fi:
  }
\cs_new_protected:Npn \__regex_chk_c_allowed:T
  {
    \if_int_compare:w \l__regex_mode_int = \c__regex_outer_mode_int
    \else:
      \if_int_compare:w \l__regex_mode_int = \c__regex_class_mode_int
      \else:
        \msg_error:nn { regex } { c-bad-mode }
        \exp_after:wN \use_i:nnn
      \fi:
    \fi:
    \use:n
  }
\cs_new_protected:Npn \__regex_mode_quit_c:
  {
    \if_int_compare:w \l__regex_mode_int = \c__regex_catcode_mode_int
```

```
      \int_set_eq:NN \l__regex_mode_int \c__regex_outer_mode_int
    \else:
      \if_int_compare:w \l__regex_mode_int =
        \c__regex_catcode_in_class_mode_int
        \int_set_eq:NN \l__regex_mode_int \c__regex_class_mode_int
      \fi:
    \fi:
  }
\cs_new_protected:Npn \__regex_compile:w
  {
    \group_begin:
      \tl_build_begin:N \l__regex_build_tl
      \int_zero:N \l__regex_group_level_int
      \int_set_eq:NN \l__regex_default_catcodes_int
        \c__regex_all_catcodes_int
      \int_set_eq:NN \l__regex_catcodes_int \l__regex_default_catcodes_int
      \cs_set:Npn \__regex_item_equal:n  { \__regex_item_caseful_equal:n }
      \cs_set:Npn \__regex_item_range:nn { \__regex_item_caseful_range:nn }
      \tl_build_put_right:Nn \l__regex_build_tl
        { \__regex_branch:n { \if_false: } \fi: }
  }
\cs_new_protected:Npn \__regex_compile_end:
  {
      \__regex_if_in_class:TF
        {
          \msg_error:nn { regex } { missing-rbrack }
          \use:c { __regex_compile_]: }
          \prg_do_nothing: \prg_do_nothing:
        }
        { }
      \if_int_compare:w \l__regex_group_level_int > \c_zero_int
        \msg_error:nne { regex } { missing-rparen }
          { \int_use:N \l__regex_group_level_int }
        \prg_replicate:nn
          \l__regex_group_level_int
          {
              \tl_build_put_right:Nn \l__regex_build_tl
                {
                  \if_false: { \fi: }
                  \if_false: { \fi: } { 1 } { 0 } \c_true_bool
                }
              \tl_build_end:N \l__regex_build_tl
              \exp_args:NNNo
            \group_end:
            \tl_build_put_right:Nn \l__regex_build_tl
              { \l__regex_build_tl }
          }
      \fi:
      \tl_build_put_right:Nn \l__regex_build_tl { \if_false: { \fi: } }
      \tl_build_end:N \l__regex_build_tl
      \exp_args:NNNe
    \group_end:
    \tl_set:Nn \l__regex_internal_regex { \l__regex_build_tl }
  }
\cs_new_protected:Npn \__regex_compile:n #1
  {
    \__regex_compile:w
      \__regex_standard_escapechar:
      \int_set_eq:NN \l__regex_mode_int \c__regex_outer_mode_int
      \__regex_escape_use:nnnn
        {
          \__regex_char_if_special:NTF ##1
            \__regex_compile_special:N \__regex_compile_raw:N ##1
        }
        {
          \__regex_char_if_alphanumeric:NTF ##1
            \__regex_compile_escaped:N \__regex_compile_raw:N ##1
        }
        { \__regex_compile_raw:N ##1 }
        { #1 }
      \prg_do_nothing: \prg_do_nothing:
      \prg_do_nothing: \prg_do_nothing:
      \int_compare:nNnT \l__regex_mode_int = \c__regex_catcode_mode_int
        { \msg_error:nn { regex } { c-trailing } }
      \int_compare:nNnT \l__regex_mode_int < \c__regex_outer_mode_int
        {
          \msg_error:nn { regex } { c-missing-rbrace }
          \__regex_compile_end_cs:
          \prg_do_nothing: \prg_do_nothing:
          \prg_do_nothing: \prg_do_nothing:
        }
    \__regex_compile_end:
  }
\cs_new_protected:Npn \__regex_compile_use:n #1
  {
    \tl_if_single_token:nT {#1}
      {
        \exp_after:wN \__regex_compile_use_aux:w
        \token_to_meaning:N #1 ~ \q__regex_nil
      }
    \__regex_compile:n {#1} \l__regex_internal_regex
  }
```

```
\cs_new_protected:Npn \__regex_compile_use_aux:w #1 ~ #2 \q__regex_nil
  {
    \str_if_eq:nnT { #1 ~ } { macro:->\__regex_branch:n }
      { \use_ii:nnn }
  }
\cs_new_protected:Npn \__regex_compile_special:N #1
  {
    \cs_if_exist_use:cF { __regex_compile_#1: }
      { \__regex_compile_raw:N #1 }
  }
\cs_new_protected:Npn \__regex_compile_escaped:N #1
  {
    \cs_if_exist_use:cF { __regex_compile_/#1: }
      { \__regex_compile_raw:N #1 }
  }
\cs_new_protected:Npn \__regex_compile_one:n #1
  {
    \__regex_mode_quit_c:
    \__regex_if_in_class:TF { }
      {
        \tl_build_put_right:Nn \l__regex_build_tl
          { \__regex_class:NnnnN \c_true_bool { \if_false: } \fi: }
      }
    \tl_build_put_right:Ne \l__regex_build_tl
      {
        \if_int_compare:w \l__regex_catcodes_int <
          \c__regex_all_catcodes_int
          \__regex_item_catcode:nT { \int_use:N \l__regex_catcodes_int }
            { \exp_not:N \exp_not:n {#1} }
        \else:
          \exp_not:N \exp_not:n {#1}
        \fi:
      }
    \int_set_eq:NN \l__regex_catcodes_int \l__regex_default_catcodes_int
    \__regex_if_in_class:TF { } { \__regex_compile_quantifier:w }
  }
\cs_new_protected:Npn \__regex_compile_abort_tokens:n #1
  {
    \use:e
      {
        \exp_args:No \tl_map_function:nN { \tl_to_str:n {#1} }
          \__regex_compile_raw:N
      }
  }
\cs_generate_variant:Nn \__regex_compile_abort_tokens:n { e }
\cs_new_protected:Npn \__regex_compile_if_quantifier:TFw #1#2#3#4
```

```
  {
    \token_if_eq_meaning:NNTF #3 \__regex_compile_special:N
      { \cs_if_exist:cTF { __regex_compile_quantifier_#4:w } }
      { \use_ii:nn }
    {#1} {#2} #3 #4
  }
\cs_new_protected:Npn \__regex_compile_quantifier:w #1#2
  {
    \token_if_eq_meaning:NNTF #1 \__regex_compile_special:N
      {
        \cs_if_exist_use:cF { __regex_compile_quantifier_#2:w }
          { \__regex_compile_quantifier_none: #1 #2 }
      }
      { \__regex_compile_quantifier_none: #1 #2 }
  }
\cs_new_protected:Npn \__regex_compile_quantifier_none:
  {
    \tl_build_put_right:Nn \l__regex_build_tl
      { \if_false: { \fi: } { 1 } { 0 } \c_false_bool }
  }
\cs_new_protected:Npn \__regex_compile_quantifier_abort:eNN #1#2#3
  {
    \__regex_compile_quantifier_none:
    \msg_warning:nnee { regex } { invalid-quantifier } {#1} {#3}
    \__regex_compile_abort_tokens:e {#1}
    #2 #3
  }
\cs_new_protected:Npn \__regex_compile_quantifier_laziness:nnNN #1#2#3#4
  {
    \__regex_two_if_eq:NNNNTF #3 #4 \__regex_compile_special:N ?
      {
        \tl_build_put_right:Nn \l__regex_build_tl
          { \if_false: { \fi: } { #1 } { #2 } \c_true_bool }
      }
      {
        \tl_build_put_right:Nn \l__regex_build_tl
          { \if_false: { \fi: } { #1 } { #2 } \c_false_bool }
        #3 #4
      }
  }
\cs_new_protected:cpn { __regex_compile_quantifier_?:w }
  { \__regex_compile_quantifier_laziness:nnNN { 0 } { 1 } }
\cs_new_protected:cpn { __regex_compile_quantifier_*:w }
  { \__regex_compile_quantifier_laziness:nnNN { 0 } { -1 } }
\cs_new_protected:cpn { __regex_compile_quantifier_+:w }
  { \__regex_compile_quantifier_laziness:nnNN { 1 } { -1 } }
```

```
\cs_new_protected:cpn { __regex_compile_quantifier_ \c_left_brace_str :w }
  {
    \__regex_get_digits:NTFw \l__regex_internal_a_int
      { \__regex_compile_quantifier_braced_auxi:w }
      { \__regex_compile_quantifier_abort:eNN { \c_left_brace_str } }
  }
\cs_new_protected:Npn \__regex_compile_quantifier_braced_auxi:w #1#2
  {
    \str_case_e:nnF { #1 #2 }
      {
        { \__regex_compile_special:N \c_right_brace_str }
          {
            \exp_args:No \__regex_compile_quantifier_laziness:nnNN
              { \int_use:N \l__regex_internal_a_int } 0
          }
        { \__regex_compile_special:N , }
          {
            \__regex_get_digits:NTFw \l__regex_internal_b_int
              { \__regex_compile_quantifier_braced_auxiii:w }
              { \__regex_compile_quantifier_braced_auxii:w }
          }
      }
      {
        \__regex_compile_quantifier_abort:eNN
          { \c_left_brace_str \int_use:N \l__regex_internal_a_int }
        #1 #2
      }
  }
\cs_new_protected:Npn \__regex_compile_quantifier_braced_auxii:w #1#2
  {
    \__regex_two_if_eq:NNNNTF #1 #2 \__regex_compile_special:N \c_right_brace_str
      {
        \exp_args:No \__regex_compile_quantifier_laziness:nnNN
          { \int_use:N \l__regex_internal_a_int } { -1 }
      }
      {
        \__regex_compile_quantifier_abort:eNN
          { \c_left_brace_str \int_use:N \l__regex_internal_a_int , }
        #1 #2
      }
  }
\cs_new_protected:Npn \__regex_compile_quantifier_braced_auxiii:w #1#2
  {
    \__regex_two_if_eq:NNNNTF #1 #2 \__regex_compile_special:N \c_right_brace_str
      {
        \if_int_compare:w \l__regex_internal_a_int >
```

```
          \l__regex_internal_b_int
          \msg_error:nnee { regex } { backwards-quantifier }
            { \int_use:N \l__regex_internal_a_int }
            { \int_use:N \l__regex_internal_b_int }
          \int_zero:N \l__regex_internal_b_int
        \else:
          \int_sub:Nn \l__regex_internal_b_int \l__regex_internal_a_int
        \fi:
        \exp_args:Noo \__regex_compile_quantifier_laziness:nnNN
          { \int_use:N \l__regex_internal_a_int }
          { \int_use:N \l__regex_internal_b_int }
      }
      {
        \__regex_compile_quantifier_abort:eNN
          {
            \c_left_brace_str
            \int_use:N \l__regex_internal_a_int ,
            \int_use:N \l__regex_internal_b_int
          }
        #1 #2
      }
  }
\cs_new_protected:Npn \__regex_compile_raw_error:N #1
  {
    \msg_error:nne { regex } { bad-escape } {#1}
    \__regex_compile_raw:N #1
  }
\cs_new_protected:Npn \__regex_compile_raw:N #1#2#3
  {
    \__regex_if_in_class:TF
      {
        \__regex_two_if_eq:NNNNTF #2 #3 \__regex_compile_special:N -
          { \__regex_compile_range:Nw #1 }
          {
            \__regex_compile_one:n
              { \__regex_item_equal:n { \int_value:w `#1 } }
            #2 #3
          }
      }
      {
        \__regex_compile_one:n
          { \__regex_item_equal:n { \int_value:w `#1 } }
        #2 #3
      }
  }
\cs_new_protected:Npn \__regex_if_end_range:NNTF #1#2
```

```
  {
    \if_meaning:w \__regex_compile_raw:N #1
    \else:
      \if_meaning:w \__regex_compile_special:N #1
        \if_charcode:w ] #2
          \use_i:nn
        \fi:
      \else:
        \exp_after:wN \exp_after:wN \exp_after:wN \use_iii:nnn
      \fi:
    \fi:
    \use_i:nn
  }
\cs_new_protected:Npn \__regex_compile_range:Nw #1#2#3
  {
    \__regex_if_end_range:NNTF #2 #3
      {
        \if_int_compare:w `#1 > `#3 \exp_stop_f:
          \msg_error:nnee { regex } { range-backwards } {#1} {#3}
        \else:
          \tl_build_put_right:Ne \l__regex_build_tl
            {
              \if_int_compare:w `#1 = `#3 \exp_stop_f:
                \__regex_item_equal:n
              \else:
                \__regex_item_range:nn { \int_value:w `#1 }
              \fi:
              { \int_value:w `#3 }
            }
        \fi:
      }
      {
        \msg_warning:nnee { regex } { range-missing-end }
          {#1} { \c_backslash_str #3 }
        \tl_build_put_right:Ne \l__regex_build_tl
          {
            \__regex_item_equal:n { \int_value:w `#1 \exp_stop_f: }
            \__regex_item_equal:n { \int_value:w `- \exp_stop_f: }
          }
        #2#3
      }
  }
\cs_new_protected:cpe { __regex_compile_.: }
  {
    \exp_not:N \__regex_if_in_class:TF
      { \__regex_compile_raw:N . }
      { \__regex_compile_one:n \exp_not:c { __regex_prop_.: } }
  }
\cs_new_protected:cpn { __regex_prop_.: }
  {
    \if_int_compare:w \l__regex_curr_char_int > - 2 \exp_stop_f:
      \exp_after:wN \__regex_break_true:w
    \fi:
  }
\cs_set_protected:Npn \__regex_tmp:w #1#2
  {
    \cs_new_protected:cpe { __regex_compile_/#1: }
      { \__regex_compile_one:n \exp_not:c { __regex_prop_#1: } }
    \cs_new_protected:cpe { __regex_compile_/#2: }
      {
        \__regex_compile_one:n
          { \__regex_item_reverse:n { \exp_not:c { __regex_prop_#1: } } }
      }
  }
\__regex_tmp:w d D
\__regex_tmp:w h H
\__regex_tmp:w s S
\__regex_tmp:w v V
\__regex_tmp:w w W
\cs_new_protected:cpn { __regex_compile_/N: }
  { \__regex_compile_one:n \__regex_prop_N: }
\cs_new_protected:Npn \__regex_compile_anchor_letter:NNN #1#2#3
  {
    \__regex_if_in_class_or_catcode:TF { \__regex_compile_raw_error:N #1 }
      {
        \tl_build_put_right:Nn \l__regex_build_tl
          { \__regex_assertion:Nn #2 {#3} }
      }
  }
\cs_new_protected:cpn { __regex_compile_/A: }
  { \__regex_compile_anchor_letter:NNN A \c_true_bool \__regex_A_test: }
\cs_new_protected:cpn { __regex_compile_/G: }
  { \__regex_compile_anchor_letter:NNN G \c_true_bool \__regex_G_test: }
\cs_new_protected:cpn { __regex_compile_/Z: }
  { \__regex_compile_anchor_letter:NNN Z \c_true_bool \__regex_Z_test: }
\cs_new_protected:cpn { __regex_compile_/z: }
  { \__regex_compile_anchor_letter:NNN z \c_true_bool \__regex_Z_test: }
\cs_new_protected:cpn { __regex_compile_/b: }
  { \__regex_compile_anchor_letter:NNN b \c_true_bool \__regex_b_test: }
\cs_new_protected:cpn { __regex_compile_/B: }
  { \__regex_compile_anchor_letter:NNN B \c_false_bool \__regex_b_test: }
\cs_set_protected:Npn \__regex_tmp:w #1#2
```

```
  {
    \cs_new_protected:cpn { __regex_compile_#1: }
      {
        \__regex_if_in_class_or_catcode:TF { \__regex_compile_raw:N #1 }
          {
            \tl_build_put_right:Nn \l__regex_build_tl
              { \__regex_assertion:Nn \c_true_bool {#2} }
          }
      }
  }
\exp_args:Ne \__regex_tmp:w { \iow_char:N \^ } { \__regex_A_test: }
\exp_args:Ne \__regex_tmp:w { \iow_char:N \$ } { \__regex_Z_test: }
\cs_new_protected:cpn { __regex_compile_]: }
  {
    \__regex_if_in_class:TF
      {
        \if_int_compare:w \l__regex_mode_int >
          \c__regex_catcode_in_class_mode_int
          \tl_build_put_right:Nn \l__regex_build_tl { \if_false: { \fi: } }
        \fi:
        \tex_advance:D \l__regex_mode_int - 15 \exp_stop_f:
        \tex_divide:D \l__regex_mode_int 13 \exp_stop_f:
        \if_int_odd:w \l__regex_mode_int \else:
          \exp_after:wN \__regex_compile_quantifier:w
        \fi:
      }
      { \__regex_compile_raw:N ] }
  }
\cs_new_protected:cpn { __regex_compile_[: }
  {
    \__regex_if_in_class:TF
      { \__regex_compile_class_posix_test:w }
      {
        \__regex_if_within_catcode:TF
          {
            \exp_after:wN \__regex_compile_class_catcode:w
              \int_use:N \l__regex_catcodes_int \__regex_sep:
          }
          { \__regex_compile_class_normal:w }
      }
  }
\cs_new_protected:Npn \__regex_compile_class_normal:w
  {
    \__regex_compile_class:TFNN
      { \__regex_class:NnnnN \c_true_bool }
      { \__regex_class:NnnnN \c_false_bool }
  }
\cs_new_protected:Npn \__regex_compile_class_catcode:w #1 \__regex_sep:
  {
    \if_int_compare:w \l__regex_mode_int = \c__regex_catcode_mode_int
      \tl_build_put_right:Nn \l__regex_build_tl
        { \__regex_class:NnnnN \c_true_bool { \if_false: } \fi: }
    \fi:
    \int_set_eq:NN \l__regex_catcodes_int \l__regex_default_catcodes_int
    \__regex_compile_class:TFNN
      { \__regex_item_catcode:nT {#1} }
      { \__regex_item_catcode_reverse:nT {#1} }
  }
\cs_new_protected:Npn \__regex_compile_class:TFNN #1#2#3#4
  {
    \l__regex_mode_int = \int_value:w \l__regex_mode_int 3 \exp_stop_f:
    \__regex_two_if_eq:NNNNTF #3 #4 \__regex_compile_special:N ^
      {
        \tl_build_put_right:Nn \l__regex_build_tl { #2 { \if_false: } \fi: }
        \__regex_compile_class:NN
      }
      {
        \tl_build_put_right:Nn \l__regex_build_tl { #1 { \if_false: } \fi: }
        \__regex_compile_class:NN #3 #4
      }
  }
\cs_new_protected:Npn \__regex_compile_class:NN #1#2
  {
    \token_if_eq_charcode:NNTF #2 ]
      { \__regex_compile_raw:N #2 }
      { #1 #2 }
  }
\cs_new_protected:Npn \__regex_compile_class_posix_test:w #1#2
  {
    \token_if_eq_meaning:NNT \__regex_compile_special:N #1
      {
        \str_case:nn { #2 }
          {
            : { \__regex_compile_class_posix:NNNNw }
            = {
                \msg_warning:nne { regex }
                  { posix-unsupported } { = }
              }
            . {
                \msg_warning:nne { regex }
                  { posix-unsupported } { . }
              }
          }
      }
    \__regex_compile_raw:N [ #1 #2
  }
\cs_new_protected:Npn \__regex_compile_class_posix:NNNNw #1#2#3#4#5#6
  {
    \__regex_two_if_eq:NNNNTF #5 #6 \__regex_compile_special:N ^
      {
        \bool_set_false:N \l__regex_internal_bool
        \__kernel_tl_set:Nx \l__regex_internal_a_tl { \if_false: } \fi:
          \__regex_compile_class_posix_loop:w
      }
      {
        \bool_set_true:N \l__regex_internal_bool
        \__kernel_tl_set:Nx \l__regex_internal_a_tl { \if_false: } \fi:
          \__regex_compile_class_posix_loop:w #5 #6
      }
  }
\cs_new:Npn \__regex_compile_class_posix_loop:w #1#2
  {
    \token_if_eq_meaning:NNTF \__regex_compile_raw:N #1
      { #2 \__regex_compile_class_posix_loop:w }
      { \if_false: { \fi: } \__regex_compile_class_posix_end:w #1 #2 }
  }
\cs_new_protected:Npn \__regex_compile_class_posix_end:w #1#2#3#4
  {
    \__regex_two_if_eq:NNNNTF #1 #2 \__regex_compile_special:N :
      { \__regex_two_if_eq:NNNNTF #3 #4 \__regex_compile_special:N ] }
      { \use_ii:nn }
      {
        \cs_if_exist:cTF { __regex_posix_ \l__regex_internal_a_tl : }
          {
            \__regex_compile_one:n
              {
                \bool_if:NTF \l__regex_internal_bool \use:n \__regex_item_reverse:n
                { \exp_not:c { __regex_posix_ \l__regex_internal_a_tl : } }
              }
          }
          {
            \msg_warning:nne { regex } { posix-unknown }
              { \l__regex_internal_a_tl }
            \__regex_compile_abort_tokens:e
              {
                [: \bool_if:NF \l__regex_internal_bool { ^ }
                \l__regex_internal_a_tl :]
              }
          }
      }
      {
        \msg_error:nnee { regex } { posix-missing-close }
          { [: \l__regex_internal_a_tl } { #2 #4 }
        \__regex_compile_abort_tokens:e { [: \l__regex_internal_a_tl }
        #1 #2 #3 #4
      }
  }
\cs_new_protected:Npn \__regex_compile_group_begin:N #1
  {
    \tl_build_put_right:Nn \l__regex_build_tl { #1 { \if_false: } \fi: }
    \__regex_mode_quit_c:
    \group_begin:
      \tl_build_begin:N \l__regex_build_tl
      \int_set_eq:NN \l__regex_default_catcodes_int \l__regex_catcodes_int
      \int_incr:N \l__regex_group_level_int
      \tl_build_put_right:Nn \l__regex_build_tl
        { \__regex_branch:n { \if_false: } \fi: }
  }
\cs_new_protected:Npn \__regex_compile_group_end:
  {
    \if_int_compare:w \l__regex_group_level_int > \c_zero_int
        \tl_build_put_right:Nn \l__regex_build_tl { \if_false: { \fi: } }
        \tl_build_end:N \l__regex_build_tl
        \exp_args:NNNe
      \group_end:
      \tl_build_put_right:Nn \l__regex_build_tl { \l__regex_build_tl }
      \int_set_eq:NN \l__regex_catcodes_int \l__regex_default_catcodes_int
      \exp_after:wN \__regex_compile_quantifier:w
    \else:
      \msg_warning:nn { regex } { extra-rparen }
      \exp_after:wN \__regex_compile_raw:N \exp_after:wN )
    \fi:
  }
\cs_new_protected:cpn { __regex_compile_(: }
  {
    \__regex_if_in_class:TF { \__regex_compile_raw:N ( }
      {
        \if_int_compare:w \l__regex_mode_int =
          \c__regex_catcode_in_class_mode_int
          \msg_error:nn { regex } { c-lparen-in-class }
          \exp_after:wN \__regex_compile_raw:N \exp_after:wN (
        \else:
          \exp_after:wN \__regex_compile_lparen:w
        \fi:
      }
  }
\cs_new_protected:Npn \__regex_compile_lparen:w #1#2#3#4
  {
    \__regex_two_if_eq:NNNNTF #1 #2 \__regex_compile_special:N ?
      {
        \cs_if_exist_use:cF
          { __regex_compile_special_group_\token_to_str:N #4 :w }
          {
            \msg_warning:nne { regex } { special-group-unknown }
              { (? #4 }
            \__regex_compile_group_begin:N \__regex_group:nnnN
              \__regex_compile_raw:N ? #3 #4
          }
      }
      {
        \__regex_compile_group_begin:N \__regex_group:nnnN
          #1 #2 #3 #4
      }
  }
\cs_new_protected:cpn { __regex_compile_|: }
  {
    \__regex_if_in_class:TF { \__regex_compile_raw:N | }
      {
        \tl_build_put_right:Nn \l__regex_build_tl
          { \if_false: { \fi: } \__regex_branch:n { \if_false: } \fi: }
      }
  }
\cs_new_protected:cpn { __regex_compile_): }
  {
    \__regex_if_in_class:TF { \__regex_compile_raw:N ) }
      { \__regex_compile_group_end: }
  }
\cs_new_protected:cpn { __regex_compile_special_group_::w }
  { \__regex_compile_group_begin:N \__regex_group_no_capture:nnnN }
\cs_new_protected:cpn { __regex_compile_special_group_|:w }
  { \__regex_compile_group_begin:N \__regex_group_resetting:nnnN }
\cs_new_protected:Npn \__regex_compile_special_group_i:w #1#2
  {
    \__regex_two_if_eq:NNNNTF #1 #2 \__regex_compile_special:N )
      {
        \cs_set:Npn \__regex_item_equal:n
          { \__regex_item_caseless_equal:n }
        \cs_set:Npn \__regex_item_range:nn
          { \__regex_item_caseless_range:nn }
      }
      {
        \msg_warning:nne { regex } { unknown-option } { (?i #2 }
        \__regex_compile_raw:N (
        \__regex_compile_raw:N ?
        \__regex_compile_raw:N i
        #1 #2
      }
  }
\cs_new_protected:cpn { __regex_compile_special_group_-:w } #1#2#3#4
  {
    \__regex_two_if_eq:NNNNTF #1 #2 \__regex_compile_raw:N i
      { \__regex_two_if_eq:NNNNTF #3 #4 \__regex_compile_special:N ) }
      { \use_ii:nn }
      {
        \cs_set:Npn \__regex_item_equal:n
          { \__regex_item_caseful_equal:n }
        \cs_set:Npn \__regex_item_range:nn
          { \__regex_item_caseful_range:nn }
      }
      {
        \msg_warning:nne { regex } { unknown-option } { (?-#2#4 }
        \__regex_compile_raw:N (
        \__regex_compile_raw:N ?
        \__regex_compile_raw:N -
        #1 #2 #3 #4
      }
  }
\cs_new_protected:cpn { __regex_compile_/c: }
  { \__regex_chk_c_allowed:T { \__regex_compile_c_test:NN } }
\cs_new_protected:Npn \__regex_compile_c_test:NN #1#2
  {
    \token_if_eq_meaning:NNTF #1 \__regex_compile_raw:N
      {
        \int_if_exist:cTF { c__regex_catcode_#2_int }
          {
            \int_set_eq:Nc \l__regex_catcodes_int
              { c__regex_catcode_#2_int }
            \l__regex_mode_int
              = \if_case:w \l__regex_mode_int
                  \c__regex_catcode_mode_int
                \else:
                  \c__regex_catcode_in_class_mode_int
                \fi:
            \token_if_eq_charcode:NNT C #2 { \__regex_compile_c_C:NN }
          }
      }
      { \cs_if_exist_use:cF { __regex_compile_c_#2:w } }
          {
            \msg_error:nne { regex } { c-missing-category } {#2}
            #1 #2
          }
  }
\cs_new_protected:Npn \__regex_compile_c_C:NN #1#2
  {
    \token_if_eq_meaning:NNTF #1 \__regex_compile_special:N
      {
        \token_if_eq_charcode:NNTF #2 .
          { \use_none:n }
          { \token_if_eq_charcode:NNF #2 ( } % )
      }
      { \use:n }
    { \msg_error:nnn { regex } { c-C-invalid } {#2} }
    #1 #2
  }
\cs_new_protected:cpn { __regex_compile_c_[:w } #1#2
  {
    \l__regex_mode_int
      = \if_case:w \l__regex_mode_int
          \c__regex_catcode_mode_int
        \else:
          \c__regex_catcode_in_class_mode_int
        \fi:
    \int_zero:N \l__regex_catcodes_int
    \__regex_two_if_eq:NNNNTF #1 #2 \__regex_compile_special:N ^
      {
        \bool_set_false:N \l__regex_catcodes_bool
        \__regex_compile_c_lbrack_loop:NN
      }
      {
        \bool_set_true:N \l__regex_catcodes_bool
        \__regex_compile_c_lbrack_loop:NN
        #1 #2
      }
  }
\cs_new_protected:Npn \__regex_compile_c_lbrack_loop:NN #1#2
  {
    \token_if_eq_meaning:NNTF #1 \__regex_compile_raw:N
      {
        \int_if_exist:cTF { c__regex_catcode_#2_int }
          {
            \exp_args:Nc \__regex_compile_c_lbrack_add:N
              { c__regex_catcode_#2_int }
            \__regex_compile_c_lbrack_loop:NN
          }
      }
      {
        \token_if_eq_charcode:NNTF #2 ]
          { \__regex_compile_c_lbrack_end: }
      }
          {
            \msg_error:nne { regex } { c-missing-rbrack } {#2}
            \__regex_compile_c_lbrack_end:
            #1 #2
          }
  }
\cs_new_protected:Npn \__regex_compile_c_lbrack_add:N #1
  {
    \if_int_odd:w \__regex_int_eval:w \l__regex_catcodes_int / #1 \scan_stop:
    \else:
      \int_add:Nn \l__regex_catcodes_int {#1}
    \fi:
  }
\cs_new_protected:Npn \__regex_compile_c_lbrack_end:
  {
    \if_meaning:w \c_false_bool \l__regex_catcodes_bool
      \int_set:Nn \l__regex_catcodes_int
        { \c__regex_all_catcodes_int - \l__regex_catcodes_int }
    \fi:
  }
\cs_new_protected:cpn { __regex_compile_c_ \c_left_brace_str :w }
  {
    \__regex_compile:w
      \__regex_disable_submatches:
      \l__regex_mode_int
        = \if_case:w \l__regex_mode_int
            \c__regex_cs_mode_int
          \else:
            \c__regex_cs_in_class_mode_int
          \fi:
  }
\cs_new_protected:cpn { __regex_compile_ \c_left_brace_str : }
  {
    \__regex_if_in_cs:TF
      { \msg_error:nnn { regex } { cu-lbrace } { c } }
      { \exp_after:wN \__regex_compile_raw:N \c_left_brace_str }
  }
\flag_new:N \l__regex_cs_flag
\cs_new_protected:cpn { __regex_compile_ \c_right_brace_str : }
```

```
  {
    \__regex_if_in_cs:TF
      { \__regex_compile_end_cs: }
      { \exp_after:wN \__regex_compile_raw:N \c_right_brace_str }
  }
\cs_new_protected:Npn \__regex_compile_end_cs:
  {
    \__regex_compile_end:
    \flag_clear:N \l__regex_cs_flag
    \__kernel_tl_set:Nx \l__regex_internal_a_tl
      {
        \exp_after:wN \__regex_compile_cs_aux:Nn \l__regex_internal_regex
        \q__regex_nil \q__regex_nil \q__regex_recursion_stop
      }
    \exp_args:Ne \__regex_compile_one:n
      {
        \flag_if_raised:NTF \l__regex_cs_flag
          { \__regex_item_cs:n { \exp_not:o \l__regex_internal_regex } }
          {
            \__regex_item_exact_cs:n
              { \tl_tail:N \l__regex_internal_a_tl }
          }
      }
  }
\cs_new:Npn \__regex_compile_cs_aux:Nn #1#2
  {
    \cs_if_eq:NNTF #1 \__regex_branch:n
      {
        \scan_stop:
        \__regex_compile_cs_aux:NNnnnN #2
        \q__regex_nil \q__regex_nil \q__regex_nil
        \q__regex_nil \q__regex_nil \q__regex_nil \q__regex_recursion_stop
        \__regex_compile_cs_aux:Nn
      }
      {
        \__regex_quark_if_nil:NF #1 { \flag_ensure_raised:N \l__regex_cs_flag }
        \__regex_use_none_delimit_by_q_recursion_stop:w
      }
  }
\cs_new:Npn \__regex_compile_cs_aux:NNnnnN #1#2#3#4#5#6
  {
    \bool_lazy_all:nTF
      {
        { \cs_if_eq_p:NN #1 \__regex_class:NnnnN }
        {#2}
        { \tl_if_head_eq_meaning_p:nN {#3} \__regex_item_caseful_equal:n }
        { \int_compare_p:nNn { \tl_count:n {#3} } = { 2 } }
        { \int_compare_p:nNn {#5} = \c_zero_int }
      }
      {
        \prg_replicate:nn {#4}
          { \char_generate:nn { \use_ii:nn #3 } {12} }
        \__regex_compile_cs_aux:NNnnnN
      }
      {
        \__regex_quark_if_nil:NF #1
          {
            \flag_ensure_raised:N \l__regex_cs_flag
            \__regex_use_i_delimit_by_q_recursion_stop:nw
          }
        \__regex_use_none_delimit_by_q_recursion_stop:w
      }
  }
\cs_new_protected:cpn { __regex_compile_/u: } #1#2
  {
    \__regex_if_in_class_or_catcode:TF
      { \__regex_compile_raw_error:N u #1 #2 }
      {
        \__regex_two_if_eq:NNNNTF #1 #2 \__regex_compile_raw:N r
          { \__regex_compile_u_brace:NNN \__regex_compile_ur_end: }
          { \__regex_compile_u_brace:NNN \__regex_compile_u_end: #1 #2 }
      }
  }
\cs_new:Npn \__regex_compile_u_brace:NNN #1#2#3
  {
    \__regex_two_if_eq:NNNNTF #2 #3 \__regex_compile_special:N \c_left_brace_str
      {
        \tl_set:Nn \l__regex_internal_b_tl {#1}
        \__kernel_tl_set:Nx \l__regex_internal_a_tl { \if_false: } \fi:
        \__regex_compile_u_loop:NN
      }
      {
        \msg_error:nn { regex } { u-missing-lbrace }
        \token_if_eq_meaning:NNTF #1 \__regex_compile_ur_end:
          { \__regex_compile_raw:N u \__regex_compile_raw:N r }
          { \__regex_compile_raw:N u }
        #2 #3
      }
  }
\cs_new:Npn \__regex_compile_u_loop:NN #1#2
  {
    \token_if_eq_meaning:NNTF #1 \__regex_compile_raw:N
```

```
      { #2 \__regex_compile_u_loop:NN }
      {
        \token_if_eq_meaning:NNTF #1 \__regex_compile_special:N
          {
            \exp_after:wN \token_if_eq_charcode:NNTF \c_right_brace_str #2
              { \if_false: { \fi: } \l__regex_internal_b_tl }
              {
                \if_charcode:w \c_left_brace_str #2
                  \msg_expandable_error:nnn { regex } { cu-lbrace } { u }
                \else:
                  #2
                \fi:
                \__regex_compile_u_loop:NN
              }
          }
          {
            \if_false: { \fi: }
            \msg_error:nne { regex } { u-missing-rbrace } {#2}
            \l__regex_internal_b_tl
            #1 #2
          }
      }
  }
\cs_new_protected:Npn \__regex_compile_ur_end:
  {
    \group_begin:
      \cs_set:Npn \__regex_group:nnnN { \__regex_group_no_capture:nnnN }
      \cs_set:Npn \__regex_group_resetting:nnnN { \__regex_group_no_capture:nnnN }
      \exp_args:NNe
    \group_end:
    \__regex_compile_ur:n { \use:c { \l__regex_internal_a_tl } }
  }
\cs_new_protected:Npn \__regex_compile_ur:n #1
  {
    \tl_if_empty:oTF { \__regex_compile_ur_aux:w #1 {} ? ? \q__regex_nil }
      { \__regex_compile_if_quantifier:TFw }
      { \use_i:nn }
          {
            \tl_build_put_right:Nn \l__regex_build_tl
              { \__regex_group_no_capture:nnnN { \if_false: } \fi: #1 }
            \__regex_compile_quantifier:w
          }
          { \tl_build_put_right:Nn \l__regex_build_tl { \use_ii:nn #1 } }
  }
\cs_new:Npn \__regex_compile_ur_aux:w \__regex_branch:n #1#2#3 \q__regex_nil {#2}
\cs_new_protected:Npn \__regex_compile_u_end:
  {
    \__regex_compile_if_quantifier:TFw
      {
        \tl_build_put_right:Nn \l__regex_build_tl
          {
            \__regex_group_no_capture:nnnN { \if_false: } \fi:
            \__regex_branch:n { \if_false: } \fi:
          }
        \__regex_compile_u_payload:
        \tl_build_put_right:Nn \l__regex_build_tl { \if_false: { \fi: } }
        \__regex_compile_quantifier:w
      }
      { \__regex_compile_u_payload: }
  }
\cs_new_protected:Npn \__regex_compile_u_payload:
  {
    \tl_set:Nv \l__regex_internal_a_tl { \l__regex_internal_a_tl }
    \if_int_compare:w \l__regex_mode_int = \c__regex_outer_mode_int
      \__regex_compile_u_not_cs:
    \else:
      \__regex_compile_u_in_cs:
    \fi:
  }
\cs_new_protected:Npn \__regex_compile_u_in_cs:
  {
    \__kernel_tl_gset:Nx \g__regex_internal_tl
      {
        \exp_args:No \__kernel_str_to_other_fast:n
          { \l__regex_internal_a_tl }
      }
    \tl_build_put_right:Ne \l__regex_build_tl
      {
        \tl_map_function:NN \g__regex_internal_tl
          \__regex_compile_u_in_cs_aux:n
      }
  }
\cs_new:Npn \__regex_compile_u_in_cs_aux:n #1
  {
    \__regex_class:NnnnN \c_true_bool
      { \__regex_item_caseful_equal:n { \int_value:w `#1 } }
      { 1 } { 0 } \c_false_bool
  }
\cs_new_protected:Npn \__regex_compile_u_not_cs:
  {
    \tl_analysis_map_inline:Nn \l__regex_internal_a_tl
      {
        \tl_build_put_right:Ne \l__regex_build_tl
          {
            \__regex_class:NnnnN \c_true_bool
              {
                \if_int_compare:w "##3 = \c_zero_int
                  \__regex_item_exact_cs:n
                    { \exp_after:wN \cs_to_str:N ##1 }
                \else:
                  \__regex_item_exact:nn { \int_value:w "##3 } { ##2 }
                \fi:
              }
              { 1 } { 0 } \c_false_bool
          }
      }
  }
\cs_new_protected:cpn { __regex_compile_/K: }
  {
    \int_compare:nNnTF \l__regex_mode_int = \c__regex_outer_mode_int
      { \tl_build_put_right:Nn \l__regex_build_tl { \__regex_command_K: } }
      { \__regex_compile_raw_error:N K }
  }
\cs_new:Npn \__regex_clean_bool:n #1
  {
    \tl_if_single:nTF {#1}
      { \bool_if:NTF #1 \c_true_bool \c_false_bool }
      { \c_true_bool }
  }
\cs_new:Npn \__regex_clean_int:n #1
  {
    \tl_if_head_eq_meaning:nNTF {#1} -
      { - \exp_args:No \__regex_clean_int:n { \use_none:n #1 } }
      { \int_eval:n { 0 \str_map_function:nN {#1} \__regex_clean_int_aux:N } }
  }
\cs_new:Npn \__regex_clean_int_aux:N #1
  {
    \if_int_compare:w \c_one_int < 1 #1 ~
      #1
    \else:
      \str_map_break:n
    \fi:
  }
\cs_new:Npn \__regex_clean_regex:n #1
  {
    \__regex_clean_regex_loop:w #1
    \__regex_branch:n { \q_recursion_tail } \q_recursion_stop
  }
```

```
\cs_new:Npn \__regex_clean_regex_loop:w #1 \__regex_branch:n #2
  {
    \quark_if_recursion_tail_stop:n {#2}
    \__regex_branch:n { \__regex_clean_branch:n {#2} }
    \__regex_clean_regex_loop:w
  }
\cs_new:Npn \__regex_clean_branch:n #1
  {
    \__regex_clean_branch_loop:n #1
    ? ? ? ? ? ? \prg_break_point:
  }
\cs_new:Npn \__regex_clean_branch_loop:n #1
  {
    \tl_if_single:nF {#1} \prg_break:
    \token_case_meaning:NnF #1
      {
        \__regex_command_K: { #1 \__regex_clean_branch_loop:n }
        \__regex_assertion:Nn { #1 \__regex_clean_assertion:Nn }
        \__regex_class:NnnnN { #1 \__regex_clean_class:NnnnN }
        \__regex_group:nnnN { #1 \__regex_clean_group:nnnN }
        \__regex_group_no_capture:nnnN { #1 \__regex_clean_group:nnnN }
        \__regex_group_resetting:nnnN { #1 \__regex_clean_group:nnnN }
      }
      \prg_break:
  }
\cs_new:Npn \__regex_clean_assertion:Nn #1#2
  {
    \__regex_clean_bool:n {#1}
    \tl_if_single:nF {#2} { { \__regex_A_test: } \prg_break: }
    \token_case_meaning:NnTF #2
      {
        \__regex_A_test: { }
        \__regex_G_test: { }
        \__regex_Z_test: { }
        \__regex_b_test: { }
      }
      { {#2} }
      { { \__regex_A_test: } \prg_break: }
    \__regex_clean_branch_loop:n
  }
\cs_new:Npn \__regex_clean_class:NnnnN #1#2#3#4#5
  {
    \__regex_clean_bool:n {#1}
    { \__regex_clean_class:n {#2} }
    { \int_max:nn \c_zero_int { \__regex_clean_int:n {#3} } }
    { \int_max:nn { -\c_one_int } { \__regex_clean_int:n {#4} } }
    \__regex_clean_bool:n {#5}
    \__regex_clean_branch_loop:n
  }
\cs_new:Npn \__regex_clean_group:nnnN #1#2#3#4
  {
    { \__regex_clean_regex:n {#1} }
    { \int_max:nn \c_zero_int { \__regex_clean_int:n {#2} } }
    { \int_max:nn { -\c_one_int } { \__regex_clean_int:n {#3} } }
    \__regex_clean_bool:n {#4}
    \__regex_clean_branch_loop:n
  }
\cs_new:Npn \__regex_clean_class:n #1
  { \__regex_clean_class_loop:nnn #1 ????? \prg_break_point: }
\cs_new:Npn \__regex_clean_class_loop:nnn #1#2#3
  {
    \tl_if_single:nF {#1} \prg_break:
    \token_case_meaning:NnTF #1
      {
        \__regex_item_cs:n { #1 { \__regex_clean_regex:n {#2} } }
        \__regex_item_exact_cs:n { #1 { \__regex_clean_exact_cs:n {#2} } }
        \__regex_item_caseful_equal:n { #1 { \__regex_clean_int:n {#2} } }
        \__regex_item_caseless_equal:n { #1 { \__regex_clean_int:n {#2} } }
        \__regex_item_reverse:n { #1 { \__regex_clean_class:n {#2} } }
      }
      { \__regex_clean_class_loop:nnn {#3} }
      {
        \token_case_meaning:NnTF #1
          {
            \__regex_item_caseful_range:nn { }
            \__regex_item_caseless_range:nn { }
            \__regex_item_exact:nn { }
          }
          {
            #1 { \__regex_clean_int:n {#2} } { \__regex_clean_int:n {#3} }
            \__regex_clean_class_loop:nnn
          }
          {
            \token_case_meaning:NnTF #1
              {
                \__regex_item_catcode:nT { }
                \__regex_item_catcode_reverse:nT { }
              }
              {
                #1 { \__regex_clean_int:n {#2} } { \__regex_clean_class:n {#3} }
                \__regex_clean_class_loop:nnn
              }
              {
                \exp_args:Ne \str_case:nnTF
                  {
                    \exp_args:Ne \str_range:nnn
                      { \cs_to_str:N #1 } \c_one_int { 13 }
                  }
                  {
                    { __regex_prop_ } { }
                    { __regex_posix } { }
                  }
                  {
                    #1
                    \__regex_clean_class_loop:nnn {#2} {#3}
                  }
                  \prg_break:
              }
          }
      }
  }
\cs_new:Npn \__regex_clean_exact_cs:n #1
  {
    \exp_last_unbraced:Nf \use_none:n
      {
        \__regex_clean_exact_cs:w #1
        \scan_stop: \q_recursion_tail \scan_stop:
        \q_recursion_stop
      }
  }
\cs_new:Npn \__regex_clean_exact_cs:w #1 \scan_stop:
  {
    \quark_if_recursion_tail_stop:n {#1}
    \scan_stop: \tl_to_str:n {#1}
    \__regex_clean_exact_cs:w
  }
\cs_new_protected:Npn \__regex_show:N #1
  {
    \group_begin:
      \tl_build_begin:N \l__regex_build_tl
      \cs_set_protected:Npn \__regex_branch:n
        {
          \seq_pop_right:NN \l__regex_show_prefix_seq
            \l__regex_internal_a_tl
          \__regex_show_one:n { +-branch }
          \seq_put_right:No \l__regex_show_prefix_seq
            \l__regex_internal_a_tl
          \use:n
        }
      \cs_set_protected:Npn \__regex_group:nnnN
        { \__regex_show_group_aux:nnnnN { } }
      \cs_set_protected:Npn \__regex_group_no_capture:nnnN
        { \__regex_show_group_aux:nnnnN { ~(no~capture) } }
      \cs_set_protected:Npn \__regex_group_resetting:nnnN
        { \__regex_show_group_aux:nnnnN { ~(resetting) } }
      \cs_set_eq:NN \__regex_class:NnnnN \__regex_show_class:NnnnN
      \cs_set_protected:Npn \__regex_command_K:
        { \__regex_show_one:n { reset~match~start~(\iow_char:N\\K) } }
      \cs_set_protected:Npn \__regex_assertion:Nn ##1##2
        {
          \__regex_show_one:n
            { \bool_if:NF ##1 { negative~ } assertion:~##2 }
        }
      \cs_set:Npn \__regex_b_test: { word~boundary }
      \cs_set:Npn \__regex_Z_test: { anchor~at~end~(\iow_char:N\\Z) }
      \cs_set:Npn \__regex_A_test: { anchor~at~start~(\iow_char:N\\A) }
      \cs_set:Npn \__regex_G_test: { anchor~at~start~of~match~(\iow_char:N\\G) }
      \cs_set_protected:Npn \__regex_item_caseful_equal:n ##1
        { \__regex_show_one:n { char~code~\__regex_show_char:n{##1} } }
      \cs_set_protected:Npn \__regex_item_caseful_range:nn ##1##2
        {
          \__regex_show_one:n
            { range~[\__regex_show_char:n{##1}, \__regex_show_char:n{##2}] }
        }
      \cs_set_protected:Npn \__regex_item_caseless_equal:n ##1
        { \__regex_show_one:n { char~code~\__regex_show_char:n{##1}~(caseless) } }
      \cs_set_protected:Npn \__regex_item_caseless_range:nn ##1##2
        {
          \__regex_show_one:n
            { Range~[\__regex_show_char:n{##1}, \__regex_show_char:n{##2}]~(caseless) }
        }
      \cs_set_protected:Npn \__regex_item_catcode:nT
        { \__regex_show_item_catcode:NnT \c_true_bool }
      \cs_set_protected:Npn \__regex_item_catcode_reverse:nT
        { \__regex_show_item_catcode:NnT \c_false_bool }
      \cs_set_protected:Npn \__regex_item_reverse:n
        { \__regex_show_scope:nn { Reversed~match } }
      \cs_set_protected:Npn \__regex_item_exact:nn ##1##2
        { \__regex_show_one:n { char~\__regex_show_char:n{##2},~catcode~##1 } }
      \cs_set_eq:NN \__regex_item_exact_cs:n \__regex_show_item_exact_cs:n
      \cs_set_protected:Npn \__regex_item_cs:n
        { \__regex_show_scope:nn { control~sequence } }
      \cs_set:cpn { __regex_prop_.: } { \__regex_show_one:n { any~token } }
      \seq_clear:N \l__regex_show_prefix_seq
```

```
      \__regex_show_push:n { ~ }
      \cs_if_exist_use:N #1
      \tl_build_end:N \l__regex_build_tl
      \exp_args:NNNo
    \group_end:
    \tl_set:Nn \l__regex_internal_a_tl { \l__regex_build_tl }
  }
\cs_new:Npn \__regex_show_char:n #1
  {
    \int_eval:n {#1}
    \int_compare:nT { 32 <= #1 <= 126 }
      { ~ ( \char_generate:nn {#1} {12} ) }
  }
\cs_new_protected:Npn \__regex_show_one:n #1
  {
    \int_incr:N \l__regex_show_lines_int
    \tl_build_put_right:Ne \l__regex_build_tl
      {
        \exp_not:N \iow_newline:
        \seq_map_function:NN \l__regex_show_prefix_seq \use:n
        #1
      }
  }
\cs_new_protected:Npn \__regex_show_push:n #1
  { \seq_put_right:Ne \l__regex_show_prefix_seq { #1 ~ } }
\cs_new_protected:Npn \__regex_show_pop:
  { \seq_pop_right:NN \l__regex_show_prefix_seq \l__regex_internal_a_tl }
\cs_new_protected:Npn \__regex_show_scope:nn #1#2
  {
    \__regex_show_one:n {#1}
    \__regex_show_push:n { ~ }
    #2
    \__regex_show_pop:
  }
\cs_new_protected:Npn \__regex_show_group_aux:nnnnN #1#2#3#4#5
  {
    \__regex_show_one:n { ,-group~begin #1 }
    \__regex_show_push:n { | }
    \use_ii:nn #2
    \__regex_show_pop:
    \__regex_show_one:n
      { `-group~end \__regex_msg_repeated:nnN {#3} {#4} #5 }
  }
\cs_new:Npn \__regex_show_class:NnnnN #1#2#3#4#5
  {
    \group_begin:
```

```
   \tl_build_begin:N \l__regex_build_tl
   \int_zero:N \l__regex_show_lines_int
   \__regex_show_push:n {~}
   #2
  \int_compare:nTF { \l__regex_show_lines_int = \c_zero_int }
   {
    \group_end:
    \__regex_show_one:n { \bool_if:NTF #1 { Fail } { Pass } }
   }
   {
    \bool_if:nTF
     { #1 && \int_compare_p:n { \l__regex_show_lines_int = \c_one_int } }
     {
      \group_end:
      #2
      \tl_build_put_right:Nn \l__regex_build_tl
       { \__regex_msg_repeated:nnN {#3} {#4} #5 }
     }
     {
       \tl_build_end:N \l__regex_build_tl
       \exp_args:NNNo
      \group_end:
      \tl_set:Nn \l__regex_internal_a_tl \l__regex_build_tl
      \__regex_show_one:n
       {
        \bool_if:NTF #1 { Match } { Don't~match }
        \__regex_msg_repeated:nnN {#3} {#4} #5
       }
      \tl_build_put_right:Ne \l__regex_build_tl
       { \exp_not:o \l__regex_internal_a_tl }
     }
   }
 }
\cs_new_protected:Npn \__regex_show_item_catcode:NnT #1#2
 {
  \seq_set_split:Nnn \l__regex_internal_seq { } { CBEMTPUDSLOA }
  \seq_set_filter:NNn \l__regex_internal_seq \l__regex_internal_seq
   { \int_if_odd_p:n { #2 / \int_use:c { c__regex_catcode_##1_int } } }
  \__regex_show_scope:nn
   {
    categories~
    \seq_map_function:NN \l__regex_internal_seq \use:n
    , ~
    \bool_if:NF #1 { negative~ } class
   }
 }
```

```
\cs_new_protected:Npn \__regex_show_item_exact_cs:n #1
  {
    \seq_set_split:Nnn \l__regex_internal_seq { \scan_stop: } {#1}
    \seq_set_map_e:NNn \l__regex_internal_seq
      \l__regex_internal_seq { \iow_char:N\\##1 }
    \__regex_show_one:n
      { control~sequence~ \seq_use:Nn \l__regex_internal_seq { ~or~ } }
  }
\int_new:N  \l__regex_min_state_int
\int_set:Nn \l__regex_min_state_int { 1 }
\int_new:N  \l__regex_max_state_int
\int_new:N  \l__regex_left_state_int
\int_new:N  \l__regex_right_state_int
\seq_new:N  \l__regex_left_state_seq
\seq_new:N  \l__regex_right_state_seq
\int_new:N  \l__regex_capturing_group_int
\cs_new_protected:Npn \__regex_build:n
  { \__regex_build_aux:Nn \c_true_bool }
\cs_new_protected:Npn \__regex_build:N
  { \__regex_build_aux:NN \c_true_bool }
\cs_new_protected:Npn \__regex_build_aux:Nn #1#2
  {
    \__regex_compile:n {#2}
    \__regex_build_aux:NN #1 \l__regex_internal_regex
  }
\cs_new_protected:Npn \__regex_build_aux:NN #1#2
  {
    \__regex_standard_escapechar:
    \int_zero:N \l__regex_capturing_group_int
    \int_set_eq:NN \l__regex_max_state_int \l__regex_min_state_int
    \__regex_build_new_state:
    \__regex_build_new_state:
    \__regex_toks_put_right:Nn \l__regex_left_state_int
      { \__regex_action_start_wildcard:N #1 }
    \__regex_group:nnnN {#2} { 1 } { 0 } \c_false_bool
    \__regex_toks_put_right:Nn \l__regex_right_state_int
      { \__regex_action_success: }
  }
\int_new:N \g__regex_case_int
\int_new:N \l__regex_case_max_group_int
\cs_new_protected:Npn \__regex_case_build:n #1
  {
    \__regex_case_build_aux:Nn \c_true_bool {#1}
    \int_gzero:N \g__regex_case_int
  }
\cs_generate_variant:Nn \__regex_case_build:n { e }
```

```
\cs_new_protected:Npn \__regex_case_build_aux:Nn #1#2
  {
    \__regex_standard_escapechar:
    \int_set_eq:NN \l__regex_max_state_int \l__regex_min_state_int
    \__regex_build_new_state:
    \__regex_build_new_state:
    \__regex_toks_put_right:Nn \l__regex_left_state_int
      { \__regex_action_start_wildcard:N #1 }
    %
    \__regex_build_new_state:
    \__regex_toks_put_left:Ne \l__regex_left_state_int
      { \__regex_action_submatch:nN \c_zero_int < }
    \__regex_push_lr_states:
    \int_zero:N \l__regex_case_max_group_int
    \int_gzero:N \g__regex_case_int
    \tl_map_inline:nn {#2}
      {
        \int_gincr:N \g__regex_case_int
        \__regex_case_build_loop:n {##1}
      }
    \int_set_eq:NN \l__regex_capturing_group_int \l__regex_case_max_group_int
    \__regex_pop_lr_states:
  }
\cs_new_protected:Npn \__regex_case_build_loop:n #1
  {
    \int_set_eq:NN \l__regex_capturing_group_int \c_one_int
    \__regex_compile_use:n {#1}
    \int_set:Nn \l__regex_case_max_group_int
      { \int_max:nn \l__regex_case_max_group_int \l__regex_capturing_group_int }
    \seq_pop:NN \l__regex_right_state_seq \l__regex_internal_a_tl
    \int_set:Nn \l__regex_right_state_int \l__regex_internal_a_tl
    \__regex_toks_put_left:Ne \l__regex_right_state_int
      {
        \__regex_action_submatch:nN \c_zero_int >
        \int_gset:Nn \g__regex_case_int
          { \int_use:N \g__regex_case_int }
        \__regex_action_success:
      }
    \__regex_toks_clear:N \l__regex_max_state_int
    \seq_push:No \l__regex_right_state_seq
      { \int_use:N \l__regex_max_state_int }
    \int_incr:N \l__regex_max_state_int
  }
\cs_new_protected:Npn \__regex_build_for_cs:n #1
  {
    \int_set_eq:NN \l__regex_min_state_int \l__regex_max_state_int
    \__regex_build_new_state:
    \__regex_build_new_state:
    \__regex_push_lr_states:
    #1
    \__regex_pop_lr_states:
    \__regex_toks_put_right:Nn \l__regex_right_state_int
      {
        \if_int_compare:w -2 = \l__regex_curr_char_int
          \exp_after:wN \__regex_action_success:
        \fi:
      }
  }
\cs_new_protected:Npn \__regex_push_lr_states:
  {
    \seq_push:No \l__regex_left_state_seq
      { \int_use:N \l__regex_left_state_int }
    \seq_push:No \l__regex_right_state_seq
      { \int_use:N \l__regex_right_state_int }
  }
\cs_new_protected:Npn \__regex_pop_lr_states:
  {
    \seq_pop:NN \l__regex_left_state_seq  \l__regex_internal_a_tl
    \int_set:Nn \l__regex_left_state_int  \l__regex_internal_a_tl
    \seq_pop:NN \l__regex_right_state_seq \l__regex_internal_a_tl
    \int_set:Nn \l__regex_right_state_int \l__regex_internal_a_tl
  }
\cs_new_protected:Npn \__regex_build_transition_left:NNN #1#2#3
  { \__regex_toks_put_left:Ne  #2 { #1 { \tex_the:D \__regex_int_eval:w #3 - #2 } } }
\cs_new_protected:Npn \__regex_build_transition_right:nNn #1#2#3
  { \__regex_toks_put_right:Ne #2 { #1 { \tex_the:D \__regex_int_eval:w #3 - #2 } } }
\cs_new_protected:Npn \__regex_build_new_state:
  {
    \__regex_toks_clear:N \l__regex_max_state_int
    \int_set_eq:NN \l__regex_left_state_int \l__regex_right_state_int
    \int_set_eq:NN \l__regex_right_state_int \l__regex_max_state_int
    \int_incr:N \l__regex_max_state_int
  }
\cs_new_protected:Npn \__regex_build_transitions_laziness:NNNNN #1#2#3#4#5
  {
    \__regex_build_new_state:
    \__regex_toks_put_right:Ne \l__regex_left_state_int
      {
        \if_meaning:w \c_true_bool #1
          #2 { \tex_the:D \__regex_int_eval:w #3 - \l__regex_left_state_int }
          #4 { \tex_the:D \__regex_int_eval:w #5 - \l__regex_left_state_int }
        \else:
          #4 { \tex_the:D \__regex_int_eval:w #5 - \l__regex_left_state_int }
          #2 { \tex_the:D \__regex_int_eval:w #3 - \l__regex_left_state_int }
        \fi:
      }
  }
\cs_new_protected:Npn \__regex_class:NnnnN #1#2#3#4#5
  {
    \cs_set:Npe \__regex_tests_action_cost:n ##1
      {
        \exp_not:n { \exp_not:n {#2} }
        \bool_if:NTF #1
          { \__regex_break_point:TF { \__regex_action_cost:n {##1} } { } }
          { \__regex_break_point:TF { } { \__regex_action_cost:n {##1} } }
      }
    \if_case:w - #4 \exp_stop_f:
           \__regex_class_repeat:n   {#3}
    \or:   \__regex_class_repeat:nN  {#3}      #5
    \else: \__regex_class_repeat:nnN {#3} {#4} #5
    \fi:
  }
\cs_new:Npn \__regex_tests_action_cost:n { \__regex_action_cost:n }
\cs_new_protected:Npn \__regex_class_repeat:n #1
  {
    \prg_replicate:nn {#1}
      {
        \__regex_build_new_state:
        \__regex_build_transition_right:nNn \__regex_tests_action_cost:n
          \l__regex_left_state_int \l__regex_right_state_int
      }
  }
\cs_new_protected:Npn \__regex_class_repeat:nN #1#2
  {
    \if_int_compare:w #1 = \c_zero_int
      \__regex_build_transitions_laziness:NNNNN #2
        \__regex_action_free:n       \l__regex_right_state_int
        \__regex_tests_action_cost:n \l__regex_left_state_int
    \else:
      \__regex_class_repeat:n {#1}
      \int_set_eq:NN \l__regex_internal_a_int \l__regex_left_state_int
      \__regex_build_transitions_laziness:NNNNN #2
        \__regex_action_free:n \l__regex_right_state_int
        \__regex_action_free:n \l__regex_internal_a_int
    \fi:
  }
\cs_new_protected:Npn \__regex_class_repeat:nnN #1#2#3
  {
    \__regex_class_repeat:n {#1}
    \int_set:Nn \l__regex_internal_a_int
      { \l__regex_max_state_int + #2 - \c_one_int }
    \prg_replicate:nn { #2 }
      {
        \__regex_build_transitions_laziness:NNNNN #3
          \__regex_action_free:n       \l__regex_internal_a_int
          \__regex_tests_action_cost:n \l__regex_right_state_int
      }
  }
\cs_new_protected:Npn \__regex_group_aux:nnnnN #1#2#3#4#5
  {
      \if_int_compare:w #3 = \c_zero_int
        \__regex_build_new_state:
        \__regex_build_transition_right:nNn \__regex_action_free_group:n
          \l__regex_left_state_int \l__regex_right_state_int
      \fi:
      \__regex_build_new_state:
      \__regex_push_lr_states:
      #2
      \__regex_pop_lr_states:
      \if_case:w - #4 \exp_stop_f:
             \__regex_group_repeat:nn   {#1} {#3}
      \or:   \__regex_group_repeat:nnN  {#1} {#3}      #5
      \else: \__regex_group_repeat:nnnN {#1} {#3} {#4} #5
      \fi:
  }
\cs_new_protected:Npn \__regex_group:nnnN #1
  {
    \exp_args:No \__regex_group_aux:nnnnN
      { \int_use:N \l__regex_capturing_group_int }
      {
        \int_incr:N \l__regex_capturing_group_int
        #1
      }
  }
\cs_new_protected:Npn \__regex_group_no_capture:nnnN
  { \__regex_group_aux:nnnnN { -1 } }
\cs_new_protected:Npn \__regex_group_resetting:nnnN #1
  {
    \__regex_group_aux:nnnnN { -1 }
      {
        \exp_args:Noo \__regex_group_resetting_loop:nnNn
          { \int_use:N \l__regex_capturing_group_int }
          { \int_use:N \l__regex_capturing_group_int }
          #1
          { ?? \prg_break:n } { }
        \prg_break_point:
      }
  }
\cs_new_protected:Npn \__regex_group_resetting_loop:nnNn #1#2#3#4
  {
    \use_none:nn #3 { \int_set:Nn \l__regex_capturing_group_int {#1} }
    \int_set:Nn \l__regex_capturing_group_int {#2}
    #3 {#4}
    \exp_args:Ne \__regex_group_resetting_loop:nnNn
      { \int_max:nn {#1} \l__regex_capturing_group_int }
      {#2}
  }
\cs_new_protected:Npn \__regex_branch:n #1
  {
    \__regex_build_new_state:
    \seq_get:NN \l__regex_left_state_seq \l__regex_internal_a_tl
    \int_set:Nn \l__regex_left_state_int \l__regex_internal_a_tl
    \__regex_build_transition_right:nNn \__regex_action_free:n
      \l__regex_left_state_int \l__regex_right_state_int
    #1
    \seq_get:NN \l__regex_right_state_seq \l__regex_internal_a_tl
    \__regex_build_transition_right:nNn \__regex_action_free:n
      \l__regex_right_state_int \l__regex_internal_a_tl
  }
\cs_new_protected:Npn \__regex_group_repeat:nn #1#2
  {
    \if_int_compare:w #2 = \c_zero_int
      \int_set:Nn \l__regex_max_state_int
        { \l__regex_left_state_int - \c_one_int }
      \__regex_build_new_state:
    \else:
      \__regex_group_repeat_aux:n {#2}
      \__regex_group_submatches:nNN {#1}
        \l__regex_internal_a_int \l__regex_right_state_int
      \__regex_build_new_state:
    \fi:
  }
\cs_new_protected:Npn \__regex_group_submatches:nNN #1#2#3
  {
    \if_int_compare:w #1 > - \c_one_int
      \__regex_toks_put_left:Ne #2 { \__regex_action_submatch:nN {#1} < }
      \__regex_toks_put_left:Ne #3 { \__regex_action_submatch:nN {#1} > }
    \fi:
  }
\cs_new_protected:Npn \__regex_group_repeat_aux:n #1
```

```
  {
    \__regex_build_transition_right:nNn \__regex_action_free:n
      \l__regex_right_state_int \l__regex_max_state_int
    \int_set_eq:NN \l__regex_internal_a_int \l__regex_left_state_int
    \int_set_eq:NN \l__regex_internal_b_int \l__regex_max_state_int
    \if_int_compare:w \__regex_int_eval:w #1 > \c_one_int
      \int_set:Nn \l__regex_internal_c_int
        {
          ( #1 - \c_one_int )
          * ( \l__regex_internal_b_int - \l__regex_internal_a_int )
        }
      \int_add:Nn \l__regex_right_state_int \l__regex_internal_c_int
      \int_add:Nn \l__regex_max_state_int   \l__regex_internal_c_int
      \__regex_toks_memcpy:NNn
        \l__regex_internal_b_int
        \l__regex_internal_a_int
        \l__regex_internal_c_int
    \fi:
  }
\cs_new_protected:Npn \__regex_group_repeat:nnN #1#2#3
  {
    \if_int_compare:w #2 = \c_zero_int
      \__regex_group_submatches:nNN {#1}
        \l__regex_left_state_int \l__regex_right_state_int
      \int_set:Nn \l__regex_internal_a_int
        { \l__regex_left_state_int - \c_one_int }
      \__regex_build_transition_right:nNn \__regex_action_free:n
        \l__regex_right_state_int \l__regex_internal_a_int
      \__regex_build_new_state:
      \if_meaning:w \c_true_bool #3
        \__regex_build_transition_left:NNN \__regex_action_free:n
          \l__regex_internal_a_int \l__regex_right_state_int
      \else:
        \__regex_build_transition_right:nNn \__regex_action_free:n
          \l__regex_internal_a_int \l__regex_right_state_int
      \fi:
    \else:
      \__regex_group_repeat_aux:n {#2}
      \__regex_group_submatches:nNN {#1}
        \l__regex_internal_a_int \l__regex_right_state_int
      \if_meaning:w \c_true_bool #3
        \__regex_build_transition_right:nNn \__regex_action_free_group:n
          \l__regex_right_state_int \l__regex_internal_a_int
      \else:
        \__regex_build_transition_left:NNN \__regex_action_free_group:n
          \l__regex_right_state_int \l__regex_internal_a_int
      \fi:
      \__regex_build_new_state:
    \fi:
  }
\cs_new_protected:Npn \__regex_group_repeat:nnnN #1#2#3#4
  {
    \__regex_group_submatches:nNN {#1}
      \l__regex_left_state_int \l__regex_right_state_int
    \__regex_group_repeat_aux:n { #2 + #3 }
    \if_meaning:w \c_true_bool #4
      \int_set_eq:NN \l__regex_left_state_int \l__regex_max_state_int
      \prg_replicate:nn { #3 }
        {
          \int_sub:Nn \l__regex_left_state_int
            { \l__regex_internal_b_int - \l__regex_internal_a_int }
          \__regex_build_transition_left:NNN \__regex_action_free:n
            \l__regex_left_state_int \l__regex_max_state_int
        }
    \else:
      \prg_replicate:nn { #3 - \c_one_int }
        {
          \int_sub:Nn \l__regex_right_state_int
            { \l__regex_internal_b_int - \l__regex_internal_a_int }
          \__regex_build_transition_right:nNn \__regex_action_free:n
            \l__regex_right_state_int \l__regex_max_state_int
        }
      \if_int_compare:w #2 = \c_zero_int
        \int_set:Nn \l__regex_right_state_int
          { \l__regex_left_state_int - \c_one_int }
      \else:
        \int_sub:Nn \l__regex_right_state_int
          { \l__regex_internal_b_int - \l__regex_internal_a_int }
      \fi:
      \__regex_build_transition_right:nNn \__regex_action_free:n
        \l__regex_right_state_int \l__regex_max_state_int
    \fi:
    \__regex_build_new_state:
  }
\cs_new_protected:Npn \__regex_assertion:Nn #1#2
  {
    \__regex_build_new_state:
    \__regex_toks_put_right:Ne \l__regex_left_state_int
      {
        \exp_not:n {#2}
        \__regex_break_point:TF
          \bool_if:NF #1 { { } }
          {
            \__regex_action_free:n
              {
                \tex_the:D \__regex_int_eval:w
                  \l__regex_right_state_int - \l__regex_left_state_int
              }
          }
          \bool_if:NT #1 { { } }
      }
  }
\cs_new_protected:Npn \__regex_b_test:
  {
    \group_begin:
      \int_set_eq:NN \l__regex_curr_char_int \l__regex_last_char_int
      \__regex_prop_w:
      \__regex_break_point:TF
        { \group_end: \__regex_item_reverse:n { \__regex_prop_w: } }
        { \group_end: \__regex_prop_w: }
  }
\cs_new_protected:Npn \__regex_Z_test:
  {
    \if_int_compare:w -2 = \l__regex_curr_char_int
      \exp_after:wN \__regex_break_true:w
    \fi:
  }
\cs_new_protected:Npn \__regex_A_test:
  {
    \if_int_compare:w -2 = \l__regex_last_char_int
      \exp_after:wN \__regex_break_true:w
    \fi:
  }
\cs_new_protected:Npn \__regex_G_test:
  {
    \if_int_compare:w \l__regex_curr_pos_int = \l__regex_start_pos_int
      \exp_after:wN \__regex_break_true:w
    \fi:
  }
\cs_new_protected:Npn \__regex_command_K:
  {
    \__regex_build_new_state:
    \__regex_toks_put_right:Ne \l__regex_left_state_int
      {
        \__regex_action_submatch:nN \c_zero_int <
        \bool_set_true:N \l__regex_fresh_thread_bool
        \__regex_action_free:n
          {
            \tex_the:D \__regex_int_eval:w
              \l__regex_right_state_int - \l__regex_left_state_int
          }
        \bool_set_false:N \l__regex_fresh_thread_bool
      }
  }
\int_new:N \l__regex_min_pos_int
\int_new:N \l__regex_max_pos_int
\int_new:N \l__regex_curr_pos_int
\int_new:N \l__regex_start_pos_int
\int_new:N \l__regex_success_pos_int
\int_new:N \l__regex_curr_char_int
\int_new:N \l__regex_curr_catcode_int
\tl_new:N \l__regex_curr_token_tl
\int_new:N \l__regex_last_char_int
\int_new:N \l__regex_last_char_success_int
\int_new:N \l__regex_case_changed_char_int
\int_new:N \l__regex_curr_state_int
\tl_new:N \l__regex_curr_submatches_tl
\tl_new:N \l__regex_success_submatches_tl
\int_new:N \l__regex_step_int
\int_new:N \l__regex_min_thread_int
\int_new:N \l__regex_max_thread_int
\intarray_new:Nn \g__regex_state_active_intarray { 65536 }
\intarray_new:Nn \g__regex_thread_info_intarray { 65536 }
\tl_new:N \l__regex_matched_analysis_tl
\tl_new:N \l__regex_curr_analysis_tl
\tl_new:N \l__regex_every_match_tl
\bool_new:N \l__regex_fresh_thread_bool
\bool_new:N \l__regex_empty_success_bool
\cs_new_eq:NN \__regex_if_two_empty_matches:F \use:n
\bool_new:N \g__regex_success_bool
\bool_new:N \l__regex_saved_success_bool
\bool_new:N \l__regex_match_success_bool
\cs_new_protected:Npn \__regex_match:n #1
  {
    \__regex_match_init:
    \__regex_match_once_init:
    \tl_analysis_map_inline:nn {#1}
      { \__regex_match_one_token:nnN {##1} {##2} ##3 }
    \__regex_match_one_token:nnN { } { -2 } F
    \prg_break_point:Nn \__regex_maplike_break: { }
  }
\cs_new_protected:Npn \__regex_match_cs:n #1
  {
    \int_set_eq:NN \l__regex_min_thread_int \l__regex_max_thread_int
    \__regex_match_init:
    \__regex_match_once_init:
    \str_map_inline:nn {#1}
      {
        \tl_if_blank:nTF {##1}
          { \__regex_match_one_token:nnN {##1} {`##1} A }
          { \__regex_match_one_token:nnN {##1} {`##1} C }
      }
    \__regex_match_one_token:nnN { } { -2 } F
    \prg_break_point:Nn \__regex_maplike_break: { }
  }
\cs_new_protected:Npn \__regex_match_init:
  {
    \bool_gset_false:N \g__regex_success_bool
    \int_step_inline:nnn
      \l__regex_min_state_int { \l__regex_max_state_int - \c_one_int }
      {
        \__kernel_intarray_gset:Nnn
          \g__regex_state_active_intarray {##1} \c_one_int
      }
    \int_zero:N \l__regex_step_int
    \int_set:Nn \l__regex_min_pos_int { 2 }
    \int_set_eq:NN \l__regex_success_pos_int \l__regex_min_pos_int
    \int_set:Nn \l__regex_last_char_success_int { -2 }
    \tl_build_begin:N \l__regex_matched_analysis_tl
    \tl_clear:N \l__regex_curr_analysis_tl
    \int_set_eq:NN \l__regex_min_submatch_int \c_one_int
    \int_set_eq:NN \l__regex_submatch_int \l__regex_min_submatch_int
    \bool_set_false:N \l__regex_empty_success_bool
  }
\cs_new_protected:Npn \__regex_match_once_init:
  {
    \if_meaning:w \c_true_bool \l__regex_empty_success_bool
      \cs_set:Npn \__regex_if_two_empty_matches:F
        {
          \int_compare:nNnF
            \l__regex_start_pos_int = \l__regex_curr_pos_int
        }
    \else:
      \cs_set_eq:NN \__regex_if_two_empty_matches:F \use:n
    \fi:
    \int_set_eq:NN \l__regex_start_pos_int \l__regex_success_pos_int
    \bool_set_false:N \l__regex_match_success_bool
    \tl_set:Ne \l__regex_curr_submatches_tl
      { \prg_replicate:nn { 2 * \l__regex_capturing_group_int } { 0 , } }
    \int_set_eq:NN \l__regex_max_thread_int \l__regex_min_thread_int
    \__regex_store_state:n { \l__regex_min_state_int }
    \int_set:Nn \l__regex_curr_pos_int { \l__regex_start_pos_int - \c_one_int }
    \int_set_eq:NN \l__regex_curr_char_int \l__regex_last_char_success_int
    \tl_build_get_intermediate:NN \l__regex_matched_analysis_tl \l__regex_internal_a_tl
    \exp_args:NNf \__regex_match_once_init_aux:
    \tl_map_inline:nn
      { \exp_after:wN \l__regex_internal_a_tl \l__regex_curr_analysis_tl }
      { \__regex_match_one_token:nnN ##1 }
    \prg_break_point:Nn \__regex_maplike_break: { }
  }
\cs_new_protected:Npn \__regex_match_once_init_aux:
  {
    \tl_build_begin:N \l__regex_matched_analysis_tl
    \tl_clear:N \l__regex_curr_analysis_tl
  }
\cs_new_protected:Npn \__regex_single_match:
  {
    \tl_set:Nn \l__regex_every_match_tl
      {
        \bool_gset_eq:NN
          \g__regex_success_bool
          \l__regex_match_success_bool
        \__regex_maplike_break:
      }
  }
\cs_new_protected:Npn \__regex_multi_match:n #1
  {
    \tl_set:Nn \l__regex_every_match_tl
      {
        \if_meaning:w \c_false_bool \l__regex_match_success_bool
          \exp_after:wN \__regex_maplike_break:
        \fi:
        \bool_gset_true:N \g__regex_success_bool
        #1
        \__regex_match_once_init:
      }
  }
\cs_new_protected:Npn \__regex_match_one_token:nnN #1#2#3
  {
    \int_add:Nn \l__regex_step_int { 2 }
    \int_incr:N \l__regex_curr_pos_int
    \int_set_eq:NN \l__regex_last_char_int \l__regex_curr_char_int
    \cs_set_eq:NN \__regex_maybe_compute_ccc: \__regex_compute_case_changed_char:
    \tl_set:Nn \l__regex_curr_token_tl {#1}
    \int_set:Nn \l__regex_curr_char_int {#2}
    \int_set:Nn \l__regex_curr_catcode_int { "#3 }
    \tl_build_put_right:Ne \l__regex_matched_analysis_tl
      { \exp_not:o \l__regex_curr_analysis_tl }
    \tl_set:Nn \l__regex_curr_analysis_tl { { {#1} {#2} #3 } }
    \use:e
      {
        \int_set_eq:NN \l__regex_max_thread_int \l__regex_min_thread_int
        \int_step_function:nnN
          \l__regex_min_thread_int
          { \l__regex_max_thread_int - \c_one_int }
          \__regex_match_one_active:n
      }
    \prg_break_point:
    \bool_set_false:N \l__regex_fresh_thread_bool
    \if_int_compare:w \l__regex_max_thread_int > \l__regex_min_thread_int
      \if_int_compare:w -2 < \l__regex_curr_char_int
        \exp_after:wN \use_i:nn
      \fi:
    \fi:
    \l__regex_every_match_tl
  }
\cs_new:Npn \__regex_match_one_active:n #1
  {
    \__regex_use_state_and_submatches:w
    \__kernel_intarray_range_to_clist:Nnn
      \g__regex_thread_info_intarray
      { \c_one_int + #1 * (\l__regex_capturing_group_int * 2 + \c_one_int) }
      { (\c_one_int + #1) * (\l__regex_capturing_group_int * 2 + \c_one_int) }
    \__regex_sep:
  }
\cs_new_protected:Npn \__regex_use_state:
  {
    \__kernel_intarray_gset:Nnn \g__regex_state_active_intarray
      \l__regex_curr_state_int \l__regex_step_int
    \__regex_toks_use:w \l__regex_curr_state_int
    \__kernel_intarray_gset:Nnn \g__regex_state_active_intarray
      \l__regex_curr_state_int
      { \__regex_int_eval:w \l__regex_step_int + \c_one_int \scan_stop: }
  }
\cs_new_protected:Npn \__regex_use_state_and_submatches:w #1 , #2 \__regex_sep:
  {
    \int_set:Nn \l__regex_curr_state_int {#1}
    \if_int_compare:w
        \__kernel_intarray_item:Nn \g__regex_state_active_intarray
          \l__regex_curr_state_int
                      < \l__regex_step_int
      \tl_set:Nn \l__regex_curr_submatches_tl { #2 , }
      \exp_after:wN \__regex_use_state:
    \fi:
    \scan_stop:
  }
\cs_new_protected:Npn \__regex_action_start_wildcard:N #1
  {
    \bool_set_true:N \l__regex_fresh_thread_bool
    \__regex_action_free:n {1}
    \bool_set_false:N \l__regex_fresh_thread_bool
    \bool_if:NT #1 { \__regex_action_cost:n {0} }
  }
\cs_new_protected:Npn \__regex_action_free:n
  { \__regex_action_free_aux:nn { > \l__regex_step_int \else: } }
\cs_new_protected:Npn \__regex_action_free_group:n
  { \__regex_action_free_aux:nn { < \l__regex_step_int } }
\cs_new_protected:Npn \__regex_action_free_aux:nn #1#2
  {
    \use:e
      {
        \int_add:Nn \l__regex_curr_state_int {#2}
        \exp_not:n
          {
            \if_int_compare:w
                \__kernel_intarray_item:Nn \g__regex_state_active_intarray
                  \l__regex_curr_state_int
                #1
              \exp_after:wN \__regex_use_state:
            \fi:
          }
        \int_set:Nn \l__regex_curr_state_int
          { \int_use:N \l__regex_curr_state_int }
        \tl_set:Nn \exp_not:N \l__regex_curr_submatches_tl
          { \exp_not:o \l__regex_curr_submatches_tl }
      }
  }
\cs_new_protected:Npn \__regex_action_cost:n #1
  {
    \exp_args:No \__regex_store_state:n
      { \tex_the:D \__regex_int_eval:w \l__regex_curr_state_int + #1 }
  }
\cs_new_protected:Npn \__regex_store_state:n #1
  {
    \exp_args:No \__regex_store_submatches:nn
      \l__regex_curr_submatches_tl {#1}
    \int_incr:N \l__regex_max_thread_int
  }
```

```
\cs_new_protected:Npn \__regex_store_submatches:nn #1#2
  {
    \__kernel_intarray_gset_range_from_clist:Nnn
      \g__regex_thread_info_intarray
      {
        \__regex_int_eval:w
        \c_one_int + \l__regex_max_thread_int *
        (\l__regex_capturing_group_int * 2 + \c_one_int)
      }
      { #2 , #1 }
  }
\cs_new_protected:Npn \__regex_disable_submatches:
  {
    \cs_set_protected:Npn \__regex_store_submatches:n ##1 { }
    \cs_set_protected:Npn \__regex_action_submatch:nN ##1##2 { }
  }
\cs_new_protected:Npn \__regex_action_submatch:nN #1#2
  {
    \exp_after:wN \__regex_action_submatch_aux:w
    \l__regex_curr_submatches_tl \__regex_sep: {#1} #2
  }
\cs_new_protected:Npn \__regex_action_submatch_aux:w #1 \__regex_sep: #2#3
  {
    \tl_set:Ne \l__regex_curr_submatches_tl
      {
        \prg_replicate:nn
          { #2 \if_meaning:w > #3 + \l__regex_capturing_group_int \fi: }
          { \__regex_action_submatch_auxii:w }
        \__regex_action_submatch_auxiii:w
        #1
      }
  }
\cs_new:Npn \__regex_action_submatch_auxii:w
    #1 \__regex_action_submatch_auxiii:w #2 ,
  { #2 , #1 \__regex_action_submatch_auxiii:w }
\cs_new:Npn \__regex_action_submatch_auxiii:w #1 ,
  { \int_use:N \l__regex_curr_pos_int , }
\cs_new_protected:Npn \__regex_action_success:
  {
    \__regex_if_two_empty_matches:F
      {
        \bool_set_true:N \l__regex_match_success_bool
        \bool_set_eq:NN \l__regex_empty_success_bool
          \l__regex_fresh_thread_bool
        \int_set_eq:NN \l__regex_success_pos_int \l__regex_curr_pos_int
        \int_set_eq:NN \l__regex_last_char_success_int \l__regex_last_char_int
        \tl_build_begin:N \l__regex_matched_analysis_tl
        \tl_set_eq:NN \l__regex_success_submatches_tl
          \l__regex_curr_submatches_tl
        \prg_break:
      }
  }
\int_new:N \l__regex_replacement_csnames_int
\tl_new:N \l__regex_replacement_category_tl
\seq_new:N \l__regex_replacement_category_seq
\tl_new:N \g__regex_balance_tl
\cs_new:Npn \__regex_replacement_balance_one_match:n #1
  { - \__regex_submatch_balance:n {#1} }
\cs_new:Npn \__regex_replacement_do_one_match:n #1
  {
    \__regex_query_range:nn
      { \__kernel_intarray_item:Nn \g__regex_submatch_prev_intarray {#1} }
      { \__kernel_intarray_item:Nn \g__regex_submatch_begin_intarray {#1} }
  }
\cs_new:Npn \__regex_replacement_exp_not:N #1 { \exp_not:n {#1} }
\cs_new_eq:NN \__regex_replacement_exp_not:V \exp_not:V
\cs_new:Npn \__regex_query_range:nn #1#2
  {
    \exp_after:wN \__regex_query_range_loop:ww
    \int_value:w \__regex_int_eval:w #1 \exp_after:wN \__regex_sep:
    \int_value:w \__regex_int_eval:w #2 \__regex_sep:
    \prg_break_point:
  }
\cs_new:Npn \__regex_query_range_loop:ww #1 \__regex_sep: #2 \__regex_sep:
  {
    \if_int_compare:w #1 < #2 \exp_stop_f:
    \else:
      \prg_break:n
    \fi:
    \__regex_toks_use:w #1 \exp_stop_f:
    \exp_after:wN \__regex_query_range_loop:ww
      \int_value:w \__regex_int_eval:w #1 + \c_one_int \__regex_sep: #2 \__regex_sep:
  }
\cs_new:Npn \__regex_query_submatch:n #1
  {
    \__regex_query_range:nn
      { \__kernel_intarray_item:Nn \g__regex_submatch_begin_intarray {#1} }
      { \__kernel_intarray_item:Nn \g__regex_submatch_end_intarray {#1} }
  }
\cs_new_protected:Npn \__regex_submatch_balance:n #1
  {
    \tex_the:D \__regex_int_eval:w
```

```
      \__regex_intarray_item:NnF \g__regex_balance_intarray
        {
          \__kernel_intarray_item:Nn
            \g__regex_submatch_end_intarray {#1}
        }
        \c_zero_int
      -
      \__regex_intarray_item:NnF \g__regex_balance_intarray
        {
          \__kernel_intarray_item:Nn
            \g__regex_submatch_begin_intarray {#1}
        }
        \c_zero_int
    \scan_stop:
  }
\cs_new_protected:Npn \__regex_replacement:n
  { \__regex_replacement_apply:Nn \__regex_replacement_set:n }
\cs_new_protected:Npn \__regex_replacement_apply:Nn #1#2
  {
    \group_begin:
      \tl_build_begin:N \l__regex_build_tl
      \int_zero:N \l__regex_balance_int
      \tl_gclear:N \g__regex_balance_tl
      \__regex_escape_use:nnnn
        {
          \if_charcode:w \c_right_brace_str ##1
            \__regex_replacement_rbrace:N
          \else:
            \if_charcode:w \c_left_brace_str ##1
              \__regex_replacement_lbrace:N
            \else:
              \__regex_replacement_normal:n
            \fi:
          \fi:
          ##1
        }
        { \__regex_replacement_escaped:N ##1 }
        { \__regex_replacement_normal:n ##1 }
        {#2}
      \prg_do_nothing: \prg_do_nothing:
      \if_int_compare:w \l__regex_replacement_csnames_int > \c_zero_int
        \msg_error:nne { regex } { replacement-missing-rbrace }
          { \int_use:N \l__regex_replacement_csnames_int }
        \tl_build_put_right:Ne \l__regex_build_tl
          { \prg_replicate:nn \l__regex_replacement_csnames_int \cs_end: }
      \fi:
```

```
      \seq_if_empty:NF \l__regex_replacement_category_seq
        {
          \msg_error:nne { regex } { replacement-missing-rparen }
            { \seq_count:N \l__regex_replacement_category_seq }
          \seq_clear:N \l__regex_replacement_category_seq
        }
      \tl_gput_right:Ne \g__regex_balance_tl
        { + \int_use:N \l__regex_balance_int }
      \tl_build_end:N \l__regex_build_tl
      \exp_args:NNo
    \group_end:
    #1 \l__regex_build_tl
  }
\cs_generate_variant:Nn \__regex_replacement:n { e }
\cs_new_protected:Npn \__regex_replacement_set:n #1
  {
    \cs_set:Npn \__regex_replacement_do_one_match:n ##1
      {
        \__regex_query_range:nn
          {
            \__kernel_intarray_item:Nn
              \g__regex_submatch_prev_intarray {##1}
          }
          {
            \__kernel_intarray_item:Nn
              \g__regex_submatch_begin_intarray {##1}
          }
        #1
      }
    \exp_args:Nno \use:n
      { \cs_gset:Npn \__regex_replacement_balance_one_match:n ##1 }
      {
        \g__regex_balance_tl
        - \__regex_submatch_balance:n {##1}
      }
  }
\tl_new:N \g__regex_case_replacement_tl
\tl_new:N \g__regex_case_balance_tl
\cs_new_protected:Npn \__regex_case_replacement:n #1
  {
    \tl_gset:Nn \g__regex_case_balance_tl
      {
        \if_case:w
          \__kernel_intarray_item:Nn
            \g__regex_submatch_case_intarray {##1}
      }
    \tl_gset_eq:NN \g__regex_case_replacement_tl \g__regex_case_balance_tl
    \tl_map_tokens:nn {#1}
      { \__regex_replacement_apply:Nn \__regex_case_replacement_aux:n }
    \tl_gset:No \g__regex_balance_tl
      { \g__regex_case_balance_tl \fi: }
    \exp_args:No \__regex_replacement_set:n
      { \g__regex_case_replacement_tl \fi: }
  }
\cs_generate_variant:Nn \__regex_case_replacement:n { e }
\cs_new_protected:Npn \__regex_case_replacement_aux:n #1
  {
    \tl_gput_right:Nn \g__regex_case_replacement_tl { \or: #1 }
    \tl_gput_right:No \g__regex_case_balance_tl
      { \exp_after:wN \or: \g__regex_balance_tl }
  }
\cs_new_protected:Npn \__regex_replacement_put:n
  { \tl_build_put_right:Nn \l__regex_build_tl }
\cs_new_protected:Npn \__regex_replacement_normal:n #1
  {
    \int_compare:nNnTF \l__regex_replacement_csnames_int > \c_zero_int
      { \exp_args:No \__regex_replacement_put:n { \token_to_str:N #1 } }
      {
        \tl_if_empty:NTF \l__regex_replacement_category_tl
          { \__regex_replacement_normal_aux:N #1 }
          { % (
            \token_if_eq_charcode:NNTF #1 )
              {
                \seq_pop:NN \l__regex_replacement_category_seq
                  \l__regex_replacement_category_tl
              }
              {
                \use:c { __regex_replacement_c_ \l__regex_replacement_category_tl :w }
                ? #1
              }
          }
      }
  }
\cs_new_protected:Npn \__regex_replacement_normal_aux:N #1
  {
    \token_if_eq_charcode:NNTF #1 \c_space_token
      { \__regex_replacement_c_S:w }
      {
        \exp_after:wN \exp_after:wN
        \if_case:w \tex_catcode:D `#1 \exp_stop_f:
             \__regex_replacement_c_O:w
        \or: \__regex_replacement_c_B:w
        \or: \__regex_replacement_c_E:w
        \or: \__regex_replacement_c_M:w
        \or: \__regex_replacement_c_T:w
        \or: \__regex_replacement_c_O:w
        \or: \__regex_replacement_c_P:w
        \or: \__regex_replacement_c_U:w
        \or: \__regex_replacement_c_D:w
        \or: \__regex_replacement_c_O:w
        \or: \__regex_replacement_c_S:w
        \or: \__regex_replacement_c_L:w
        \or: \__regex_replacement_c_O:w
        \or: \__regex_replacement_c_A:w
        \else: \__regex_replacement_c_O:w
        \fi:
      }
    ? #1
  }
\cs_new_protected:Npn \__regex_replacement_escaped:N #1
  {
    \cs_if_exist_use:cF { __regex_replacement_#1:w }
      {
        \if_int_compare:w \c_one_int < 1#1 \exp_stop_f:
          \__regex_replacement_put_submatch:n {#1}
        \else:
          \__regex_replacement_normal:n {#1}
        \fi:
      }
  }
\cs_new_protected:Npn \__regex_replacement_put_submatch:n #1
  {
    \if_int_compare:w #1 < \l__regex_capturing_group_int
      \__regex_replacement_put_submatch_aux:n {#1}
    \else:
      \msg_expandable_error:nnff { regex } { submatch-too-big }
        {#1} { \int_eval:n { \l__regex_capturing_group_int - \c_one_int } }
    \fi:
  }
\cs_new_protected:Npn \__regex_replacement_put_submatch_aux:n #1
  {
    \tl_build_put_right:Nn \l__regex_build_tl
      { \__regex_query_submatch:n { \__regex_int_eval:w #1 + ##1 \scan_stop: } }
    \if_int_compare:w \l__regex_replacement_csnames_int = \c_zero_int
      \tl_gput_right:Nn \g__regex_balance_tl
        { + \__regex_submatch_balance:n { \__regex_int_eval:w #1 + ##1 \scan_stop: } }
    \fi:
  }
```

```
\cs_new_protected:Npn \__regex_replacement_g:w #1#2
  {
    \token_if_eq_meaning:NNTF #1 \__regex_replacement_lbrace:N
      { \l__regex_internal_a_int = \__regex_replacement_g_digits:NN }
      { \__regex_replacement_error:NNN g #1 #2 }
  }
\cs_new:Npn \__regex_replacement_g_digits:NN #1#2
  {
    \token_if_eq_meaning:NNTF #1 \__regex_replacement_normal:n
      {
        \if_int_compare:w \c_one_int < 1#2 \exp_stop_f:
          #2
          \exp_after:wN \use_i:nnn
          \exp_after:wN \__regex_replacement_g_digits:NN
        \else:
          \exp_stop_f:
          \exp_after:wN \__regex_replacement_error:NNN
          \exp_after:wN g
        \fi:
      }
      {
        \exp_stop_f:
        \if_meaning:w \__regex_replacement_rbrace:N #1
          \exp_args:No \__regex_replacement_put_submatch:n
            { \int_use:N \l__regex_internal_a_int }
          \exp_after:wN \use_none:nn
        \else:
          \exp_after:wN \__regex_replacement_error:NNN
          \exp_after:wN g
        \fi:
      }
    #1 #2
  }
\cs_new_protected:Npn \__regex_replacement_c:w #1#2
  {
    \token_if_eq_meaning:NNTF #1 \__regex_replacement_normal:n
      {
        \cs_if_exist:cTF { __regex_replacement_c_#2:w }
          { \__regex_replacement_cat:NNN #2 }
          { \__regex_replacement_error:NNN c #1#2 }
      }
      {
        \token_if_eq_meaning:NNTF #1 \__regex_replacement_lbrace:N
          { \__regex_replacement_cu_aux:Nw \__regex_replacement_exp_not:N }
          { \__regex_replacement_error:NNN c #1#2 }
      }
  }
\cs_new_protected:Npn \__regex_replacement_cu_aux:Nw #1
  {
    \if_case:w \l__regex_replacement_csnames_int
      \tl_build_put_right:Nn \l__regex_build_tl
        { \exp_not:n { \exp_after:wN #1 \cs:w } }
    \else:
      \tl_build_put_right:Nn \l__regex_build_tl
        { \exp_not:n { \exp_after:wN \tl_to_str:V \cs:w } }
    \fi:
    \int_incr:N \l__regex_replacement_csnames_int
  }
\cs_new_protected:Npn \__regex_replacement_u:w #1#2
  {
    \token_if_eq_meaning:NNTF #1 \__regex_replacement_lbrace:N
      { \__regex_replacement_cu_aux:Nw \__regex_replacement_exp_not:V }
      { \__regex_replacement_error:NNN u #1#2 }
  }
\cs_new_protected:Npn \__regex_replacement_rbrace:N #1
  {
    \if_int_compare:w \l__regex_replacement_csnames_int > \c_zero_int
      \tl_build_put_right:Nn \l__regex_build_tl { \cs_end: }
      \int_decr:N \l__regex_replacement_csnames_int
    \else:
      \__regex_replacement_normal:n {#1}
    \fi:
  }
\cs_new_protected:Npn \__regex_replacement_lbrace:N #1
  {
    \if_int_compare:w \l__regex_replacement_csnames_int > \c_zero_int
      \msg_error:nnn { regex } { cu-lbrace } { u }
    \else:
      \__regex_replacement_normal:n {#1}
    \fi:
  }
\cs_new_protected:Npn \__regex_replacement_cat:NNN #1#2#3
  {
    \token_if_eq_meaning:NNTF \prg_do_nothing: #3
      { \msg_error:nn { regex } { replacement-catcode-end } }
      {
        \int_compare:nNnTF \l__regex_replacement_csnames_int > \c_zero_int
          {
            \msg_error:nnnn
              { regex } { replacement-catcode-in-cs } {#1} {#3}
            #2 #3
          }
          {
            \__regex_two_if_eq:NNNNTF #2 #3 \__regex_replacement_normal:n (
              {
                \seq_push:NV \l__regex_replacement_category_seq
                  \l__regex_replacement_category_tl
                \tl_set:Nn \l__regex_replacement_category_tl {#1}
              }
              {
                \token_if_eq_meaning:NNT #2 \__regex_replacement_escaped:N
                  {
                    \__regex_char_if_alphanumeric:NTF #3
                      {
                        \msg_error:nnnn
                          { regex } { replacement-catcode-escaped }
                          {#1} {#3}
                      }
                      { }
                  }
                \use:c { __regex_replacement_c_#1:w } #2 #3
              }
          }
      }
  }
\group_begin:
  \cs_new_protected:Npn \__regex_replacement_char:nNN #1#2#3
    {
      \tex_lccode:D \c_zero_int = `#3 \scan_stop:
      \tex_lowercase:D { \__regex_replacement_put:n {#1} }
    }
  \char_set_catcode_active:N \^^@
  \cs_new_protected:Npn \__regex_replacement_c_A:w
    { \__regex_replacement_char:nNN { \exp_not:n { \exp_not:N ^^@ } } }
  \char_set_catcode_group_begin:N \^^@
  \cs_new_protected:Npn \__regex_replacement_c_B:w
    {
      \if_int_compare:w \l__regex_replacement_csnames_int = \c_zero_int
        \int_incr:N \l__regex_balance_int
      \fi:
      \__regex_replacement_char:nNN
        { \exp_not:n { \exp_after:wN ^^@ \if_false: } \fi: } }
    }
  \cs_new_protected:Npn \__regex_replacement_c_C:w #1#2
    {
      \tl_build_put_right:Nn \l__regex_build_tl
        { \exp_not:N \__regex_replacement_exp_not:N \exp_not:c {#2} }
    }
  \char_set_catcode_math_subscript:N \^^@
  \cs_new_protected:Npn \__regex_replacement_c_D:w
    { \__regex_replacement_char:nNN { ^^@ } }
  \char_set_catcode_group_end:N \^^@
  \cs_new_protected:Npn \__regex_replacement_c_E:w
    {
      \if_int_compare:w \l__regex_replacement_csnames_int = \c_zero_int
        \int_decr:N \l__regex_balance_int
      \fi:
      \__regex_replacement_char:nNN
        { \exp_not:n { \if_false: { \fi:  ^^@ } }
    }
  \char_set_catcode_letter:N \^^@
  \cs_new_protected:Npn \__regex_replacement_c_L:w
    { \__regex_replacement_char:nNN { ^^@ } }
  \char_set_catcode_math_toggle:N \^^@
  \cs_new_protected:Npn \__regex_replacement_c_M:w
    { \__regex_replacement_char:nNN { ^^@ } }
  \char_set_catcode_other:N \^^@
  \cs_new_protected:Npn \__regex_replacement_c_O:w
    { \__regex_replacement_char:nNN { ^^@ } }
  \char_set_catcode_parameter:N \^^@
  \cs_new_protected:Npn \__regex_replacement_c_P:w
    {
      \__regex_replacement_char:nNN
        { \exp_not:n { \exp_not:n { ^^@^^@^^@^^@ } } }
    }
  \cs_new_protected:Npn \__regex_replacement_c_S:w #1#2
    {
      \if_int_compare:w `#2 = \c_zero_int
        \msg_error:nn { regex } { replacement-null-space }
      \fi:
      \tex_lccode:D `\ = `#2 \scan_stop:
      \tex_lowercase:D { \__regex_replacement_put:n {~} }
    }
  \char_set_catcode_alignment:N \^^@
  \cs_new_protected:Npn \__regex_replacement_c_T:w
    { \__regex_replacement_char:nNN { ^^@ } }
  \char_set_catcode_math_superscript:N \^^@
  \cs_new_protected:Npn \__regex_replacement_c_U:w
    { \__regex_replacement_char:nNN { ^^@ } }
\group_end:
\cs_new_protected:Npn \__regex_replacement_error:NNN #1#2#3
  {
    \msg_error:nne { regex } { replacement-#1 } {#3}
    #2 #3
  }
\cs_new_protected:Npn \regex_new:N #1
  { \cs_new_eq:NN #1 \c__regex_no_match_regex }
\regex_new:N \l_tmpa_regex
\regex_new:N \l_tmpb_regex
\regex_new:N \g_tmpa_regex
\regex_new:N \g_tmpb_regex
\cs_new_protected:Npn \regex_set:Nn #1#2
  {
    \__regex_compile:n {#2}
    \tl_set_eq:NN #1 \l__regex_internal_regex
  }
\cs_new_protected:Npn \regex_gset:Nn #1#2
  {
    \__regex_compile:n {#2}
    \tl_gset_eq:NN #1 \l__regex_internal_regex
  }
\cs_new_protected:Npn \regex_const:Nn #1#2
  {
    \__regex_compile:n {#2}
    \tl_const:Ne #1 { \exp_not:o \l__regex_internal_regex }
  }
\cs_new_protected:Npn \regex_show:n { \__regex_show:Nn \msg_show:nneeee }
\cs_new_protected:Npn \regex_log:n { \__regex_show:Nn \msg_log:nneeee }
\cs_new_protected:Npn \__regex_show:Nn #1#2
  {
    \__regex_compile:n {#2}
    \__regex_show:N \l__regex_internal_regex
    #1 { regex } { show }
      { \tl_to_str:n {#2} } { }
      { \l__regex_internal_a_tl } { }
  }
\cs_new_protected:Npn \regex_show:N { \__regex_show:NN \msg_show:nneeee }
\cs_new_protected:Npn \regex_log:N { \__regex_show:NN \msg_log:nneeee }
\cs_new_protected:Npn \__regex_show:NN #1#2
  {
    \__kernel_chk_tl_type:NnnT #2 { regex }
      { \exp_args:No \__regex_clean_regex:n {#2} }
      {
        \__regex_show:N #2
        #1 { regex } { show }
          { } { \token_to_str:N #2 }
          { \l__regex_internal_a_tl } { }
      }
  }
\prg_new_protected_conditional:Npnn \regex_match:nn #1#2 { T , F , TF }
```

```
  {
    \__regex_if_match:nn { \__regex_build:n {#1} } {#2}
    \__regex_return:
  }
\prg_generate_conditional_variant:Nnn \regex_match:nn { nV } { T , F , TF }
\prg_new_protected_conditional:Npnn \regex_match:Nn #1#2 { T , F , TF }
  {
    \__regex_if_match:nn { \__regex_build:N #1 } {#2}
    \__regex_return:
  }
\prg_generate_conditional_variant:Nnn \regex_match:Nn { NV } { T , F , TF }
\cs_new_protected:Npn \regex_count:nnN #1
  { \__regex_count:nnN { \__regex_build:n {#1} } }
\cs_new_protected:Npn \regex_count:NnN #1
  { \__regex_count:nnN { \__regex_build:N #1 } }
\cs_generate_variant:Nn \regex_count:nnN { nV }
\cs_generate_variant:Nn \regex_count:NnN { NV }
\cs_new_protected:Npn \regex_match_case:nnTF #1#2#3
  {
    \__regex_match_case:nnTF {#1} {#2}
      {
        \tl_item:nn {#1} { 2 * \g__regex_case_int }
        #3
      }
  }
\cs_new_protected:Npn \regex_match_case:nn #1#2
  { \regex_match_case:nnTF {#1} {#2} { } { } }
\cs_new_protected:Npn \regex_match_case:nnT #1#2#3
  { \regex_match_case:nnTF {#1} {#2} {#3} { } }
\cs_new_protected:Npn \regex_match_case:nnF #1#2
  { \regex_match_case:nnTF {#1} {#2} { } }
\cs_set_protected:Npn \__regex_tmp:w #1#2#3
  {
    \cs_new_protected:Npn #2 ##1 { #1 { \__regex_build:n {##1} } }
    \cs_new_protected:Npn #3 ##1 { #1 { \__regex_build:N  ##1  } }
    \prg_new_protected_conditional:Npnn #2 ##1##2##3 { T , F , TF }
      { #1 { \__regex_build:n {##1} } {##2} ##3 \__regex_return: }
    \prg_new_protected_conditional:Npnn #3 ##1##2##3 { T , F , TF }
      { #1 { \__regex_build:N  ##1  } {##2} ##3 \__regex_return: }
    \cs_generate_variant:Nn #2 { nV }
    \prg_generate_conditional_variant:Nnn #2 { nV } { T , F , TF }
    \cs_generate_variant:Nn #3 { NV }
    \prg_generate_conditional_variant:Nnn #3 { NV } { T , F , TF }
  }
\__regex_tmp:w \__regex_extract_once:nnN
  \regex_extract_once:nnN \regex_extract_once:NnN
```

```
\__regex_tmp:w \__regex_extract_all:nnN
  \regex_extract_all:nnN \regex_extract_all:NnN
\__regex_tmp:w \__regex_replace_once:nnN
  \regex_replace_once:nnN \regex_replace_once:NnN
\__regex_tmp:w \__regex_replace_all:nnN
  \regex_replace_all:nnN \regex_replace_all:NnN
\__regex_tmp:w \__regex_split:nnN \regex_split:nnN \regex_split:NnN
\cs_new_protected:Npn \regex_replace_case_once:nNTF #1#2
  {
    \int_if_odd:nTF { \tl_count:n {#1} }
      {
        \msg_error:nneeee { regex } { case-odd }
          { \token_to_str:N \regex_replace_case_once:nN(TF) } { code }
          { \tl_count:n {#1} } { \tl_to_str:n {#1} }
        \use_ii:nn
      }
      {
        \__regex_replace_once_aux:nnN
          { \__regex_case_build:e { \__regex_tl_odd_items:n {#1} } }
          { \__regex_replacement:e { \tl_item:nn {#1} { 2 * \g__regex_case_int } } }
          #2
        \bool_if:NTF \g__regex_success_bool
      }
  }
\cs_new_protected:Npn \regex_replace_case_once:nN #1#2
  { \regex_replace_case_once:nNTF {#1} {#2} { } { } }
\cs_new_protected:Npn \regex_replace_case_once:nNT #1#2#3
  { \regex_replace_case_once:nNTF {#1} {#2} {#3} { } }
\cs_new_protected:Npn \regex_replace_case_once:nNF #1#2
  { \regex_replace_case_once:nNTF {#1} {#2} { } }
\cs_new_protected:Npn \regex_replace_case_all:nNTF #1#2
  {
    \int_if_odd:nTF { \tl_count:n {#1} }
      {
        \msg_error:nneeee { regex } { case-odd }
          { \token_to_str:N \regex_replace_case_all:nN(TF) } { code }
          { \tl_count:n {#1} } { \tl_to_str:n {#1} }
        \use_ii:nn
      }
      {
        \__regex_replace_all_aux:nnN
          { \__regex_case_build:e { \__regex_tl_odd_items:n {#1} } }
          { \__regex_case_replacement:e { \__regex_tl_even_items:n {#1} } }
          #2
        \bool_if:NTF \g__regex_success_bool
      }
  }
\cs_new_protected:Npn \regex_replace_case_all:nN #1#2
  { \regex_replace_case_all:nNTF {#1} {#2} { } { } }
\cs_new_protected:Npn \regex_replace_case_all:nNT #1#2#3
  { \regex_replace_case_all:nNTF {#1} {#2} {#3} { } }
\cs_new_protected:Npn \regex_replace_case_all:nNF #1#2
  { \regex_replace_case_all:nNTF {#1} {#2} { } }
\int_new:N \l__regex_match_count_int
\flag_new:N \l__regex_begin_flag
\flag_new:N \l__regex_end_flag
\int_new:N \l__regex_min_submatch_int
\int_new:N \l__regex_submatch_int
\int_new:N \l__regex_zeroth_submatch_int
\intarray_new:Nn \g__regex_submatch_prev_intarray { 65536 }
\intarray_new:Nn \g__regex_submatch_begin_intarray { 65536 }
\intarray_new:Nn \g__regex_submatch_end_intarray { 65536 }
\intarray_new:Nn \g__regex_submatch_case_intarray { 65536 }
\intarray_new:Nn \g__regex_balance_intarray { 65536 }
\int_new:N \l__regex_added_begin_int
\int_new:N \l__regex_added_end_int
\cs_new_protected:Npn \__regex_return:
  {
    \if_meaning:w \c_true_bool \g__regex_success_bool
      \prg_return_true:
    \else:
      \prg_return_false:
    \fi:
  }
\cs_new_protected:Npn \__regex_query_set:n #1
  {
    \int_zero:N \l__regex_balance_int
    \int_zero:N \l__regex_curr_pos_int
    \__regex_query_set_aux:nN { } F
    \tl_analysis_map_inline:nn {#1}
      { \__regex_query_set_aux:nN {##1} ##3 }
    \__regex_query_set_aux:nN { } F
    \int_set_eq:NN \l__regex_max_pos_int \l__regex_curr_pos_int
  }
\cs_new_protected:Npn \__regex_query_set_aux:nN #1#2
  {
    \int_incr:N \l__regex_curr_pos_int
    \__regex_toks_set:Nn \l__regex_curr_pos_int {#1}
    \__kernel_intarray_gset:Nnn \g__regex_balance_intarray
      \l__regex_curr_pos_int \l__regex_balance_int
    \if_case:w "#2 \exp_stop_f:
    \or: \int_incr:N \l__regex_balance_int
    \or: \int_decr:N \l__regex_balance_int
    \fi:
  }
\cs_new_protected:Npn \__regex_if_match:nn #1#2
  {
    \group_begin:
      \__regex_disable_submatches:
      \__regex_single_match:
      #1
      \__regex_match:n {#2}
    \group_end:
  }
\cs_new_protected:Npn \__regex_match_case:nnTF #1#2
  {
    \int_if_odd:nTF { \tl_count:n {#1} }
      {
        \msg_error:nneeee { regex } { case-odd }
          { \token_to_str:N \regex_match_case:nn(TF) } { code }
          { \tl_count:n {#1} } { \tl_to_str:n {#1} }
        \use_ii:nn
      }
      {
        \__regex_if_match:nn
          { \__regex_case_build:e { \__regex_tl_odd_items:n {#1} } }
          {#2}
        \bool_if:NTF \g__regex_success_bool
      }
  }
\cs_new:Npn \__regex_match_case_aux:nn #1#2 { \exp_not:n { {#1} } }
\cs_new_protected:Npn \__regex_count:nnN #1#2#3
  {
    \group_begin:
      \__regex_disable_submatches:
      \int_zero:N \l__regex_match_count_int
      \__regex_multi_match:n { \int_incr:N \l__regex_match_count_int }
      #1
      \__regex_match:n {#2}
      \exp_args:NNNo
    \group_end:
    \int_set:Nn #3 { \int_use:N \l__regex_match_count_int }
  }
\cs_new_protected:Npn \__regex_extract_once:nnN #1#2#3
  {
    \group_begin:
      \__regex_single_match:
      #1
      \__regex_match:n {#2}
      \__regex_extract:
      \__regex_query_set:n {#2}
    \__regex_group_end_extract_seq:N #3
  }
\cs_new_protected:Npn \__regex_extract_all:nnN #1#2#3
  {
    \group_begin:
      \__regex_multi_match:n { \__regex_extract: }
      #1
      \__regex_match:n {#2}
      \__regex_query_set:n {#2}
    \__regex_group_end_extract_seq:N #3
  }
\cs_new_protected:Npn \__regex_split:nnN #1#2#3
  {
    \group_begin:
      \__regex_multi_match:n
        {
          \if_int_compare:w
            \l__regex_start_pos_int < \l__regex_success_pos_int
            \__regex_extract:
            \__kernel_intarray_gset:Nnn \g__regex_submatch_prev_intarray
              \l__regex_zeroth_submatch_int \c_zero_int
            \__kernel_intarray_gset:Nnn \g__regex_submatch_end_intarray
              \l__regex_zeroth_submatch_int
              {
                \__kernel_intarray_item:Nn \g__regex_submatch_begin_intarray
                  \l__regex_zeroth_submatch_int
              }
            \__kernel_intarray_gset:Nnn \g__regex_submatch_begin_intarray
              \l__regex_zeroth_submatch_int
              \l__regex_start_pos_int
          \fi:
        }
      #1
      \__regex_match:n {#2}
      \__regex_query_set:n {#2}
      \__kernel_intarray_gset:Nnn \g__regex_submatch_prev_intarray
        \l__regex_submatch_int \c_zero_int
      \__kernel_intarray_gset:Nnn \g__regex_submatch_end_intarray
        \l__regex_submatch_int
        \l__regex_max_pos_int
      \__kernel_intarray_gset:Nnn \g__regex_submatch_begin_intarray
        \l__regex_submatch_int
        \l__regex_start_pos_int
      \int_incr:N \l__regex_submatch_int
      \if_meaning:w \c_true_bool \l__regex_empty_success_bool
        \if_int_compare:w \l__regex_start_pos_int = \l__regex_max_pos_int
          \int_decr:N \l__regex_submatch_int
        \fi:
      \fi:
    \__regex_group_end_extract_seq:N #3
  }
\cs_new_protected:Npn \__regex_group_end_extract_seq:N #1
  {
      \flag_clear:N \l__regex_begin_flag
      \flag_clear:N \l__regex_end_flag
      \cs_set_eq:NN \__regex_tmp:w \scan_stop:
      \__kernel_tl_gset:Nx \g__regex_internal_tl
        {
          \int_step_function:nnN \l__regex_min_submatch_int
            { \l__regex_submatch_int - \c_one_int } \__regex_extract_seq_aux:n
          \__regex_tmp:w
        }
      \int_set:Nn \l__regex_added_begin_int
        { \flag_height:N \l__regex_begin_flag }
      \int_set:Nn \l__regex_added_end_int
        { \flag_height:N \l__regex_end_flag }
      \tex_afterassignment:D \__regex_extract_check:w
      \__kernel_tl_gset:Nx \g__regex_internal_tl
        { \g__regex_internal_tl \if_false: { \fi: } }
      \int_compare:nNnT
        { \l__regex_added_begin_int + \l__regex_added_end_int } > \c_zero_int
        {
          \msg_error:nneee { regex } { result-unbalanced }
            { splitting~or~extracting~submatches }
            { \int_use:N \l__regex_added_begin_int }
            { \int_use:N \l__regex_added_end_int }
        }
    \group_end:
    \__regex_extract_seq:N #1
  }
\cs_gset_protected:Npn \__regex_extract_seq:N #1
  {
    \seq_clear:N #1
    \cs_set_eq:NN \__regex_tmp:w  \__regex_extract_seq_loop:Nw
    \exp_after:wN \__regex_extract_seq:NNn
    \exp_after:wN #1
    \g__regex_internal_tl \use_none:nnn
  }
\cs_new_protected:Npn \__regex_extract_seq:NNn #1#2#3
```

```
  { #3 #2 #1 \prg_do_nothing: }
\cs_new_protected:Npn \__regex_extract_seq_loop:Nw #1#2 \__regex_tmp:w #3
  {
    \seq_put_right:No #1 {#2}
    #3 \__regex_extract_seq_loop:Nw #1 \prg_do_nothing:
  }
\cs_new:Npn \__regex_extract_seq_aux:n #1
  {
    \__regex_tmp:w { }
    \exp_after:wN \__regex_extract_seq_aux:ww
    \int_value:w \__regex_submatch_balance:n {#1} \__regex_sep: #1 \__regex_sep:
  }
\cs_new:Npn \__regex_extract_seq_aux:ww #1 \__regex_sep: #2 \__regex_sep:
  {
    \if_int_compare:w #1 < \c_zero_int
      \prg_replicate:nn {-#1}
        {
          \flag_raise:N \l__regex_begin_flag
          \exp_not:n { { \if_false: } \fi: }
        }
    \fi:
    \__regex_query_submatch:n {#2}
    \if_int_compare:w #1 > \c_zero_int
      \prg_replicate:nn {#1}
        {
          \flag_raise:N \l__regex_end_flag
          \exp_not:n { \if_false: { \fi: } }
        }
    \fi:
  }
\cs_new_protected:Npn \__regex_extract_check:w
  {
    \exp_after:wN \__regex_extract_check:n
    \exp_after:wN { \if_false: } \fi:
  }
\cs_new_protected:Npn \__regex_extract_check:n #1
  {
    \tl_if_empty:nF {#1}
      {
        \int_incr:N \l__regex_added_begin_int
        \int_incr:N \l__regex_added_end_int
        \tex_afterassignment:D \__regex_extract_check:w
        \__kernel_tl_gset:Nx \g__regex_internal_tl
          {
            \exp_after:wN \__regex_extract_check_loop:w
            \g__regex_internal_tl
            \__regex_tmp:w \__regex_extract_check_end:w
            #1
          }
      }
  }
\cs_new:Npn \__regex_extract_check_loop:w #1 \__regex_tmp:w #2
  {
    #2
    \exp_not:o {#1}
    \__regex_tmp:w { }
    \__regex_extract_check_loop:w \prg_do_nothing:
  }
\cs_new:Npn \__regex_extract_check_end:w
    \exp_not:o #1#2 \__regex_extract_check_loop:w #3 \__regex_tmp:w
  {
    { \exp_not:o {#1} }
    #3
    \if_false: { \fi: }
    \__regex_tmp:w
  }
\cs_new_protected:Npn \__regex_extract:
  {
    \if_meaning:w \c_true_bool \g__regex_success_bool
      \int_set_eq:NN \l__regex_zeroth_submatch_int \l__regex_submatch_int
      \prg_replicate:nn \l__regex_capturing_group_int
        {
          \__kernel_intarray_gset:Nnn \g__regex_submatch_prev_intarray
            \l__regex_submatch_int \c_zero_int
          \__kernel_intarray_gset:Nnn \g__regex_submatch_case_intarray
            \l__regex_submatch_int \c_zero_int
          \int_incr:N \l__regex_submatch_int
        }
      \__kernel_intarray_gset:Nnn \g__regex_submatch_prev_intarray
        \l__regex_zeroth_submatch_int \l__regex_start_pos_int
      \__kernel_intarray_gset:Nnn \g__regex_submatch_case_intarray
        \l__regex_zeroth_submatch_int \g__regex_case_int
      \int_zero:N \l__regex_internal_a_int
      \exp_after:wN \__regex_extract_aux:w \l__regex_success_submatches_tl
        \prg_break_point: \__regex_use_none_delimit_by_q_recursion_stop:w ,
        \q__regex_recursion_stop
    \fi:
  }
\cs_new_protected:Npn \__regex_extract_aux:w #1 ,
  {
    \prg_break: #1 \prg_break_point:
    \if_int_compare:w \l__regex_internal_a_int < \l__regex_capturing_group_int
```

```
      \__kernel_intarray_gset:Nnn \g__regex_submatch_begin_intarray
        { \__regex_int_eval:w \l__regex_zeroth_submatch_int + \l__regex_internal_a_int } {#1}
    \else:
      \__kernel_intarray_gset:Nnn \g__regex_submatch_end_intarray
        {
          \__regex_int_eval:w
            \l__regex_zeroth_submatch_int + \l__regex_internal_a_int
            - \l__regex_capturing_group_int
        }
        {#1}
    \fi:
    \int_incr:N \l__regex_internal_a_int
    \__regex_extract_aux:w
  }
\cs_new_protected:Npn \__regex_replace_once:nnN #1#2
  { \__regex_replace_once_aux:nnN {#1} { \__regex_replacement:n {#2} } }
\cs_new_protected:Npn \__regex_replace_once_aux:nnN #1#2#3
  {
    \group_begin:
      \__regex_single_match:
      #1
      \exp_args:No \__regex_match:n {#3}
    \bool_if:NTF \g__regex_success_bool
      {
        \__regex_extract:
        \exp_args:No \__regex_query_set:n {#3}
        #2
        \int_set:Nn \l__regex_balance_int
          { \__regex_replacement_balance_one_match:n \l__regex_zeroth_submatch_int }
        \__kernel_tl_set:Nx \l__regex_internal_a_tl
          {
            \__regex_replacement_do_one_match:n \l__regex_zeroth_submatch_int
            \__regex_query_range:nn
              {
                \__kernel_intarray_item:Nn \g__regex_submatch_end_intarray
                  \l__regex_zeroth_submatch_int
              }
              \l__regex_max_pos_int
          }
        \__regex_group_end_replace:N #3
      }
      { \group_end: }
  }
\cs_new_protected:Npn \__regex_replace_all:nnN #1#2
  { \__regex_replace_all_aux:nnN {#1} { \__regex_replacement:n {#2} } }
```

```
\cs_new_protected:Npn \__regex_replace_all_aux:nnN #1#2#3
  {
    \group_begin:
      \__regex_multi_match:n { \__regex_extract: }
      #1
      \exp_args:No \__regex_match:n {#3}
      \exp_args:No \__regex_query_set:n {#3}
      #2
      \int_set:Nn \l__regex_balance_int
        {
          \c_zero_int
          \int_step_function:nnnN
            \l__regex_min_submatch_int
            \l__regex_capturing_group_int
            { \l__regex_submatch_int - \c_one_int }
            \__regex_replacement_balance_one_match:n
        }
      \__kernel_tl_set:Nx \l__regex_internal_a_tl
        {
          \int_step_function:nnnN
            \l__regex_min_submatch_int
            \l__regex_capturing_group_int
            { \l__regex_submatch_int - \c_one_int }
            \__regex_replacement_do_one_match:n
          \__regex_query_range:nn
            \l__regex_start_pos_int \l__regex_max_pos_int
        }
    \__regex_group_end_replace:N #3
  }
\cs_new_protected:Npn \__regex_group_end_replace:N #1
  {
    \int_set:Nn \l__regex_added_begin_int
      { \int_max:nn { - \l__regex_balance_int } \c_zero_int }
    \int_set:Nn \l__regex_added_end_int
      { \int_max:nn \l__regex_balance_int \c_zero_int }
    \__regex_group_end_replace_try:
    \int_compare:nNnT { \l__regex_added_begin_int + \l__regex_added_end_int }
                      > \c_zero_int
      {
        \msg_error:nneee { regex } { result-unbalanced }
          { replacing } { \int_use:N \l__regex_added_begin_int }
          { \int_use:N \l__regex_added_end_int }
      }
    \group_end:
    \tl_set_eq:NN #1 \g__regex_internal_tl
  }
```

```
\cs_new_protected:Npn \__regex_group_end_replace_try:
  {
    \tex_afterassignment:D \__regex_group_end_replace_check:w
    \__kernel_tl_gset:Nx \g__regex_internal_tl
      {
        \prg_replicate:nn \l__regex_added_begin_int { { \if_false: } \fi: }
        \l__regex_internal_a_tl
        \prg_replicate:nn \l__regex_added_end_int { \if_false: { \fi: } }
        \if_false: { \fi: }
      }
  }
\cs_new_protected:Npn \__regex_group_end_replace_check:w
  {
    \exp_after:wN \__regex_group_end_replace_check:n
    \exp_after:wN { \if_false: } \fi:
  }
\cs_new_protected:Npn \__regex_group_end_replace_check:n #1
  {
    \tl_if_empty:nF {#1}
      {
        \int_incr:N \l__regex_added_begin_int
        \int_incr:N \l__regex_added_end_int
        \__regex_group_end_replace_try:
      }
  }
\tl_new:N \l__regex_peek_true_tl
\tl_new:N \l__regex_peek_false_tl
\tl_new:N \l__regex_replacement_tl
\tl_new:N \l__regex_input_tl
\cs_new_eq:NN \__regex_input_item:n ?
\cs_new_protected:Npn \peek_regex:nTF #1
  {
    \__regex_peek:nnTF
      { \__regex_build_aux:Nn \c_false_bool {#1} }
      { \__regex_peek_end: }
  }
\cs_new_protected:Npn \peek_regex:nT #1#2
  { \peek_regex:nTF {#1} {#2} { } }
\cs_new_protected:Npn \peek_regex:nF #1 { \peek_regex:nTF {#1} { } }
\cs_new_protected:Npn \peek_regex:NTF #1
  {
    \__regex_peek:nnTF
      { \__regex_build_aux:NN \c_false_bool #1 }
      { \__regex_peek_end: }
  }
\cs_new_protected:Npn \peek_regex:NT #1#2
```

```
  { \peek_regex:NTF #1 {#2} { } }
\cs_new_protected:Npn \peek_regex:NF #1 { \peek_regex:NTF {#1} { } }
\cs_new_protected:Npn \peek_regex_remove_once:nTF #1
  {
    \__regex_peek:nnTF
      { \__regex_build_aux:Nn \c_false_bool {#1} }
      { \__regex_peek_remove_end:n {##1} }
  }
\cs_new_protected:Npn \peek_regex_remove_once:nT #1#2
  { \peek_regex_remove_once:nTF {#1} {#2} { } }
\cs_new_protected:Npn \peek_regex_remove_once:nF #1
  { \peek_regex_remove_once:nTF {#1} { } }
\cs_new_protected:Npn \peek_regex_remove_once:NTF #1
  {
    \__regex_peek:nnTF
      { \__regex_build_aux:NN \c_false_bool #1 }
      { \__regex_peek_remove_end:n {##1} }
  }
\cs_new_protected:Npn \peek_regex_remove_once:NT #1#2
  { \peek_regex_remove_once:NTF #1 {#2} { } }
\cs_new_protected:Npn \peek_regex_remove_once:NF #1
  { \peek_regex_remove_once:NTF #1 { } }
\cs_new_protected:Npn \__regex_peek:nnTF #1
  {
    \__regex_peek_aux:nnTF
      {
        \__regex_disable_submatches:
        #1
      }
  }
\cs_new_protected:Npn \__regex_peek_aux:nnTF #1#2#3#4
  {
    \group_begin:
      \tl_set:Nn \l__regex_peek_true_tl { \group_end: #3 }
      \tl_set:Nn \l__regex_peek_false_tl { \group_end: #4 }
      \__regex_single_match:
      #1
      \__regex_match_init:
      \tl_build_begin:N \l__regex_input_tl
      \__regex_match_once_init:
      \peek_analysis_map_inline:n
        {
          \tl_build_put_right:Nn \l__regex_input_tl
            { \__regex_input_item:n {##1} }
          \__regex_match_one_token:nnN {##1} {##2} ##3
          \use_none:nnn
          \prg_break_point:Nn \__regex_maplike_break:
            { \peek_analysis_map_break:n {#2} }
        }
  }
\cs_new_protected:Npn \__regex_peek_end:
  {
    \bool_if:NTF \g__regex_success_bool
      { \__regex_peek_reinsert:N \l__regex_peek_true_tl }
      { \__regex_peek_reinsert:N \l__regex_peek_false_tl }
  }
\cs_new_protected:Npn \__regex_peek_remove_end:n #1
  {
    \bool_if:NTF \g__regex_success_bool
      { \exp_args:NNo \use:nn \l__regex_peek_true_tl {#1} }
      { \__regex_peek_reinsert:N \l__regex_peek_false_tl }
  }
\cs_new_protected:Npn \__regex_peek_reinsert:N #1
  {
    \tl_build_end:N \l__regex_input_tl
    \cs_set_eq:NN \__regex_input_item:n \__regex_reinsert_item:n
    \exp_after:wN #1 \exp:w \l__regex_input_tl \exp_end:
  }
\cs_new_protected:Npn \__regex_reinsert_item:n #1
  {
    \exp_after:wN \exp_after:wN
    \exp_after:wN \exp_end:
    \exp_after:wN \exp_after:wN
    #1
    \exp:w
  }
\cs_new_protected:Npn \peek_regex_replace_once:nnTF #1
  { \__regex_peek_replace:nnTF { \__regex_build_aux:Nn \c_false_bool {#1} } }
\cs_new_protected:Npn \peek_regex_replace_once:nnT #1#2#3
  { \peek_regex_replace_once:nnTF {#1} {#2} {#3} { } }
\cs_new_protected:Npn \peek_regex_replace_once:nnF #1#2
  { \peek_regex_replace_once:nnTF {#1} {#2} { } }
\cs_new_protected:Npn \peek_regex_replace_once:nn #1#2
  { \peek_regex_replace_once:nnTF {#1} {#2} { } { } }
\cs_new_protected:Npn \peek_regex_replace_once:NnTF #1
  { \__regex_peek_replace:nnTF { \__regex_build_aux:NN \c_false_bool #1 } }
\cs_new_protected:Npn \peek_regex_replace_once:NnT #1#2#3
  { \peek_regex_replace_once:NnTF #1 {#2} {#3} { } }
\cs_new_protected:Npn \peek_regex_replace_once:NnF #1#2
  { \peek_regex_replace_once:NnTF #1 {#2} { } }
\cs_new_protected:Npn \peek_regex_replace_once:Nn #1#2
  { \peek_regex_replace_once:NnTF #1 {#2} { } { } }
```

```
\cs_new_protected:Npn \__regex_peek_replace:nnTF #1#2
  {
    \tl_set:Nn \l__regex_replacement_tl {#2}
    \__regex_peek_aux:nnTF {#1} { \__regex_peek_replace_end: }
  }
\cs_new_protected:Npn \__regex_peek_replace_end:
  {
    \bool_if:NTF \g__regex_success_bool
      {
        \__regex_extract:
        \__regex_query_set_from_input_tl:
        \cs_set_eq:NN \__regex_replacement_put:n \__regex_peek_replacement_put:n
        \cs_set_eq:NN \__regex_replacement_put_submatch_aux:n
          \__regex_peek_replacement_put_submatch_aux:n
        \cs_set_eq:NN \__regex_input_item:n \__regex_reinsert_item:n
        \cs_set_eq:NN \__regex_replacement_exp_not:N \__regex_peek_replacement_token:n
        \cs_set_eq:NN \__regex_replacement_exp_not:V \__regex_peek_replacement_var:N
        \exp_args:No \__regex_replacement:n { \l__regex_replacement_tl }
        \use:e
          {
            \exp_not:n { \exp_after:wN \l__regex_peek_true_tl \exp:w }
            \__regex_replacement_do_one_match:n \l__regex_zeroth_submatch_int
            \__regex_query_range:nn
              {
                \__kernel_intarray_item:Nn \g__regex_submatch_end_intarray
                  \l__regex_zeroth_submatch_int
              }
              \l__regex_max_pos_int
            \exp_end:
          }
      }
      { \__regex_peek_reinsert:N \l__regex_peek_false_tl }
  }
\cs_new_protected:Npn \__regex_query_set_from_input_tl:
  {
    \tl_build_end:N \l__regex_input_tl
    \int_zero:N \l__regex_curr_pos_int
    \cs_set_eq:NN \__regex_input_item:n \__regex_query_set_item:n
    \__regex_query_set_item:n { }
    \l__regex_input_tl
    \__regex_query_set_item:n { }
    \int_set_eq:NN \l__regex_max_pos_int \l__regex_curr_pos_int
  }
\cs_new_protected:Npn \__regex_query_set_item:n #1
  {
    \int_incr:N \l__regex_curr_pos_int
    \__regex_toks_set:Nn \l__regex_curr_pos_int { \__regex_input_item:n {#1} }
  }
\cs_new_protected:Npn \__regex_peek_replacement_put:n #1
  {
    \if_case:w \l__regex_replacement_csnames_int
      \tl_build_put_right:Nn \l__regex_build_tl
        { \exp_not:N \__regex_reinsert_item:n {#1} }
    \else:
      \tl_build_put_right:Nn \l__regex_build_tl {#1}
    \fi:
  }
\cs_new_protected:Npn \__regex_peek_replacement_token:n #1
  { \exp_after:wN \exp_end: \exp_after:wN #1 \exp:w }
\cs_new_protected:Npn \__regex_peek_replacement_put_submatch_aux:n #1
  {
    \if_case:w \l__regex_replacement_csnames_int
      \tl_build_put_right:Nn \l__regex_build_tl
        { \__regex_query_submatch:n { \__regex_int_eval:w #1 + ##1 \scan_stop: } }
    \else:
      \tl_build_put_right:Nn \l__regex_build_tl
        {
          \exp:w
            \__regex_query_submatch:n { \__regex_int_eval:w #1 + ##1 \scan_stop: }
          \exp_end:
        }
    \fi:
  }
\cs_new_protected:Npn \__regex_peek_replacement_var:N #1
  {
    \exp_after:wN \exp_last_unbraced:NV
    \exp_after:wN \exp_end:
    \exp_after:wN #1
    \exp:w
  }
\use:e
  {
    \msg_new:nnn { regex } { trailing-backslash }
      { Trailing~'\iow_char:N\\'~in~regex~or~replacement. }
    \msg_new:nnn { regex } { x-missing-rbrace }
      {
        Missing~brace~'\iow_char:N\}'~in~regex~
        '...\iow_char:N\\x\iow_char:N\{...##1'.
      }
    \msg_new:nnn { regex } { x-overflow }
      {
        Character~code~##1~too~large~in~
        \iow_char:N\\x\iow_char:N\{##2\iow_char:N\}~regex.
      }
  }
\msg_new:nnnn { regex } { invalid-quantifier }
  { Braced~quantifier~'#1'~may~not~be~followed~by~'#2'. }
  {
    The~character~'#2'~is~invalid~in~the~braced~quantifier~'#1'.~
    The~only~valid~quantifiers~are~'*',~'?',~'+',~'{<int>}',~
    '{<min>,}'~and~'{<min>,<max>}',~optionally~followed~by~'?'.
  }
\msg_new:nnnn { regex } { missing-rbrack }
  { Missing~right~bracket~inserted~in~regular~expression. }
  {
    LaTeX~was~given~a~regular~expression~where~a~character~class~
    was~started~with~'[',~but~the~matching~']'~is~missing.
  }
\msg_new:nnnn { regex } { missing-rparen }
  {
    Missing~right~
    \int_compare:nTF { #1 = 1 } { parenthesis } { parentheses } ~
    inserted~in~regular~expression.
  }
  {
    LaTeX~was~given~a~regular~expression~with~\int_eval:n {#1} ~
    more~left~parentheses~than~right~parentheses.
  }
\msg_new:nnnn { regex } { extra-rparen }
  { Extra~right~parenthesis~ignored~in~regular~expression. }
  {
    LaTeX~came~across~a~closing~parenthesis~when~no~submatch~group~
    was~open.~The~parenthesis~will~be~ignored.
  }
\msg_new:nnnn { regex } { bad-escape }
  {
    Invalid~escape~'\iow_char:N\\#1'~
    \__regex_if_in_cs:TF { within~a~control~sequence. }
      {
        \__regex_if_in_class:TF
          { in~a~character~class. }
          { following~a~category~test. }
      }
  }
  {
    The~escape~sequence~'\iow_char:N\\#1'~may~not~appear~
    \__regex_if_in_cs:TF
      {
        within~a~control~sequence~test~introduced~by~
        '\iow_char:N\\c\iow_char:N\{'.
      }
      {
        \__regex_if_in_class:TF
          { within~a~character~class~ }
          { following~a~category~test~such~as~'\iow_char:N\\cL'~ }
        because~it~does~not~match~exactly~one~character.
      }
  }
\msg_new:nnnn { regex } { range-missing-end }
  { Invalid~end-point~for~range~'#1-#2'~in~character~class. }
  {
    The~end-point~'#2'~of~the~range~'#1-#2'~may~not~serve~as~an~
    end-point~for~a~range:~alphanumeric~characters~should~not~be~
    escaped,~and~non-alphanumeric~characters~should~be~escaped.
  }
\msg_new:nnnn { regex } { range-backwards }
  { Range~'[#1-#2]'~out~of~order~in~character~class. }
  {
    In~ranges~of~characters~'[x-y]'~appearing~in~character~classes,~
    the~first~character~code~must~not~be~larger~than~the~second.~
    Here,~'#1'~has~character~code~\int_eval:n {`#1},~while~
    '#2'~has~character~code~\int_eval:n {`#2}.
  }
\msg_new:nnnn { regex } { c-bad-mode }
  { Invalid~nested~'\iow_char:N\\c'~escape~in~regular~expression. }
  {
    The~'\iow_char:N\\c'~escape~cannot~be~used~within~
    a~control~sequence~test~'\iow_char:N\\c{...}'~
    nor~another~category~test.~
    To~combine~several~category~tests,~use~'\iow_char:N\\c[...]'.
  }
\msg_new:nnnn { regex } { c-C-invalid }
  { '\iow_char:N\\cC'~should~be~followed~by~'.'~or~'(',~not~'#1'. }
  {
    The~'\iow_char:N\\cC'~construction~restricts~the~next~item~to~be~a~
    control~sequence~or~the~next~group~to~be~made~of~control~sequences.~
    It~only~makes~sense~to~follow~it~by~'.'~or~by~a~group.
  }
\msg_new:nnnn { regex } { cu-lbrace }
  { Left~braces~must~be~escaped~in~'\iow_char:N\\#1{...}'. }
  {
    Constructions~such~as~'\iow_char:N\\#1{...\iow_char:N\{...}'~are~
    not~allowed~and~should~be~replaced~by~
    '\iow_char:N\\#1{...\token_to_str:N\{...}'.
  }
\msg_new:nnnn { regex } { c-lparen-in-class }
  { Catcode~test~cannot~apply~to~group~in~character~class }
  {
    Construction~such~as~'\iow_char:N\\cL(abc)'~are~not~allowed~inside~a~
    class~'[...]'~because~classes~do~not~match~multiple~characters~at~once.
  }
\msg_new:nnnn { regex } { c-missing-rbrace }
  { Missing~right~brace~inserted~for~'\iow_char:N\\c'~escape. }
  {
    LaTeX~was~given~a~regular~expression~where~a~
    '\iow_char:N\\c\iow_char:N\{...'~construction~was~not~ended~
    with~a~closing~brace~'\iow_char:N\}'.
  }
\msg_new:nnnn { regex } { c-missing-rbrack }
  { Missing~right~bracket~inserted~for~'\iow_char:N\\c'~escape. }
  {
    A~construction~'\iow_char:N\\c[...'~appears~in~a~
    regular~expression,~but~the~closing~']'~is~not~present.
  }
\msg_new:nnnn { regex } { c-missing-category }
  { Invalid~character~'#1'~following~'\iow_char:N\\c'~escape. }
  {
    In~regular~expressions,~the~'\iow_char:N\\c'~escape~sequence~
    may~only~be~followed~by~a~left~brace,~a~left~bracket,~or~a~
    capital~letter~representing~a~character~category,~namely~
    one~of~'ABCDELMOPSTU'.
  }
\msg_new:nnnn { regex } { c-trailing }
  { Trailing~category~code~escape~'\iow_char:N\\c'... }
  {
    A~regular~expression~ends~with~'\iow_char:N\\c'~followed~
    by~a~letter.~It~will~be~ignored.
  }
\msg_new:nnnn { regex } { u-missing-lbrace }
  { Missing~left~brace~following~'\iow_char:N\\u'~escape. }
  {
    The~'\iow_char:N\\u'~escape~sequence~must~be~followed~by~
    a~brace~group~with~the~name~of~the~variable~to~use.
  }
\msg_new:nnnn { regex } { u-missing-rbrace }
  { Missing~right~brace~inserted~for~'\iow_char:N\\u'~escape. }
  {
    LaTeX~
    \str_if_eq:eeTF { } {#2}
      { reached~the~end~of~the~string~ }
      { encountered~an~escaped~alphanumeric~character '\iow_char:N\\#2'~ }
    when~parsing~the~argument~of~an~
    '\iow_char:N\\u\iow_char:N\{...\}'~escape.
  }
\msg_new:nnnn { regex } { posix-unsupported }
  { POSIX~collating~element~'[#1 ~ #1]'~not~supported. }
  {
    The~'[.foo.]'~and~'[=bar=]'~syntaxes~have~a~special~meaning~
    in~POSIX~regular~expressions.~This~is~not~supported~by~LaTeX.~
    Maybe~you~forgot~to~escape~a~left~bracket~in~a~character~class?
  }
\msg_new:nnnn { regex } { posix-unknown }
  { POSIX~class~'[:#1:]'~unknown. }
  {
    '[:#1:]'~is~not~among~the~known~POSIX~classes~
    '[:alnum:]',~'[:alpha:]',~'[:ascii:]',~'[:blank:]',~
    '[:cntrl:]',~'[:digit:]',~'[:graph:]',~'[:lower:]',~
    '[:print:]',~'[:punct:]',~'[:space:]',~'[:upper:]',~
    '[:word:]',~and~'[:xdigit:]'.
  }
\msg_new:nnnn { regex } { posix-missing-close }
  { Missing~closing~':]'~for~POSIX~class. }
  { The~POSIX~syntax~'#1'~must~be~followed~by~':]',~not~'#2'. }
\msg_new:nnnn { regex } { result-unbalanced }
  { Missing~brace~inserted~when~#1. }
  {
    LaTeX~was~asked~to~do~some~regular~expression~operation,~
    and~the~resulting~token~list~would~not~have~the~same~number~
    of~begin-group~and~end-group~tokens.~Braces~were~inserted:~
    #2~left,~#3~right.
  }
\msg_new:nnnn { regex } { unknown-option }
  { Unknown~option~'#1'~for~regular~expressions. }
  {
    The~only~available~option~is~'case-insensitive',~toggled~by~
    '(?i)'~and~'(?-i)'.
  }
\msg_new:nnnn { regex } { special-group-unknown }
  { Unknown~special~group~'#1~...'~in~a~regular~expression. }
  {
    The~only~valid~constructions~starting~with~'(?'~are~
    '(?:~...~)',~'(?|~...~)',~'(?i)',~and~'(?-i)'.
  }
\msg_new:nnnn { regex } { replacement-c }
  { Misused~'\iow_char:N\\c'~command~in~a~replacement~text. }
  {
    In~a~replacement~text,~the~'\iow_char:N\\c'~escape~sequence~
    can~be~followed~by~one~of~the~letters~'ABCDELMOPSTU'~
    or~a~brace~group,~not~by~'#1'.
  }
\msg_new:nnnn { regex } { replacement-u }
  { Misused~'\iow_char:N\\u'~command~in~a~replacement~text. }
  {
    In~a~replacement~text,~the~'\iow_char:N\\u'~escape~sequence~
    must~be~~followed~by~a~brace~group~holding~the~name~of~the~
    variable~to~use.
  }
\msg_new:nnnn { regex } { replacement-g }
  {
    Missing~brace~for~the~'\iow_char:N\\g'~construction~
    in~a~replacement~text.
  }
  {
    In~the~replacement~text~for~a~regular~expression~search,~
    submatches~are~represented~either~as~'\iow_char:N \\g{dd..d}',~
    or~'\\d',~where~'d'~are~single~digits.~Here,~a~brace~is~missing.
  }
\msg_new:nnnn { regex } { replacement-catcode-end }
  {
    Missing~character~for~the~'\iow_char:N\\c<category><character>'~
    construction~in~a~replacement~text.
  }
  {
    In~a~replacement~text,~the~'\iow_char:N\\c'~escape~sequence~
    can~be~followed~by~one~of~the~letters~'ABCDELMOPSTU'~representing~
    the~character~category.~Then,~a~character~must~follow.~LaTeX~
    reached~the~end~of~the~replacement~when~looking~for~that.
  }
\msg_new:nnnn { regex } { replacement-catcode-escaped }
  {
    Escaped~letter~or~digit~after~category~code~in~replacement~text.
  }
  {
    In~a~replacement~text,~the~'\iow_char:N\\c'~escape~sequence~
    can~be~followed~by~one~of~the~letters~'ABCDELMOPSTU'~representing~
    the~character~category.~Then,~a~character~must~follow,~not~
    '\iow_char:N\\#2'.
  }
\msg_new:nnnn { regex } { replacement-catcode-in-cs }
  {
    Category~code~'\iow_char:N\\c#1#3'~ignored~inside~
    '\iow_char:N\\c\{...\}'~in~a~replacement~text.
  }
  {
    In~a~replacement~text,~the~category~codes~of~the~argument~of~
    '\iow_char:N\\c\{...\}'~are~ignored~when~building~the~control~
    sequence~name.
  }
\msg_new:nnnn { regex } { replacement-null-space }
  { TeX~cannot~build~a~space~token~with~character~code~0. }
  {
    You~asked~for~a~character~token~with~category~space,~
    and~character~code~0,~for~instance~through~
    '\iow_char:N\\cS\iow_char:N\\x00'.~
    This~specific~case~is~impossible~and~will~be~replaced~
    by~a~normal~space.
  }
\msg_new:nnnn { regex } { replacement-missing-rbrace }
  { Missing~right~brace~inserted~in~replacement~text. }
  {
    There~ \int_compare:nTF { #1 = 1 } { was } { were } ~ #1~
    missing~right~\int_compare:nTF { #1 = 1 } { brace } { braces } .
  }
\msg_new:nnnn { regex } { replacement-missing-rparen }
  { Missing~right~parenthesis~inserted~in~replacement~text. }
  {
    There~ \int_compare:nTF { #1 = 1 } { was } { were } ~ #1~
    missing~right~
    \int_compare:nTF { #1 = 1 } { parenthesis } { parentheses } .
  }
\msg_new:nnn { regex } { submatch-too-big }
  { Submatch~#1~used~but~regex~only~has~#2~group(s) }
\msg_new:nnnn { regex } { backwards-quantifier }
  { Quantifer~"{#1,#2}"~is~backwards. }
  { The~values~given~in~a~quantifier~must~be~in~order. }
\msg_new:nnnn { regex } { case-odd }
  { #1~with~odd~number~of~items }
  {
    There~must~be~a~#2~part~for~each~regex:~
    found~odd~number~of~items~(#3)~in\\
    \iow_indent:n {#4}
  }
\msg_new:nnn { regex } { show }
  {
    >~Compiled~regex~
    \tl_if_empty:nTF {#1} { variable~ #2 } { {#1} } :
    #3
  }
```

```
\prop_gput:Nnn \g_msg_module_name_prop { regex } { LaTeX }
\prop_gput:Nnn \g_msg_module_type_prop { regex } { }
\cs_new:Npn \__regex_msg_repeated:nnN #1#2#3
  {
    \str_if_eq:eeF { #1 #2 } { 1 0 }
      {
        , ~ repeated ~
        \int_case:nnF {#2}
          {
            { -1 } { #1~or~more~times,~\bool_if:NTF #3 { lazy } { greedy } }
            {  0 } { #1~times }
          }
          {
            between~#1~and~\int_eval:n {#1+#2}~times,~
            \bool_if:NTF #3 { lazy } { greedy }
          }
      }
  }
\cs_new_protected:Npn \__regex_trace_push:nnN #1#2#3
  { \__regex_trace:nne {#1} {#2} { entering~ \token_to_str:N #3 } }
\cs_new_protected:Npn \__regex_trace_pop:nnN #1#2#3
  { \__regex_trace:nne {#1} {#2} { leaving~ \token_to_str:N #3 } }
\cs_new_protected:Npn \__regex_trace:nne #1#2#3
  {
    \int_compare:nNnF
      { \int_use:c { g__regex_trace_#1_int } } < {#2}
      { \iow_term:e { Trace:~#3 } }
  }
\int_new:N \g__regex_trace_regex_int
\cs_new_protected:Npn \__regex_trace_states:n #1
  {
    \int_step_inline:nnn
      \l__regex_min_state_int
      { \l__regex_max_state_int - \c_one_int }
      {
        \__regex_trace:nne { regex } {#1}
          { \iow_char:N \\toks ##1 = { \__regex_toks_use:w ##1 } }
      }
  }
%% File: l3box.dtx
\cs_new_eq:NN \__box_dim_eval:w \tex_dimexpr:D
\cs_new:Npn \__box_dim_eval:n #1
  { \__box_dim_eval:w #1 \scan_stop: }
\cs_new_protected:Npn \__kernel_kern:n #1
  { \tex_kern:D \__box_dim_eval:n {#1} }
\cs_new_protected:Npn \box_new:N #1
```

```
  {
    \__kernel_chk_if_free_cs:N #1
    \cs:w newbox \cs_end: #1
  }
\cs_generate_variant:Nn \box_new:N { c }
\cs_new_protected:Npn \box_clear:N #1
  { \box_set_eq:NN  #1 \c_empty_box }
\cs_new_protected:Npn \box_gclear:N #1
  { \box_gset_eq:NN #1 \c_empty_box }
\cs_generate_variant:Nn \box_clear:N  { c }
\cs_generate_variant:Nn \box_gclear:N { c }
\cs_new_protected:Npn \box_clear_new:N #1
  { \box_if_exist:NTF #1 { \box_clear:N #1 } { \box_new:N #1 } }
\cs_new_protected:Npn \box_gclear_new:N #1
  { \box_if_exist:NTF #1 { \box_gclear:N #1 } { \box_new:N #1 } }
\cs_generate_variant:Nn \box_clear_new:N  { c }
\cs_generate_variant:Nn \box_gclear_new:N { c }
\cs_new_protected:Npn \box_set_eq:NN #1#2
  { \tex_setbox:D #1 \tex_copy:D #2 }
\cs_new_protected:Npn \box_gset_eq:NN #1#2
  { \tex_global:D \tex_setbox:D #1 \tex_copy:D #2 }
\cs_generate_variant:Nn \box_set_eq:NN  { c , Nc , cc }
\cs_generate_variant:Nn \box_gset_eq:NN { c , Nc , cc }
\cs_new_protected:Npn \box_set_eq_drop:NN #1#2
  { \tex_setbox:D #1 \tex_box:D #2 }
\cs_new_protected:Npn \box_gset_eq_drop:NN #1#2
  { \tex_global:D \tex_setbox:D #1 \tex_box:D #2 }
\cs_generate_variant:Nn \box_set_eq_drop:NN  { c , Nc , cc }
\cs_generate_variant:Nn \box_gset_eq_drop:NN { c , Nc , cc }
\prg_new_eq_conditional:NNn \box_if_exist:N \cs_if_exist:N
  { TF , T , F , p }
\prg_new_eq_conditional:NNn \box_if_exist:c \cs_if_exist:c
  { TF , T , F , p }
\cs_new_eq:NN \box_ht:N \tex_ht:D
\cs_new_eq:NN \box_dp:N \tex_dp:D
\cs_new_eq:NN \box_wd:N \tex_wd:D
\cs_generate_variant:Nn \box_ht:N { c }
\cs_generate_variant:Nn \box_dp:N { c }
\cs_generate_variant:Nn \box_wd:N { c }
\cs_new_protected:Npn \box_ht_plus_dp:N #1
  { \__box_dim_eval:n { \box_ht:N #1 + \box_dp:N #1 } }
\cs_generate_variant:Nn \box_ht_plus_dp:N { c }
\cs_new_protected:Npn \box_set_dp:Nn #1#2
  {
    \tex_setbox:D #1 = \tex_copy:D #1
    \box_dp:N #1 \__box_dim_eval:n {#2}
  }
\cs_generate_variant:Nn \box_set_dp:Nn { c }
\cs_new_protected:Npn \box_gset_dp:Nn #1#2
  { \box_dp:N #1 \__box_dim_eval:n {#2} }
\cs_generate_variant:Nn \box_gset_dp:Nn { c }
\cs_new_protected:Npn \box_set_ht:Nn #1#2
  {
    \tex_setbox:D #1 = \tex_copy:D #1
    \box_ht:N #1 \__box_dim_eval:n {#2}
  }
\cs_generate_variant:Nn \box_set_ht:Nn { c }
\cs_new_protected:Npn \box_gset_ht:Nn #1#2
  { \box_ht:N #1 \__box_dim_eval:n {#2} }
\cs_generate_variant:Nn \box_gset_ht:Nn { c }
\cs_new_protected:Npn \box_set_wd:Nn #1#2
  {
    \tex_setbox:D #1 = \tex_copy:D #1
    \box_wd:N #1 \__box_dim_eval:n {#2}
  }
\cs_generate_variant:Nn \box_set_wd:Nn { c }
\cs_new_protected:Npn \box_gset_wd:Nn #1#2
  { \box_wd:N #1 \__box_dim_eval:n {#2} }
\cs_generate_variant:Nn \box_gset_wd:Nn { c }
\cs_new_eq:NN \box_use_drop:N \tex_box:D
\cs_new_eq:NN \box_use:N \tex_copy:D
\cs_generate_variant:Nn \box_use_drop:N { c }
\cs_generate_variant:Nn \box_use:N { c }
\cs_new_protected:Npn \box_move_left:nn #1#2
  { \tex_moveleft:D \__box_dim_eval:n {#1} #2 }
\cs_new_protected:Npn \box_move_right:nn #1#2
  { \tex_moveright:D \__box_dim_eval:n {#1} #2 }
\cs_new_protected:Npn \box_move_up:nn #1#2
  { \tex_raise:D \__box_dim_eval:n {#1} #2 }
\cs_new_protected:Npn \box_move_down:nn #1#2
  { \tex_lower:D \__box_dim_eval:n {#1} #2 }
\cs_new_eq:NN \if_hbox:N      \tex_ifhbox:D
\cs_new_eq:NN \if_vbox:N      \tex_ifvbox:D
\cs_new_eq:NN \if_box_empty:N \tex_ifvoid:D
\prg_new_conditional:Npnn \box_if_horizontal:N #1 { p , T , F , TF }
  { \if_hbox:N #1 \prg_return_true: \else: \prg_return_false: \fi: }
\prg_new_conditional:Npnn \box_if_vertical:N #1 { p , T , F , TF }
  { \if_vbox:N #1 \prg_return_true: \else: \prg_return_false: \fi: }
\prg_generate_conditional_variant:Nnn \box_if_horizontal:N
  { c } { p , T , F , TF }
\prg_generate_conditional_variant:Nnn \box_if_vertical:N
  { c } { p , T , F , TF }
```

```
\prg_new_conditional:Npnn \box_if_empty:N #1 { p , T , F , TF }
  { \if_box_empty:N #1 \prg_return_true: \else: \prg_return_false: \fi: }
\prg_generate_conditional_variant:Nnn \box_if_empty:N
  { c } { p , T , F , TF }
\cs_new_protected:Npn \box_set_to_last:N #1
  { \tex_setbox:D #1 \tex_lastbox:D }
\cs_new_protected:Npn \box_gset_to_last:N #1
  { \tex_global:D \tex_setbox:D #1 \tex_lastbox:D }
\cs_generate_variant:Nn \box_set_to_last:N  { c }
\cs_generate_variant:Nn \box_gset_to_last:N { c }
\box_new:N \c_empty_box
\box_new:N \l_tmpa_box
\box_new:N \l_tmpb_box
\box_new:N \g_tmpa_box
\box_new:N \g_tmpb_box
\cs_new_protected:Npn \box_show:N #1
  { \box_show:Nnn #1 \c_max_int \c_max_int }
\cs_generate_variant:Nn \box_show:N { c }
\cs_new_protected:Npn \box_show:Nnn #1#2#3
  { \__box_show:NNff 1 #1 { \int_eval:n {#2} } { \int_eval:n {#3} } }
\cs_generate_variant:Nn \box_show:Nnn { c }
\cs_new_protected:Npn \box_log:N #1
  { \box_log:Nnn #1 \c_max_int \c_max_int }
\cs_generate_variant:Nn \box_log:N { c }
\cs_new_protected:Npn \box_log:Nnn
  { \exp_args:No \__box_log:nNnn { \tex_the:D \tex_interactionmode:D } }
\cs_new_protected:Npn \__box_log:nNnn #1#2#3#4
  {
    \int_gset:Nn \tex_interactionmode:D { 0 }
    \__box_show:NNff 0 #2 { \int_eval:n {#3} } { \int_eval:n {#4} }
    \int_gset:Nn \tex_interactionmode:D {#1}
  }
\cs_generate_variant:Nn \box_log:Nnn { c }
\cs_new_protected:Npn \__box_show:NNnn #1#2#3#4
  {
    \box_if_exist:NTF #2
      {
        \group_begin:
          \int_set:Nn \tex_showboxbreadth:D {#3}
          \int_set:Nn \tex_showboxdepth:D   {#4}
          \int_set:Nn \tex_tracingonline:D  {#1}
          \int_set:Nn \tex_errorcontextlines:D { -1 }
          \tex_showbox:D \use:n {#2}
        \group_end:
      }
      {
        \msg_error:nne { kernel } { variable-not-defined }
          { \token_to_str:N #2 }
      }
  }
\cs_generate_variant:Nn \__box_show:NNnn { NNff }
\cs_new_protected:Npn \hbox:n #1
  { \tex_hbox:D \scan_stop: { \color_group_begin: #1 \color_group_end: } }
\cs_new_protected:Npn \hbox_set:Nn #1#2
  {
    \tex_setbox:D #1 \tex_hbox:D
      { \color_group_begin: #2 \color_group_end: }
  }
\cs_new_protected:Npn \hbox_gset:Nn #1#2
  {
    \tex_global:D \tex_setbox:D #1 \tex_hbox:D
      { \color_group_begin: #2 \color_group_end: }
  }
\cs_generate_variant:Nn \hbox_set:Nn { c }
\cs_generate_variant:Nn \hbox_gset:Nn { c }
\cs_new_protected:Npn \hbox_set_to_wd:Nnn #1#2#3
  {
    \tex_setbox:D #1 \tex_hbox:D to \__box_dim_eval:n {#2}
      { \color_group_begin: #3 \color_group_end: }
  }
\cs_new_protected:Npn \hbox_gset_to_wd:Nnn #1#2#3
  {
    \tex_global:D \tex_setbox:D #1 \tex_hbox:D to \__box_dim_eval:n {#2}
      { \color_group_begin: #3 \color_group_end: }
  }
\cs_generate_variant:Nn \hbox_set_to_wd:Nnn { c }
\cs_generate_variant:Nn \hbox_gset_to_wd:Nnn { c }
\cs_new_protected:Npn \hbox_set:Nw  #1
  {
    \tex_setbox:D #1 \tex_hbox:D
      \c_group_begin_token
        \color_group_begin:
  }
\cs_new_protected:Npn \hbox_gset:Nw  #1
  {
    \tex_global:D \tex_setbox:D #1 \tex_hbox:D
      \c_group_begin_token
        \color_group_begin:
  }
\cs_generate_variant:Nn \hbox_set:Nw  { c }
\cs_generate_variant:Nn \hbox_gset:Nw { c }
\cs_new_protected:Npn \hbox_set_end:
  {
      \color_group_end:
    \c_group_end_token
  }
\cs_new_eq:NN \hbox_gset_end: \hbox_set_end:
\cs_new_protected:Npn \hbox_set_to_wd:Nnw #1#2
  {
    \tex_setbox:D #1 \tex_hbox:D to \__box_dim_eval:n {#2}
      \c_group_begin_token
        \color_group_begin:
  }
\cs_new_protected:Npn \hbox_gset_to_wd:Nnw #1#2
  {
    \tex_global:D \tex_setbox:D #1 \tex_hbox:D to \__box_dim_eval:n {#2}
      \c_group_begin_token
        \color_group_begin:
  }
\cs_generate_variant:Nn \hbox_set_to_wd:Nnw  { c }
\cs_generate_variant:Nn \hbox_gset_to_wd:Nnw { c }
\cs_new_protected:Npn \hbox_to_wd:nn #1#2
  {
    \tex_hbox:D to \__box_dim_eval:n {#1}
      { \color_group_begin: #2 \color_group_end: }
  }
\cs_new_protected:Npn \hbox_to_zero:n #1
  {
    \tex_hbox:D to \c_zero_dim
      { \color_group_begin: #1 \color_group_end: }
  }
\cs_new_protected:Npn \hbox_overlap_center:n  #1
  { \hbox_to_zero:n { \tex_hss:D #1 \tex_hss:D } }
\cs_new_protected:Npn \hbox_overlap_left:n  #1
  { \hbox_to_zero:n { \tex_hss:D #1 } }
\cs_new_protected:Npn \hbox_overlap_right:n #1
  { \hbox_to_zero:n { #1 \tex_hss:D } }
\cs_new_eq:NN \hbox_unpack:N \tex_unhcopy:D
\cs_new_eq:NN \hbox_unpack_drop:N \tex_unhbox:D
\cs_generate_variant:Nn \hbox_unpack:N { c }
\cs_generate_variant:Nn \hbox_unpack_drop:N { c }
\cs_new_protected:Npn \vbox:n #1
  { \tex_vbox:D { \color_group_begin: #1 \par \color_group_end: } }
\cs_new_protected:Npn \vbox_top:n #1
  { \tex_vtop:D { \color_group_begin: #1 \par \color_group_end: } }
\cs_new_protected:Npn \vbox_to_ht:nn #1#2
  {
    \tex_vbox:D to \__box_dim_eval:n {#1}
```

```
      { \color_group_begin: #2 \par \color_group_end: }
  }
\cs_new_protected:Npn \vbox_to_zero:n #1
  {
    \tex_vbox:D to \c_zero_dim
      { \color_group_begin: #1 \par \color_group_end: }
  }
\cs_new_protected:Npn \vbox_set:Nn #1#2
  {
    \tex_setbox:D #1 \tex_vbox:D
      { \color_group_begin: #2 \par \color_group_end: }
  }
\cs_new_protected:Npn \vbox_gset:Nn #1#2
  {
    \tex_global:D \tex_setbox:D #1 \tex_vbox:D
      { \color_group_begin: #2 \par \color_group_end: }
  }
\cs_generate_variant:Nn \vbox_set:Nn  { c }
\cs_generate_variant:Nn \vbox_gset:Nn { c }
\cs_new_protected:Npn \vbox_set_top:Nn #1#2
  {
    \tex_setbox:D #1 \tex_vtop:D
      { \color_group_begin: #2 \par \color_group_end: }
  }
\cs_new_protected:Npn \vbox_gset_top:Nn #1#2
  {
    \tex_global:D \tex_setbox:D #1 \tex_vtop:D
      { \color_group_begin: #2 \par \color_group_end: }
  }
\cs_generate_variant:Nn \vbox_set_top:Nn { c }
\cs_generate_variant:Nn \vbox_gset_top:Nn { c }
\cs_new_protected:Npn \vbox_set_to_ht:Nnn #1#2#3
  {
    \tex_setbox:D #1 \tex_vbox:D to \__box_dim_eval:n {#2}
      { \color_group_begin: #3 \par \color_group_end: }
  }
\cs_new_protected:Npn \vbox_gset_to_ht:Nnn #1#2#3
  {
    \tex_global:D \tex_setbox:D #1 \tex_vbox:D to \__box_dim_eval:n {#2}
      { \color_group_begin: #3 \par \color_group_end: }
  }
\cs_generate_variant:Nn \vbox_set_to_ht:Nnn  { c }
\cs_generate_variant:Nn \vbox_gset_to_ht:Nnn { c }
\cs_new_protected:Npn \vbox_set:Nw #1
  {
    \tex_setbox:D #1 \tex_vbox:D
```

```
      \c_group_begin_token
        \color_group_begin:
  }
\cs_new_protected:Npn \vbox_gset:Nw #1
  {
    \tex_global:D \tex_setbox:D #1 \tex_vbox:D
      \c_group_begin_token
        \color_group_begin:
  }
\cs_generate_variant:Nn \vbox_set:Nw  { c }
\cs_generate_variant:Nn \vbox_gset:Nw { c }
\cs_new_protected:Npn \vbox_set_end:
  {
        \par
      \color_group_end:
    \c_group_end_token
  }
\cs_new_eq:NN \vbox_gset_end: \vbox_set_end:
\cs_new_protected:Npn \vbox_set_to_ht:Nnw #1#2
  {
    \tex_setbox:D #1 \tex_vbox:D to \__box_dim_eval:n {#2}
      \c_group_begin_token
        \color_group_begin:
  }
\cs_new_protected:Npn \vbox_gset_to_ht:Nnw #1#2
  {
    \tex_global:D \tex_setbox:D #1 \tex_vbox:D to \__box_dim_eval:n {#2}
      \c_group_begin_token
        \color_group_begin:
  }
\cs_generate_variant:Nn \vbox_set_to_ht:Nnw  { c }
\cs_generate_variant:Nn \vbox_gset_to_ht:Nnw { c }
\cs_new_eq:NN \vbox_unpack:N \tex_unvcopy:D
\cs_new_eq:NN \vbox_unpack_drop:N \tex_unvbox:D
\cs_generate_variant:Nn \vbox_unpack:N { c }
\cs_generate_variant:Nn \vbox_unpack_drop:N { c }
\cs_new_protected:Npn \vbox_set_split_to_ht:NNn #1#2#3
  { \tex_setbox:D #1 \tex_vsplit:D #2 to \__box_dim_eval:n {#3} }
\cs_generate_variant:Nn \vbox_set_split_to_ht:NNn { c , Nc , cc }
\cs_new_protected:Npn \vbox_gset_split_to_ht:NNn #1#2#3
  {
    \tex_global:D \tex_setbox:D #1
      \tex_vsplit:D #2 to \__box_dim_eval:n {#3}
  }
\cs_generate_variant:Nn \vbox_gset_split_to_ht:NNn { c , Nc , cc }
\fp_new:N \l__box_angle_fp
```

```
\fp_new:N \l__box_cos_fp
\fp_new:N \l__box_sin_fp
\dim_new:N \l__box_top_dim
\dim_new:N \l__box_bottom_dim
\dim_new:N \l__box_left_dim
\dim_new:N \l__box_right_dim
\dim_new:N \l__box_top_new_dim
\dim_new:N \l__box_bottom_new_dim
\dim_new:N \l__box_left_new_dim
\dim_new:N \l__box_right_new_dim
\box_new:N \l__box_internal_box
\cs_new_protected:Npn \box_rotate:Nn #1#2
  { \__box_rotate:NnN #1 {#2} \hbox_set:Nn }
\cs_generate_variant:Nn \box_rotate:Nn { c }
\cs_new_protected:Npn \box_grotate:Nn #1#2
  { \__box_rotate:NnN #1 {#2} \hbox_gset:Nn }
\cs_generate_variant:Nn \box_grotate:Nn { c }
\cs_new_protected:Npn \__box_rotate:NnN #1#2#3
  {
    #3 #1
      {
        \fp_set:Nn \l__box_angle_fp {#2}
        \fp_set:Nn \l__box_sin_fp { sind ( \l__box_angle_fp ) }
        \fp_set:Nn \l__box_cos_fp { cosd ( \l__box_angle_fp ) }
        \__box_rotate:N #1
      }
  }
\cs_new_protected:Npn \__box_rotate:N #1
  {
    \dim_set:Nn \l__box_top_dim    {  \box_ht:N #1 }
    \dim_set:Nn \l__box_bottom_dim { -\box_dp:N #1 }
    \dim_set:Nn \l__box_right_dim  {  \box_wd:N #1 }
    \dim_zero:N \l__box_left_dim
    \fp_compare:nNnTF \l__box_sin_fp > \c_zero_fp
      {
        \fp_compare:nNnTF \l__box_cos_fp > \c_zero_fp
          { \__box_rotate_quadrant_one: }
          { \__box_rotate_quadrant_two: }
      }
      {
        \fp_compare:nNnTF \l__box_cos_fp < \c_zero_fp
          { \__box_rotate_quadrant_three: }
          { \__box_rotate_quadrant_four: }
      }
    \hbox_set:Nn \l__box_internal_box { \box_use:N #1 }
    \hbox_set:Nn \l__box_internal_box
```

```
      {
        \__kernel_kern:n { -\l__box_left_new_dim }
        \hbox:n
          {
            \__box_backend_rotate:Nn
              \l__box_internal_box
              \l__box_angle_fp
          }
      }
    \box_set_ht:Nn \l__box_internal_box {  \l__box_top_new_dim }
    \box_set_dp:Nn \l__box_internal_box { -\l__box_bottom_new_dim }
    \box_set_wd:Nn \l__box_internal_box
      { \l__box_right_new_dim - \l__box_left_new_dim }
    \box_use_drop:N \l__box_internal_box
  }
\cs_new_protected:Npn \__box_rotate_xdir:nnN #1#2#3
  {
    \dim_set:Nn #3
      {
        \fp_to_dim:n
          {
              \l__box_cos_fp * \dim_to_fp:n {#1}
            - \l__box_sin_fp * \dim_to_fp:n {#2}
          }
      }
  }
\cs_new_protected:Npn \__box_rotate_ydir:nnN #1#2#3
  {
    \dim_set:Nn #3
      {
        \fp_to_dim:n
          {
              \l__box_sin_fp * \dim_to_fp:n {#1}
            + \l__box_cos_fp * \dim_to_fp:n {#2}
          }
      }
  }
\cs_new_protected:Npn \__box_rotate_quadrant_one:
  {
    \__box_rotate_ydir:nnN \l__box_right_dim \l__box_top_dim
      \l__box_top_new_dim
    \__box_rotate_ydir:nnN \l__box_left_dim  \l__box_bottom_dim
      \l__box_bottom_new_dim
    \__box_rotate_xdir:nnN \l__box_left_dim  \l__box_top_dim
      \l__box_left_new_dim
    \__box_rotate_xdir:nnN \l__box_right_dim \l__box_bottom_dim
```

```
      \l__box_right_new_dim
  }
\cs_new_protected:Npn \__box_rotate_quadrant_two:
  {
    \__box_rotate_ydir:nnN \l__box_right_dim \l__box_bottom_dim
      \l__box_top_new_dim
    \__box_rotate_ydir:nnN \l__box_left_dim  \l__box_top_dim
      \l__box_bottom_new_dim
    \__box_rotate_xdir:nnN \l__box_right_dim  \l__box_top_dim
      \l__box_left_new_dim
    \__box_rotate_xdir:nnN \l__box_left_dim   \l__box_bottom_dim
      \l__box_right_new_dim
  }
\cs_new_protected:Npn \__box_rotate_quadrant_three:
  {
    \__box_rotate_ydir:nnN \l__box_left_dim  \l__box_bottom_dim
      \l__box_top_new_dim
    \__box_rotate_ydir:nnN \l__box_right_dim \l__box_top_dim
      \l__box_bottom_new_dim
    \__box_rotate_xdir:nnN \l__box_right_dim \l__box_bottom_dim
      \l__box_left_new_dim
    \__box_rotate_xdir:nnN \l__box_left_dim   \l__box_top_dim
      \l__box_right_new_dim
  }
\cs_new_protected:Npn \__box_rotate_quadrant_four:
  {
    \__box_rotate_ydir:nnN \l__box_left_dim  \l__box_top_dim
      \l__box_top_new_dim
    \__box_rotate_ydir:nnN \l__box_right_dim \l__box_bottom_dim
      \l__box_bottom_new_dim
    \__box_rotate_xdir:nnN \l__box_left_dim  \l__box_bottom_dim
      \l__box_left_new_dim
    \__box_rotate_xdir:nnN \l__box_right_dim \l__box_top_dim
      \l__box_right_new_dim
  }
\fp_new:N \l__box_scale_x_fp
\fp_new:N \l__box_scale_y_fp
\cs_new_protected:Npn \box_resize_to_wd_and_ht_plus_dp:Nnn #1#2#3
  {
    \__box_resize_to_wd_and_ht_plus_dp:NnnN #1 {#2} {#3}
      \hbox_set:Nn
  }
\cs_generate_variant:Nn \box_resize_to_wd_and_ht_plus_dp:Nnn { c }
\cs_new_protected:Npn \box_gresize_to_wd_and_ht_plus_dp:Nnn #1#2#3
  {
    \__box_resize_to_wd_and_ht_plus_dp:NnnN #1 {#2} {#3}
```

```
      \hbox_gset:Nn
  }
\cs_generate_variant:Nn \box_gresize_to_wd_and_ht_plus_dp:Nnn { c }
\cs_new_protected:Npn \__box_resize_to_wd_and_ht_plus_dp:NnnN #1#2#3#4
  {
    #4 #1
      {
        \__box_resize_set_corners:N #1
        \fp_set:Nn \l__box_scale_x_fp
          { \dim_to_fp:n {#2} / \dim_to_fp:n { \l__box_right_dim } }
        \fp_set:Nn \l__box_scale_y_fp
          {
              \dim_to_fp:n {#3}
            / \dim_to_fp:n { \l__box_top_dim - \l__box_bottom_dim }
          }
        \__box_resize:N #1
      }
  }
\cs_new_protected:Npn \__box_resize_set_corners:N #1
  {
    \dim_set:Nn \l__box_top_dim    {  \box_ht:N #1 }
    \dim_set:Nn \l__box_bottom_dim { -\box_dp:N #1 }
    \dim_set:Nn \l__box_right_dim  {  \box_wd:N #1 }
    \dim_zero:N \l__box_left_dim
  }
\cs_new_protected:Npn \__box_resize:N #1
  {
    \__box_resize:NNN \l__box_right_new_dim
      \l__box_scale_x_fp \l__box_right_dim
    \__box_resize:NNN \l__box_bottom_new_dim
      \l__box_scale_y_fp \l__box_bottom_dim
    \__box_resize:NNN \l__box_top_new_dim
      \l__box_scale_y_fp \l__box_top_dim
    \__box_resize_common:N #1
  }
\cs_new_protected:Npn \__box_resize:NNN #1#2#3
  {
    \dim_set:Nn #1
      { \fp_to_dim:n { \fp_abs:n { #2 } * \dim_to_fp:n { #3 } } }
  }
\cs_new_protected:Npn \box_resize_to_ht:Nn #1#2
  { \__box_resize_to_ht:NnN #1 {#2} \hbox_set:Nn }
\cs_generate_variant:Nn \box_resize_to_ht:Nn { c }
\cs_new_protected:Npn \box_gresize_to_ht:Nn #1#2
  { \__box_resize_to_ht:NnN #1 {#2} \hbox_gset:Nn }
\cs_generate_variant:Nn \box_gresize_to_ht:Nn { c }
```

```
\cs_new_protected:Npn \__box_resize_to_ht:NnN #1#2#3
  {
    #3 #1
      {
        \__box_resize_set_corners:N #1
        \fp_set:Nn \l__box_scale_y_fp
          {
              \dim_to_fp:n {#2}
            / \dim_to_fp:n { \l__box_top_dim }
          }
        \fp_set_eq:NN \l__box_scale_x_fp \l__box_scale_y_fp
        \__box_resize:N #1
      }
  }
\cs_new_protected:Npn \box_resize_to_ht_plus_dp:Nn #1#2
  { \__box_resize_to_ht_plus_dp:NnN #1 {#2} \hbox_set:Nn }
\cs_generate_variant:Nn \box_resize_to_ht_plus_dp:Nn { c }
\cs_new_protected:Npn \box_gresize_to_ht_plus_dp:Nn #1#2
  { \__box_resize_to_ht_plus_dp:NnN #1 {#2} \hbox_gset:Nn }
\cs_generate_variant:Nn \box_gresize_to_ht_plus_dp:Nn { c }
\cs_new_protected:Npn \__box_resize_to_ht_plus_dp:NnN #1#2#3
  {
    #3 #1
      {
        \__box_resize_set_corners:N #1
        \fp_set:Nn \l__box_scale_y_fp
          {
              \dim_to_fp:n {#2}
            / \dim_to_fp:n { \l__box_top_dim - \l__box_bottom_dim }
          }
        \fp_set_eq:NN \l__box_scale_x_fp \l__box_scale_y_fp
        \__box_resize:N #1
      }
  }
\cs_new_protected:Npn \box_resize_to_wd:Nn #1#2
  { \__box_resize_to_wd:NnN #1 {#2} \hbox_set:Nn }
\cs_generate_variant:Nn \box_resize_to_wd:Nn { c }
\cs_new_protected:Npn \box_gresize_to_wd:Nn #1#2
  { \__box_resize_to_wd:NnN #1 {#2} \hbox_gset:Nn }
\cs_generate_variant:Nn \box_gresize_to_wd:Nn { c }
\cs_new_protected:Npn \__box_resize_to_wd:NnN #1#2#3
  {
    #3 #1
      {
        \__box_resize_set_corners:N #1
        \fp_set:Nn \l__box_scale_x_fp
```

```
          { \dim_to_fp:n {#2} / \dim_to_fp:n { \l__box_right_dim } }
        \fp_set_eq:NN \l__box_scale_y_fp \l__box_scale_x_fp
        \__box_resize:N #1
      }
  }
\cs_new_protected:Npn \box_resize_to_wd_and_ht:Nnn #1#2#3
  { \__box_resize_to_wd_and_ht:NnnN #1 {#2} {#3} \hbox_set:Nn }
\cs_generate_variant:Nn \box_resize_to_wd_and_ht:Nnn { c }
\cs_new_protected:Npn \box_gresize_to_wd_and_ht:Nnn #1#2#3
  { \__box_resize_to_wd_and_ht:NnnN #1 {#2} {#3} \hbox_gset:Nn }
\cs_generate_variant:Nn \box_gresize_to_wd_and_ht:Nnn { c }
\cs_new_protected:Npn \__box_resize_to_wd_and_ht:NnnN #1#2#3#4
  {
    #4 #1
      {
        \__box_resize_set_corners:N #1
        \fp_set:Nn \l__box_scale_x_fp
          { \dim_to_fp:n {#2} / \dim_to_fp:n { \l__box_right_dim } }
        \fp_set:Nn \l__box_scale_y_fp
          {
              \dim_to_fp:n {#3}
            / \dim_to_fp:n { \l__box_top_dim }
          }
        \__box_resize:N #1
      }
  }
\cs_new_protected:Npn \box_scale:Nnn #1#2#3
  { \__box_scale:NnnN #1 {#2} {#3} \hbox_set:Nn }
\cs_generate_variant:Nn \box_scale:Nnn { c }
\cs_new_protected:Npn \box_gscale:Nnn #1#2#3
  { \__box_scale:NnnN #1 {#2} {#3} \hbox_gset:Nn }
\cs_generate_variant:Nn \box_gscale:Nnn { c }
\cs_new_protected:Npn \__box_scale:NnnN #1#2#3#4
  {
    #4 #1
      {
        \fp_set:Nn \l__box_scale_x_fp {#2}
        \fp_set:Nn \l__box_scale_y_fp {#3}
        \__box_scale:N #1
      }
  }
\cs_new_protected:Npn \__box_scale:N #1
  {
    \dim_set:Nn \l__box_top_dim    {  \box_ht:N #1 }
    \dim_set:Nn \l__box_bottom_dim { -\box_dp:N #1 }
    \dim_set:Nn \l__box_right_dim  {  \box_wd:N #1 }
    \dim_zero:N \l__box_left_dim
    \dim_set:Nn \l__box_top_new_dim
      { \fp_abs:n { \l__box_scale_y_fp } \l__box_top_dim }
    \dim_set:Nn \l__box_bottom_new_dim
      { \fp_abs:n { \l__box_scale_y_fp } \l__box_bottom_dim }
    \dim_set:Nn \l__box_right_new_dim
      { \fp_abs:n { \l__box_scale_x_fp } \l__box_right_dim }
    \__box_resize_common:N #1
  }
\cs_new_protected:Npn \box_autosize_to_wd_and_ht:Nnn #1#2#3
  { \__box_autosize:NnnnN #1 {#2} {#3} { \box_ht:N #1 } \hbox_set:Nn }
\cs_generate_variant:Nn \box_autosize_to_wd_and_ht:Nnn { c }
\cs_new_protected:Npn \box_gautosize_to_wd_and_ht:Nnn #1#2#3
  { \__box_autosize:NnnnN #1 {#2} {#3} { \box_ht:N #1 } \hbox_gset:Nn }
\cs_generate_variant:Nn \box_gautosize_to_wd_and_ht:Nnn { c }
\cs_new_protected:Npn \box_autosize_to_wd_and_ht_plus_dp:Nnn #1#2#3
  {
    \__box_autosize:NnnnN #1 {#2} {#3} { \box_ht:N #1 + \box_dp:N #1 }
      \hbox_set:Nn
  }
\cs_generate_variant:Nn \box_autosize_to_wd_and_ht_plus_dp:Nnn { c }
\cs_new_protected:Npn \box_gautosize_to_wd_and_ht_plus_dp:Nnn #1#2#3
  {
    \__box_autosize:NnnnN #1 {#2} {#3} { \box_ht:N #1 + \box_dp:N #1 }
      \hbox_gset:Nn
  }
\cs_generate_variant:Nn \box_gautosize_to_wd_and_ht_plus_dp:Nnn { c }
\cs_new_protected:Npn \__box_autosize:NnnnN #1#2#3#4#5
  {
    #5 #1
      {
        \fp_set:Nn \l__box_scale_x_fp { ( \dim_to_fp:n {#2} ) / \box_wd:N #1 }
        \fp_set:Nn \l__box_scale_y_fp
          { (  \dim_to_fp:n {#3} ) / ( \dim_to_fp:n {#4} ) }
        \fp_compare:nNnTF \l__box_scale_x_fp > \l__box_scale_y_fp
          { \fp_set_eq:NN \l__box_scale_x_fp \l__box_scale_y_fp }
          { \fp_set_eq:NN \l__box_scale_y_fp \l__box_scale_x_fp }
        \__box_scale:N #1
      }
  }
\cs_new_protected:Npn \__box_resize_common:N #1
  {
    \hbox_set:Nn \l__box_internal_box
      {
        \__box_backend_scale:Nnn
          #1
          \l__box_scale_x_fp
          \l__box_scale_y_fp
      }
    \fp_compare:nNnTF \l__box_scale_y_fp > \c_zero_fp
      {
        \box_set_ht:Nn \l__box_internal_box { \l__box_top_new_dim }
        \box_set_dp:Nn \l__box_internal_box { -\l__box_bottom_new_dim }
      }
      {
        \box_set_dp:Nn \l__box_internal_box { \l__box_top_new_dim }
        \box_set_ht:Nn \l__box_internal_box { -\l__box_bottom_new_dim }
      }
    \fp_compare:nNnTF \l__box_scale_x_fp < \c_zero_fp
      {
        \hbox_to_wd:nn { \l__box_right_new_dim }
          {
            \__kernel_kern:n { \l__box_right_new_dim }
            \box_use_drop:N \l__box_internal_box
            \tex_hss:D
          }
      }
      {
        \box_set_wd:Nn \l__box_internal_box { \l__box_right_new_dim }
        \hbox:n
          {
            \__kernel_kern:n { 0pt }
            \box_use_drop:N \l__box_internal_box
            \tex_hss:D
          }
      }
  }
\cs_new_protected:Npn \box_set_clipped:N #1
  { \hbox_set:Nn #1 { \__box_backend_clip:N #1 } }
\cs_generate_variant:Nn \box_set_clipped:N { c }
\cs_new_protected:Npn \box_gset_clipped:N #1
  { \hbox_gset:Nn #1 { \__box_backend_clip:N #1 } }
\cs_generate_variant:Nn \box_gset_clipped:N { c }
\cs_new_protected:Npn \box_set_trim:Nnnnn #1#2#3#4#5
  { \__box_set_trim:NnnnnN #1 {#2} {#3} {#4} {#5} \box_set_eq:NN }
\cs_generate_variant:Nn \box_set_trim:Nnnnn { c }
\cs_new_protected:Npn \box_gset_trim:Nnnnn #1#2#3#4#5
  { \__box_set_trim:NnnnnN #1 {#2} {#3} {#4} {#5} \box_gset_eq:NN }
\cs_generate_variant:Nn \box_gset_trim:Nnnnn { c }
\cs_new_protected:Npn \__box_set_trim:NnnnnN #1#2#3#4#5#6
  {
    \hbox_set:Nn \l__box_internal_box
```

```
   {
    \__kernel_kern:n { -#2 }
    \box_use:N #1
    \__kernel_kern:n { -#4 }
   }
  \dim_compare:nNnTF { \box_dp:N #1 } > {#3}
   {
    \hbox_set:Nn \l__box_internal_box
     {
      \box_move_down:nn \c_zero_dim
       { \box_use_drop:N \l__box_internal_box }
     }
    \box_set_dp:Nn \l__box_internal_box { \box_dp:N #1 - (#3) }
   }
   {
    \hbox_set:Nn \l__box_internal_box
     {
      \box_move_down:nn { (#3) - \box_dp:N #1 }
       { \box_use_drop:N \l__box_internal_box }
     }
    \box_set_dp:Nn \l__box_internal_box \c_zero_dim
   }
  \dim_compare:nNnTF { \box_ht:N \l__box_internal_box } > {#5}
   {
    \hbox_set:Nn \l__box_internal_box
     {
      \box_move_up:nn \c_zero_dim
       { \box_use_drop:N \l__box_internal_box }
     }
    \box_set_ht:Nn \l__box_internal_box
     { \box_ht:N \l__box_internal_box - (#5) }
   }
   {
    \hbox_set:Nn \l__box_internal_box
     {
      \box_move_up:nn { (#5) - \box_ht:N \l__box_internal_box }
       { \box_use_drop:N \l__box_internal_box }
     }
    \box_set_ht:Nn \l__box_internal_box \c_zero_dim
   }
  #6 #1 \l__box_internal_box
 }
\cs_new_protected:Npn \box_set_viewport:Nnnnn #1#2#3#4#5
 { \__box_set_viewport:NnnnnN #1 {#2} {#3} {#4} {#5} \box_set_eq:NN }
\cs_generate_variant:Nn \box_set_viewport:Nnnnn { c }
\cs_new_protected:Npn \box_gset_viewport:Nnnnn #1#2#3#4#5
```

```
  { \__box_set_viewport:NnnnnN #1 {#2} {#3} {#4} {#5} \box_gset_eq:NN }
\cs_generate_variant:Nn \box_gset_viewport:Nnnnn { c }
\cs_new_protected:Npn \__box_set_viewport:NnnnnN #1#2#3#4#5#6
  {
    \hbox_set:Nn \l__box_internal_box
      {
        \__kernel_kern:n { -#2 }
        \box_use:N #1
        \__kernel_kern:n { #4 - \box_wd:N #1 }
      }
    \dim_compare:nNnTF {#3} < \c_zero_dim
      {
        \hbox_set:Nn \l__box_internal_box
          {
            \box_move_down:nn \c_zero_dim
              { \box_use_drop:N \l__box_internal_box }
          }
        \box_set_dp:Nn \l__box_internal_box { - \__box_dim_eval:n {#3} }
      }
      {
        \hbox_set:Nn \l__box_internal_box
          { \box_move_down:nn {#3} { \box_use_drop:N \l__box_internal_box } }
        \box_set_dp:Nn \l__box_internal_box \c_zero_dim
      }
    \dim_compare:nNnTF {#5} > \c_zero_dim
      {
        \hbox_set:Nn \l__box_internal_box
          {
            \box_move_up:nn \c_zero_dim
              { \box_use_drop:N \l__box_internal_box }
          }
        \box_set_ht:Nn \l__box_internal_box
          {
            (#5)
            \dim_compare:nNnT {#3} > \c_zero_dim
              { - (#3) }
          }
      }
      {
        \hbox_set:Nn \l__box_internal_box
          {
            \box_move_up:nn { - \__box_dim_eval:n {#5} }
              { \box_use_drop:N \l__box_internal_box }
          }
        \box_set_ht:Nn \l__box_internal_box \c_zero_dim
      }
    #6 #1 \l__box_internal_box
  }
%% File: l3color.dtx
\cs_new_eq:NN \color_group_begin: \group_begin:
\cs_new_eq:NN \color_group_end:   \group_end:
\cs_new_protected:Npn \color_ensure_current:
  { \__color_select:N \l__color_current_tl }
\scan_new:N \s__color_stop
\cs_new_protected:Npn \__color_select:N #1
  {
    \exp_after:wN \__color_select:nn #1
    \group_insert_after:N \__color_backend_reset:
  }
\cs_new_protected:Npn \__color_select_math:N #1
  { \exp_after:wN \__color_select:nn #1 }
\cs_new_protected:Npn \__color_select:nn #1#2
  { \use:c { __color_backend_select_ #1 :n } {#2} }
\tl_new:N \l__color_current_tl
\tl_set:Nn \l__color_current_tl { { gray } { 0 } }
\int_new:N \l__color_internal_int
\tl_new:N \l__color_internal_tl
\scan_new:N \s__color_mark
\bool_new:N \l__color_ignore_error_bool
\prg_new_conditional:Npnn \color_if_exist:n #1 { p , T, F, TF }
  {
    \prop_if_exist:cTF { l__color_named_ #1 _prop }
      {
        \prop_if_empty:cTF { l__color_named_ #1 _prop }
          \prg_return_false:
          \prg_return_true:
      }
      \prg_return_false:
  }
\cs_new:Npn \__color_model:N #1 { \exp_after:wN \use_i:nn #1 }
\cs_new:Npn \__color_values:N #1 { \exp_after:wN \use_ii:nn #1 }
\cs_new_protected:Npn \__color_extract:nNN #1#2#3
  {
    \tl_set_eq:Nc #2 { l__color_named_ #1 _tl }
    \prop_get:cVN { l__color_named_ #1 _prop } #2 #3
  }
\cs_generate_variant:Nn \__color_extract:nNN { V }
\cs_new_protected:Npn \__color_convert:nnN #1#2#3
  { \__color_convert:nnVN {#1} {#2} #3 #3 }
\cs_generate_variant:Nn \__color_convert:nnN { VV }
\cs_generate_variant:Nn \exp_last_unbraced:Nf { c }
\cs_new_protected:Npn \__color_convert:nnnN #1#2#3#4
```

```
  {
    \tl_set:Ne #4
      {
        \cs_if_exist_use:cTF { __color_convert_ #1 _ #2 :w }
          { #3 \s__color_stop }
          {
            \cs_if_exist:cTF { __color_convert_ \use:c { c__color_fallback_ #1 _tl } _ #2 :w }
              {
                \exp_last_unbraced:cf
                  { __color_convert_ \use:c { c__color_fallback_ #1 _tl } _ #2 :w }
                  { \use:c { __color_convert_ #1 _ \use:c { c__color_fallback_ #1 _tl } :w } #3 \s__color_stop }
                  \s__color_stop
              }
              {
                \exp_last_unbraced:cf
                  { __color_convert_ \use:c { c__color_fallback_ #2 _tl } _ #2 :w }
                  {
                    \cs_if_exist_use:cTF { __color_convert_ #1 _ \use:c { c__color_fallback_ #2 _tl } :w }
                      { #3 \s__color_stop }
                      {
                        \exp_last_unbraced:cf
                          { __color_convert_ \use:c { c__color_fallback_ #1 _tl } _ \use:c { c__color_fallback_ #2 _tl } :w }
                          { \use:c { __color_convert_ #1 _ \use:c { c__color_fallback_ #1 _tl } :w } #3 \s__color_stop }
                          \s__color_stop
                      }
                  }
                  \s__color_stop
              }
          }
      }
  }
\cs_generate_variant:Nn \__color_convert:nnnN { nV , nnV }
\cs_new:Npn \__color_convert_gray_gray:w #1 \s__color_stop
  { #1 }
\cs_new:Npn \__color_convert_gray_rgb:w #1 \s__color_stop
  { #1 ~ #1 ~ #1 }
\cs_new:Npn \__color_convert_gray_cmyk:w #1 \s__color_stop
  { 0 ~ 0 ~ 0 ~ \fp_eval:n { 1 - #1 } }
\cs_new:Npn \__color_convert_rgb_gray:w #1 ~ #2 ~ #3 \s__color_stop
  { \fp_eval:n { 0.3 * #1 + 0.59 * #2 + 0.11 * #3 } }
\cs_new:Npn \__color_convert_rgb_rgb:w #1 \s__color_stop
```

```
  { #1 }
\cs_new:Npn \__color_convert_rgb_cmyk:w #1 ~ #2 ~ #3 \s__color_stop
  {
    \exp_args:Neee \__color_convert_rgb_cmyk:nnn
      { \fp_eval:n { 1 - #1 } }
      { \fp_eval:n { 1 - #2 } }
      { \fp_eval:n { 1 - #3 } }
  }
\cs_new:Npn \__color_convert_rgb_cmyk:nnn #1#2#3
  {
    \exp_args:Ne \__color_convert_rgb_cmyk:nnnn
      { \fp_eval:n { min( #1, #2 , #3 ) } } {#1} {#2} {#3}
  }
\cs_new:Npn \__color_convert_rgb_cmyk:nnnn #1#2#3#4
  {
    \fp_eval:n { min ( 1 , max ( 0 , #2 - #1 ) ) } \c_space_tl
    \fp_eval:n { min ( 1 , max ( 0 , #3 - #1 ) ) } \c_space_tl
    \fp_eval:n { min ( 1 , max ( 0 , #4 - #1 ) ) } \c_space_tl
    #1
  }
\cs_new:Npn \__color_convert_cmyk_gray:w #1 ~ #2 ~ #3 ~ #4 \s__color_stop
  { \fp_eval:n { 1 - min ( 1 , 0.3 * #1 + 0.59 * #2 + 0.11 * #3 + #4 ) } }
\cs_new:Npn \__color_convert_cmyk_rgb:w #1 ~ #2 ~ #3 ~ #4 \s__color_stop
  {
    \fp_eval:n { 1 - min ( 1 , #1 + #4 ) } \c_space_tl
    \fp_eval:n { 1 - min ( 1 , #2 + #4 ) } \c_space_tl
    \fp_eval:n { 1 - min ( 1 , #3 + #4 ) }
  }
\cs_new:Npn \__color_convert_cmyk_cmyk:w #1 \s__color_stop
  { #1 }
\tl_new:N \l__color_model_tl
\tl_new:N \l__color_value_tl
\tl_new:N \l__color_next_model_tl
\tl_new:N \l__color_next_value_tl
\cs_new_protected:Npe \__color_parse:nN #1#2
  {
    \tl_set:Ne \exp_not:c { l__color_named_ . _tl }
      { \exp_not:N \__color_model:N \exp_not:N \l__color_current_tl }
    \prop_put:NVe \exp_not:c { l__color_named_ . _prop }
      \exp_not:c { l__color_named_ . _tl }
      { \exp_not:N \__color_values:N \exp_not:N \l__color_current_tl }
    \exp_not:N \exp_args:Ne \exp_not:N \__color_parse_aux:nN
      { \exp_not:N \tl_to_str:n {#1} } #2
  }
\cs_new_protected:Npn \__color_parse_aux:nN #1#2
  {
    \color_if_exist:nTF {#1}
      { \__color_parse_set_eq:Nn #2 {#1} }
      { \__color_parse:Nw #2#1 ! \s__color_stop }
    \__color_check_model:N #2
  }
\cs_new_protected:Npn \__color_parse_set_eq:Nn #1#2
  {
    \tl_if_empty:NTF \l_color_fixed_model_tl
      { \exp_args:Nv \__color_parse_set_eq:nNn { l__color_named_ #2 _tl } }
      { \exp_args:NV \__color_parse_set_eq:nNn \l_color_fixed_model_tl }
        #1 {#2}
  }
\cs_new_protected:Npn \__color_parse_set_eq:nNn #1#2#3
  {
    \prop_get:cnNTF
      { l__color_named_ #3 _prop } {#1}
      \l__color_value_tl
      { \tl_set:Ne #2 { {#1} { \l__color_value_tl } } }
      {
        \tl_set_eq:Nc \l__color_model_tl { l__color_named_ #3 _tl }
        \prop_get:cVN { l__color_named_ #3 _prop } \l__color_model_tl
          \l__color_value_tl
        \__color_convert:nnN
          \l__color_model_tl {#1} \l__color_value_tl
        \tl_set:Ne #2
          {
            {#1}
            { \l__color_value_tl }
          }
      }
  }
\cs_new_protected:Npn \__color_parse:Nw #1#2 ! #3 \s__color_stop
  {
    \color_if_exist:nTF {#2}
      {
        \tl_if_blank:nTF {#3}
          { \__color_parse_set_eq:Nn #1 {#2} }
          { \__color_parse_loop_init:Nnn #1 {#2} {#3} }
      }
      {
        \msg_error:nnn { color } { unknown-color } {#2}
        \tl_set:Nn \l__color_current_tl { { gray } { 0 } }
      }
  }
\cs_new_protected:Npn \__color_parse_loop_init:Nnn #1#2#3
  {
    \group_begin:
      \__color_extract:nNN {#2} \l__color_model_tl \l__color_value_tl
      \__color_parse_loop:w #3 ! ! ! ! \s__color_stop
      \tl_set:Ne \l__color_internal_tl
        { { \l__color_model_tl } { \l__color_value_tl } }
    \exp_args:NNNV \group_end:
    \tl_set:Nn #1 \l__color_internal_tl
  }
\cs_new_protected:Npn \__color_parse_loop:w #1 ! #2 ! #3 ! #4 ! #5 \s__color_stop
  {
    \tl_if_blank:nF {#1}
      {
        \bool_lazy_and:nnTF
          { \fp_compare_p:nNn {#1} > { 0 } }
          { \fp_compare_p:nNn {#1} < { 100 } }
          {
            \use:e
              {
                \__color_parse_loop:nn {#1}
                  { \tl_if_blank:nTF {#2} { white } {#2} }
              }
          }
          { \__color_parse_loop_check:nn {#1} {#2} }
      }
    \tl_if_blank:nF {#3}
      { \__color_parse_loop:w #3 ! #4 ! #5 \s__color_stop }
    \__color_parse_end:
  }
\cs_new_protected:Npn \__color_parse_loop_check:nn #1#2
  {
    \bool_if:NF \l__color_ignore_error_bool
      {
        \bool_lazy_or:nnT
          { \fp_compare_p:nNn {#1} < { 0 } }
          { \fp_compare_p:nNn {#1} > { 100 } }
          { \msg_error:nnnnn { color } { out-of-range } {#1} { 0 } { 100 } }
      }
    \fp_compare:nNnF {#1} > \c_zero_fp
      {
        \tl_if_blank:nTF {#2}
          { \__color_extract:nNN { white } }
          { \__color_extract:nNN {#2} }
            \l__color_model_tl \l__color_value_tl
      }
  }
\cs_new_protected:Npn \__color_parse_loop:nn #1#2
```

```
  {
    \color_if_exist:nTF {#2}
      {
        \__color_extract:nNN {#2} \l__color_next_model_tl \l__color_next_value_tl
        \tl_if_eq:NNF \l__color_model_tl \l__color_next_model_tl
          {
            \str_if_eq:VnTF \l__color_model_tl { gray }
              { \__color_parse_gray:n {#2} }
              { \__color_parse_std:n {#2} }
          }
        \tl_set:Ne \l__color_value_tl
          {
            \__color_parse_mix:NVVn
              \l__color_model_tl \l__color_value_tl \l__color_next_value_tl {#1}
          }
      }
      {
        \msg_error:nnn { color } { unknown-color } {#2}
        \__color_extract:nNN { black } \l__color_model_tl \l__color_value_tl
        \__color_parse_break:w
      }
  }
\cs_new_protected:Npn \__color_parse_gray:n #1
  {
    \tl_set_eq:NN \l__color_model_tl \l__color_next_model_tl
    \tl_set:Nn \l__color_next_model_tl { gray }
    \exp_args:NnV \__color_convert:nnN { gray } \l__color_model_tl
      \l__color_value_tl
    \prop_get:cVN { l__color_named_ #1 _prop } \l__color_model_tl
      \l__color_next_value_tl
  }
\cs_new_protected:Npn \__color_parse_std:n #1
  {
    \prop_get:cVNF { l__color_named_ #1 _prop }
      \l__color_model_tl
      \l__color_next_value_tl
        {
          \__color_convert:VVN
            \l__color_next_model_tl
            \l__color_model_tl
            \l__color_next_value_tl
        }
  }
\cs_new_protected:Npn \__color_parse_break:w #1 \__color_parse_end: { }
\cs_new_protected:Npn \__color_parse_end: { }
\cs_new:Npn \__color_parse_mix:Nnnn #1#2#3#4
```

```
  {
    \exp_args:Nf \__color_parse_mix:nNnn
      { \fp_eval:n { #4 / 100 } }
      #1 {#2} {#3}
  }
\cs_generate_variant:Nn \__color_parse_mix:Nnnn { NVV }
\cs_new:Npn \__color_parse_mix:nNnn #1#2#3#4
  {
    \use:c { __color_parse_mix_ #2 :nw } {#1}
      #3 \s__color_mark #4 \s__color_stop
  }
\cs_new:Npn \__color_parse_mix_gray:nw #1#2 \s__color_mark #3 \s__color_stop
  { \fp_eval:n { #2 * #1 + #3 * ( 1 - #1 ) } }
\cs_new:Npn \__color_parse_mix_rgb:nw
  #1#2 ~ #3 ~ #4 \s__color_mark #5 ~ #6 ~ #7 \s__color_stop
  {
    \fp_eval:n { #2 * #1 + #5 * ( 1 - #1 ) } \c_space_tl
    \fp_eval:n { #3 * #1 + #6 * ( 1 - #1 ) } \c_space_tl
    \fp_eval:n { #4 * #1 + #7 * ( 1 - #1 ) }
  }
\cs_new:Npn \__color_parse_mix_cmyk:nw
  #1#2 ~ #3 ~ #4 ~ #5 \s__color_mark #6 ~ #7 ~ #8 ~ #9 \s__color_stop
  {
    \fp_eval:n { #2 * #1 + #6 * ( 1 - #1 ) } \c_space_tl
    \fp_eval:n { #3 * #1 + #7 * ( 1 - #1 ) } \c_space_tl
    \fp_eval:n { #4 * #1 + #8 * ( 1 - #1 ) } \c_space_tl
    \fp_eval:n { #5 * #1 + #9 * ( 1 - #1 ) }
  }
\cs_new:Npn \__color_parse_model_gray:w #1 , #2 \s__color_stop
  { { gray } { \__color_parse_number:n {#1} } }
\cs_new:Npn \__color_parse_model_rgb:w #1 , #2 , #3 , #4 \s__color_stop
  {
    { rgb }
    {
      \__color_parse_number:n {#1} ~
      \__color_parse_number:n {#2} ~
      \__color_parse_number:n {#3}
    }
  }
\cs_new:Npn \__color_parse_model_cmyk:w #1 , #2 , #3 , #4 , #5 \s__color_stop
  {
    { cmyk }
    {
      \__color_parse_number:n {#1} ~
      \__color_parse_number:n {#2} ~
      \__color_parse_number:n {#3} ~
      \__color_parse_number:n {#4}
    }
  }
\cs_new:Npn \__color_parse_number:n #1
  {  \__color_parse_number:w #1 . 0 . \s__color_stop }
\cs_new:Npn \__color_parse_number:w #1 . #2 . #3 \s__color_stop
  { \tl_if_blank:nTF {#1} { 0 } {#1} . #2 }
\cs_new:Npn \__color_parse_model_Gray:w #1 , #2 \s__color_stop
  { { gray } { \fp_eval:n { #1 / 15 } } }
\cs_new:Npn \__color_parse_model_hsb:w #1 , #2 , #3 , #4 \s__color_stop
  { \__color_parse_model_hsb:nnn {#1} {#2} {#3} }
\cs_new:Npn \__color_parse_model_Hsb:w #1 , #2 , #3 , #4 \s__color_stop
  {
    \exp_args:Ne \__color_parse_model_hsb:nnn { \fp_eval:n { #1 / 360 } }
      {#2} {#3}
  }
\cs_new:Npn \__color_parse_model_hsb:nnn #1#2#3
  {
    { rgb }
    {
      \exp_args:Ne \__color_parse_model_hsb_aux:nnn
        { \fp_eval:n { 6 * (#1) } } {#2} {#3}
    }
  }
\cs_new:Npn \__color_parse_model_hsb_aux:nnn #1#2#3
  {
    \exp_args:Nee \__color_parse_model_hsb_aux:nnnn
      { \fp_eval:n { floor(#1) } } { \fp_eval:n { #1 - floor(#1) } }
      {#2} {#3}
  }
\cs_new:Npn \__color_parse_model_hsb_aux:nnnn #1#2#3#4
  {
    \use:e
      {
        \exp_not:N \__color_parse_model_hsb_aux:nnnnn
          { \__color_parse_number:n {#4} }
          { \fp_eval:n { round(#4 * (1 - #3) ,5) } }
          { \fp_eval:n { round(#4 * ( 1 - #3 * #2 ) ,5) } }
          { \fp_eval:n { round(#4 * ( 1 - #3 * (1 - #2) ) ,5) } }
          {#1}
      }
  }
\cs_new:Npn \__color_parse_model_hsb_aux:nnnnn #1#2#3#4#5
  { \use:c { __color_parse_model_hsb_ #5 :nnnn } {#1} {#2} {#3} {#4} }
\cs_new:cpn { __color_parse_model_hsb_0:nnnn } #1#2#3#4 { #1 ~ #4 ~ #2 }
\cs_new:cpn { __color_parse_model_hsb_1:nnnn } #1#2#3#4 { #3 ~ #1 ~ #2 }
```

```
\cs_new:cpn { __color_parse_model_hsb_2:nnnn } #1#2#3#4 { #2 ~ #1 ~ #4 }
\cs_new:cpn { __color_parse_model_hsb_3:nnnn } #1#2#3#4 { #2 ~ #3 ~ #1 }
\cs_new:cpn { __color_parse_model_hsb_4:nnnn } #1#2#3#4 { #4 ~ #2 ~ #1 }
\cs_new:cpn { __color_parse_model_hsb_5:nnnn } #1#2#3#4 { #1 ~ #2 ~ #3 }
\cs_new:cpn { __color_parse_model_hsb_6:nnnn } #1#2#3#4 { #1 ~ #2 ~ #2 }
\cs_new:Npn \__color_parse_model_HSB:w #1 , #2 , #3 , #4 \s__color_stop
  {
    \exp_args:Neee \__color_parse_model_hsb:nnn
      { \fp_eval:n { round((#1) / 240,5) } }
      { \fp_eval:n { round((#2) / 240,5) } }
      { \fp_eval:n { round((#3) / 240,5) } }
  }
\cs_new:Npn \__color_parse_model_HTML:w #1 , #2 \s__color_stop
  { \__color_parse_model_HTML_aux:w #1 0 0 0 0 0 0 \s__color_stop }
\cs_new:Npn \__color_parse_model_HTML_aux:w #1#2#3#4#5#6#7 \s__color_stop
  {
    { rgb }
    {
      \fp_eval:n { round(\int_from_hex:n {#1#2} / 255,5) } ~
      \fp_eval:n { round(\int_from_hex:n {#3#4} / 255,5) } ~
      \fp_eval:n { round(\int_from_hex:n {#5#6} / 255,5) }
    }
  }
\cs_new:Npn \__color_parse_model_RGB:w #1 , #2 , #3 , #4 \s__color_stop
  {
    { rgb }
    {
      \fp_eval:n { round((#1) / 255,5) } ~
      \fp_eval:n { round((#2) / 255,5) } ~
      \fp_eval:n { round((#3) / 255,5) }
    }
  }
\cs_new:Npn \__color_parse_model_wave:w #1 , #2 \s__color_stop
  {
    { rgb }
    {
      \fp_compare:nNnTF {#1} < { 420 }
        { \__color_parse_model_wave_auxi:nn {#1} { 0.3 + 0.7 * (#1 - 380) / 40 }
        }
        {
          \fp_compare:nNnTF {#1} > { 700 }
            { \__color_parse_model_wave_auxi:nn {#1} { 0.3 + 0.7 * (#1 - 780) / -80 } }
            { \__color_parse_model_wave_auxi:nn {#1} { 1 } }
        }
    }
  }
```

```
\cs_new:Npn \__color_parse_model_wave_auxi:nn #1#2
  {
    \fp_compare:nNnTF {#1} < { 440 }
      {
        \__color_parse_model_wave_auxii:nn
          { 4 + \__color_parse_model_wave_rho:n { (#1 - 440) / -60 } }
          {#2}
      }
      {
        \fp_compare:nNnTF {#1} < { 490 }
          {
            \__color_parse_model_wave_auxii:nn
              { 4 - \__color_parse_model_wave_rho:n { (#1 - 440) / 50 } }
              {#2}
          }
          {
            \fp_compare:nNnTF {#1} < { 510 }
              {
                \__color_parse_model_wave_auxii:nn
                  { 2 + \__color_parse_model_wave_rho:n { (#1 - 510) / -20 } }
                  {#2}
              }
              {
                \fp_compare:nNnTF {#1} < { 580 }
                  {
                    \__color_parse_model_wave_auxii:nn
                      { 2 - \__color_parse_model_wave_rho:n { (#1 - 510) / 70 } }
                      {#2}
                  }
                  {
                    \fp_compare:nNnTF {#1} < { 645 }
                      {
                        \__color_parse_model_wave_auxii:nn
                          { \__color_parse_model_wave_rho:n { (#1 - 645) / -65 } }
                          {#2}
                      }
                      { \__color_parse_model_wave_auxii:nn { 0 } {#2} }
                  }
              }
          }
      }
  }
\cs_new:Npn \__color_parse_model_wave_auxii:nn #1#2
  {
    \exp_args:Neee \__color_parse_model_hsb_aux:nnn
      { \fp_eval:n {#1} }
      { 1 }
      { \__color_parse_model_wave_rho:n {#2} }
  }
\cs_new:Npn \__color_parse_model_wave_rho:n #1
  { \fp_eval:n { min(1, max(0,#1) ) } }
\cs_new:Npn \__color_parse_model_cmy:w #1 , #2 , #3 , #4 \s__color_stop
  {
    { cmyk }
    { \__color_convert_rgb_cmyk:nnn {#1} {#2} {#3} }
  }
\cs_new:Npn \__color_parse_model_tHsb:w #1 , #2 , #3 , #4 \s__color_stop
  {
    \exp_args:Ne \__color_parse_model_hsb:nnn
      { \__color_parse_model_tHsb:n {#1} } {#2} {#3}
  }
\cs_new:Npn \__color_parse_model_tHsb:n #1
  {
    \__color_parse_model_tHsb:nw {#1}
        0 ,   0 ;
       60 ,  30 ;
      120 ,  60 ;
      180 , 120 ;
      210 , 180 ;
      240 , 240 ;
      360 , 360 ;
      \q_recursion_tail , ;
      \q_recursion_stop
  }
\cs_new:Npn \__color_parse_model_tHsb:nw #1 #2 , #3 ; #4 , #5 ;
  {
    \quark_if_recursion_tail_stop_do:nn {#4} { 0 }
    \fp_compare:nNnTF {#1} > {#4}
      { \__color_parse_model_tHsb:nw {#1} #4 , #5 ; }
      {
        \use_i_delimit_by_q_recursion_stop:nw
          { \fp_eval:n { ((#1 - #2) / (#4 - #2) * (#5 - #3) + #3) / 360 } }
      }
  }
\cs_new:cpn { __color_parse_model_&spot:w } #1 , #2 \s__color_stop
  { { gray } { #1 } }
\tl_new:N \l_color_fixed_model_tl
\cs_new_protected:Npn \__color_check_model:N #1
  {
    \tl_if_empty:NF \l_color_fixed_model_tl
      {
        \exp_after:wN \__color_check_model:nn #1
        \tl_if_eq:NNF \l__color_model_tl \l_color_fixed_model_tl
          {
            \__color_convert:VVN \l__color_model_tl \l_color_fixed_model_tl
              \l__color_value_tl
          }
        \tl_set:Ne #1
          { { \l_color_fixed_model_tl } { \l__color_value_tl } }
      }
  }
\cs_new_protected:Npn \__color_check_model:nn #1#2
  {
    \tl_set:Nn \l__color_model_tl {#1}
    \tl_set:Nn \l__color_value_tl {#2}
  }
\cs_new_protected:Npe \__color_finalise_current:
  {
    \tl_set:Ne \exp_not:c { l__color_named_ . _tl }
      { \exp_not:N \__color_model:N \exp_not:N \l__color_current_tl }
    \prop_clear:N \exp_not:c { l__color_named_ . _prop }
    \prop_put:NVe \exp_not:c { l__color_named_ . _prop }
      \exp_not:c { l__color_named_ . _tl }
      { \exp_not:N \__color_values:N \exp_not:N \l__color_current_tl }
  }
\cs_new_protected:Npn \color_select:n #1
  {
    \__color_parse:nN {#1} \l__color_current_tl
    \__color_finalise_current:
    \__color_select:N \l__color_current_tl
  }
\cs_generate_variant:Nn \color_select:n { V }
\cs_new_protected:Npn \color_select:nn #1#2
  {
    \__color_select_main:Nnn \l__color_current_tl {#1} {#2}
    \__color_finalise_current:
    \__color_select:N \l__color_current_tl
  }
\cs_generate_variant:Nn \color_select:nn { nV , V , VV }
\cs_new_protected:Npn \__color_select_main:Nnn #1#2#3
  {
    \use:e
      {
        \exp_not:N \__color_select_main:Nw \exp_not:N #1
          \exp_not:n {#2} / / \exp_not:N \s__color_mark
          #3 / / \exp_not:N \s__color_stop
      }
  }
```

```
\cs_new_protected:Npn \__color_select_main:Nw
  #1 #2 / #3 / #4 \s__color_mark #5 / #6 / #7 \s__color_stop
  {
    \__color_select:nnN {#2} {#5} #1
    \bool_lazy_or:nnF
      { \tl_if_empty_p:N \l_color_fixed_model_tl }
      { \str_if_eq_p:nV {#2} \l_color_fixed_model_tl }
      { \__color_select_loop:Nw #1 #3 / #4 \s__color_mark #6 / #7 \s__color_stop }
  }
\cs_new_protected:Npn \__color_select_loop:Nw
  #1 #2 / #3 \s__color_mark #4 / #5 \s__color_stop
  {
    \str_if_eq:nVTF {#2} \l_color_fixed_model_tl
      { \__color_select:nnN {#2} {#4} #1 }
      {
        \tl_if_blank:nTF {#2}
          { \exp_after:wN \__color_select_swap:Nnn \exp_after:wN #1 #1 }
          { \__color_select_loop:Nw #1 #3 \s__color_mark #5 \s__color_stop }
      }
  }
\cs_new_protected:Npn \__color_select:nnN #1#2#3
  {
    \cs_if_exist:cTF { __color_parse_model_ #1 :w }
      {
        \tl_set:Ne #3
          { \use:c { __color_parse_model_ #1 :w } #2 , 0 , 0 , 0 , 0 \s__color_stop }
      }
      { \msg_error:nnn { color } { unknown-model } {#1} }
  }
\cs_new_protected:Npn \__color_select_swap:Nnn #1#2#3
  {
    \__color_convert:nVnN {#2} \l_color_fixed_model_tl {#3} \l__color_value_tl
    \tl_set:Ne #1
      { { \l_color_fixed_model_tl } { \l__color_value_tl } }
  }
\tl_new:N \l_color_math_active_tl
\tl_set:Nn \l_color_math_active_tl { ' }
\seq_new:N \g__color_math_seq
\cs_new_protected:Npn \color_math:nn #1#2
  {
    \__color_math:nn {#2}
      { \__color_parse:nN {#1} \l__color_current_tl }
  }
\cs_new_protected:Npn \color_math:nnn #1#2#3
  {
    \__color_math:nn {#3}
```

```
      { \__color_select_main:Nnn \l__color_current_tl {#1} {#2} }
  }
\cs_new_protected:Npn \__color_math:nn #1#2
  {
    \seq_gpush:NV \g__color_math_seq \l__color_current_tl
    #2
    \__color_select_math:N \l__color_current_tl
    #1
    \__color_math_scan:w
  }
\cs_new_protected:Npn \__color_math_scan:w
  {
    \peek_remove_filler:n
      {
        \group_align_safe_begin:
        \peek_catcode:NTF \c_alignment_token
          {
            \group_align_safe_end:
            \__color_math_scan_end:
          }
          {
            \group_align_safe_end:
            \__color_math_scan_auxi:
          }
      }
  }
\cs_new_protected:Npn \__color_math_scan_auxi:
  {
    \token_case_catcode:NnTF \l_peek_token
      {
        \c_math_subscript_token   { }
        \c_math_superscript_token { }
      }
      { \__color_math_scripts:Nw }
      {
        \token_case_meaning:NnTF \l_peek_token
          {
            \tex_limits:D        { \tex_limits:D }
            \tex_nolimits:D      { \tex_nolimits:D }
            \tex_displaylimits:D { \tex_displaylimits:D }
          }
          { \__color_math_scan:w \use_none:n }
          { \__color_math_scan_auxii: }
      }
  }
\cs_new_protected:Npn \__color_math_scan_auxii:
  {
    \tl_map_inline:Nn \l_color_math_active_tl
      {
        \token_if_eq_meaning:NNT \l_peek_token ##1
          {
            \tl_map_break:n
              {
                \use_i:nn
                  { \__color_math_scan_auxiii:N ##1 }
              }
          }
        \__color_math_scan_end:
      }
  }
\cs_new_protected:Npn \__color_math_scan_auxiii:N #1
  {
    \exp_after:wN \exp_after:wN \exp_after:wN \__color_math_scan:w
      \char_generate:nn { `#1 } { 13 }
  }
\cs_new_protected:Npn \__color_math_scan_end:
  {
    \__color_backend_reset:
    \seq_gpop:NN \g__color_math_seq \l__color_current_tl
  }
\cs_new_protected:Npn \__color_math_scripts:Nw #1
  {
    #1
    \c_group_begin_token
      \c_group_begin_token
        \seq_get:NN \g__color_math_seq \l__color_current_tl
        \__color_select:N \l__color_current_tl
        \group_insert_after:N \c_group_end_token
        \group_insert_after:N \__color_math_scan:w
    \peek_remove_filler:n
      {
        \peek_catcode_remove:NF \c_group_begin_token
          { \__color_math_script_aux:N }
      }
  }
\cs_new_protected:Npn \__color_math_script_aux:N #1 { #1 \c_group_end_token }
\cs_new_protected:Npn \color_fill:n #1
  {
    \__color_parse:nN {#1} \l__color_current_tl
    \exp_after:wN \__color_draw:nnn \l__color_current_tl { fill }
  }
\cs_new_protected:Npn \color_stroke:n #1
```

```
  {
    \__color_parse:nN {#1} \l__color_current_tl
    \exp_after:wN \__color_draw:nnn \l__color_current_tl { stroke }
  }
\cs_new_protected:Npn \color_fill:nn #1#2
  {
    \__color_select_main:Nnn \l__color_current_tl {#1} {#2}
    \exp_after:wN \__color_draw:nnn \l__color_current_tl { fill }
  }
\cs_new_protected:Npn \color_stroke:nn #1#2
  {
    \__color_select_main:Nnn \l__color_current_tl {#1} {#2}
    \exp_after:wN \__color_draw:nnn \l__color_current_tl { stroke }
  }
\cs_new_protected:Npn \__color_draw:nnn #1#2#3
  {
    \use:c { __color_backend_ #3 _ #1 :n } {#2}
    \exp_args:Nc \group_insert_after:N { __color_backend_ #3 _ reset: }
  }
\tl_new:N \l__color_named_tl
\cs_new_protected:Npn \color_set:nn #1#2
  {
    \exp_args:NV \__color_set:nnn
      \l_color_fixed_model_tl {#1} {#2}
  }
\cs_new_protected:Npn \__color_set:nnn #1#2#3
  {
    \tl_clear:N \l_color_fixed_model_tl
    \__color_set:nn {#2} {#3}
    \tl_set:Nn \l_color_fixed_model_tl {#1}
  }
\cs_new_protected:Npn \__color_set:nn #1#2
  {
    \str_if_eq:nnF {#1} { . }
      {
        \__color_parse:nN {#2} \l__color_named_tl
        \tl_clear_new:c { l__color_named_ #1 _tl }
        \tl_set:ce { l__color_named_ #1 _tl }
          { \__color_model:N \l__color_named_tl }
        \prop_clear_new:c { l__color_named_ #1 _prop }
        \prop_put:cve { l__color_named_ #1 _prop } { l__color_named_ #1 _tl }
          { \__color_values:N \l__color_named_tl }
        \__color_set:nnw {#1} {#2} #2 ! \s__color_stop
      }
  }
\cs_new_protected:Npn \__color_set:nnw #1#2#3 ! #4 \s__color_stop
```

```
  {
    \clist_map_inline:nn { cmyk , gray , rgb }
      {
        \prop_get:cnNT { l__color_named_ #3 _prop } {##1} \l__color_internal_tl
          {
            \prop_if_in:cnF { l__color_named_ #1 _prop } {##1}
              {
                \group_begin:
                  \bool_set_true:N \l__color_ignore_error_bool
                  \tl_set:cn { l__color_named_ #3 _tl } {##1}
                  \__color_parse:nN {#2} \l__color_internal_tl
                \exp_args:NNNV \group_end:
                \tl_set:Nn \l__color_internal_tl \l__color_internal_tl
                \prop_put:cee { l__color_named_ #1 _prop }
                  { \__color_model:N \l__color_internal_tl }
                  { \__color_values:N \l__color_internal_tl }
              }
          }
      }
  }
\cs_new_protected:Npn \color_set:nnn #1#2#3
  {
    \str_if_eq:nnF {#1} { . }
      {
        \tl_clear_new:c { l__color_named_ #1 _tl }
        \prop_clear_new:c { l__color_named_ #1 _prop }
        \exp_args:Ne \__color_set_aux:nnn { \tl_to_str:n {#2} }
          {#1} {#3}
      }
  }
\cs_new_protected:Npe \__color_set_aux:nnn #1#2#3
  {
    \exp_not:N \__color_set_colon:nnw {#2} {#3}
      #1 \c_colon_str \c_colon_str \exp_not:N \s__color_stop
  }
\use:e
  {
    \cs_new_protected:Npn \exp_not:N \__color_set_colon:nnw
      #1#2 #3 \c_colon_str #4 \c_colon_str
      #5 \exp_not:N \s__color_stop
  }
  {
    \tl_if_blank:nTF {#4}
      { \__color_set_loop:nw {#1} #3 }
      { \__color_set_loop:nw {#1} #4 }
        / / \s__color_mark #2 / / \s__color_stop
  }
\cs_new_protected:Npn \__color_set_loop:nw
  #1#2 / #3 \s__color_mark #4 / #5 \s__color_stop
  {
    \tl_if_blank:nF {#2}
      {
        \__color_select:nnN {#2} {#4} \l__color_named_tl
        \tl_set:Ne \l__color_internal_tl { \__color_model:N \l__color_named_tl }
        \tl_if_empty:cT { l__color_named_ #1 _tl }
          { \tl_set_eq:cN { l__color_named_ #1 _tl } \l__color_internal_tl }
        \prop_put:cVe { l__color_named_ #1 _prop } \l__color_internal_tl
          { \__color_values:N \l__color_named_tl }
        \__color_set_loop:nw {#1} #3 \s__color_mark #5 \s__color_stop
      }
  }
\cs_new_protected:Npn \color_set_eq:nn #1#2
  {
    \color_if_exist:nTF {#2}
      {
        \tl_clear_new:c { l__color_named_ #1 _tl }
        \prop_clear_new:c { l__color_named_ #1 _prop }
        \str_if_eq:nnTF {#2} { . }
          {
            \tl_set:ce { l__color_named_ #1 _tl }
              { \__color_model:N \l__color_current_tl }
            \prop_put:cve { l__color_named_ #1 _prop } { l__color_named_ #1 _tl }
              { \__color_values:N \l__color_current_tl }
          }
          {
            \tl_set_eq:cc { l__color_named_ #1 _tl } { l__color_named_ #2 _tl }
            \prop_set_eq:cc { l__color_named_ #1 _prop } { l__color_named_ #2 _prop }
          }
      }
      {
        \msg_error:nnn { color } { unknown-color } {#2}
      }
  }
\color_set:nnn { black } { gray } { 0 }
\color_set:nnn { white } { gray } { 1 }
\color_set:nnn { cyan }    { cmyk } { 1 , 0 , 0 , 0 }
\color_set:nnn { magenta } { cmyk } { 0 , 1 , 0 , 0 }
\color_set:nnn { yellow }  { cmyk } { 0 , 0 , 1 , 0 }
\color_set:nnn { red }   { rgb } { 1 , 0 , 0 }
\color_set:nnn { green } { rgb } { 0 , 1 , 0 }
\color_set:nnn { blue }  { rgb } { 0 , 0 , 1 }
\prop_new:c { l__color_named_._prop }
```

```
\tl_new:c { l__color_named_._tl }
\tl_set:ce { l__color_named_._tl } { \__color_model:N \l__color_current_tl }
\prop_put:cve { l__color_named_._prop } { l__color_named_._tl }
  { \__color_values:N \l__color_current_tl }
\cs_new_protected:Npn \color_export:nnN #1#2#3
  {
    \group_begin:
      \tl_if_exist:cT { c__color_export_ #2 _tl }
        { \tl_set_eq:Nc \l_color_fixed_model_tl { c__color_export_ #2 _tl } }
      \__color_parse:nN {#1} #3
      \__color_export:nN {#2} #3
    \exp_args:NNNV \group_end:
    \tl_set:Nn #3 #3
  }
\cs_new_protected:Npn \color_export:nnnN #1#2#3#4
  {
    \__color_select_main:Nnn #4 {#1} {#2}
    \__color_export:nN {#3} #4
  }
\cs_new_protected:Npn \__color_export:nN #1#2
  { \exp_after:wN \__color_export:nnnN #2 {#1} #2 }
\cs_new:Npn \__color_export:nnnN #1#2#3#4
  {
    \cs_if_exist_use:cF { __color_export_format_ #3 :nnN }
      {
        \msg_error:nnn { color } { unknown-export-format } {#3}
        \use_none:nnn
      }
        {#1} {#2} #4
  }
\cs_new_protected:Npn \__color_export_format_backend:nnN #1#2#3
  { \tl_set:Nn #3 { {#1} {#2} } }
\cs_new_protected:Npn \__color_export:nnnNN #1#2#3#4#5
  {
    \str_if_eq:nnTF {#2} {#1}
      { #5 #4 #3 \s__color_stop }
      {
        \__color_convert:nnnN {#2} {#1} {#3} #4
        \exp_after:wN #5 \exp_after:wN #4
          #4 \s__color_stop
      }
  }
\tl_const:cn { c__color_export_comma-sep-cmyk_tl } { cmyk }
\tl_const:cn { c__color_export_comma-sep-rgb_tl } { rgb }
\tl_const:Nn \c__color_export_HTML_tl { rgb }
\tl_const:cn { c__color_export_space-sep-cmyk_tl } { cmyk }
```

```
\tl_const:cn { c__color_export_space-sep-rgb_tl } { rgb }
\group_begin:
  \cs_set_protected:Npn \__color_tmp:w #1#2
    {
      \cs_new_protected:cpe { __color_export_format_ #1 :nnN } ##1##2##3
        {
          \exp_not:N \__color_export:nnnNN {#2} {##1} {##2} ##3
            \exp_not:c { __color_export_ #1 :Nw }
        }
    }
  \__color_tmp:w { comma-sep-cmyk } { cmyk }
  \__color_tmp:w { comma-sep-rgb }  { rgb }
  \__color_tmp:w { HTML }           { rgb }
  \__color_tmp:w { space-sep-cmyk } { cmyk }
  \__color_tmp:w { space-sep-rgb }  { rgb }
\group_end:
\cs_new_protected:cpn { __color_export_comma-sep-cmyk:Nw }
  #1#2 ~ #3 ~ #4 ~ #5 \s__color_stop
  { \tl_set:Nn #1 { #2 , #3 , #4 , #5 } }
\cs_new_protected:cpn { __color_export_space-sep-cmyk:Nw } #1#2 \s__color_stop
  { \tl_set:Nn #1 {#2} }
\cs_new_protected:cpn { __color_export_comma-sep-rgb:Nw } #1#2 ~ #3 ~ #4 \s__color_stop
  { \tl_set:Ne #1 { #2 , #3 , #4 } }
\cs_new_protected:Npn \__color_export_HTML:Nw #1#2 ~ #3 ~ #4 \s__color_stop
  {
    \tl_set:Ne #1
      {
        \__color_export_HTML:n {#2}
        \__color_export_HTML:n {#3}
        \__color_export_HTML:n {#4}
      }
  }
\cs_new:Npn \__color_export_HTML:n #1
  {
    \fp_compare:nNnTF {#1} = { 0 }
      { 00 }
      {
        \fp_compare:nNnT { #1 * 255 } < { 16 } { 0 }
        \int_to_Hex:n { \fp_to_int:n { #1 * 255 } }
      }
  }
\cs_new_protected:cpn { __color_export_space-sep-rgb:Nw } #1#2 \s__color_stop
  { \tl_set:Nn #1 {#2} }
\prop_new:N \l__color_internal_prop
\int_new:N \g__color_model_int
\tl_const:Nn \c__color_fallback_cmyk_tl { cmyk }
```

```
\tl_const:Nn \c__color_fallback_gray_tl { gray }
\tl_const:Nn \c__color_fallback_rgb_tl { rgb }
\prop_new:N \g__color_colorants_prop
\prop_gput:Nnn \g__color_colorants_prop { black }   { Black }
\prop_gput:Nnn \g__color_colorants_prop { blue }    { Blue }
\prop_gput:Nnn \g__color_colorants_prop { cyan }    { Cyan }
\prop_gput:Nnn \g__color_colorants_prop { green }   { Green }
\prop_gput:Nnn \g__color_colorants_prop { magenta } { Magenta }
\prop_gput:Nnn \g__color_colorants_prop { none }    { None }
\prop_gput:Nnn \g__color_colorants_prop { red }     { Red }
\prop_gput:Nnn \g__color_colorants_prop { yellow }  { Yellow }
\tl_const:Nn \c__color_model_whitepoint_CIELAB_a_tl      { 1.0985 ~ 1 ~ 0.3558 }
\tl_const:Nn \c__color_model_whitepoint_CIELAB_b_tl      { 0.9807 ~ 1 ~ 1.1822 }
\tl_const:Nn \c__color_model_whitepoint_CIELAB_e_tl      { 1 ~ 1 ~ 1 }
\tl_const:cn { c__color_model_whitepoint_CIELAB_d50_tl } { 0.9642 ~ 1 ~ 0.8251 }
\tl_const:cn { c__color_model_whitepoint_CIELAB_d55_tl } { 0.9568 ~ 1 ~ 0.9214 }
\tl_const:cn { c__color_model_whitepoint_CIELAB_d65_tl } { 0.9504 ~ 1 ~ 1.0888 }
\tl_const:cn { c__color_model_whitepoint_CIELAB_d75_tl } { 0.9497 ~ 1 ~ 1.2261 }
\tl_const:Nn \c__color_model_range_CIELAB_tl { 0 ~ 100 ~ -128 ~ 127 ~ -128 ~ 127 }
\prop_new:N \g__color_alternative_model_prop
\clist_map_inline:nn { cyan , magenta , yellow , black }
  { \prop_gput:Nnn \g__color_alternative_model_prop {#1} { cmyk } }
\clist_map_inline:nn { red , green , blue }
  { \prop_gput:Nnn \g__color_alternative_model_prop {#1} { rgb } }
\prop_new:N \g__color_alternative_values_prop
\prop_gput:Nnn \g__color_alternative_values_prop { cyan }    {  1 , 0 , 0 , 0 }
\prop_gput:Nnn \g__color_alternative_values_prop { magenta } {  0 , 1 , 0 , 0 }
\prop_gput:Nnn \g__color_alternative_values_prop { yellow }  {  0 , 0 , 1 , 0 }
\prop_gput:Nnn \g__color_alternative_values_prop { black }   {  0 , 0 , 0 , 1 }
\prop_gput:Nnn \g__color_alternative_values_prop { red }   {  1 , 0 , 0 }
\prop_gput:Nnn \g__color_alternative_values_prop { green } {  0 , 1 , 0 }
\prop_gput:Nnn \g__color_alternative_values_prop { blue }  {  0 , 0 , 1 }
\cs_new_protected:Npn \color_model_new:nnn #1#2#3
  {
    \exp_args:Nee \__color_model_new:nnn
      { \tl_to_str:n {#1} }
      { \str_casefold:n {#2} } {#3}
  }
\cs_new_protected:Npn \__color_model_new:nnn #1#2#3
  {
    \cs_if_exist:cTF { __color_parse_model_ #1 :w }
      {
        \msg_error:nnn { color } { model-already-defined } {#1}
      }
      {
        \cs_if_exist:cTF { __color_model_ #2 :n }
```

```
     {
      \prop_set_from_keyval:Nn \l__color_internal_prop {#3}
      \use:c { __color_model_ #2 :n } {#1}
     }
     {
      \msg_error:nnn { color } { unknown-model-type } {#2}
     }
   }
 }
\cs_new_protected:Npn \__color_model_init:nnn #1#2#3
 {
  \int_gincr:N \g__color_model_int
  \clist_map_inline:nn { fill , stroke , select }
   {
    \cs_new_protected:cpe { __color_backend_ ##1 _ #1 :n } ####1
     {
      \exp_not:c { __color_backend_ ##1 _ #2 :nn }
       { color \int_use:N \g__color_model_int } {####1}
     }
   }
  \cs_new_protected:cpe { __color_model_ #1 _white: }
   {
    \prop_put:Nnn \exp_not:N \l__color_named_white_prop {#1}
     { \exp_not:n {#3} }
    \exp_not:N \int_compare:nNnF { \tex_currentgrouplevel:D } = 0
     { \group_insert_after:N \exp_not:c { __color_model_ #1 _ white: } }
   }
  \use:c { __color_model_ #1 _white: }
 }
\cs_generate_variant:Nn \__color_model_init:nnn { nne }
\cs_new_protected:Npn \__color_model_separation:n #1
 {
  \prop_get:NnNTF \l__color_internal_prop { name }
   \l__color_internal_tl
   {
    \exp_args:NV \__color_model_separation:nn
     \l__color_internal_tl {#1}
   }
   {
    \msg_error:nnn { color }
     { separation-requires-name } {#1}
   }
 }
\cs_new_protected:Npn \__color_model_separation:nn #1#2
 {
  \prop_get:NnNTF \l__color_internal_prop { alternative-model }
```

```
      \l__color_internal_tl
      {
        \exp_args:NV \__color_model_separation:nnn
          \l__color_internal_tl {#2} {#1}
      }
      {
        \msg_error:nnn { color }
          { separation-alternative-model } {#2}
      }
  }
\cs_new_protected:Npn \__color_model_separation:nnn #1#2#3
  {
    \cs_if_exist:cTF { __color_model_separation_ #1 :nnnnnn }
      {
        \prop_get:NnNTF \l__color_internal_prop { alternative-values }
          \l__color_internal_tl
          {
            \exp_after:wN \__color_model_separation:w \l__color_internal_tl
              , 0 , 0 , 0 , 0 \s__color_stop {#2} {#3} {#1}
          }
          {
            \msg_error:nnn { color }
              { separation-alternative-values } {#2}
          }
      }
      {
        \msg_error:nnn { color }
          { unknown-alternative-model } {#1}
      }
  }
\cs_new_protected:Npn \__color_model_separation:w
  #1 , #2 , #3 , #4 , #5 \s__color_stop #6#7#8
  {
    \__color_model_init:nnn {#6} { separation } { 0 }
    \cs_new_eq:cN { __color_parse_mix_ #6 :nw } \__color_parse_mix_gray:nw
    \cs_new:cpn { __color_parse_model_ #6 :w } ##1 , ##2 \s__color_stop
      { {#6} { \__color_parse_number:n {##1} } }
    \use:c { __color_model_separation_ #8 :nnnnnn }
      {#6} {#7} {#1} {#2} {#3} {#4}
    \prop_gput:Nnn \g__color_alternative_model_prop {#6} {#8}
    \prop_gput:Nne \g__color_colorants_prop {#6}
      { \str_convert_pdfname:n {#7} }
  }
\cs_new_protected:Npn \__color_model_separation_cmyk:nnnnnn #1#2#3#4#5#6
  {
    \tl_const:cn { c__color_fallback_ #1 _tl } { cmyk }
    \cs_new:cpn { __color_convert_ #1 _cmyk:w } ##1 \s__color_stop
      {
        \fp_eval:n {##1 * #3} ~
        \fp_eval:n {##1 * #4} ~
        \fp_eval:n {##1 * #5} ~
        \fp_eval:n {##1 * #6}
      }
    \cs_new:cpn { __color_convert_cmyk_ #1 :w } ##1 \s__color_stop { 1 }
    \prop_gput:Nnn \g__color_alternative_values_prop {#1} { #3 , #4 , #5 , #6 }
    \__color_backend_separation_init:nnnnn {#2} { /DeviceCMYK } { }
      { 0 ~ 0 ~ 0 ~ 0 } { #3 ~ #4 ~ #5 ~ #6 }
  }
\cs_new_protected:Npn \__color_model_separation_rgb:nnnnnn #1#2#3#4#5#6
  {
    \tl_const:cn { c__color_fallback_ #1 _tl } { rgb }
    \cs_new:cpn { __color_convert_ #1 _rgb:w } ##1 \s__color_stop
      {
        \fp_eval:n {##1 * #3} ~
        \fp_eval:n {##1 * #4} ~
        \fp_eval:n {##1 * #5}
      }
    \cs_new:cpn { __color_convert_rgb_ #1 :w } ##1 \s__color_stop { 1 }
    \prop_gput:Nnn \g__color_alternative_values_prop {#1} { #3 , #4 , #5 }
    \__color_backend_separation_init:nnnnn {#2} { /DeviceRGB } { }
      { 0 ~ 0 ~ 0 } { #3 ~ #4 ~ #5 }
  }
\cs_new_protected:Npn \__color_model_separation_gray:nnnnnn #1#2#3#4#5#6
  {
    \tl_const:cn { c__color_fallback_ #1 _tl } { gray }
    \cs_new:cpn { __color_convert_ #1 _gray:w } ##1 \s__color_stop
      { \fp_eval:n {##1 * #3} }
    \cs_new:cpn { __color_convert_gray_ #1 :w } ##1 \s__color_stop { 1 }
    \prop_gput:Nnn \g__color_alternative_values_prop {#1} {#3}
    \__color_backend_separation_init:nnnnn {#2} { /DeviceGray } { } { 0 } {#3}
  }
\cs_new_protected:Npn \__color_model_convert:nnn #1#2#3
  {
    \cs_new:cpe { __color_convert_ #1 _ #3 :w } ##1 \s__color_stop
      {
        \exp_not:N \exp_args:NNe \exp_not:N \use:nn
        \exp_not:c { __color_convert_  #2 _ #3 :w }
          { \exp_not:c { __color_convert_ #1 _ #2 :w } ##1 \s__color_stop }
          \c_space_tl \exp_not:N \s__color_stop
      }
  }
\cs_new_protected:Npn \__color_model_separation_CIELAB:nnnnnn #1#2#3#4#5#6
```

```
  {
    \prop_get:NnNF \l__color_internal_prop { illuminant }
      \l__color_internal_tl
      {
        \msg_error:nnn { color }
          { CIELAB-requires-illuminant } {#1}
        \tl_set:Nn \l__color_internal_tl { d50 }
      }
    \exp_args:NV \__color_model_separation_CIELAB:nnnnnnn
      \l__color_internal_tl {#1} {#2} {#3} {#4} {#5} {#6}
  }
\cs_new_protected:Npn \__color_model_separation_CIELAB:nnnnnnn #1#2#3#4#5#6#7
  {
    \tl_if_exist:cTF { c__color_model_whitepoint_CIELAB_ #1 _tl }
      {
        \__color_backend_separation_init_CIELAB:nnn {#1} {#3} { #4 ~ #5 ~ #6 }
        \tl_const:cn { c__color_fallback_ #2 _tl } { gray }
        \cs_new:cpn { __color_convert_ #2 _gray:w } ##1 \s__color_stop
          { 0 }
        \cs_new:cpn { __color_convert_gray_ #2 :w } ##1 \s__color_stop
          { 1 }
      }
      {
        \msg_error:nnn { color }
          { unknown-CIELAB-illuminant } {#1}
      }
  }
\cs_new_protected:Npn \__color_model_devicen:n #1
  {
    \prop_get:NnNTF \l__color_internal_prop { names }
      \l__color_internal_tl
      {
        \exp_args:NV \__color_model_devicen:nn
          \l__color_internal_tl {#1}
      }
      {
        \msg_error:nnn { color }
          { DeviceN-requires-names } {#1}
      }
  }
\cs_new_protected:Npn \__color_model_devicen:nn #1#2
  {
    \tl_clear:N \l__color_model_tl
    \clist_map_inline:nn {#1}
      {
        \prop_get:NnNTF \g__color_alternative_model_prop {##1}
```

```
          \l__color_internal_tl
          {
            \tl_if_empty:NTF \l__color_model_tl
              { \tl_set_eq:NN \l__color_model_tl \l__color_internal_tl }
              {
                \str_if_eq:VVF \l__color_model_tl \l__color_internal_tl
                  {
                    \msg_error:nnn { color }
                      { DeviceN-inconsistent-alternative }
                      {#2}
                    \clist_map_break:n { \use_none:nnnn }
                  }
              }
          }
          {
            \str_if_eq:nnF {##1} { none }
              {
                \msg_error:nnn { color }
                  { DeviceN-no-alternative }
                  {#2}
              }
          }
      }
    \tl_if_empty:NTF \l__color_model_tl
      {
        \msg_error:nnn { color }
          { DeviceN-no-alternative } {#2}
      }
      { \exp_args:NV \__color_model_devicen:nnn \l__color_model_tl {#1} {#2} }
  }
\cs_new_protected:Npn \__color_model_devicen:nnn #1#2#3
  {
    \exp_args:Ne \__color_model_devicen:nnnn
      { \clist_count:n {#2} } {#1} {#2} {#3}
  }
\cs_new_protected:Npn \__color_model_devicen:nnnn #1#2#3#4
  {
    \__color_model_init:nne {#4} { devicen }
      {
        0 \prg_replicate:nn { #1 - 1 } { ~ 0 }
      }
    \cs_if_exist_use:cF { __color_model_devicen_parse_ #1 :nn }
      { \__color_model_devicen_parse_generic:nn }
        {#4} {#1}
    \__color_model_devicen_init:nnn {#1} {#2} {#3}
    \__color_model_devicen_convert:nnne {#4} {#2} {#3}
```

```
      {
        1 \prg_replicate:nn { #1 - 1 } { ~ 1 }
      }
  }
\cs_new_protected:cpn { __color_model_devicen_parse_1:nn } #1#2
  {
    \cs_new:cpn { __color_parse_model_ #1 :w  } ##1 , ##2 \s__color_stop
      { {#1} { \__color_parse_number:n {##1} } }
    \cs_new_eq:cN { __color_parse_mix_ #1 :nw  } \__color_parse_mix_gray:nw
  }
\cs_new_protected:cpn { __color_model_devicen_parse_2:nn } #1#2
  {
    \cs_new:cpn { __color_parse_model_ #1 :w  } ##1 , ##2 , ##3 \s__color_stop
      { {#1} { \__color_parse_number:n {##1} ~ \__color_parse_number:n {##2} } }
    \cs_new:cpn { __color_parse_mix_ #1 :nw }
      ##1##2 ~ ##3 \s__color_mark ##4 ~ ##5 \s__color_stop
      {
        \fp_eval:n { ##2 * ##1 + ##4 * ( 1 - ##1 ) } \c_space_tl
        \fp_eval:n { ##3 * ##1 + ##5 * ( 1 - ##1 ) }
      }
  }
\cs_new_protected:cpn { __color_model_devicen_parse_3:nn } #1#2
  {
    \cs_new:cpn { __color_parse_model_ #1 :w  } ##1 , ##2 , ##3 , ##4 \s__color_stop
      {
        {#1}
        {
          \__color_parse_number:n {##1} ~
          \__color_parse_number:n {##2} ~
          \__color_parse_number:n {##3}
        }
      }
    \cs_new_eq:cN { __color_parse_mix_ #1 :nw  } \__color_parse_mix_rgb:nw
  }
\cs_new_protected:cpn { __color_model_devicen_parse_4:nn } #1#2
  {
    \cs_new:cpn { __color_parse_model_ #1 :w  }
      ##1 , ##2 , ##3 , ##4 , ##5 \s__color_stop
      {
        {#1}
        {
          \__color_parse_number:n {##1} ~
          \__color_parse_number:n {##2} ~
          \__color_parse_number:n {##3} ~
          \__color_parse_number:n {##4}
        }
      }
    \cs_new_eq:cN { __color_parse_mix_ #1 :nw } \__color_parse_mix_cmyk:nw
  }
\cs_new_protected:Npn \__color_model_devicen_parse_generic:nn #1#2
  {
    \cs_new:cpn { __color_parse_model_ #1 :w  } ##1 , ##2 \s__color_stop
      {
        {#1}
        { \__color_model_devicen_parse:nw {#2} ##1 , ##2 , \q_nil , \s__color_stop }
      }
    \cs_new:cpe { __color_parse_mix_ #1 :nw }
      ##1 ##2 \s__color_mark ##3 \s__color_stop
      {
        \exp_not:N \__color_model_devicen_mix:nw {##1}
          ##2 \c_space_tl \exp_not:N \q_nil \c_space_tl \exp_not:N \s__color_mark
          ##3 \c_space_tl \exp_not:N \q_nil \c_space_tl \exp_not:N \s__color_stop
      }
  }
\cs_new:Npn \__color_model_devicen_parse:nw #1#2 , #3 \s__color_stop
  {
    \int_compare:nNnT {#1} > 0
      {
        \quark_if_nil:nTF {#2}
          { \prg_replicate:nn {#1} { 0 ~ } }
          {
            \__color_parse_number:n {#2}
            \int_compare:nNnT {#1} > 1 { ~ }
            \exp_args:Nf \__color_model_devicen_parse:nw
              { \int_eval:n { #1 - 1 } } #3 \s__color_stop
          }
      }
  }
\cs_new:Npn \__color_model_devicen_mix:nw #1#2 ~ #3 \s__color_mark #4 ~ #5 \s__color_stop
  {
    \fp_eval:n { #2 * #1 + #4 * ( 1 - #1 ) }
    \quark_if_nil:oF { \tl_head:w #3 \q_stop }
      {
        \c_space_tl
        \__color_model_devicen_mix:nw {#1} #3 \s__color_mark #5 \s__color_stop
      }
  }
\cs_new_protected:Npn \__color_model_devicen_init:nnn #1#2#3
  {
    \exp_args:Ne \__color_model_devicen_init:nnnn
      {
        \str_case:nn {#2}
```

```
          {
            { cmyk } { 4 }
            { gray } { 1 }
            { rgb }  { 3 }
          }
      }
      {#1} {#2} {#3}
  }
\cs_new_protected:Npn \__color_model_devicen_init:nnnn #1#2#3#4
  {
    \tl_set:Ne \l__color_internal_tl
      { \prg_replicate:nn {#1} { 1.0 ~ }   }
    \int_zero:N \l__color_internal_int
    \clist_map_inline:nn {#4}
      {
        \int_incr:N \l__color_internal_int
        \prop_get:NnN \g__color_alternative_values_prop {##1}
          \l__color_value_tl
        \exp_after:wN \__color_model_devicen_transform:w
          \l__color_value_tl , 0 , 0 , 0 , \s__color_stop {#1} {#2}
      }
    \tl_put_right:Ne \l__color_internal_tl
      {
        \prg_replicate:nn {#1}
          { neg ~ 1.0 ~ add ~ #1 ~ -1 ~ roll ~ }
        \int_eval:n { #2 + #1 } ~ #1 ~ roll
        \prg_replicate:nn {#2} { ~ pop } ~
        #1 ~ 1 ~ roll
      }
    \use:e
      {
        \__color_backend_devicen_init:nnn
          {
            \clist_map_function:nN {#4}
              \__color_model_devicen_colorant:n
          }
          {
            \str_case:nn {#3}
              {
                { cmyk } { /DeviceCMYK }
                { gray } { /DeviceGray }
                { rgb }  { /DeviceRGB }
              }
          }
          { \exp_not:V \l__color_internal_tl }
      }
  }
\cs_new_protected:Npn \__color_model_devicen_transform:w
  #1 , #2 , #3 , #4 , #5 \s__color_stop #6#7
  {
    \use:c { __color_model_devicen_transform_ #6 :nnnnn }
      {#1} {#2} {#3} {#4} {#7}
  }
\cs_new_protected:cpn { __color_model_devicen_transform_1:nnnnn } #1#2#3#4#5
  { \__color_model_devicen_transform:nnn {#5} { 1 } {#1} }
\cs_new_protected:cpn { __color_model_devicen_transform_3:nnnnn } #1#2#3#4#5
  {
    \clist_map_inline:nn { #1 , #2 , #3 }
      { \__color_model_devicen_transform:nnn {#5} { 3 } {##1} }
  }
\cs_new_protected:cpn { __color_model_devicen_transform_4:nnnnn } #1#2#3#4#5
  {
    \clist_map_inline:nn { #1 , #2 , #3 , #4 }
      { \__color_model_devicen_transform:nnn {#5} { 4 } {##1} }
  }
\cs_new_protected:Npn \__color_model_devicen_transform:nnn #1#2#3
  {
    \tl_put_right:Ne \l__color_internal_tl
      {
        \fp_compare:nNnF {#3} = \c_zero_fp
          {
            \int_eval:n { #1 - \l__color_internal_int + #2 } ~ index ~
              -#3 ~ mul ~ 1.0 ~ add ~ mul ~
          }
        #2 ~ -1 ~ roll ~
      }
  }
\cs_new:Npn \__color_model_devicen_colorant:n #1
  {
    / \prop_item:Nn \g__color_colorants_prop {#1} ~
  }
\cs_new_protected:Npn \__color_model_devicen_convert:nnnn #1#2#3
  {
    \use:c { __color_model_devicen_convert_ #2 :nnn } {#1} {#3}
  }
\cs_generate_variant:Nn \__color_model_devicen_convert:nnnn { nnne }
\cs_new_protected:Npn \__color_model_devicen_convert_cmyk:nnn #1#2
  {
    \tl_const:cn { c__color_fallback_ #1 _tl } { cmyk }
    \__color_model_devicen_convert:nnnnn {#1} { cmyk } { 4 } {#2}
  }
\cs_new_protected:Npn \__color_model_devicen_convert_gray:nnn #1#2
```

```
  {
    \tl_const:cn { c__color_fallback_ #1 _tl } { gray }
    \__color_model_devicen_convert:nnnnn {#1} { gray } { 1 } {#2}
  }
\cs_new_protected:Npn \__color_model_devicen_convert_rgb:nnn #1#2
  {
    \tl_const:cn { c__color_fallback_ #1 _tl } { rgb }
    \__color_model_devicen_convert:nnnnn {#1} { rgb } { 3 } {#2}
  }
\cs_new_protected:Npn \__color_model_devicen_convert:nnnnn #1#2#3#4#5
  {
    \cs_new:cpn { __color_convert_ #2 _ #1 :w } ##1 \s__color_stop {#5}
    \cs_new:cpe { __color_convert_ #1 _ #2 :w } ##1 \s__color_stop
      {
        \exp_not:c { __color_convert_devicen_ #2 : \prg_replicate:nn {#3} { n } w }
          \prg_replicate:nn {#3} { { 1 } }
          ##1 ~ \exp_not:N \s__color_mark
          \clist_map_function:nN {#4} \__color_model_devicen_convert:n
          {}
          \exp_not:N \s__color_stop
      }
  }
\cs_new:Npn \__color_model_devicen_convert:n #1
  {
    {
      \exp_args:Ne \__color_model_devicen_convert_aux:n
        { \prop_item:Nn \g__color_alternative_values_prop {#1} }
    }
  }
\cs_new:Npn \__color_model_devicen_convert_aux:n #1
  { \__color_model_devicen_convert_aux:w #1 , , , , \s__color_stop }
\cs_new:Npn \__color_model_devicen_convert_aux:w #1 , #2 , #3 , #4 , #5 \s__color_stop
  {
    {#1}
    \tl_if_blank:nF {#2}
      {
        {#2}
        \tl_if_blank:nF {#3}
          {
            {#3}
            \tl_if_blank:nF {#4} { {#4} }
          }
      }
  }
\cs_new:Npn \__color_convert_devicen_cmyk:nnnnw
  #1#2#3#4#5 ~ #6 \s__color_mark #7#8 \s__color_stop
  {
    \__color_convert_devicen_cmyk:nnnnnnnnn {#5} {#1} {#2} {#3} {#4} #7
      #6 \s__color_mark #8 \s__color_stop
  }
\cs_new:Npn \__color_convert_devicen_cmyk:nnnnnnnnn #1#2#3#4#5#6#7#8#9
  {
    \use:e
      {
        \exp_not:N \__color_convert_devicen_cmyk_aux:nnnnw
          { \fp_eval:n { #2 * (1 - (#1 * #6)) } }
          { \fp_eval:n { #3 * (1 - (#1 * #7)) } }
          { \fp_eval:n { #4 * (1 - (#1 * #8)) } }
          { \fp_eval:n { #5 * (1 - (#1 * #9)) } }
      }
  }
\cs_new:Npn \__color_convert_devicen_cmyk_aux:nnnnw
  #1#2#3#4 #5 \s__color_mark #6 \s__color_stop
  {
    \tl_if_blank:nTF {#5}
      {
        \fp_eval:n { 1 - #1 } ~
        \fp_eval:n { 1 - #2 } ~
        \fp_eval:n { 1 - #3 } ~
        \fp_eval:n { 1 - #4 }
      }
      {
        \__color_convert_devicen_cmyk:nnnnw {#1} {#2} {#3} {#4}
          #5 \s__color_mark #6 \s__color_stop
      }
  }
\cs_new:Npn \__color_convert_devicen_gray:nw
  #1#2 ~ #3 \s__color_mark #4#5 \s__color_stop
  {
    \__color_convert_devicen_gray:nnn {#2} {#1} #4
      #3 \s__color_mark #5 \s__color_stop
  }
\cs_new:Npn \__color_convert_devicen_gray:nnn #1#2#3
  {
    \exp_arsgs:Ne \__color_convert_devicen_gray_aux:nw
      { \fp_eval:n { #2 * (1 - (#1 * #3)) } }
  }
\cs_new:Npn \__color_convert_devicen_gray_aux:nw
  #1 #2 \s__color_mark #3 \s__color_stop
  {
    \tl_if_blank:nTF {#2}
      { \fp_eval:n { 1 - #1 } }
      {
        \__color_convert_devicen_gray:nw {#1}
          #2 \s__color_mark #3 \s__color_stop
      }
  }
\cs_new:Npn \__color_convert_devicen_rgb:nnnw
  #1#2#3#4 ~ #5 \s__color_mark #6#7 \s__color_stop
  {
    \__color_convert_devicen_rgb:nnnnnnn {#4} {#1} {#2} {#3} #6
      #5 \s__color_mark #7 \s__color_stop
  }
\cs_new:Npn \__color_convert_devicen_rgb:nnnnnnn #1#2#3#4#5#6#7
  {
    \use:e
      {
        \exp_not:N \__color_convert_devicen_rgb_aux:nnnw
          { \fp_eval:n { #2 * (1 - (#1 * #5)) } }
          { \fp_eval:n { #3 * (1 - (#1 * #6)) } }
          { \fp_eval:n { #4 * (1 - (#1 * #7)) } }
      }
  }
\cs_new:Npn \__color_convert_devicen_rgb_aux:nnnw
  #1#2#3 #4 \s__color_mark #5 \s__color_stop
  {
    \tl_if_blank:nTF {#4}
      {
        \fp_eval:n { 1 - #1 } ~
        \fp_eval:n { 1 - #2 } ~
        \fp_eval:n { 1 - #3 }
      }
      {
        \__color_convert_devicen_rgb:nnnw {#1} {#2} {#3}
          #4 \s__color_mark #5 \s__color_stop
      }
  }
\prop_const_from_keyval:Nn \c__color_icc_colorspace_signatures_prop
  {
    47524159 = {1} {1} {0} {},
    52474220 = {3} {0~0~0} {1~1~1} {},
    434D594B = {4} {0~0~0~1} {0~0~0~0} {},
    4C616220 = {3} {0~0~0} {100~0~0} {0~100~-128~127~-128~127}
  }
\cs_new_protected:Npn \__color_model_iccbased:n #1
  {
    \prop_get:NnNTF \l__color_internal_prop { file }
      \l__color_internal_tl
      {
        \exp_args:NV \__color_model_iccbased:nn
          \l__color_internal_tl {#1}
      }
      {
        \msg_error:nnn { color }
          { ICCBased-requires-file } {#1}
      }
  }
\cs_new_protected:Npn \__color_model_iccbased:nn #1#2
  {
    \prop_get:NeNTF \c__color_icc_colorspace_signatures_prop
      { \file_hex_dump:nnn { #1 } { 17 } { 20 } } \l__color_internal_tl
      {
        \exp_last_unbraced:NV \__color_model_iccbased_aux:nnnnnn
          \l__color_internal_tl { #2 } { #1 }
      }
      {
        \msg_error:nnn { color }
        { ICCBased-unsupported-colorspace } {#2}
      }
  }
\cs_new_protected:Npn \__color_model_iccbased_aux:nnnnnn #1#2#3#4#5#6
  {
    \__color_model_init:nnn {#5} { iccbased } {#3}
    \tl_const:cn { c__color_fallback_ #5 _tl } { gray }
    \cs_new:cpn { __color_convert_ #5 _gray:w } ##1 \s__color_stop { 0 }
    \cs_new:cpn { __color_convert_gray_ #5 :w } ##1 \s__color_stop { #2 }
    \use:c { __color_model_devicen_parse_ #1 :nn } {#5} {#1}
    \exp_args:Ne \__color_backend_iccbased_init:nnn
      { \file_full_name:n {#6} } {#1} {#4}
  }
\cs_new_protected:Npn \color_profile_apply:nn #1#2
  {
    \exp_args:Ne \__color_profile_apply:nn
      { \file_full_name:n {#1} } {#2}
  }
\cs_new_protected:Npn \__color_profile_apply:nn #1#2
  {
    \cs_if_exist_use:cF { __color_profile_apply_ \tl_to_str:n {#2} :n }
      {
        \msg_error:nnn { color } { ICC-Device-unknown } {#2}
        \use_none:n
      }
        {#1}
  }
```

```
\cs_new_protected:Npn \__color_profile_apply_gray:n #1
  {
    \int_gincr:N \g__color_model_int
    \__color_backend_iccbased_device:nnn {#1} { Gray } { 1 }
  }
\cs_new_protected:Npn \__color_profile_apply_rgb:n #1
  {
    \int_gincr:N \g__color_model_int
    \__color_backend_iccbased_device:nnn {#1} { RGB } { 3 }
  }
\cs_new_protected:Npn \__color_profile_apply_cmyk:n #1
  {
    \int_gincr:N \g__color_model_int
    \__color_backend_iccbased_device:nnn {#1} { CMYK } { 4 }
  }
\cs_new_protected:Npn \color_show:n
  { \__color_show:Nn \msg_show:nneeee }
\cs_new_protected:Npn \color_log:n
  { \__color_show:Nn \msg_log:nneeee }
\cs_new_protected:Npn \__color_show:Nn #1#2
  {
    #1 { color } { show }
      {#2}
      {
        \color_if_exist:nT {#2}
          {
            \exp_args:Nv \__color_show:n { l__color_named_ #2 _tl }
            \prop_map_function:cN
              { l__color_named_ #2 _prop }
              \msg_show_item_unbraced:nn
          }
      }
      { }
      { }
  }
\cs_new:Npn \__color_show:n #1
  {
    \msg_show_item_unbraced:nn { model } {#1}
  }
\msg_new:nnnn { color } { CIELAB-requires-illuminant }
  { CIELAB~color~space~'#1'~require~an~illuminant. }
  {
    LaTeX~has~been~asked~to~create~a~separation~color~space~using~
    CIELAB~specifications,~but~no~\\ \\
    \iow_indent:n { illuminant~=~<basis> }
    \\ \\
    key~was~given~with~the~correct~information.~LaTeX~will~use~illuminant~
    'd50'~for~recovery.
  }
\msg_new:nnnn { color } { conversion-not-available }
  { No~model~conversion~available~from~'#1'~to~'#2'. }
  {
    LaTeX~has~been~asked~to~convert~a~color~from~model~'#1'~
    to~model'#2',~but~there~is~no~method~available~to~do~that.
  }
\msg_new:nnnn { color } { DeviceN-inconsistent-alternative }
  { DeviceN~color~spaces~require~a~single~alternative~space. }
  {
    LaTeX~has~been~asked~to~create~a~DeviceN~color~space~'#1',~
    but~the~constituent~colors~do~not~have~a~common~alternative~
    color.
  }
\msg_new:nnnn { color } { DeviceN-no-alternative }
  { DeviceN~color~spaces~require~an~alternative~space. }
  {
    LaTeX~has~been~asked~to~create~a~DeviceN~color~space~'#1',~
    but~the~constituent~colors~do~not~all~have~a~device-based~alternative.
  }
\msg_new:nnnn { color } { DeviceN-requires-names }
  { DeviceN~color~space~'#1'~require~a~list~of~names. }
  {
    LaTeX~has~been~asked~to~create~a~DeviceN~color~space,~
    but~no~\\ \\
    \iow_indent:n { names~=~<names> }
    \\ \\
    key~was~given~with~the~correct~information.
  }
\msg_new:nnnn { color } { ICC-Device-unknown }
  { Unknown~device~color~space~'#1'. }
  {
    LaTeX~has~been~asked~to~apply~an~ICC~profile~but~the~device~color~space~
    '#1'~is~unknown.
  }
\msg_new:nnnn { color } { ICCBased-unsupported-colorspace }
  { ICCBased~color~space~'#1'~uses~an~unsupported~data~color~space. }
  {
    LaTeX~has~been~asked~to~create~a~ICCBased~colorspace,~but~the~
    used~data~colorspace~is~not~supported.~ICC~profiles~used~for~
    defining~a~ICCBased~colorspace~should~use~a~Lab,~RGB,~or~
    CMYK~data~colorspace.~LaTeX~will~ignore~this~request.
  }
\msg_new:nnnn { color } { ICCBased-requires-file }
```

```
  { ICCBased~color~space~'#1'~require~an~file. }
  {
    LaTeX~has~been~asked~to~create~an~ICCBased~color~space,~but~no~\\ \\
    \iow_indent:n { file~=~<name> }
    \\ \\
    key~was~given~with~the~correct~information.~LaTeX~will~ignore~this~
    request.
  }
\msg_new:nnnn { color } { model-already-defined }
  { Color~model~'#1'~already~defined. }
  {
    LaTeX~was~asked~to~define~a~new~color~model~called~'#1',~but~
    this~color~model~already~exists.
  }
\msg_new:nnnn { color } { out-of-range }
  { Input~value~#1~out~of~range~[#2,~#3]. }
  {
    LaTeX~was~expecting~a~value~in~the~range~[#2,~#3]~as~part~of~a~color,~
    but~you~gave~#1.~LaTeX~will~assume~you~meant~the~limit~of~the~range~
    and~continue.
  }
\msg_new:nnnn { color } { separation-alternative-model }
  { Separation~color~space~'#1'~require~an~alternative~model. }
  {
    LaTeX~has~been~asked~to~create~a~separation~color~space,~
    but~no~\\ \\
    \iow_indent:n { alternative-model~=~<model> }
    \\ \\
    key~was~given~with~the~correct~information.
  }
\msg_new:nnnn { color } { separation-alternative-values }
  { Separation~color~space~'#1'~require~values~for~the~alternative~space. }
  {
    LaTeX~has~been~asked~to~create~a~separation~color~space,~
    but~no~\\ \\
    \iow_indent:n { alternative-values~=~<model> }
    \\ \\
    key~was~given~with~the~correct~information.
  }
\msg_new:nnnn { color } { separation-requires-name }
  { Separation~color~space~'#1'~require~a~formal~name. }
  {
    LaTeX~has~been~asked~to~create~a~separation~color~space,~
    but~no~\\ \\
    \iow_indent:n { name~=~<formal~name> }
    \\ \\
    key~was~given~with~the~correct~information.
  }
\msg_new:nnn { color } { unhandled-model }
  {
    Unhandled~color~model~in~LaTeX2e~value~"#1":
    \\ \\
    falling~back~on~grayscale.
  }
\msg_new:nnnn { color } { unknown-color }
  { Unknown~color~'#1'. }
  {
    LaTeX~has~been~asked~to~use~a~color~named~'#1',~
    but~this~has~never~been~defined.
  }
\msg_new:nnnn { color } { unknown-alternative-model }
  { Separation~color~space~'#1'~require~an~valid~alternative~space. }
  {
    LaTeX~has~been~asked~to~create~a~separation~color~space,~
    but~the~model~given~as\\ \\
    \iow_indent:n { alternative-model~=~<model> }
    \\ \\
    is~unknown.
  }
\msg_new:nnnn { color } { unknown-export-format }
  { Unknown~export~format~'#1'. }
  {
    LaTeX~has~been~asked~to~export~a~color~in~format~'#1',~
    but~this~has~never~been~defined.
  }
\msg_new:nnnn { color } { unknown-CIELAB-illuminant }
  { Unknown~illuminant~model~'#1'. }
  {
    LaTeX~has~been~asked~to~use~create~a~color~space~using~CIELAB~
    illuminant~'#1',~but~this~does~not~exist.
  }
\msg_new:nnnn { color } { unknown-model }
  { Unknown~color~model~'#1'. }
  {
    LaTeX~has~been~asked~to~use~a~color~model~called~'#1',~
    but~this~model~is~not~set~up.
  }
\msg_new:nnnn { color } { unknown-model-type }
  { Unknown~color~model~type~'#1'. }
  {
    LaTeX~has~been~asked~to~create~a~new~color~model~called~'#1',~
    but~this~type~of~model~was~never~set~up.
  }
\prop_gput:Nnn \g_msg_module_name_prop { color } { LaTeX }
\prop_gput:Nnn \g_msg_module_type_prop { color } { }
\msg_new:nnn { color } { show }
  {
    The~color~#1~
    \tl_if_empty:nTF {#2}
      { is~undefined. }
      { has~the~properties: #2 }
  }
%% File: l3graphics.dtx
\cs_if_exist:NT \@expl@finalise@setup@@
  {
    \tl_gput_right:Nn \@expl@finalise@setup@@
      { \declare@file@substitution { l3graphics.sty } { null.tex } }
  }
\dim_new:N \l__graphics_internal_dim
\ior_new:N \l__graphics_internal_ior
\tl_new:N  \l__graphics_internal_tl
\scan_new:N \s__graphics_stop
\tl_new:N \l__graphics_pagebox_tl
\keys_define:nn { graphics }
  {
    decodearray .str_set:N =
      \l__graphics_decodearray_str ,
    draft .bool_set:N =
      \l__graphics_draft_bool ,
    interpolate .bool_set:N =
      \l__graphics_interpolate_bool ,
    pagebox .choices:nn =
      { art , bleed , crop , media , trim }
      {
        \tl_set:Ne \l__graphics_pagebox_tl
          { \l_keys_choice_tl box }
      } ,
    pagebox .initial:n =
      crop ,
    page .int_set:N =
      \l__graphics_page_int ,
    pdf-attr .str_set:N =
      \l__graphics_pdf_str ,
    type . str_set:N =
      \l__graphics_type_str
  }
\dim_new:N \l__graphics_llx_dim
\dim_new:N \l__graphics_lly_dim
```

```
\dim_new:N \l__graphics_urx_dim
\dim_new:N \l__graphics_ury_dim
\cs_new_protected:Npn \__graphics_bb_save:n #1
  {
    \dim_if_exist:cTF { c__graphics_ #1 _urx_dim }
      { \msg_error:nnn { graphic } { bb-already-cached } {#1} }
      {
        \dim_compare:nNnF \l__graphics_llx_dim = { 0pt }
          { \dim_const:cn { c__graphics_ #1 _llx_dim } { \l__graphics_llx_dim } }
        \dim_compare:nNnF \l__graphics_lly_dim = { 0pt }
          { \dim_const:cn { c__graphics_ #1 _lly_dim } { \l__graphics_lly_dim } }
        \dim_const:cn { c__graphics_ #1 _urx_dim } { \l__graphics_urx_dim }
        \dim_const:cn { c__graphics_ #1 _ury_dim } { \l__graphics_ury_dim }
      }
  }
\cs_generate_variant:Nn \__graphics_bb_save:n { e }
\cs_new_protected:Npn \__graphics_bb_restore:nF #1#2
  {
    \dim_if_exist:cTF { c__graphics_ #1 _urx_dim }
      {
        \dim_set_eq:Nc \l__graphics_urx_dim { c__graphics_ #1 _urx_dim }
        \dim_set_eq:Nc \l__graphics_ury_dim { c__graphics_ #1 _ury_dim }
        \dim_if_exist:cTF { c__graphics_ #1 _llx_dim }
          { \dim_set_eq:Nc \l__graphics_llx_dim { c__graphics_ #1 _llx_dim } }
          { \dim_zero:N \l__graphics_llx_dim }
        \dim_if_exist:cTF { c__graphics_ #1 _lly_dim }
          { \dim_set_eq:Nc \l__graphics_lly_dim { c__graphics_ #1 _lly_dim } }
          { \dim_zero:N \l__graphics_lly_dim }
      }
      {#2}
  }
\cs_generate_variant:Nn \__graphics_bb_restore:nF { e }
\cs_new_protected:Npn \__graphics_extract_bb:n #1
  {
    \int_compare:nNnTF \l__graphics_page_int > 0
      { \__graphics_extract_bb_auxi:Vn \l__graphics_page_int {#1} }
      { \__graphics_extract_bb_auxii:nnn {#1} { } { } }
  }
\cs_new_protected:Npn \__graphics_extract_bb_auxi:nn #1#2
  { \__graphics_extract_bb_auxii:nnn {#2} { :P #1 } { -p~#1~ } }
\cs_generate_variant:Nn \__graphics_extract_bb_auxi:nn { Vn }
\cs_new_protected:Npn \__graphics_extract_bb_auxii:nnn #1#2#3
  {
    \tl_if_empty:NTF \l__graphics_pagebox_tl
      { \__graphics_extract_bb_auxiv:nnn }
      { \__graphics_extract_bb_auxiii:Vnnn \l__graphics_pagebox_tl }
      {#1} {#2} {#3}
  }
\cs_new_protected:Npn \__graphics_extract_bb_auxiii:nnnn #1#2#3#4
  { \__graphics_extract_bb_auxiv:nnn {#2} { : #1 #3 } { #4 -B~#1~ } }
\cs_generate_variant:Nn \__graphics_extract_bb_auxiii:nnnn { V }
\cs_new_protected:Npn \__graphics_extract_bb_auxiv:nnn #1#2#3
  {
    \__graphics_read_bb_auxi:nnnn {#1} {#2}
      { \ior_shell_open:Nn \l__graphics_internal_ior { extractbb~#3-O~#1 } }
      { pipe-failed }
  }
\cs_new_protected:Npn \__graphics_read_bb:n #1
  {
    \__graphics_read_bb_auxi:nnnn {#1} { }
      { \ior_open:Nn \l__graphics_internal_ior {#1} }
      { graphic-not-found }
  }
\cs_new_protected:Npn \__graphics_read_bb_auxi:nnnn #1#2#3#4
  {
    \__graphics_bb_restore:nF {#1#2}
      { \__graphics_read_bb_auxii:nnnn {#3} {#4} {#1} {#2} }
  }
\cs_new_protected:Npe \__graphics_read_bb_auxii:nnnn #1#2#3#4
  {
    #1
    \exp_not:N \ior_if_eof:NTF \exp_not:N \l__graphics_internal_ior
      { \msg_error:nnn { graphics } {#2} {#3} }
      {
        \ior_str_map_inline:Nn \exp_not:N \l__graphics_internal_ior
          {
            \exp_not:N \__graphics_read_bb_auxiii:w
              ##1 ~ \c_colon_str \s__graphics_stop
          }
        \__graphics_bb_save:n {#3#4}
      }
    \ior_close:N \exp_not:N \l__graphics_internal_ior
  }
\use:e
  {
    \cs_new_protected:Npn \exp_not:N \__graphics_read_bb_auxiii:w
      #1 \c_colon_str #2 \s__graphics_stop
      {
        \exp_not:N \str_if_eq:nnT
          { \c_percent_str \c_percent_str BoundingBox }
          {#1}
          { \exp_not:N \__graphics_read_bb_auxiv:w #2 ( ) \s__graphics_stop }
      }
  }
\cs_new_protected:Npn \__graphics_read_bb_auxiv:w #1 ( #2 ) #3 \s__graphics_stop
  {
    \str_if_eq:nnF {#2} { atend }
      {
        \tl_set_rescan:Nne \l__graphics_internal_tl
          {
            \char_set_catcode_space:n {  9 }
            \char_set_catcode_space:n { 32 }
          }
          { \use:n #1 }
        \exp_after:wN \__graphics_read_bb_auxv:w \l__graphics_internal_tl \s__graphics_stop
      }
  }
\cs_new_protected:Npn \__graphics_read_bb_auxv:w #1~#2~#3~#4~#5 \s__graphics_stop
  {
    \dim_set:Nn \l__graphics_llx_dim { #1 bp }
    \dim_set:Nn \l__graphics_lly_dim { #2 bp }
    \dim_set:Nn \l__graphics_urx_dim { #3 bp }
    \dim_set:Nn \l__graphics_ury_dim { #4 bp }
    \ior_map_break:
  }
\str_new:N \l__graphics_final_name_str
\str_new:N \l__graphics_full_name_str
\box_new:N \l__graphics_internal_box
\str_new:N \l__graphics_dir_str
\str_new:N \l__graphics_name_str
\str_new:N \l__graphics_ext_str
\seq_new:N \l_graphics_search_path_seq
\seq_new:N \l_graphics_search_ext_seq
\prop_new:N \l_graphics_ext_type_prop
\prop_put:Nnn \l_graphics_ext_type_prop { .ps } { eps }
\seq_new:N \g__graphics_record_seq
\cs_new_protected:Npn \graphics_include:nn #1#2
  {
    \group_begin:
      \keys_set:nn { graphics } {#1}
      \seq_set_eq:NN \l_file_search_path_seq \l_graphics_search_path_seq
      \file_get_full_name:nNTF {#2} \l__graphics_full_name_str
        {
          \str_if_eq:eeTF { \l__graphics_full_name_str } { #2 .tex }
            { \msg_error:nnn { graphics } { graphic-not-found } {#2} }
            { \__graphics_include: }
        }
        { \msg_error:nnn { graphics } { graphic-not-found } {#2} }
    \group_end:
  }
\cs_generate_variant:Nn \graphics_include:nn { nV }
\cs_new_protected:Npn \__graphics_include:
  {
    \str_if_empty:NTF \l__graphics_type_str
      {
        \file_parse_full_name:VNNN \l__graphics_full_name_str
          \l__graphics_dir_str \l__graphics_name_str \l__graphics_ext_str
        \__graphics_include_auxi:e
          {
            \exp_args:Ne \str_tail:n
              { \str_casefold:V \l__graphics_ext_str }
          }
      }
      { \__graphics_include_auxi:e { \l__graphics_type_str } }
  }
\cs_new_protected:Npn \__graphics_include_auxi:n #1
  {
    \prop_get:NnNF \l_graphics_ext_type_prop { .#1 } \l__graphics_internal_tl
      { \tl_set:Nn \l__graphics_internal_tl {#1} }
    \exp_args:NV \__graphics_include_auxii:n \l__graphics_internal_tl
  }
\cs_generate_variant:Nn \__graphics_include_auxi:n { e }
\cs_new_protected:Npn \__graphics_include_auxii:n #1
  {
    \mode_leave_vertical:
    \cs_if_exist:cTF { __graphics_backend_include_ #1 :n }
      {
        \tl_set_eq:NN \l__graphics_final_name_str \l__graphics_full_name_str
        \str_set:Ne \l__graphics_full_name_str
          { \exp_args:NV \__kernel_file_name_quote:n \l__graphics_full_name_str }
        \exp_args:NnV \use:c { __graphics_backend_getbb_ #1 :n }
          \l__graphics_full_name_str
        \seq_gput_right:NV \g__graphics_record_seq \l__graphics_final_name_str
        \clist_if_exist:NT \@filelist
          { \exp_args:NV \@addtofilelist \l__graphics_final_name_str }
        \bool_if:NTF \l__graphics_draft_bool
          { \__graphics_include_auxiii:n }
          { \__graphics_include_auxiv:n }
            {#1}
      }
      { \msg_error:nnn { graphics } { unsupported-graphic-type } {#1} }
  }
\cs_new_protected:Npn \__graphics_include_auxiii:n #1
```

```
  {
    \hbox_to_wd:nn { \l__graphics_urx_dim - \l__graphics_llx_dim }
      {
        \tex_vrule:D
        \tex_hss:D
        \vbox_to_ht:nn
          { \l__graphics_ury_dim - \l__graphics_lly_dim }
          {
            \tex_hrule:D width
              \dim_eval:n { \l__graphics_urx_dim - \l__graphics_llx_dim }
            \tex_vss:D
            \hbox_to_wd:nn
              { \l__graphics_urx_dim - \l__graphics_llx_dim }
              {
                \ttfamily
                \tex_hss:D \l__graphics_full_name_str \tex_hss:D
              }
            \tex_vss:D
            \tex_hrule:D
          }
        \tex_hss:D
        \tex_vrule:D
      }
  }
\cs_new_protected:Npn \__graphics_include_auxiv:n #1
  {
    \hbox_set:Nn \l__graphics_internal_box
      {
        \exp_args:NnV \use:c { __graphics_backend_include_ #1 :n }
          \l__graphics_full_name_str
      }
    \box_set_dp:Nn \l__graphics_internal_box { 0pt }
    \box_set_ht:Nn \l__graphics_internal_box
      { \l__graphics_ury_dim - \l__graphics_lly_dim }
    \box_set_wd:Nn \l__graphics_internal_box
      { \l__graphics_urx_dim - \l__graphics_llx_dim }
    \box_use_drop:N \l__graphics_internal_box
  }
\cs_new_protected:Npn \graphics_show_list: { \__graphics_list:N \msg_show:nneeee }
\cs_new_protected:Npn \graphics_log_list: { \__graphics_list:N \msg_log:nneeee }
\cs_new_protected:Npn \__graphics_list:N #1
  {
    \seq_remove_duplicates:N \g__graphics_record_seq
    #1 { kernel } { file-list }
      { \seq_map_function:NN \g__graphics_record_seq \__graphics_list_aux:n }
        { } { } { }
  }
\cs_new:Npn \__graphics_list_aux:n #1 { \iow_newline: #1 }
\cs_new_protected:Npn \graphics_get_full_name:nN #1#2
  {
    \graphics_get_full_name:nNF {#1} #2
      { \tl_set:Nn #2 { \q_no_value } }
  }
\prg_new_protected_conditional:Npnn \graphics_get_full_name:nN #1#2
  { T , F , TF }
  {
    \group_begin:
      \seq_set_eq:NN \l_file_search_path_seq \l_graphics_search_path_seq
      \file_get_full_name:nNTF {#1} \l__graphics_full_name_str
        {
          \str_if_eq:eeTF { \l__graphics_full_name_str } { #1 .tex }
            { \__graphics_get_full_name:n {#1} }
            {
              \file_parse_full_name:VNNN \l__graphics_full_name_str
                \l__graphics_dir_str \l__graphics_name_str \l__graphics_ext_str
              \seq_map_inline:Nn \l_graphics_search_ext_seq
                {
                  \str_if_eq:nVT {##1} \l__graphics_ext_str
                    { \seq_map_break:n { \use_none:nn } }
                }
                  \__graphics_get_full_name:n {#1}
            }
        }
        { \__graphics_get_full_name:n {#1} }
    \exp_args:NNNV \group_end:
    \tl_set:Nn #2 \l__graphics_full_name_str
    \tl_if_empty:NTF #2
      { \prg_return_false: }
      { \prg_return_true: }
  }
\cs_new_protected:Npn \__graphics_get_full_name:n #1
  {
    \str_clear:N \l__graphics_full_name_str
    \seq_map_inline:Nn \l_graphics_search_ext_seq
      {
        \file_get_full_name:nNT { #1 ##1 } \l__graphics_full_name_str
          { \seq_map_break:n { \use_none:nn } }
      }
    \use:n
      { \str_clear:N \l__graphics_full_name_str }
  }
\cs_new_protected:Npn \graphics_get_pagecount:nN #1#2
```

```
  {
    \group_begin:
      \seq_set_eq:NN \l_file_search_path_seq \l_graphics_search_path_seq
      \file_get_full_name:nNTF {#1} \l__graphics_full_name_str
        {
          \int_if_exist:cF { c__graphics_ \l__graphics_full_name_str _pages_int }
            {
              \exp_args:NV \__graphics_backend_get_pagecount:n
                \l__graphics_full_name_str
            }
          \tl_set:Nv #2 { c__graphics_ \l__graphics_full_name_str _pages_int }
        }
        {
          \tl_set:Nn #2 { 0 }
          \msg_error:nnn { graphics } { graphic-not-found } {#1}
        }
    \exp_args:NNNV \group_end:
    \tl_set:Nn #2 #2
  }
\cs_new_protected:Npe \__graphics_get_pagecount:n #1
  {
    \exp_not:N \ior_shell_open:Nn \exp_not:N \l__graphics_internal_ior
      { extractbb~-O~#1 }
    \exp_not:N \ior_if_eof:NTF \exp_not:N \l__graphics_internal_ior
      { \msg_error:nnn { graphics } { pipe-failed } }
      {
        \ior_str_map_inline:Nn \exp_not:N \l__graphics_internal_ior
          {
            \exp_not:N \__graphics_get_pagecount:nw {#1}
              ##1 ~ \c_colon_str \c_colon_str \s__graphics_stop
          }
        \exp_not:N \int_if_exist:cF { c__graphics_ #1 _pages_int }
          { \int_const:cn { c__graphics_ #1 _pages_int } { 1 } }
      }
    \ior_close:N \exp_not:N \l__graphics_internal_ior
  }
\use:e
  {
    \cs_new_protected:Npn \exp_not:N \__graphics_get_pagecount:nw
      #1#2 \c_colon_str #3 \c_colon_str #4 \s__graphics_stop
      {
        \exp_not:N \str_if_eq:nnT
          { \c_percent_str \c_percent_str Pages }
          {#2}
          {
            \int_const:cn { c__graphics_ #1 _pages_int } {#3}
            \exp_not:N \ior_map_break:
          }
      }
  }
\msg_new:nnnn { graphics } { graphic-not-found }
  { Image~file~'#1'~not~found. }
  {
    LaTeX~tried~to~open~graphic~file~'#1',~
    but~the~file~could~not~be~read.
  }
\msg_new:nnnn { graphics } { pipe-failed }
  { Cannot~run~piped~system~commands. }
  {
    LaTeX~tried~to~call~a~system~process~but~this~was~not~possible.\\
    Try~the~"--shell-escape"~(or~"--enable-pipes")~option.
  }
\msg_new:nnnn { graphics } { unsupported-graphic-type }
  { Image~type~'#1'~not~supported~by~current~driver. }
  {
    LaTeX~was~asked~to~include~an~graphic~of~type~'#1',~
    but~this~is~not~supported~by~the~current~driver~(production~route).
  }
%% File: l3opacity.dtx
\cs_if_exist:NT \@expl@finalise@setup@@
  {
    \tl_gput_right:Nn \@expl@finalise@setup@@
      { \declare@file@substitution { l3opacity.sty } { null.tex } }
  }
\fp_new:N \l__opacity_tmp_fp
\cs_new_protected:Npn \opacity_select:n #1
  { \__opacity_select:nN {#1} \__opacity_backend_select:n }
\cs_new_protected:Npn \opacity_fill:n #1
  { \__opacity_select:nN {#1} \__opacity_backend_fill:n }
\cs_new_protected:Npn \opacity_stroke:n #1
  { \__opacity_select:nN {#1} \__opacity_backend_stroke:n }
\cs_new_protected:Npn \__opacity_select:nN #1#2
  {
    \fp_set:Nn \l__opacity_tmp_fp { #1 }
    \bool_lazy_or:nnTF
      { \fp_compare_p:nNn \l__opacity_tmp_fp < \c_zero_fp }
      { \fp_compare_p:nNn \l__opacity_tmp_fp > \c_one_fp }
      { \msg_error:nnn { opacity } { out-of-range } {#1} }
      { \exp_args:Ne #2 { \fp_use:N \l__opacity_tmp_fp } }
  }
\msg_new:nnnn { opacity } { out-of-range }
  { Opacity~value~out~of~range. }
  {
    LaTeX~was~asked~to~set~opacity~of~#1,~but~only~values~in~the~range~
    0~to~1~are~supported.
  }
%% File: l3pdf.dtx
\scan_new:N \s__pdf_stop
\bool_new:N \g__pdf_init_bool
\bool_lazy_and:nnT
  { \str_if_eq_p:Vn \fmtname { LaTeX2e } }
  { \tl_if_exist_p:N \@expl@finalise@setup@@ }
  {
    \tl_gput_right:Nn \@expl@finalise@setup@@
      {
        \tl_gput_right:Nn \@kernel@after@begindocument
          { \bool_gset_true:N \g__pdf_init_bool }
      }
  }
\cs_new_protected:Npn \pdf_uncompress:
  {
    \bool_if:NF \g__pdf_init_bool
      {
        \__pdf_backend_compresslevel:n { 0 }
        \__pdf_backend_compress_objects:n { \c_false_bool }
      }
  }
\int_new:N \g__pdf_backend_object_int
\cs_new_protected:Npn \pdf_object_new:n #1
  {
    \__pdf_backend_object_new:
    \__pdf_object_record:nN {#1} \g__pdf_backend_object_int
  }
\cs_new_protected:Npn \pdf_object_write:nnn #1#2#3
  {
    \exp_args:Ne \__pdf_backend_object_write:nnn
      { \__pdf_object_retrieve:n {#1} } {#2} {#3}
    \bool_gset_true:N \g__pdf_init_bool
  }
\cs_generate_variant:Nn \pdf_object_write:nnn { nne , nnx }
\cs_new:Npn \pdf_object_ref:n #1
  {
    \exp_args:Ne \__pdf_backend_object_ref:n
      { \__pdf_object_retrieve:n {#1} }
  }
\cs_new:Npn \__kernel_pdf_object_id:n #1
  {
    \exp_args:Ne \__pdf_backend_object_id:n
```

```
      { \__pdf_object_retrieve:n {#1} }
  }
\sys_if_engine_luatex:F
  {
    \cs_new_protected:Npn \__pdf_object_record:nN #1#2
      {
        \int_const:cn
          { c__pdf_object_ #1 _int } {#2}
      }
    \cs_new:Npn \__pdf_object_retrieve:n #1
      {
        \int_if_exist:cTF { c__pdf_object_ #1 _int }
          {
            \int_use:c
              { c__pdf_object_ #1 _int }
          }
          { 0 }
      }
  }
\prg_new_conditional:Npnn \pdf_object_if_exist:n #1 { p , T , F , TF }
  {
    \int_compare:nNnTF { \__pdf_object_retrieve:n {#1} } = 0
      \prg_return_false:
      \prg_return_true:
  }
\cs_new_protected:Npn \pdf_object_new_indexed:nn #1#2
  {
    \__pdf_backend_object_new:
    \__pdf_object_record:neN {#1}
      { \int_eval:n {#2} } \g__pdf_backend_object_int
  }
\cs_new_protected:Npn \pdf_object_write_indexed:nnnn #1#2#3#4
  {
    \exp_args:Ne \__pdf_backend_object_write:nnn
      { \__pdf_object_retrieve:ne {#1} { \int_eval:n {#2} } } {#3} {#4}
    \bool_gset_true:N \g__pdf_init_bool
  }
\cs_generate_variant:Nn \pdf_object_write_indexed:nnnn { nnne }
\cs_new:Npn \pdf_object_ref_indexed:nn #1#2
  {
    \exp_args:Ne \__pdf_backend_object_ref:n
      { \__pdf_object_retrieve:ne {#1} { \int_eval:n {#2} } }
  }
\cs_new:Npn \__kernel_pdf_object_id_indexed:nn #1#2
  {
    \exp_args:Ne \__pdf_backend_object_id:n
```

```
      { \__pdf_object_retrieve:ne {#1} { \int_eval:n {#2} } }
  }
\sys_if_engine_luatex:F
  {
    \cs_new_protected:Npn \__pdf_object_record:nnN #1#2#3
      {
        \use:e
          {
            \__pdf_object_record:NnN
               \__pdf_object_index_split:nn {#1} {#2}
               \exp_not:N #3
          }
      }
    \cs_new_protected:Npn \__pdf_object_record:NnN #1#2#3
      {
        \intarray_if_exist:NF #1
          { \intarray_new:Nn #1 \c__pdf_object_block_size_int }
        \intarray_gset:Nnn #1 {#2} #3
      }
    \cs_new:Npn \__pdf_object_retrieve:nn #1#2
      {
        \use:e
          {
            \exp_not:N \__pdf_object_retrieve:Nn
              \__pdf_object_index_split:nn {#1} {#2}
          }
      }
    \cs_new:Npn \__pdf_object_retrieve:Nn #1#2
      { \intarray_item:Nn #1 {#2} }
    \cs_new:Npn \__pdf_object_index_split:nn #1#2
      {
        \exp_not:c
          {
            g__pdf_object_ #1 _
            \int_eval:n
              {
                \int_div_truncate:nn { #2 - 1 }
                  \c__pdf_object_block_size_int + 1
              }
            _intarray
          }
          {
            \int_eval:n
              { \int_mod:nn { #2 - 1 } \c__pdf_object_block_size_int + 1 }
          }
      }
    \int_const:Nn \c__pdf_object_block_size_int { 10000 }
  }
\cs_generate_variant:Nn \__pdf_object_record:nnN { ne }
\cs_generate_variant:Nn \__pdf_object_retrieve:nn { ne }
\cs_new_protected:Npn \pdf_object_unnamed_write:nn #1#2
  {
    \exp_args:Ne \__pdf_backend_object_now:nn {#1} {#2}
    \bool_gset_true:N \g__pdf_init_bool
  }
\cs_generate_variant:Nn \pdf_object_unnamed_write:nn { ne , nx }
\cs_new:Npn \pdf_object_ref_last: { \__pdf_backend_object_last: }
\cs_new:Npn \pdf_pageobject_ref:n #1
  { \exp_args:Ne \__pdf_backend_pageobject_ref:n {#1} }
\prg_new_conditional:Npnn \pdf_version_compare:Nn #1#2 { p , T , F , TF }
  { \use:c { __pdf_version_compare_ #1 :w } #2 . . \s__pdf_stop }
\cs_new:cpn { __pdf_version_compare_=:w } #1 . #2 . #3 \s__pdf_stop
  {
    \bool_lazy_and:nnTF
      { \int_compare_p:nNn \__pdf_backend_version_major: = {#1} }
      { \int_compare_p:nNn \__pdf_backend_version_minor: = {#2} }
      { \prg_return_true: }
      { \prg_return_false: }
  }
\cs_new:cpn { __pdf_version_compare_<:w } #1 . #2 . #3 \s__pdf_stop
  {
    \bool_lazy_or:nnTF
      { \int_compare_p:nNn \__pdf_backend_version_major: < {#1} }
      {
        \bool_lazy_and_p:nn
          { \int_compare_p:nNn \__pdf_backend_version_major: = {#1} }
          { \int_compare_p:nNn \__pdf_backend_version_minor: < {#2} }
      }
      { \prg_return_true: }
      { \prg_return_false: }
  }
\cs_new:cpn { __pdf_version_compare_>:w } #1 . #2 . #3 \s__pdf_stop
  {
    \bool_lazy_or:nnTF
      { \int_compare_p:nNn \__pdf_backend_version_major: > {#1} }
      {
        \bool_lazy_and_p:nn
          { \int_compare_p:nNn \__pdf_backend_version_major: = {#1} }
          { \int_compare_p:nNn \__pdf_backend_version_minor: > {#2} }
      }
      { \prg_return_true: }
      { \prg_return_false: }
  }
\cs_new_protected:Npn \pdf_version_gset:n #1
  { \__pdf_version_gset:w  #1 . . \s__pdf_stop }
\cs_new_protected:Npn \pdf_version_min_gset:n #1
  {
    \pdf_version_compare:NnT < {#1}
      { \__pdf_version_gset:w  #1 . . \s__pdf_stop }
  }
\cs_new_protected:Npn \__pdf_version_gset:w  #1 . #2 . #3\s__pdf_stop
  {
    \bool_if:NF \g__pdf_init_bool
      {
        \__pdf_backend_version_major_gset:n {#1}
        \__pdf_backend_version_minor_gset:n {#2}
      }
  }
\cs_new:Npn \pdf_version:
  { \__pdf_backend_version_major: . \__pdf_backend_version_minor: }
\cs_new:Npn \pdf_version_major: { \__pdf_backend_version_major: }
\cs_new:Npn \pdf_version_minor: { \__pdf_backend_version_minor: }
\cs_new_protected:Npn \pdf_pagesize_gset:nn #1#2
  { \__pdf_backend_pagesize_gset:nn {#1} {#2} }
\cs_new_protected:Npn \pdf_destination:nn #1#2
  { \__pdf_backend_destination:nn {#1} {#2} }
\cs_new_protected:Npn \pdf_destination:nnnn #1#2#3#4
  {
    \hbox_to_zero:n
      { \__pdf_backend_destination:nnnn {#1} {#2} {#3} {#4} }
  }
\cs_if_exist:NT \@kernel@before@begindocument
  {
    \tl_gput_right:Nn \@kernel@before@begindocument
      {
        \bool_lazy_all:nT
          {
            { \cs_if_exist_p:N \stockheight }
            { \cs_if_exist_p:N \stockwidth }
            { \cs_if_exist_p:N \IfDocumentMetadataTF }
            { \IfDocumentMetadataTF { \c_true_bool } { \c_false_bool } }
            { \int_compare_p:nNn \tex_mag:D = { 1000 } }
          }
          {
            \bool_lazy_and:nnTF
              { \dim_compare_p:nNn \stockheight > { 0pt } }
              { \dim_compare_p:nNn \stockwidth  > { 0pt } }
              {
                \__pdf_backend_pagesize_gset:nn
                  \stockwidth \stockheight
              }
              {
                \bool_lazy_or:nnF
                  { \dim_compare_p:nNn \stockheight < { 0pt } }
                  { \dim_compare_p:nNn \stockwidth  < { 0pt } }
                  {
                    \bool_lazy_and:nnT
                      { \dim_compare_p:nNn \paperheight > { 0pt } }
                      { \dim_compare_p:nNn \paperwidth  > { 0pt } }
                      {
                        \__pdf_backend_pagesize_gset:nn
                          \paperwidth \paperheight
                      }
                  }
              }
          }
      }
  }
%% File: l3coffins.dtx
\box_new:N \l__coffin_internal_box
\dim_new:N \l__coffin_internal_dim
\tl_new:N  \l__coffin_internal_tl
\prop_const_from_keyval:Nn \c__coffin_corners_prop
  {
    tl = { 0pt } { 0pt } ,
    tr = { 0pt } { 0pt } ,
    bl = { 0pt } { 0pt } ,
    br = { 0pt } { 0pt } ,
  }
\prop_const_from_keyval:Nn \c__coffin_poles_prop
  {
    l  = { 0pt } { 0pt } { 0pt } { 1000pt } ,
    hc = { 0pt } { 0pt } { 0pt } { 1000pt } ,
    r  = { 0pt } { 0pt } { 0pt } { 1000pt } ,
    b  = { 0pt } { 0pt } { 1000pt } { 0pt } ,
    vc = { 0pt } { 0pt } { 1000pt } { 0pt } ,
    t  = { 0pt } { 0pt } { 1000pt } { 0pt } ,
    B  = { 0pt } { 0pt } { 1000pt } { 0pt } ,
    H  = { 0pt } { 0pt } { 1000pt } { 0pt } ,
    T  = { 0pt } { 0pt } { 1000pt } { 0pt } ,
  }
\fp_new:N \l__coffin_slope_A_fp
\fp_new:N \l__coffin_slope_B_fp
\bool_new:N \l__coffin_error_bool
```

```
\dim_new:N \l__coffin_offset_x_dim
\dim_new:N \l__coffin_offset_y_dim
\tl_new:N \l__coffin_pole_a_tl
\tl_new:N \l__coffin_pole_b_tl
\dim_new:N \l__coffin_x_dim
\dim_new:N \l__coffin_y_dim
\dim_new:N \l__coffin_x_prime_dim
\dim_new:N \l__coffin_y_prime_dim
\cs_new_eq:NN \__coffin_to_value:N \tex_number:D
\prg_new_conditional:Npnn \coffin_if_exist:N #1 { p , T , F , TF }
  {
    \cs_if_exist:NTF #1
      {
        \cs_if_exist:cTF { coffin ~ \__coffin_to_value:N #1 ~ poles }
          { \prg_return_true: }
          { \prg_return_false: }
      }
      { \prg_return_false: }
  }
\prg_generate_conditional_variant:Nnn \coffin_if_exist:N
  { c } { p , T , F , TF }
\cs_new_protected:Npn \__coffin_if_exist:NT #1#2
  {
    \coffin_if_exist:NTF #1
      { #2 }
      {
        \msg_error:nne { coffin } { unknown }
          { \token_to_str:N #1 }
      }
  }
\cs_new_protected:Npn \coffin_clear:N #1
  {
    \__coffin_if_exist:NT #1
      {
        \box_clear:N #1
        \__coffin_reset_structure:N #1
      }
  }
\cs_generate_variant:Nn \coffin_clear:N { c }
\cs_new_protected:Npn \coffin_gclear:N #1
  {
    \__coffin_if_exist:NT #1
      {
        \box_gclear:N #1
        \__coffin_greset_structure:N #1
      }
  }
\cs_generate_variant:Nn \coffin_gclear:N { c }
\cs_new_protected:Npn \coffin_new:N #1
  {
    \box_new:N #1
    \debug_suspend:
    \prop_gclear_new:c { coffin ~ \__coffin_to_value:N #1 ~ corners }
    \prop_gclear_new:c { coffin ~ \__coffin_to_value:N #1 ~ poles }
    \prop_gset_eq:cN { coffin ~ \__coffin_to_value:N #1 ~ corners }
      \c__coffin_corners_prop
    \prop_gset_eq:cN { coffin ~ \__coffin_to_value:N #1 ~ poles }
      \c__coffin_poles_prop
    \debug_resume:
  }
\cs_generate_variant:Nn \coffin_new:N { c }
\cs_new_protected:Npn \hcoffin_set:Nn #1#2
  {
    \__coffin_if_exist:NT #1
      {
        \hbox_set:Nn #1
          {
            \color_ensure_current:
            #2
          }
        \coffin_reset_poles:N #1
      }
  }
\cs_generate_variant:Nn \hcoffin_set:Nn { c }
\cs_new_protected:Npn \hcoffin_gset:Nn #1#2
  {
    \__coffin_if_exist:NT #1
      {
        \hbox_gset:Nn #1
          {
            \color_ensure_current:
            #2
          }
        \coffin_greset_poles:N #1
      }
  }
\cs_generate_variant:Nn \hcoffin_gset:Nn { c }
\cs_new_protected:Npn \vcoffin_set:Nnn #1#2#3
  {
    \__coffin_set_vertical:NnnNNN #1 {#2} {#3}
      \vbox_set:Nn \coffin_reset_poles:N \__coffin_set_pole:Nnn
  }
```

```
\cs_generate_variant:Nn \vcoffin_set:Nnn { c }
\cs_new_protected:Npn \vcoffin_gset:Nnn #1#2#3
  {
    \__coffin_set_vertical:NnnNNN #1 {#2} {#3}
      \vbox_gset:Nn \coffin_greset_poles:N \__coffin_gset_pole:Nnn
  }
\cs_generate_variant:Nn \vcoffin_gset:Nnn { c }
\cs_new_protected:Npn \__coffin_set_vertical:NnnNNN #1#2#3#4#5#6
  {
    \__coffin_if_exist:NT #1
      {
        #4 #1
          {
            \dim_set:Nn \tex_hsize:D {#2}
            \__coffin_set_vertical_aux:
            #3
          }
        #5 #1
        \vbox_set_top:Nn \l__coffin_internal_box { \vbox_unpack:N #1 }
        #6 #1 { T }
          {
            { 0pt }
            {
              \dim_eval:n
                { \box_ht:N #1 - \box_ht:N \l__coffin_internal_box }
            }
            { 1000pt }
            { 0pt }
          }
        \box_clear:N \l__coffin_internal_box
      }
  }
\cs_new_protected:Npe \__coffin_set_vertical_aux:
  {
    \bool_lazy_and:nnT
      { \cs_if_exist_p:N \fmtname }
      { \str_if_eq_p:Vn \fmtname { LaTeX2e } }
      {
        \dim_set_eq:NN \exp_not:N \linewidth \tex_hsize:D
        \dim_set_eq:NN \exp_not:N \columnwidth \tex_hsize:D
      }
  }
\cs_new_protected:Npn \hcoffin_set:Nw #1
  {
    \__coffin_if_exist:NT #1
      {
        \hbox_set:Nw #1 \color_ensure_current:
          \cs_set_protected:Npn \hcoffin_set_end:
            {
              \hbox_set_end:
              \coffin_reset_poles:N #1
            }
      }
  }
\cs_generate_variant:Nn \hcoffin_set:Nw { c }
\cs_new_protected:Npn \hcoffin_gset:Nw #1
  {
    \__coffin_if_exist:NT #1
      {
        \hbox_gset:Nw #1 \color_ensure_current:
          \cs_set_protected:Npn \hcoffin_gset_end:
            {
              \hbox_gset_end:
              \coffin_greset_poles:N #1
            }
      }
  }
\cs_generate_variant:Nn \hcoffin_gset:Nw { c }
\cs_new_protected:Npn \hcoffin_set_end: { }
\cs_new_protected:Npn \hcoffin_gset_end: { }
\cs_new_protected:Npn \vcoffin_set:Nnw #1#2
  {
    \__coffin_set_vertical:NnNNNNw #1 {#2} \vbox_set:Nw
      \vcoffin_set_end:
      \vbox_set_end: \coffin_reset_poles:N \__coffin_set_pole:Nnn
  }
\cs_generate_variant:Nn \vcoffin_set:Nnw { c }
\cs_new_protected:Npn \vcoffin_gset:Nnw #1#2
  {
    \__coffin_set_vertical:NnNNNNw #1 {#2} \vbox_gset:Nw
      \vcoffin_gset_end:
      \vbox_gset_end: \coffin_greset_poles:N \__coffin_gset_pole:Nnn
  }
\cs_generate_variant:Nn \vcoffin_gset:Nnw { c }
\cs_new_protected:Npn \__coffin_set_vertical:NnNNNNw #1#2#3#4#5#6#7
  {
    \__coffin_if_exist:NT #1
      {
        #3 #1
          \dim_set:Nn \tex_hsize:D {#2}
          \__coffin_set_vertical_aux:
          \cs_set_protected:Npn #4
```

```
            {
              #5
              #6 #1
              \vbox_set_top:Nn \l__coffin_internal_box { \vbox_unpack:N #1 }
              #7 #1 { T }
                {
                  { 0pt }
                  {
                    \dim_eval:n
                      { \box_ht:N #1 - \box_ht:N \l__coffin_internal_box }
                  }
                  { 1000pt }
                  { 0pt }
                }
              \box_clear:N \l__coffin_internal_box
            }
      }
  }
\cs_new_protected:Npn \vcoffin_set_end: { }
\cs_new_protected:Npn \vcoffin_gset_end: { }
\cs_new_protected:Npn \coffin_set_eq:NN #1#2
  {
    \__coffin_if_exist:NT #2
      {
        \box_set_eq:NN #1 #2
        \prop_set_eq:cc { coffin ~ \__coffin_to_value:N #1 ~ corners }
          { coffin ~ \__coffin_to_value:N #2 ~ corners }
        \prop_set_eq:cc { coffin ~ \__coffin_to_value:N #1 ~ poles }
          { coffin ~ \__coffin_to_value:N #2 ~ poles }
      }
  }
\cs_generate_variant:Nn \coffin_set_eq:NN { c , Nc , cc }
\cs_new_protected:Npn \coffin_gset_eq:NN #1#2
  {
    \__coffin_if_exist:NT #2
      {
        \box_gset_eq:NN #1 #2
        \prop_gset_eq:cc { coffin ~ \__coffin_to_value:N #1 ~ corners }
          { coffin ~ \__coffin_to_value:N #2 ~ corners }
        \prop_gset_eq:cc { coffin ~ \__coffin_to_value:N #1 ~ poles }
          { coffin ~ \__coffin_to_value:N #2 ~ poles }
      }
  }
\cs_generate_variant:Nn \coffin_gset_eq:NN { c , Nc , cc }
\coffin_new:N \c_empty_coffin
\coffin_new:N \l__coffin_aligned_coffin
```

```
\coffin_new:N \l__coffin_aligned_internal_coffin
\coffin_new:N \l_tmpa_coffin
\coffin_new:N \l_tmpb_coffin
\coffin_new:N \g_tmpa_coffin
\coffin_new:N \g_tmpb_coffin
\cs_new_eq:NN \coffin_dp:N \box_dp:N
\cs_new_eq:NN \coffin_dp:c \box_dp:c
\cs_new_eq:NN \coffin_ht:N \box_ht:N
\cs_new_eq:NN \coffin_ht:c \box_ht:c
\cs_new_eq:NN \coffin_ht_plus_dp:N \box_ht_plus_dp:N
\cs_new_eq:NN \coffin_ht_plus_dp:c \box_ht_plus_dp:c
\cs_new_eq:NN \coffin_wd:N \box_wd:N
\cs_new_eq:NN \coffin_wd:c \box_wd:c
\cs_new_protected:Npn \__coffin_get_pole:NnN #1#2#3
  {
    \prop_get:cnNF
      { coffin ~ \__coffin_to_value:N #1 ~ poles } {#2} #3
      {
        \msg_error:nnee { coffin } { unknown-pole }
          { \exp_not:n {#2} } { \token_to_str:N #1 }
        \tl_set:Nn #3 { { 0pt } { 0pt } { 0pt } { 0pt } }
      }
  }
\cs_new_protected:Npn \__coffin_reset_structure:N #1
  {
    \prop_set_eq:cN { coffin ~ \__coffin_to_value:N #1 ~ corners }
      \c__coffin_corners_prop
    \prop_set_eq:cN { coffin ~ \__coffin_to_value:N #1 ~ poles }
      \c__coffin_poles_prop
  }
\cs_new_protected:Npn \__coffin_greset_structure:N #1
  {
    \prop_gset_eq:cN { coffin ~ \__coffin_to_value:N #1 ~ corners }
      \c__coffin_corners_prop
    \prop_gset_eq:cN { coffin ~ \__coffin_to_value:N #1 ~ poles }
      \c__coffin_poles_prop
  }
\cs_new_protected:Npn \coffin_set_horizontal_pole:Nnn #1#2#3
  { \__coffin_set_horizontal_pole:NnnN #1 {#2} {#3} \prop_put:cne }
\cs_generate_variant:Nn \coffin_set_horizontal_pole:Nnn { c }
\cs_new_protected:Npn \coffin_gset_horizontal_pole:Nnn #1#2#3
  { \__coffin_set_horizontal_pole:NnnN #1 {#2} {#3} \prop_gput:cne }
\cs_generate_variant:Nn \coffin_gset_horizontal_pole:Nnn { c }
\cs_new_protected:Npn \__coffin_set_horizontal_pole:NnnN #1#2#3#4
  {
    \__coffin_if_exist:NT #1
```

```
      {
        #4 { coffin ~ \__coffin_to_value:N #1 ~ poles }
          {#2}
          {
            { 0pt } { \dim_eval:n {#3} }
            { 1000pt } { 0pt }
          }
      }
  }
\cs_new_protected:Npn \coffin_set_vertical_pole:Nnn #1#2#3
  { \__coffin_set_vertical_pole:NnnN #1 {#2} {#3} \prop_put:cne }
\cs_generate_variant:Nn \coffin_set_vertical_pole:Nnn { c }
\cs_new_protected:Npn \coffin_gset_vertical_pole:Nnn #1#2#3
  { \__coffin_set_vertical_pole:NnnN #1 {#2} {#3} \prop_gput:cne }
  \cs_generate_variant:Nn \coffin_gset_vertical_pole:Nnn { c }
\cs_new_protected:Npn \__coffin_set_vertical_pole:NnnN #1#2#3#4
  {
    \__coffin_if_exist:NT #1
      {
        #4 { coffin ~ \__coffin_to_value:N #1 ~ poles }
          {#2}
          {
            { \dim_eval:n {#3} } { 0pt }
            { 0pt } { 1000pt }
          }
      }
  }
\cs_new_protected:Npn \__coffin_set_pole:Nnn #1#2#3
  {
    \prop_put:cne { coffin ~ \__coffin_to_value:N #1 ~ poles }
      {#2} {#3}
  }
\cs_new_protected:Npn \__coffin_gset_pole:Nnn #1#2#3
  {
    \prop_gput:cne { coffin ~ \__coffin_to_value:N #1 ~ poles }
      {#2} {#3}
  }
\cs_new_protected:Npn \coffin_reset_poles:N #1
  {
    \__coffin_reset_structure:N #1
    \__coffin_update_corners:N #1
    \__coffin_update_poles:N #1
  }
\cs_new_protected:Npn \coffin_greset_poles:N #1
  {
    \__coffin_greset_structure:N #1
    \__coffin_gupdate_corners:N #1
    \__coffin_gupdate_poles:N #1
  }
\cs_new_protected:Npn \__coffin_update_corners:N #1
  { \__coffin_update_corners:NN #1 \prop_put:Nne }
\cs_new_protected:Npn \__coffin_gupdate_corners:N #1
  { \__coffin_update_corners:NN #1 \prop_gput:Nne }
\cs_new_protected:Npn \__coffin_update_corners:NN #1#2
  {
    \exp_args:Nc \__coffin_update_corners:NNN
      { coffin ~ \__coffin_to_value:N #1 ~ corners }
      #1 #2
  }
\cs_new_protected:Npn \__coffin_update_corners:NNN #1#2#3
  {
    #3 #1
      { tl }
      { { 0pt } { \dim_eval:n { \box_ht:N #2 } } }
    #3 #1
      { tr }
      {
        { \dim_eval:n { \box_wd:N #2 } }
        { \dim_eval:n { \box_ht:N #2 } }
      }
    #3 #1
      { bl }
      { { 0pt } { \dim_eval:n { -\box_dp:N #2 } } }
    #3 #1
      { br }
      {
        { \dim_eval:n { \box_wd:N #2 } }
        { \dim_eval:n { -\box_dp:N #2 } }
      }
  }
\cs_new_protected:Npn \__coffin_update_poles:N #1
  { \__coffin_update_poles:NN #1 \prop_put:Nne }
\cs_new_protected:Npn \__coffin_gupdate_poles:N #1
  { \__coffin_update_poles:NN #1 \prop_gput:Nne }
\cs_new_protected:Npn \__coffin_update_poles:NN #1#2
  {
    \exp_args:Nc \__coffin_update_poles:NNN
      { coffin ~ \__coffin_to_value:N #1 ~ poles }
      #1 #2
  }
\cs_new_protected:Npn \__coffin_update_poles:NNN #1#2#3
  {
    #3 #1 { hc }
      {
        { \dim_eval:n { 0.5 \box_wd:N #2 } }
        { 0pt } { 0pt } { 1000pt }
      }
    #3 #1 { r }
      {
        { \dim_eval:n { \box_wd:N #2 } }
        { 0pt } { 0pt } { 1000pt }
      }
    #3 #1 { vc }
      {
        { 0pt }
        { \dim_eval:n { ( \box_ht:N #2 - \box_dp:N #2 ) / 2 } }
        { 1000pt }
        { 0pt }
      }
    #3 #1 { t }
      {
        { 0pt }
        { \dim_eval:n { \box_ht:N #2 } }
        { 1000pt }
        { 0pt }
      }
    #3 #1 { b }
      {
        { 0pt }
        { \dim_eval:n { -\box_dp:N #2 } }
        { 1000pt }
        { 0pt }
      }
  }
\cs_new_protected:Npn \__coffin_calculate_intersection:Nnn #1#2#3
  {
    \__coffin_get_pole:NnN #1 {#2} \l__coffin_pole_a_tl
    \__coffin_get_pole:NnN #1 {#3} \l__coffin_pole_b_tl
    \bool_set_false:N \l__coffin_error_bool
    \exp_last_two_unbraced:Noo
      \__coffin_calculate_intersection:nnnnnnnn
        \l__coffin_pole_a_tl \l__coffin_pole_b_tl
    \bool_if:NT \l__coffin_error_bool
      {
        \msg_error:nn { coffin } { no-pole-intersection }
        \dim_zero:N \l__coffin_x_dim
        \dim_zero:N \l__coffin_y_dim
      }
```

```
 }
\cs_new_protected:Npn \__coffin_calculate_intersection:nnnnnnnn
 #1#2#3#4#5#6#7#8
 {
  \dim_compare:nNnTF {#3} = \c_zero_dim
   {
    \dim_set:Nn \l__coffin_x_dim {#1}
    \dim_compare:nNnTF {#7} = \c_zero_dim
     { \bool_set_true:N \l__coffin_error_bool }
     {
      \dim_set:Nn \l__coffin_y_dim
       {
        \dim_compare:nNnTF {#8} = \c_zero_dim
         {#6}
         {
          \fp_to_dim:n
           {
             ( \dim_to_fp:n {#8} / \dim_to_fp:n {#7} )
            * ( \dim_to_fp:n {#1} - \dim_to_fp:n {#5} )
            + \dim_to_fp:n {#6}
           }
         }
       }
     }
   }
   {
    \dim_compare:nNnTF {#4} = \c_zero_dim
     {
      \dim_set:Nn \l__coffin_y_dim {#2}
      \dim_compare:nNnTF {#8} = { \c_zero_dim }
       { \bool_set_true:N \l__coffin_error_bool }
       {
        \dim_set:Nn \l__coffin_x_dim
         {
          \dim_compare:nNnTF {#7} = \c_zero_dim
           {#5}
           {
            \fp_to_dim:n
             {
               ( \dim_to_fp:n {#7} / \dim_to_fp:n {#8} )
              * ( \dim_to_fp:n {#4} - \dim_to_fp:n {#6} )
              + \dim_to_fp:n {#5}
             }
           }
         }
       }
```

```
          }
          {
            \use:e
              {
                \__coffin_calculate_intersection:nnnnnn
                  { \dim_to_fp:n {#4} / \dim_to_fp:n {#3} }
                  { \dim_to_fp:n {#8} / \dim_to_fp:n {#7} }
              }
                {#1} {#2} {#5} {#6}
          }
      }
  }
\cs_new_protected:Npn \__coffin_calculate_intersection:nnnnnn #1#2#3#4#5#6
  {
    \fp_compare:nNnTF {#1} = {#2}
      { \bool_set_true:N \l__coffin_error_bool }
      {
        \dim_set:Nn \l__coffin_x_dim
          {
            \fp_to_dim:n
              {
                (
                    #1 * \dim_to_fp:n {#3}
                  - #2 * \dim_to_fp:n {#5}
                  - \dim_to_fp:n {#4}
                  + \dim_to_fp:n {#6}
                )
                /
                ( #1 - #2 )
              }
          }
        \dim_set:Nn \l__coffin_y_dim
          {
            \fp_to_dim:n
              {
                  #1 * ( \l__coffin_x_dim - \dim_to_fp:n {#3} )
                + \dim_to_fp:n {#4}
              }
          }
      }
  }
\fp_new:N \l__coffin_sin_fp
\fp_new:N \l__coffin_cos_fp
\prop_new:N \l__coffin_bounding_prop
\prop_new:N \l__coffin_corners_prop
\prop_new:N \l__coffin_poles_prop
```

```
\dim_new:N \l__coffin_bounding_shift_dim
\dim_new:N \l__coffin_left_corner_dim
\dim_new:N \l__coffin_right_corner_dim
\dim_new:N \l__coffin_bottom_corner_dim
\dim_new:N \l__coffin_top_corner_dim
\cs_new_protected:Npn \coffin_rotate:Nn #1#2
  { \__coffin_rotate:NnNNN #1 {#2} \box_rotate:Nn \prop_set_eq:cN \hbox_set:Nn }
\cs_generate_variant:Nn \coffin_rotate:Nn { c }
\cs_new_protected:Npn \coffin_grotate:Nn #1#2
  { \__coffin_rotate:NnNNN #1 {#2} \box_grotate:Nn \prop_gset_eq:cN \hbox_gset:Nn }
\cs_generate_variant:Nn \coffin_grotate:Nn { c }
\cs_new_protected:Npn \__coffin_rotate:NnNNN #1#2#3#4#5
  {
    \fp_set:Nn \l__coffin_sin_fp { sind ( #2 ) }
    \fp_set:Nn \l__coffin_cos_fp { cosd ( #2 ) }
    \prop_set_eq:Nc \l__coffin_corners_prop
      { coffin ~ \__coffin_to_value:N #1 ~ corners }
    \prop_set_eq:Nc \l__coffin_poles_prop
      { coffin ~ \__coffin_to_value:N #1 ~ poles }
    \prop_map_inline:Nn \l__coffin_corners_prop
      { \__coffin_rotate_corner:Nnnn #1 {##1} ##2 }
    \prop_map_inline:Nn \l__coffin_poles_prop
      { \__coffin_rotate_pole:Nnnnnn #1 {##1} ##2 }
    \__coffin_set_bounding:N #1
    \prop_map_inline:Nn \l__coffin_bounding_prop
      { \__coffin_rotate_bounding:nnn {##1} ##2 }
    \__coffin_find_corner_maxima:N #1
    \__coffin_find_bounding_shift:
    #3 #1 {#2}
    \hbox_set:Nn \l__coffin_internal_box
      {
        \__kernel_kern:n
            { \l__coffin_bounding_shift_dim - \l__coffin_left_corner_dim }
        \box_move_down:nn { \l__coffin_bottom_corner_dim }
          { \box_use:N #1 }
      }
    \box_set_ht:Nn \l__coffin_internal_box
      { \l__coffin_top_corner_dim - \l__coffin_bottom_corner_dim }
    \box_set_dp:Nn \l__coffin_internal_box { 0pt }
    \box_set_wd:Nn \l__coffin_internal_box
      { \l__coffin_right_corner_dim - \l__coffin_left_corner_dim }
    #5 #1 { \box_use_drop:N \l__coffin_internal_box }
    \prop_map_inline:Nn \l__coffin_corners_prop
      { \__coffin_shift_corner:Nnnn #1 {##1} ##2 }
    \prop_map_inline:Nn \l__coffin_poles_prop
      { \__coffin_shift_pole:Nnnnnn #1 {##1} ##2 }
    #4 { coffin ~ \__coffin_to_value:N #1 ~ corners }
      \l__coffin_corners_prop
    #4 { coffin ~ \__coffin_to_value:N #1 ~ poles }
      \l__coffin_poles_prop
  }
\cs_new_protected:Npn \__coffin_set_bounding:N #1
  {
    \prop_put:Nne \l__coffin_bounding_prop { tl }
      { { 0pt } { \dim_eval:n { \box_ht:N #1 } } }
    \prop_put:Nne \l__coffin_bounding_prop { tr }
      {
        { \dim_eval:n { \box_wd:N #1 } }
        { \dim_eval:n { \box_ht:N #1 } }
      }
    \dim_set:Nn \l__coffin_internal_dim { -\box_dp:N #1 }
    \prop_put:Nne \l__coffin_bounding_prop { bl }
      { { 0pt } { \dim_use:N \l__coffin_internal_dim } }
    \prop_put:Nne \l__coffin_bounding_prop { br }
      {
        { \dim_eval:n { \box_wd:N #1 } }
        { \dim_use:N \l__coffin_internal_dim }
      }
  }
\cs_new_protected:Npn \__coffin_rotate_bounding:nnn #1#2#3
  {
    \__coffin_rotate_vector:nnNN {#2} {#3} \l__coffin_x_dim \l__coffin_y_dim
    \prop_put:Nne \l__coffin_bounding_prop {#1}
      { { \dim_use:N \l__coffin_x_dim } { \dim_use:N \l__coffin_y_dim } }
  }
\cs_new_protected:Npn \__coffin_rotate_corner:Nnnn #1#2#3#4
  {
    \__coffin_rotate_vector:nnNN {#3} {#4} \l__coffin_x_dim \l__coffin_y_dim
    \prop_put:Nne \l__coffin_corners_prop {#2}
      { { \dim_use:N \l__coffin_x_dim } { \dim_use:N \l__coffin_y_dim } }
  }
\cs_new_protected:Npn \__coffin_rotate_pole:Nnnnnn #1#2#3#4#5#6
  {
    \__coffin_rotate_vector:nnNN {#3} {#4} \l__coffin_x_dim \l__coffin_y_dim
    \__coffin_rotate_vector:nnNN {#5} {#6}
      \l__coffin_x_prime_dim \l__coffin_y_prime_dim
    \prop_put:Nne \l__coffin_poles_prop {#2}
      {
        { \dim_use:N \l__coffin_x_dim } { \dim_use:N \l__coffin_y_dim }
        { \dim_use:N \l__coffin_x_prime_dim }
        { \dim_use:N \l__coffin_y_prime_dim }
      }
  }
\cs_new_protected:Npn \__coffin_rotate_vector:nnNN #1#2#3#4
  {
    \dim_set:Nn #3
      {
        \fp_to_dim:n
          {
              \dim_to_fp:n {#1} * \l__coffin_cos_fp
            - \dim_to_fp:n {#2} * \l__coffin_sin_fp
          }
      }
    \dim_set:Nn #4
      {
        \fp_to_dim:n
          {
              \dim_to_fp:n {#1} * \l__coffin_sin_fp
            + \dim_to_fp:n {#2} * \l__coffin_cos_fp
          }
      }
  }
\cs_new_protected:Npn \__coffin_find_corner_maxima:N #1
  {
    \dim_set:Nn \l__coffin_top_corner_dim   { -\c_max_dim }
    \dim_set:Nn \l__coffin_right_corner_dim { -\c_max_dim }
    \dim_set:Nn \l__coffin_bottom_corner_dim { \c_max_dim }
    \dim_set:Nn \l__coffin_left_corner_dim   { \c_max_dim }
    \prop_map_inline:Nn \l__coffin_corners_prop
      { \__coffin_find_corner_maxima_aux:nn ##2 }
  }
\cs_new_protected:Npn \__coffin_find_corner_maxima_aux:nn #1#2
  {
    \dim_set:Nn \l__coffin_left_corner_dim
      { \dim_min:nn { \l__coffin_left_corner_dim } {#1} }
    \dim_set:Nn \l__coffin_right_corner_dim
      { \dim_max:nn { \l__coffin_right_corner_dim } {#1} }
    \dim_set:Nn \l__coffin_bottom_corner_dim
      { \dim_min:nn { \l__coffin_bottom_corner_dim } {#2} }
    \dim_set:Nn \l__coffin_top_corner_dim
      { \dim_max:nn { \l__coffin_top_corner_dim } {#2} }
  }
\cs_new_protected:Npn \__coffin_find_bounding_shift:
  {
    \dim_set:Nn \l__coffin_bounding_shift_dim { \c_max_dim }
    \prop_map_inline:Nn \l__coffin_bounding_prop
      { \__coffin_find_bounding_shift_aux:nn ##2 }
  }
```

```
\cs_new_protected:Npn \__coffin_find_bounding_shift_aux:nn #1#2
  {
    \dim_set:Nn \l__coffin_bounding_shift_dim
      { \dim_min:nn { \l__coffin_bounding_shift_dim } {#1} }
  }
\cs_new_protected:Npn \__coffin_shift_corner:Nnnn #1#2#3#4
  {
    \prop_put:Nne \l__coffin_corners_prop {#2}
      {
        { \dim_eval:n { #3 - \l__coffin_left_corner_dim } }
        { \dim_eval:n { #4 - \l__coffin_bottom_corner_dim } }
      }
  }
\cs_new_protected:Npn \__coffin_shift_pole:Nnnnnn #1#2#3#4#5#6
  {
    \prop_put:Nne \l__coffin_poles_prop {#2}
      {
        { \dim_eval:n { #3 - \l__coffin_left_corner_dim } }
        { \dim_eval:n { #4 - \l__coffin_bottom_corner_dim } }
        {#5} {#6}
      }
  }
\fp_new:N \l__coffin_scale_x_fp
\fp_new:N \l__coffin_scale_y_fp
\dim_new:N \l__coffin_scaled_total_height_dim
\dim_new:N \l__coffin_scaled_width_dim
\cs_new_protected:Npn \coffin_resize:Nnn #1#2#3
  {
    \__coffin_resize:NnnNN #1 {#2} {#3}
      \box_resize_to_wd_and_ht_plus_dp:Nnn
      \prop_set_eq:cN
  }
\cs_generate_variant:Nn \coffin_resize:Nnn { c }
\cs_new_protected:Npn \coffin_gresize:Nnn #1#2#3
  {
    \__coffin_resize:NnnNN #1 {#2} {#3}
      \box_gresize_to_wd_and_ht_plus_dp:Nnn
      \prop_gset_eq:cN
  }
\cs_generate_variant:Nn \coffin_gresize:Nnn { c }
\cs_new_protected:Npn \__coffin_resize:NnnNN #1#2#3#4#5
  {
    \fp_set:Nn \l__coffin_scale_x_fp
      { \dim_to_fp:n {#2} / \dim_to_fp:n { \coffin_wd:N #1 } }
    \fp_set:Nn \l__coffin_scale_y_fp
      {
          \dim_to_fp:n {#3}
        / \dim_to_fp:n { \coffin_ht:N #1 + \coffin_dp:N #1 }
      }
    #4 #1 {#2} {#3}
    \__coffin_resize_common:NnnN #1 {#2} {#3} #5
  }
\cs_new_protected:Npn \__coffin_resize_common:NnnN #1#2#3#4
  {
    \prop_set_eq:Nc \l__coffin_corners_prop
      { coffin ~ \__coffin_to_value:N #1 ~ corners }
    \prop_set_eq:Nc \l__coffin_poles_prop
      { coffin ~ \__coffin_to_value:N #1 ~ poles }
    \prop_map_inline:Nn \l__coffin_corners_prop
      { \__coffin_scale_corner:Nnnn #1 {##1} ##2 }
    \prop_map_inline:Nn \l__coffin_poles_prop
      { \__coffin_scale_pole:Nnnnnn #1 {##1} ##2 }
    \fp_compare:nNnT \l__coffin_scale_x_fp < \c_zero_fp
      {
        \prop_map_inline:Nn \l__coffin_corners_prop
          { \__coffin_x_shift_corner:Nnnn #1 {##1} ##2 }
        \prop_map_inline:Nn \l__coffin_poles_prop
          { \__coffin_x_shift_pole:Nnnnnn #1 {##1} ##2 }
      }
    #4 { coffin ~ \__coffin_to_value:N #1 ~ corners }
      \l__coffin_corners_prop
    #4 { coffin ~ \__coffin_to_value:N #1 ~ poles }
      \l__coffin_poles_prop
  }
\cs_new_protected:Npn \coffin_scale:Nnn #1#2#3
  { \__coffin_scale:NnnNN #1 {#2} {#3} \box_scale:Nnn \prop_set_eq:cN }
\cs_generate_variant:Nn \coffin_scale:Nnn { c }
\cs_new_protected:Npn \coffin_gscale:Nnn #1#2#3
  { \__coffin_scale:NnnNN #1 {#2} {#3} \box_gscale:Nnn \prop_gset_eq:cN }
\cs_generate_variant:Nn \coffin_gscale:Nnn { c }
\cs_new_protected:Npn \__coffin_scale:NnnNN #1#2#3#4#5
  {
    \fp_set:Nn \l__coffin_scale_x_fp {#2}
    \fp_set:Nn \l__coffin_scale_y_fp {#3}
    #4 #1 { \l__coffin_scale_x_fp } { \l__coffin_scale_y_fp }
    \dim_set:Nn \l__coffin_internal_dim
      { \coffin_ht:N #1 + \coffin_dp:N #1 }
    \dim_set:Nn \l__coffin_scaled_total_height_dim
      { \fp_abs:n { \l__coffin_scale_y_fp } \l__coffin_internal_dim }
    \dim_set:Nn \l__coffin_scaled_width_dim
      { -\fp_abs:n { \l__coffin_scale_x_fp  } \coffin_wd:N #1 }
    \__coffin_resize_common:NnnN #1
```

```
      { \l__coffin_scaled_width_dim } { \l__coffin_scaled_total_height_dim }
      #5
  }
\cs_new_protected:Npn \__coffin_scale_vector:nnNN #1#2#3#4
  {
    \dim_set:Nn #3
      { \fp_to_dim:n { \dim_to_fp:n {#1} * \l__coffin_scale_x_fp } }
    \dim_set:Nn #4
      { \fp_to_dim:n { \dim_to_fp:n {#2} * \l__coffin_scale_y_fp } }
  }
\cs_new_protected:Npn \__coffin_scale_corner:Nnnn #1#2#3#4
  {
    \__coffin_scale_vector:nnNN {#3} {#4} \l__coffin_x_dim \l__coffin_y_dim
    \prop_put:Nne \l__coffin_corners_prop {#2}
      { { \dim_use:N \l__coffin_x_dim } { \dim_use:N \l__coffin_y_dim } }
  }
\cs_new_protected:Npn \__coffin_scale_pole:Nnnnnn #1#2#3#4#5#6
  {
    \__coffin_scale_vector:nnNN {#3} {#4} \l__coffin_x_dim \l__coffin_y_dim
    \prop_put:Nne \l__coffin_poles_prop {#2}
      {
        { \dim_use:N \l__coffin_x_dim } { \dim_use:N \l__coffin_y_dim }
        {#5} {#6}
      }
  }
\cs_new_protected:Npn \__coffin_x_shift_corner:Nnnn #1#2#3#4
  {
    \prop_put:Nne \l__coffin_corners_prop {#2}
      {
        { \dim_eval:n { #3 + \box_wd:N #1 } } {#4}
      }
  }
\cs_new_protected:Npn \__coffin_x_shift_pole:Nnnnnn #1#2#3#4#5#6
  {
    \prop_put:Nne \l__coffin_poles_prop {#2}
      {
        { \dim_eval:n { #3 + \box_wd:N #1 } } {#4}
        {#5} {#6}
      }
  }
\cs_new_protected:Npn \coffin_join:NnnNnnnn #1#2#3#4#5#6#7#8
  {
    \__coffin_join:NnnNnnnnN #1 {#2} {#3} #4 {#5} {#6} {#7} {#8}
      \coffin_set_eq:NN
  }
\cs_generate_variant:Nn \coffin_join:NnnNnnnn { c , Nnnc , cnnc }
```

```
\cs_new_protected:Npn \coffin_gjoin:NnnNnnnn #1#2#3#4#5#6#7#8
  {
    \__coffin_join:NnnNnnnnN #1 {#2} {#3} #4 {#5} {#6} {#7} {#8}
      \coffin_gset_eq:NN
  }
\cs_generate_variant:Nn \coffin_gjoin:NnnNnnnn { c , Nnnc , cnnc }
\cs_new_protected:Npn \__coffin_join:NnnNnnnnN #1#2#3#4#5#6#7#8#9
  {
    \__coffin_align:NnnNnnnnN
      #1 {#2} {#3} #4 {#5} {#6} {#7} {#8} \l__coffin_aligned_coffin
    \hbox_set:Nn \l__coffin_aligned_coffin
      {
        \dim_compare:nNnT { \l__coffin_offset_x_dim } < \c_zero_dim
          { \__kernel_kern:n { -\l__coffin_offset_x_dim } }
        \hbox_unpack:N \l__coffin_aligned_coffin
        \dim_set:Nn \l__coffin_internal_dim
          { \l__coffin_offset_x_dim - \box_wd:N #1 + \box_wd:N #4 }
        \dim_compare:nNnT \l__coffin_internal_dim < \c_zero_dim
          { \__kernel_kern:n { -\l__coffin_internal_dim } }
      }
    \__coffin_reset_structure:N \l__coffin_aligned_coffin
    \prop_clear:c
      {
        coffin ~ \__coffin_to_value:N \l__coffin_aligned_coffin
        \c_space_tl corners
      }
    \__coffin_update_poles:N \l__coffin_aligned_coffin
    \dim_compare:nNnTF \l__coffin_offset_x_dim < \c_zero_dim
      {
        \__coffin_offset_poles:Nnn #1 { -\l__coffin_offset_x_dim } { 0pt }
        \__coffin_offset_poles:Nnn #4 { 0pt } { \l__coffin_offset_y_dim }
        \__coffin_offset_corners:Nnn #1 { -\l__coffin_offset_x_dim } { 0pt }
        \__coffin_offset_corners:Nnn #4 { 0pt } { \l__coffin_offset_y_dim }
      }
      {
        \__coffin_offset_poles:Nnn #1 { 0pt } { 0pt }
        \__coffin_offset_poles:Nnn #4
          { \l__coffin_offset_x_dim } { \l__coffin_offset_y_dim }
        \__coffin_offset_corners:Nnn #1 { 0pt } { 0pt }
        \__coffin_offset_corners:Nnn #4
          { \l__coffin_offset_x_dim } { \l__coffin_offset_y_dim }
      }
    \__coffin_update_vertical_poles:NNN #1 #4 \l__coffin_aligned_coffin
    #9 #1 \l__coffin_aligned_coffin
  }
\cs_new_protected:Npn \coffin_attach:NnnNnnnn #1#2#3#4#5#6#7#8
```

```
  {
    \__coffin_attach:NnnNnnnnN #1 {#2} {#3} #4 {#5} {#6} {#7} {#8}
      \coffin_set_eq:NN
  }
\cs_generate_variant:Nn \coffin_attach:NnnNnnnn { c , Nnnc , cnnc }
\cs_new_protected:Npn \coffin_gattach:NnnNnnnn #1#2#3#4#5#6#7#8
  {
    \__coffin_attach:NnnNnnnnN #1 {#2} {#3} #4 {#5} {#6} {#7} {#8}
      \coffin_gset_eq:NN
  }
\cs_generate_variant:Nn \coffin_gattach:NnnNnnnn { c , Nnnc , cnnc }
\cs_new_protected:Npn \__coffin_attach:NnnNnnnnN #1#2#3#4#5#6#7#8#9
  {
    \__coffin_align:NnnNnnnnN
      #1 {#2} {#3} #4 {#5} {#6} {#7} {#8} \l__coffin_aligned_coffin
    \box_set_ht:Nn \l__coffin_aligned_coffin { \box_ht:N #1 }
    \box_set_dp:Nn \l__coffin_aligned_coffin { \box_dp:N #1 }
    \box_set_wd:Nn \l__coffin_aligned_coffin { \box_wd:N #1 }
    \__coffin_reset_structure:N \l__coffin_aligned_coffin
    \prop_set_eq:cc
      {
        coffin ~ \__coffin_to_value:N \l__coffin_aligned_coffin
        \c_space_tl corners
      }
      { coffin ~ \__coffin_to_value:N #1 ~ corners }
    \__coffin_update_poles:N  \l__coffin_aligned_coffin
    \__coffin_offset_poles:Nnn #1 { 0pt } { 0pt }
    \__coffin_offset_poles:Nnn #4
      { \l__coffin_offset_x_dim } { \l__coffin_offset_y_dim }
    \__coffin_update_vertical_poles:NNN #1 #4 \l__coffin_aligned_coffin
    #9 #1 \l__coffin_aligned_coffin
  }
\cs_new_protected:Npn \__coffin_attach_mark:NnnNnnnn #1#2#3#4#5#6#7#8
  {
    \__coffin_align:NnnNnnnnN
      #1 {#2} {#3} #4 {#5} {#6} {#7} {#8} \l__coffin_aligned_coffin
    \box_set_ht:Nn \l__coffin_aligned_coffin { \box_ht:N #1 }
    \box_set_dp:Nn \l__coffin_aligned_coffin { \box_dp:N #1 }
    \box_set_wd:Nn \l__coffin_aligned_coffin { \box_wd:N #1 }
    \box_set_eq:NN #1 \l__coffin_aligned_coffin
  }
\cs_new_protected:Npn \__coffin_align:NnnNnnnnN #1#2#3#4#5#6#7#8#9
  {
    \__coffin_calculate_intersection:Nnn #4 {#5} {#6}
    \dim_set:Nn \l__coffin_x_prime_dim { \l__coffin_x_dim }
    \dim_set:Nn \l__coffin_y_prime_dim { \l__coffin_y_dim }
    \__coffin_calculate_intersection:Nnn #1 {#2} {#3}
    \dim_set:Nn \l__coffin_offset_x_dim
      { \l__coffin_x_dim - \l__coffin_x_prime_dim + #7 }
    \dim_set:Nn \l__coffin_offset_y_dim
      { \l__coffin_y_dim - \l__coffin_y_prime_dim + #8 }
    \hbox_set:Nn \l__coffin_aligned_internal_coffin
      {
        \box_use:N #1
        \__kernel_kern:n { -\box_wd:N #1 }
        \__kernel_kern:n { \l__coffin_offset_x_dim }
        \box_move_up:nn { \l__coffin_offset_y_dim } { \box_use:N #4 }
      }
    \coffin_set_eq:NN #9 \l__coffin_aligned_internal_coffin
  }
\cs_new_protected:Npn \__coffin_offset_poles:Nnn #1#2#3
  {
    \prop_map_inline:cn { coffin ~ \__coffin_to_value:N #1 ~ poles }
      { \__coffin_offset_pole:Nnnnnnn #1 {##1} ##2 {#2} {#3} }
  }
\cs_new_protected:Npn \__coffin_offset_pole:Nnnnnnn #1#2#3#4#5#6#7#8
  {
    \dim_set:Nn \l__coffin_x_dim { #3 + #7 }
    \dim_set:Nn \l__coffin_y_dim { #4 + #8 }
    \tl_if_in:nnTF {#2} { - }
      { \tl_set:Nn \l__coffin_internal_tl { {#2} } }
      { \tl_set:Nn \l__coffin_internal_tl { { #1 - #2 } } }
    \exp_last_unbraced:NNo \__coffin_set_pole:Nnn \l__coffin_aligned_coffin
      { \l__coffin_internal_tl }
      {
        { \dim_use:N \l__coffin_x_dim } { \dim_use:N \l__coffin_y_dim }
        {#5} {#6}
      }
  }
\cs_new_protected:Npn \__coffin_offset_corners:Nnn #1#2#3
  {
    \prop_map_inline:cn { coffin ~ \__coffin_to_value:N #1 ~ corners }
      { \__coffin_offset_corner:Nnnnn #1 {##1} ##2 {#2} {#3} }
  }
\cs_new_protected:Npn \__coffin_offset_corner:Nnnnn #1#2#3#4#5#6
  {
    \prop_put:cne
      {
        coffin ~ \__coffin_to_value:N \l__coffin_aligned_coffin
        \c_space_tl corners
      }
      { #1 - #2 }
      {
        { \dim_eval:n { #3 + #5 } }
        { \dim_eval:n { #4 + #6 } }
      }
  }
\cs_new_protected:Npn \__coffin_update_vertical_poles:NNN #1#2#3
  {
    \__coffin_get_pole:NnN #3 { #1 -T } \l__coffin_pole_a_tl
    \__coffin_get_pole:NnN #3 { #2 -T } \l__coffin_pole_b_tl
    \exp_last_two_unbraced:Noo \__coffin_update_T:nnnnnnnnN
      \l__coffin_pole_a_tl \l__coffin_pole_b_tl #3
    \__coffin_get_pole:NnN #3 { #1 -B } \l__coffin_pole_a_tl
    \__coffin_get_pole:NnN #3 { #2 -B } \l__coffin_pole_b_tl
    \exp_last_two_unbraced:Noo \__coffin_update_B:nnnnnnnnN
      \l__coffin_pole_a_tl \l__coffin_pole_b_tl #3
  }
\cs_new_protected:Npn \__coffin_update_T:nnnnnnnnN #1#2#3#4#5#6#7#8#9
  {
    \dim_compare:nNnTF {#2} < {#6}
      {
        \__coffin_set_pole:Nnn #9 { T }
          { { 0pt } {#6} { 1000pt } { 0pt } }
      }
      {
        \__coffin_set_pole:Nnn #9 { T }
          { { 0pt } {#2} { 1000pt } { 0pt } }
      }
  }
\cs_new_protected:Npn \__coffin_update_B:nnnnnnnnN #1#2#3#4#5#6#7#8#9
  {
    \dim_compare:nNnTF {#2} < {#6}
      {
        \__coffin_set_pole:Nnn #9 { B }
          { { 0pt } {#2}  { 1000pt } { 0pt } }
      }
      {
        \__coffin_set_pole:Nnn #9 { B }
          { { 0pt } {#6} { 1000pt } { 0pt } }
      }
  }
\coffin_new:N \c__coffin_empty_coffin
\tex_setbox:D \c__coffin_empty_coffin = \tex_hbox:D { }
\cs_new_protected:Npn \coffin_typeset:Nnnnn #1#2#3#4#5
  {
    \mode_leave_vertical:
    \__coffin_align:NnnNnnnnN \c__coffin_empty_coffin { H } { l }
```

```
      #1 {#2} {#3} {#4} {#5} \l__coffin_aligned_coffin
    \box_use_drop:N \l__coffin_aligned_coffin
  }
\cs_generate_variant:Nn \coffin_typeset:Nnnnn { c }
\coffin_new:N \l__coffin_display_coffin
\coffin_new:N \l__coffin_display_coord_coffin
\coffin_new:N \l__coffin_display_pole_coffin
\prop_new:N \l__coffin_display_handles_prop
\prop_put:Nnn \l__coffin_display_handles_prop { tl }
  { { b } { r } { -1 } { 1 } }
\prop_put:Nnn \l__coffin_display_handles_prop { thc }
  { { b } { hc } { 0 } { 1 } }
\prop_put:Nnn \l__coffin_display_handles_prop { tr }
  { { b } { l } { 1 } { 1 } }
\prop_put:Nnn \l__coffin_display_handles_prop { vcl }
  { { vc } { r } { -1 } { 0 } }
\prop_put:Nnn \l__coffin_display_handles_prop { vchc }
  { { vc } { hc } { 0 } { 0 } }
\prop_put:Nnn \l__coffin_display_handles_prop { vcr }
  { { vc } { l } { 1 } { 0 } }
\prop_put:Nnn \l__coffin_display_handles_prop { bl }
  { { t } { r } { -1 } { -1 } }
\prop_put:Nnn \l__coffin_display_handles_prop { bhc }
  { { t } { hc } { 0 } { -1 } }
\prop_put:Nnn \l__coffin_display_handles_prop { br }
  { { t } { l } { 1 } { -1 } }
\prop_put:Nnn \l__coffin_display_handles_prop { Tl }
  { { t } { r } { -1 } { -1 } }
\prop_put:Nnn \l__coffin_display_handles_prop { Thc }
  { { t } { hc } { 0 } { -1 } }
\prop_put:Nnn \l__coffin_display_handles_prop { Tr }
  { { t } { l } { 1 } { -1 } }
\prop_put:Nnn \l__coffin_display_handles_prop { Hl }
  { { vc } { r } { -1 } { 1 } }
\prop_put:Nnn \l__coffin_display_handles_prop { Hhc }
  { { vc } { hc } { 0 } { 1 } }
\prop_put:Nnn \l__coffin_display_handles_prop { Hr }
  { { vc } { l } { 1 } { 1 } }
\prop_put:Nnn \l__coffin_display_handles_prop { Bl }
  { { b } { r } { -1 } { -1 } }
\prop_put:Nnn \l__coffin_display_handles_prop { Bhc }
  { { b } { hc } { 0 } { -1 } }
\prop_put:Nnn \l__coffin_display_handles_prop { Br }
  { { b } { l } { 1 } { -1 } }
\dim_new:N  \l__coffin_display_offset_dim
\dim_set:Nn \l__coffin_display_offset_dim { 2pt }
```

```
\dim_new:N \l__coffin_display_x_dim
\dim_new:N \l__coffin_display_y_dim
\prop_new:N \l__coffin_display_poles_prop
\tl_new:N \l__coffin_display_font_tl
\bool_lazy_and:nnT
  { \cs_if_exist_p:N \fmtname }
  { \str_if_eq_p:Vn \fmtname { LaTeX2e } }
  {
    \tl_set:Nn \l__coffin_display_font_tl
      { \sffamily \tiny }
  }
\cs_new_protected:Npn \__coffin_rule:nn #1#2
  {
    \mode_leave_vertical:
    \hbox:n { \tex_vrule:D width #1 height #2 \scan_stop: }
  }
\cs_new_protected:Npn \coffin_mark_handle:Nnnn #1#2#3#4
  {
    \hcoffin_set:Nn \l__coffin_display_pole_coffin
      {
        \color_select:n {#4}
        \__coffin_rule:nn { 1pt } { 1pt }
      }
    \__coffin_attach_mark:NnnNnnnn #1 {#2} {#3}
      \l__coffin_display_pole_coffin { hc } { vc } { 0pt } { 0pt }
    \hcoffin_set:Nn \l__coffin_display_coord_coffin
      {
        \color_select:n {#4}
        \l__coffin_display_font_tl
        ( \tl_to_str:n { #2 , #3 } )
      }
    \prop_get:NnN \l__coffin_display_handles_prop
      { #2 #3 } \l__coffin_internal_tl
    \quark_if_no_value:NTF \l__coffin_internal_tl
      {
        \prop_get:NnN \l__coffin_display_handles_prop
          { #3 #2 } \l__coffin_internal_tl
        \quark_if_no_value:NTF \l__coffin_internal_tl
          {
            \__coffin_attach_mark:NnnNnnnn #1 {#2} {#3}
              \l__coffin_display_coord_coffin { l } { vc }
                { 1pt } { 0pt }
          }
          {
            \exp_last_unbraced:No \__coffin_mark_handle_aux:nnnnNnn
              \l__coffin_internal_tl #1 {#2} {#3}
          }
      }
      {
        \exp_last_unbraced:No \__coffin_mark_handle_aux:nnnnNnn
          \l__coffin_internal_tl #1 {#2} {#3}
      }
  }
\cs_new_protected:Npn \__coffin_mark_handle_aux:nnnnNnn #1#2#3#4#5#6#7
  {
    \__coffin_attach_mark:NnnNnnnn #5 {#6} {#7}
      \l__coffin_display_coord_coffin {#1} {#2}
      { #3 \l__coffin_display_offset_dim }
      { #4 \l__coffin_display_offset_dim }
  }
\cs_generate_variant:Nn \coffin_mark_handle:Nnnn { c }
\cs_new_protected:Npn \coffin_display_handles:Nn #1#2
  {
    \hcoffin_set:Nn \l__coffin_display_pole_coffin
      {
        \color_select:n {#2}
        \__coffin_rule:nn { 1pt } { 1pt }
      }
    \prop_set_eq:Nc \l__coffin_display_poles_prop
      { coffin ~ \__coffin_to_value:N #1 ~ poles }
    \__coffin_get_pole:NnN #1 { H } \l__coffin_pole_a_tl
    \__coffin_get_pole:NnN #1 { T } \l__coffin_pole_b_tl
    \tl_if_eq:NNT \l__coffin_pole_a_tl \l__coffin_pole_b_tl
      { \prop_remove:Nn \l__coffin_display_poles_prop { T } }
    \__coffin_get_pole:NnN #1 { B } \l__coffin_pole_b_tl
    \tl_if_eq:NNT \l__coffin_pole_a_tl \l__coffin_pole_b_tl
      { \prop_remove:Nn \l__coffin_display_poles_prop { B } }
    \coffin_set_eq:NN \l__coffin_display_coffin #1
    \prop_map_inline:Nn \l__coffin_display_poles_prop
      {
        \prop_remove:Nn \l__coffin_display_poles_prop {##1}
        \__coffin_display_handles_aux:nnnnnn {##1} ##2 {#2}
      }
    \box_use_drop:N \l__coffin_display_coffin
  }
\cs_new_protected:Npn \__coffin_display_handles_aux:nnnnnn #1#2#3#4#5#6
  {
    \prop_map_inline:Nn \l__coffin_display_poles_prop
      {
        \bool_set_false:N \l__coffin_error_bool
        \__coffin_calculate_intersection:nnnnnnnn {#2} {#3} {#4} {#5} ##2
        \bool_if:NF \l__coffin_error_bool
```

```
          {
            \dim_set:Nn \l__coffin_display_x_dim { \l__coffin_x_dim }
            \dim_set:Nn \l__coffin_display_y_dim { \l__coffin_y_dim }
            \__coffin_display_attach:Nnnnn
              \l__coffin_display_pole_coffin { hc } { vc }
              { 0pt } { 0pt }
            \hcoffin_set:Nn \l__coffin_display_coord_coffin
              {
                \color_select:n {#6}
                \l__coffin_display_font_tl
                ( \tl_to_str:n { #1 , ##1 } )
              }
            \prop_get:NnN \l__coffin_display_handles_prop
              { #1 ##1 } \l__coffin_internal_tl
            \quark_if_no_value:NTF \l__coffin_internal_tl
              {
                \prop_get:NnN \l__coffin_display_handles_prop
                  { ##1 #1 } \l__coffin_internal_tl
                \quark_if_no_value:NTF \l__coffin_internal_tl
                  {
                    \__coffin_display_attach:Nnnnn
                      \l__coffin_display_coord_coffin { l } { vc }
                      { 1pt } { 0pt }
                  }
                  {
                    \exp_last_unbraced:No
                      \__coffin_display_handles_aux:nnnn
                      \l__coffin_internal_tl
                  }
              }
              {
                \exp_last_unbraced:No \__coffin_display_handles_aux:nnnn
                  \l__coffin_internal_tl
              }
          }
      }
  }
\cs_new_protected:Npn \__coffin_display_handles_aux:nnnn #1#2#3#4
  {
    \__coffin_display_attach:Nnnnn
      \l__coffin_display_coord_coffin {#1} {#2}
      { #3 \l__coffin_display_offset_dim }
      { #4 \l__coffin_display_offset_dim }
  }
\cs_generate_variant:Nn \coffin_display_handles:Nn { c }
\cs_new_protected:Npn \__coffin_display_attach:Nnnnn #1#2#3#4#5
```

```
  {
    \__coffin_calculate_intersection:Nnn #1 {#2} {#3}
    \dim_set:Nn \l__coffin_x_prime_dim { \l__coffin_x_dim }
    \dim_set:Nn \l__coffin_y_prime_dim { \l__coffin_y_dim }
    \dim_set:Nn \l__coffin_offset_x_dim
      { \l__coffin_display_x_dim - \l__coffin_x_prime_dim + #4 }
    \dim_set:Nn \l__coffin_offset_y_dim
      { \l__coffin_display_y_dim - \l__coffin_y_prime_dim + #5 }
    \hbox_set:Nn \l__coffin_aligned_coffin
      {
        \box_use:N \l__coffin_display_coffin
        \__kernel_kern:n { -\box_wd:N \l__coffin_display_coffin }
        \__kernel_kern:n { \l__coffin_offset_x_dim }
        \box_move_up:nn { \l__coffin_offset_y_dim } { \box_use:N #1 }
      }
    \box_set_ht:Nn \l__coffin_aligned_coffin
      { \box_ht:N \l__coffin_display_coffin }
    \box_set_dp:Nn \l__coffin_aligned_coffin
      { \box_dp:N \l__coffin_display_coffin }
    \box_set_wd:Nn \l__coffin_aligned_coffin
      { \box_wd:N \l__coffin_display_coffin }
    \box_set_eq:NN \l__coffin_display_coffin \l__coffin_aligned_coffin
  }
\cs_new_protected:Npn \coffin_show_structure:N
  { \__coffin_show_structure:NN \msg_show:nneeee }
\cs_generate_variant:Nn \coffin_show_structure:N { c }
\cs_new_protected:Npn \coffin_log_structure:N
  { \__coffin_show_structure:NN \msg_log:nneeee }
\cs_generate_variant:Nn \coffin_log_structure:N { c }
\cs_new_protected:Npn \__coffin_show_structure:NN #1#2
  {
    \__coffin_if_exist:NT #2
      {
        #1 { coffin } { show }
          { \token_to_str:N #2 }
          {
            \iow_newline: >~ ht ~=~ \dim_eval:n { \coffin_ht:N #2 }
            \iow_newline: >~ dp ~=~ \dim_eval:n { \coffin_dp:N #2 }
            \iow_newline: >~ wd ~=~ \dim_eval:n { \coffin_wd:N #2 }
          }
          {
            \prop_map_function:cN
              { coffin ~ \__coffin_to_value:N #2 ~ poles }
              \msg_show_item_unbraced:nn
          }
          { }
      }
  }
\cs_new_protected:Npn \coffin_show:N #1
  { \coffin_show:Nnn #1 \c_max_int \c_max_int }
\cs_generate_variant:Nn \coffin_show:N { c }
\cs_new_protected:Npn \coffin_log:N #1
  { \coffin_log:Nnn #1 \c_max_int \c_max_int }
\cs_generate_variant:Nn \coffin_log:N { c }
\cs_new_protected:Npn \coffin_show:Nnn
  { \__coffin_show:NNNnn \msg_term:nneeee \box_show:Nnn }
\cs_generate_variant:Nn \coffin_show:Nnn { c }
\cs_new_protected:Npn \coffin_log:Nnn
  { \__coffin_show:NNNnn \msg_log:nneeee \box_show:Nnn }
\cs_generate_variant:Nn \coffin_log:Nnn { c }
\cs_new_protected:Npn \__coffin_show:NNNnn #1#2#3#4#5
  {
    \__coffin_if_exist:NT #3
      {
        \__coffin_show_structure:NN #1 #3
        #2 #3 {#4} {#5}
      }
  }
\msg_new:nnnn { coffin } { no-pole-intersection }
  { No~intersection~between~coffin~poles. }
  {
    LaTeX~was~asked~to~find~the~intersection~between~two~poles,~
    but~they~do~not~have~a~unique~meeting~point:~
    the~value~(0pt,~0pt)~will~be~used.
  }
\msg_new:nnnn { coffin } { unknown }
  { Unknown~coffin~'#1'. }
  { The~coffin~'#1'~was~never~defined. }
\msg_new:nnnn { coffin } { unknown-pole }
  { Pole~'#1'~unknown~for~coffin~'#2'. }
  {
    LaTeX~was~asked~to~find~a~typesetting~pole~for~a~coffin,~
    but~either~the~coffin~does~not~exist~or~the~pole~name~is~wrong.
  }
\msg_new:nnn { coffin } { show }
  {
    Size~of~coffin~#1 : #2 \\
    Poles~of~coffin~#1 : #3 .
  }
%% File: l3luatex.dtx
\cs_new_eq:NN \__lua_escape:n  \tex_luaescapestring:D
\cs_new_eq:NN \__lua_now:n     \tex_directlua:D
```

```
\cs_new_eq:NN \__lua_shipout:n \tex_latelua:D
\cs_undefine:N \lua_escape:e
\cs_undefine:N \lua_now:e
\cs_new:Npn \lua_now:e #1 { \__lua_now:n {#1} }
\cs_new:Npn \lua_now:n #1 { \lua_now:e { \exp_not:n {#1} } }
\cs_new_protected:Npn \lua_shipout_e:n #1 { \__lua_shipout:n {#1} }
\cs_new_protected:Npn \lua_shipout:n #1
  { \lua_shipout_e:n { \exp_not:n {#1} } }
\cs_new:Npn \lua_escape:e #1 { \__lua_escape:n {#1} }
\cs_new:Npn \lua_escape:n #1 { \lua_escape:e { \exp_not:n {#1} } }
\str_new:N \l__lua_err_msg_str
\cs_new_protected:Npn \lua_load_module:n #1
  {
    \bool_if:nF { \__lua_load_module_p:n { #1 } }
      {
        \msg_error:nnnV
          { luatex } { module-not-found } { #1 } \l__lua_err_msg_str
      }
  }
\sys_if_engine_luatex:F
  {
    \clist_map_inline:nn
      {
        \lua_escape:n , \lua_escape:e ,
        \lua_now:n , \lua_now:e
      }
      {
        \cs_gset:Npn #1 ##1
          {
            \msg_expandable_error:nnn
              { luatex } { luatex-required } { #1 }
          }
      }
    \clist_map_inline:nn
      { \lua_shipout_e:n , \lua_shipout:n, \lua_load_module:n }
      {
        \cs_gset_protected:Npn #1 ##1
          {
            \msg_error:nnn
              { luatex } { luatex-required } { #1 }
          }
      }
  }
\msg_new:nnnn { luatex } { luatex-required }
  { LuaTeX~engine~not~in~use!~Ignoring~#1. }
  {
    The~feature~you~are~using~is~only~available~
    with~the~LuaTeX~engine.~LaTeX3~ignored~'#1'.
  }
\msg_new:nnnn { luatex } { module-not-found }
  { Lua~module~`#1'~not~found. }
  {
    The~file~`#1.lua'~could~not~be~found.~Please~ensure~
    that~the~file~was~properly~installed~and~that~the~
    filename~database~is~current. \\ \\
    The~Lua~loader~provided~this~additional~information: \\
    #2
  }
\prop_gput:Nnn \g_msg_module_name_prop { luatex } { LaTeX }
\prop_gput:Nnn \g_msg_module_type_prop { luatex } { }
%% File: l3unicode.dtx
\sys_if_engine_opentype:TF
  {
    \cs_new:Npn \codepoint_str_generate:n #1
      {
        \int_compare:nNnTF {#1} = { `\␣ }
          { ~ }
          { \char_generate:nn {#1} { 12 } }
      }
    \cs_new:Npn \codepoint_generate:nn #1#2
      {
        \int_compare:nNnTF {#1} = { `\␣ }
          { ~ }
          {
            \__kernel_exp_not:w \exp_after:wN \exp_after:wN \exp_after:wN
              { \char_generate:nn {#1} {#2} }
          }
      }
  }
  {
    \cs_new:Npn \codepoint_str_generate:n #1
      {
        \int_compare:nNnTF {#1} = { `\␣ }
          { ~ }
          {
            \use:e
              {
                \exp_not:N \__codepoint_str_generate:nnnn
                  \__kernel_codepoint_to_bytes:n {#1}
              }
          }
      }
```

```
\cs_new:Npn \__codepoint_str_generate:nnnn #1#2#3#4
  {
    \char_generate:nn {#1} { 12 }
    \tl_if_blank:nF {#2}
      {
        \char_generate:nn {#2} { 12 }
        \tl_if_blank:nF {#3}
          {
            \char_generate:nn {#3} { 12 }
            \tl_if_blank:nF {#4}
              { \char_generate:nn {#4} { 12 } }
          }
      }
  }
\cs_new:Npn \codepoint_generate:nn #1#2
  {
    \int_compare:nNnTF {#1} = { `\␣ }
      { ~ }
      {
        \int_compare:nNnTF {#1} < { "80 }
          {
            \__kernel_exp_not:w \exp_after:wN \exp_after:wN \exp_after:wN
              { \char_generate:nn {#1} {#2} }
          }
          {
            \use:e
              {
                \exp_not:N \__codepoint_generate:nnnn
                  \__kernel_codepoint_to_bytes:n {#1}
              }
          }
      }
  }
\cs_new:Npn \__codepoint_generate:nnnn #1#2#3#4
  {
    \__kernel_exp_not:w \exp_after:wN
      {
        \tex_expanded:D
          {
            \__codepoint_generate:n {#1}
            \__codepoint_generate:n {#2}
            \tl_if_blank:nF {#3}
              {
                \__codepoint_generate:n {#3}
                \tl_if_blank:nF {#4}
                  { \__codepoint_generate:n {#4} }
```

```
         }
       }
     }
   }
  \cs_new:Npn \__codepoint_generate:n #1
   {
    \__kernel_exp_not:w \exp_after:wN \exp_after:wN \exp_after:wN
     { \char_generate:nn {#1} { 13 } }
   }
 }
\cs_new:Npn \__kernel_codepoint_to_bytes:n #1
 {
  \exp_args:Nf \__codepoint_to_bytes_auxi:n
   { \int_eval:n {#1} }
 }
\cs_new:Npn \__codepoint_to_bytes_auxi:n #1
 {
  \if_int_compare:w #1 > "80 \exp_stop_f:
   \if_int_compare:w #1 < "800 \exp_stop_f:
    \__codepoint_to_bytes_outputi:nw
     { \__codepoint_to_bytes_auxii:Nnn C {#1} { 64 } }
    \__codepoint_to_bytes_outputii:nw
     { \__codepoint_to_bytes_auxiii:n {#1} }
   \else:
    \if_int_compare:w #1 < "10000 \exp_stop_f:
     \__codepoint_to_bytes_outputi:nw
      { \__codepoint_to_bytes_auxii:Nnn E {#1} { 64 * 64 } }
     \__codepoint_to_bytes_outputii:nw
      {
       \__codepoint_to_bytes_auxiii:n
        { \int_div_truncate:nn {#1} { 64 } }
      }
     \__codepoint_to_bytes_outputiii:nw
      { \__codepoint_to_bytes_auxiii:n {#1} }
    \else:
     \__codepoint_to_bytes_outputi:nw
      {
       \__codepoint_to_bytes_auxii:Nnn F
        {#1} { 64 * 64 * 64 }
      }
     \__codepoint_to_bytes_outputii:nw
      {
       \__codepoint_to_bytes_auxiii:n
        { \int_div_truncate:nn {#1} { 64 * 64 } }
      }
     \__codepoint_to_bytes_outputiii:nw
```

```
            {
              \__codepoint_to_bytes_auxiii:n
                { \int_div_truncate:nn {#1} { 64 } }
            }
          \__codepoint_to_bytes_outputiv:nw
            { \__codepoint_to_bytes_auxiii:n {#1} }
        \fi:
      \fi:
    \else:
      \__codepoint_to_bytes_outputi:nw {#1}
    \fi:
    \__codepoint_to_bytes_end: { } { } { } { }
  }
\cs_new:Npn \__codepoint_to_bytes_auxii:Nnn #1#2#3
  {  "#10 + \int_div_truncate:nn {#2} {#3} }
\cs_new:Npn \__codepoint_to_bytes_auxiii:n #1
  { \int_mod:nn {#1} { 64 } + 128 }
\cs_new:Npn \__codepoint_to_bytes_outputi:nw
  #1 #2 \__codepoint_to_bytes_end: #3
  { \__codepoint_to_bytes_output:fnn { \int_eval:n {#1} } { } {#2} }
\cs_new:Npn \__codepoint_to_bytes_outputii:nw
  #1 #2 \__codepoint_to_bytes_end: #3#4
  { \__codepoint_to_bytes_output:fnn { \int_eval:n {#1} } { {#3} } {#2} }
\cs_new:Npn \__codepoint_to_bytes_outputiii:nw
  #1 #2 \__codepoint_to_bytes_end: #3#4#5
  {
    \__codepoint_to_bytes_output:fnn
      { \int_eval:n {#1} } { {#3} {#4} } {#2}
  }
\cs_new:Npn \__codepoint_to_bytes_outputiv:nw
  #1 #2 \__codepoint_to_bytes_end: #3#4#5#6
  {
    \__codepoint_to_bytes_output:fnn
      { \int_eval:n {#1} } { {#3} {#4} {#5} } {#2}
  }
\cs_new:Npn \__codepoint_to_bytes_output:nnn #1#2#3
  {
    #3
    \__codepoint_to_bytes_end: #2 {#1}
  }
\cs_generate_variant:Nn \__codepoint_to_bytes_output:nnn { f }
\cs_new:Npn \__codepoint_to_bytes_end: { }
\cs_new:Npn \codepoint_to_category:n #1
  {
    \cs:w
      c__codepoint_category_
      \tex_romannumeral:D
        \__kernel_codepoint_data:nn { category } {#1}
      _str
    \cs_end:
  }
\cs_new:Npn \codepoint_to_nfd:n #1
  { \exp_args:Ne \__codepoint_to_nfd:n { \int_eval:n {#1} } }
\cs_new:Npn \__codepoint_to_nfd:n #1
  { \__codepoint_to_nfd:nn {#1} { \char_value_catcode:n {#1} } }
\sys_if_engine_opentype:F
  {
    \cs_gset:Npn \__codepoint_to_nfd:n #1
      {
        \int_compare:nNnTF {#1} > { "80 }
          { \__codepoint_to_nfd:nn {#1} { 12 } }
          { \__codepoint_to_nfd:nn {#1} { \char_value_catcode:n {#1} } }
      }
  }
\cs_new:Npn \__codepoint_to_nfd:nn #1#2
  {
    \exp_args:Ne \__codepoint_to_nfd:nnn
      { \__codepoint_nfd:n {#1} } {#1} {#2}
  }
\cs_new:Npn \__codepoint_to_nfd:nnn #1#2#3 { \__codepoint_to_nfd:nnnn #1 {#2} {#3} }
\cs_new:Npn \__codepoint_to_nfd:nnnn #1#2#3#4
  {
    \int_compare:nNnTF {#1} = {#3}
      { \codepoint_generate:nn {#1} {#4} }
      {
        \__codepoint_to_nfd:nn {#1} {#4}
        \tl_if_blank:nF {#2}
          { \__codepoint_to_nfd:nn {#2} {#4} }
      }
  }
\int_const:Nn \c__codepoint_block_size_int { 64 }
\ior_new:N \g__codepoint_data_ior
\group_begin:
  \clist_map_inline:nn
    { category , grapheme , lowercase , uppercase , wordbreak }
    {
      \cs_set_nopar:cpn { l__codepoint_ #1 _block_clist } { }
      \cs_set_nopar:cpn { l__codepoint_ #1 _block_tl } { 1 }
      \cs_set_nopar:cpn { l__codepoint_ #1 _pos_tl } { 0 }
      \cs_set_nopar:cpn { l__codepoint_ #1 _next_tl } { 0 }
      \intarray_new:cn { g__codepoint_ #1 _index_intarray }
        { \int_div_truncate:nn { "110000 } \c__codepoint_block_size_int }
    }
  \cs_set_nopar:Npn \l__codepoint_matched_block_tl { 0 }
  \cs_set_nopar:Npn \l__codepoint_uppercase_default_tl { 0 }
  \cs_set_nopar:Npn \l__codepoint_lowercase_default_tl { 0 }
  \cs_set_protected:Npn \__codepoint_data_auxi:w #1#2#3
    {
      \quark_if_recursion_tail_stop:n {#3}
      \cs_set_nopar:cpn { l__codepoint_ #2 _ #3 _tl } {#1}
      \str_const:cn { c__codepoint_ #2 _ \tex_romannumeral:D #1 _str } {#3}
      \exp_args:Ne \__codepoint_data_auxi:w { \int_eval:n { #1 + 1 } } {#2}
    }
  \__codepoint_data_auxi:w { 1 } { category }
    { Lu } { Ll } { Lt } { Lm } { Lo }
    { Mn } { Me } { Mc }
    { Nd } { Nl } { No }
    { Zs } { Zl } { Zp }
    { Cc } { Cf } { Co } { Cs } { Cn }
    { Pd } { Ps } { Pe } { Pc } { Po } { Pi } { Pf }
    { Sm } { Sc } { Sk } { So }
    \q_recursion_tail
    \q_recursion_stop
  \cs_set_eq:NN \l__codepoint_category_default_tl \l__codepoint_category_Cn_tl
  \__codepoint_data_auxi:w { 1 } { grapheme }
    { Control }
    { CR } { LF } { ZWJ }
    { Extend }
    { L } { LV } { LVT } { T } { V }
    { Prepend }
    { Regional_Indicator }
    { SpacingMark }
    { Other }
    \q_recursion_tail
    \q_recursion_stop
  \cs_set_eq:NN \l__codepoint_grapheme_default_tl \l__codepoint_grapheme_Other_tl
  \__codepoint_data_auxi:w { 1 } { wordbreak }
    { Double_Quote } { Single_Quote }
    { CR } { LF } { Newline }
    { WSegSpace } { ZWJ }
    { Extend } { ExtendNumLet }
    { Regional_Indicator }
    { Format }
    { Katakana }
    { ALetter } { MidLetter } { Hebrew_Letter }
    { Numeric }{ MidNum } { MidNumLet }
    { Other }
    \q_recursion_tail
    \q_recursion_stop
  \cs_set_eq:NN \l__codepoint_wordbreak_default_tl \l__codepoint_wordbreak_Other_tl
  \cs_set_protected:Npn \__codepoint_data_auxi:w #1 ;~ #2 ~ #3 \q_stop
    { \__codepoint_data_auxii:w #1 .. \q_stop {#2} }
  \cs_set_protected:Npn \__codepoint_data_auxii:w #1 .. #2 \q_stop
    { \__codepoint_data_auxiii:w #1 ~ .. #2 ~ \q_stop }
  \cs_set_protected:Npn \__codepoint_data_auxiii:w #1 ~ #2 .. #3 ~ #4 \q_stop #5#6
    {
      \cs_set_nopar:cpe { l__codepoint_ #6 _ \tex_romannumeral:D "#1 _tl }
        {
          {#3}
          { \use:c { l__codepoint_ #6 _ #5 _tl } }
        }
    }
  \cs_set_protected:Npn \__codepoint_data_auxvi:w #1#2
    {
      \ior_open:Nn \g__codepoint_data_ior {#1}
      \ior_str_map_inline:Nn \g__codepoint_data_ior
        {
          \str_if_eq:eeF { \tl_head:w ##1 \c_hash_str \q_stop } { \c_hash_str }
            {
              \tl_if_blank:nF {##1}
                { \__codepoint_data_auxi:w ##1 \q_stop {#2} }
            }
        }
      \ior_close:N \g__codepoint_data_ior
    }
  \__codepoint_data_auxvi:w { GraphemeBreakProperty.txt } { grapheme }
  \__codepoint_data_auxvi:w { WordBreakProperty.txt } { wordbreak }
  \cs_set_protected:Npn \__codepoint_data_property:nnnn #1#2#3#4
    {
      \int_compare:nNnT {#3} > { \use:c { l__codepoint_ #4 _next_tl } }
        {
          \__codepoint_range:nnv {#3} {#4}
            { l__codepoint_ #4 _default_tl }
        }
      \__codepoint_add:nn {#4} {#2}
      \tl_set:ce { l__codepoint_ #4 _next_tl } { \int_eval:n { #3 + 1 } }
      \tl_if_blank:nF {#1}
        {
          \__codepoint_range:nnn {"#1} {#4} {#2}
          \__codepoint_add:nn {#4} {#2}
          \tl_set:ce { l__codepoint_ #4 _next_tl } { \int_eval:n { "#1 + 1 } }
        }
    }
  \cs_set_protected:Npn \__codepoint_data_auxi:w
```

```
    #1 ; #2 ; #3 ; #4 ; #5 ; #6 ; #7 ; #8 ; #9 ;
    {
      \tl_if_blank:nF {#6}
        {
          \tl_if_head_eq_charcode:nNF {#6}  < % >
            { \__codepoint_data_auxii:w #1 ; #6 ~ \q_stop }
        }
      \__codepoint_data_auxiii:w #1 ; #2 ; #3 ;
    }
  \cs_set_protected:Npn \__codepoint_data_auxii:w #1 ; #2 ~ #3 \q_stop
    {
      \tl_const:ce
        { c__codepoint_nfd_ \codepoint_str_generate:n {"#1} _tl }
        {
          {"#2}
          { \tl_if_blank:nF {#3} {"#3} }
        }
    }
  \cs_set_protected:Npn \__codepoint_data_auxiii:w
    #1 ; #2 ; #3 ; #4 ; #5 ; #6 ; #7 ; #8 ; #9 ~ \q_stop
    {
      \use:e
        {
          \__codepoint_data_auxiv:w
            #1 ; #2 ;
            \__codepoint_data_category:n {#3} ;
            \__codepoint_data_offset:nn {#1} {#7} ;
            \__codepoint_data_offset:nn {#1} {#8} ;
            #9;
        }
    }
  \cs_set:Npn \__codepoint_data_category:n #1
    { \use:c { l__codepoint_category_ #1 _tl } }
  \cs_set:Npn \__codepoint_data_offset:nn #1#2
    {
      \tl_if_blank:nTF {#2}
        { 0 }
        { \int_eval:n { "#2 - "#1 } }
    }
  \cs_set_protected:Npn \__codepoint_data_auxiv:w #1 ; #2 ; #3 ; #4 ; #5 ; #6 ;
    {
      \int_compare:nNnT {"#1} > \l__codepoint_category_next_tl
        {
          \__codepoint_data_auxv:nnnnw {#1} {#3} {#4} {#5}
            #2 Last> \q_stop
        }
      \__codepoint_add:nn { category } {#3}
      \__codepoint_add:nn { uppercase } {#4}
      \__codepoint_add:nn { lowercase } {#5}
      \int_compare:nNnF {#4} = { \__codepoint_data_offset:nn {#1} {#6} }
        {
          \tl_const:ce
            { c__codepoint_titlecase_ \codepoint_str_generate:n {"#1} _tl }
            { {"#6} { } { } }
        }
      \__codepoint_data_auxvi:nn { grapheme } {"#1}
      \__codepoint_data_auxvi:nn { wordbreak } {"#1}
      \int_compare:nNnT {"#1} = { "AC00 }
        {
          \int_step_inline:nnn { "AC01 } { "D7A2 }
            { \__codepoint_data_auxvi:nn { grapheme } {##1} }
        }
      \tl_set:Ne \l__codepoint_category_next_tl
        { \int_eval:n { "#1 + 1 } }
      \tl_set_eq:NN \l__codepoint_lowercase_next_tl \l__codepoint_category_next_tl
      \tl_set_eq:NN \l__codepoint_uppercase_next_tl \l__codepoint_category_next_tl
    }
  \cs_set_protected:Npn \__codepoint_add:nn #1#2
    {
      \clist_put_right:cn { l__codepoint_ #1 _block_clist } {#2}
      \int_compare:nNnT { \clist_count:c { l__codepoint_ #1 _block_clist } }
        = \c__codepoint_block_size_int
        { \__codepoint_save_blocks:nn {#1} { 1 } }
    }
  \cs_set_protected:Npn \__codepoint_data_auxv:nnnnw #1#2#3#4#5 Last> #6 \q_stop
    {
      \tl_if_blank:nTF {#6}
        {
          \__codepoint_range:nno {"#1} { category }
            \l__codepoint_category_default_tl
          \__codepoint_range:nnn {"#1} { uppercase } { 0 }
          \__codepoint_range:nnn {"#1} { lowercase } { 0 }
        }
        {
          \__codepoint_range:nnn {"#1} { category } {#2}
          \__codepoint_range:nnn {"#1} { uppercase } {#3}
          \__codepoint_range:nnn {"#1} { lowercase } {#4}
        }
    }
  \cs_set_protected:Npn \__codepoint_data_auxvi:nn #1#2
    {
      \cs_if_exist:cT
      { l__codepoint_ #1 _ \tex_romannumeral:D #2 _tl }
      {
        \exp_after:wN \exp_after:wN \exp_after:wN \__codepoint_data_property:nnnn
          \cs:w
            l__codepoint_ #1 _ \tex_romannumeral:D #2 _tl
              \cs_end: {#2} {#1}
      }
  }
\cs_set_protected:Npn \__codepoint_range:nnn #1#2
  {
    \exp_args:Nf \__codepoint_range_aux:nnn
      { \int_eval:n { #1 - \use:c { l__codepoint_ #2 _next_tl } } }
      {#2}
  }
\cs_set_protected:Npn \__codepoint_range:nno { \exp_args:Nnno \__codepoint_range:nnn }
\cs_set_protected:Npn \__codepoint_range:nnv { \exp_args:Nnnv \__codepoint_range:nnn }
\cs_set_protected:Npn \__codepoint_range_aux:nnn #1#2
  {
    \exp_args:Nf \__codepoint_range:nnnn
      {
        \int_min:nn
          {#1}
          {
            \c__codepoint_block_size_int
            - \clist_count:c { l__codepoint_ #2 _block_clist }
          }
      }
      {#1} {#2}
  }
\cs_set_protected:Npn \__codepoint_range:nnnn #1#2#3#4
  {
    \prg_replicate:nn {#1}
      { \clist_put_right:cn { l__codepoint_ #3 _block_clist } {#4} }
  \int_compare:nNnT { \clist_count:c { l__codepoint_ #3 _block_clist } }
    = \c__codepoint_block_size_int
    { \__codepoint_save_blocks:nn {#3} { 1 } }
  \int_compare:nNnF
    { \int_div_truncate:nn { #2 - #1 } \c__codepoint_block_size_int } = 0
    {
      \tl_set:ce { l__codepoint_ #3 _block_clist }
        {
          \exp_args:NNe \use:nn \use_none:n
            { \prg_replicate:nn { \c__codepoint_block_size_int } { , #4 } }
        }
      \__codepoint_save_blocks:nn {#3}
        { \int_div_truncate:nn { (#2 - #1) } \c__codepoint_block_size_int }
    }
  \prg_replicate:nn
    { \int_mod:nn { #2 - #1 } \c__codepoint_block_size_int }
    { \clist_put_right:ce { l__codepoint_ #3 _block_clist } {#4} }
  }
\cs_set_protected:Npn \__codepoint_save_blocks:nn #1#2
  {
    \tl_set_eq:Nc \l__codepoint_matched_block_tl { l__codepoint_ #1 _block_tl }
    \int_step_inline:nn { \tl_use:c { l__codepoint_ #1 _block_tl } - 1 }
      {
        \tl_if_eq:ccT { l__codepoint_ #1 _block_clist }
          { l__codepoint_ #1 _block_ ##1 _clist }
          { \tl_set:Nn \l__codepoint_matched_block_tl {##1} }
      }
    \int_compare:nNnT
      { \tl_use:c { l__codepoint_ #1 _block_tl } } = \l__codepoint_matched_block_tl
        {
          \clist_set_eq:cc
            {
              l__codepoint_ #1 _block_
              \tl_use:c { l__codepoint_ #1 _block_tl } _clist
            }
            { l__codepoint_ #1 _block_clist }
          \tl_set:ce { l__codepoint_ #1 _block_tl }
            { \int_eval:n { \tl_use:c { l__codepoint_ #1 _block_tl } + 1 } }
        }
      \int_step_inline:nnn
        { \tl_use:c { l__codepoint_ #1 _pos_tl } + 1 }
        { \tl_use:c { l__codepoint_ #1 _pos_tl } + #2 }
        {
          \exp_args:Nc \__kernel_intarray_gset:Nnn
            { g__codepoint_ #1 _index_intarray }
            {##1}
            \l__codepoint_matched_block_tl
        }
      \tl_set:ce { l__codepoint_ #1 _pos_tl }
        { \int_eval:n { \tl_use:c { l__codepoint_ #1 _pos_tl } + #2 } }
    \clist_clear:c { l__codepoint_ #1 _block_clist }
  }
\cs_set_protected:Npn \__codepoint_finalise_blocks:n #1
  {
    \clist_map_inline:nn {#1}
      {
        \exp_args:Nnnv \__codepoint_range:nnn { "110000 } {##1}
          { l__codepoint_ ##1 _default_tl }
        \__codepoint_finalise_blocks_aux:n {##1}
      }
  }
\cs_set_protected:Npn \__codepoint_finalise_blocks_aux:n #1
  {
    \cs_gset_eq:cc { c__codepoint_ #1 _index_intarray } { g__codepoint_ #1 _index_intarray }
    \cs_undefine:c { g__codepoint_ #1 _index_intarray }
    \intarray_new:cn { g__codepoint_ #1 _blocks_intarray }
      { ( \tl_use:c { l__codepoint_ #1 _block_tl } - 1 ) * \c__codepoint_block_size_int }
    \int_step_inline:nn { \tl_use:c { l__codepoint_ #1 _block_tl } - 1 }
      {
        \exp_args:Nv \__codepoint_finalise_blocks:nnn
          { l__codepoint_ #1 _block_ ##1 _clist }
          {##1} {#1}
      }
    \cs_gset_eq:cc { c__codepoint_ #1 _blocks_intarray }
      { g__codepoint_ #1 _blocks_intarray }
    \cs_undefine:c { g__codepoint_ #1 _blocks_intarray }
  }
\cs_set_protected:Npn \__codepoint_finalise_blocks:nnn #1#2#3
  {
    \exp_args:Nnf \__codepoint_finalise_blocks:nnnw { 1 }
      { \int_eval:n { ( #2 - 1 ) * \c__codepoint_block_size_int } }
      {#3}
      #1 , \q_recursion_tail , \q_recursion_stop
  }
\cs_set_protected:Npn \__codepoint_finalise_blocks:nnnw #1#2#3#4 ,
  {
    \quark_if_recursion_tail_stop:n {#4}
    \intarray_gset:cnn { g__codepoint_ #3 _blocks_intarray }
      { #1 + #2 }
      {#4}
    \exp_args:Nf \__codepoint_finalise_blocks:nnnw
      { \int_eval:n { #1 + 1 } } {#2} {#3}
  }
\ior_open:Nn \g__codepoint_data_ior { UnicodeData.txt }
\char_set_catcode_space:n { `\  }%
\ior_map_variable:NNn \g__codepoint_data_ior \l__codepoint_tmpa_tl
  {%
    \if_meaning:w \l__codepoint_tmpa_tl \c_space_tl
      \exp_after:wN \ior_map_break:
    \fi:
    \exp_after:wN \__codepoint_data_auxi:w \l__codepoint_tmpa_tl \q_stop
  }%
\char_set_catcode_ignore:n { `\  }%
\__codepoint_finalise_blocks:n
  { category , grapheme , lowercase , uppercase , wordbreak }
\group_end:
\cs_new:Npn \__kernel_codepoint_data:nn #1#2
{
  \exp_args:Nf \__codepoint_data:nnn
    {
      \int_eval:n
        {
          \c__codepoint_block_size_int *
            (
              \intarray_item:cn { c__codepoint_ #1 _index_intarray }
                {
                  \int_div_truncate:nn {#2}
                    \c__codepoint_block_size_int
                  + 1
                }
                - 1
            )
        }
    }
    {#2} {#1}
}
\cs_new:Npn \__codepoint_data:nnn #1#2#3
{
  \intarray_item:cn { c__codepoint_ #3 _blocks_intarray }
    { #1 + \int_mod:nn {#2} \c__codepoint_block_size_int + 1 }
}
\group_begin:
\ior_open:Nn \g__codepoint_data_ior { CaseFolding.txt }
\cs_set_protected:Npn \__codepoint_data_auxi:w #1 ;~ #2 ;~ #3 ; #4 \q_stop
  {
    \if:w \tl_head:n { #2 ? } C
      \reverse_if:N \if_int_compare:w
        \int_eval:n { \__kernel_codepoint_data:nn { lowercase } {"#1} + "#1 }
          = "#3 ~
        \tl_const:ce
          { c__codepoint_casefold_ \codepoint_str_generate:n {"#1} _tl }
          { {"#3} { } { } }
      \fi:
    \else:
      \if:w \tl_head:n { #2 ? } F
        \__codepoint_data_auxii:w #1 ~ #3 ~ \q_stop
      \fi:
    \fi:
  }
```

```
\cs_set_protected:Npn \__codepoint_data_auxii:w #1 ~ #2 ~ #3 ~ #4 \q_stop
  {
    \tl_const:ce { c__codepoint_casefold_ \codepoint_str_generate:n {"#1} _tl }
      {
        {"#2}
        {"#3}
        { \tl_if_blank:nF {#4} { " \int_to_Hex:n {"#4} } }
      }
  }
\ior_str_map_inline:Nn \g__codepoint_data_ior
  {
    \reverse_if:N \if:w \c_hash_str \tl_head:w #1 \c_hash_str \q_stop
      \__codepoint_data_auxi:w #1 \q_stop
    \fi:
  }
\ior_close:N \g__codepoint_data_ior
\ior_open:Nn \g__codepoint_data_ior { SpecialCasing.txt }
\cs_set_protected:Npn \__codepoint_data_auxi:w
  #1 ;~ #2 ;~ #3 ;~ #4 ; #5 \q_stop
  {
    \use:n { \__codepoint_data_auxii:w #1 ~ lower ~ #2 ~ } ~ \q_stop
    \use:n { \__codepoint_data_auxii:w #1 ~ upper ~ #4 ~ } ~ \q_stop
    \str_if_eq:nnF {#3} {#4}
      { \use:n { \__codepoint_data_auxii:w #1 ~ title ~ #3 ~ } ~ \q_stop }
  }
\cs_set_protected:Npn \__codepoint_data_auxii:w
  #1 ~ #2 ~ #3 ~ #4 ~ #5 \q_stop
  {
    \tl_if_empty:nF {#4}
      {
        \tl_const:ce { c__codepoint_ #2 case_ \codepoint_str_generate:n {"#1} _tl }
          {
            {"#3}
            {"#4}
            { \tl_if_blank:nF {#5} {"#5} }
          }
      }
  }
\ior_str_map_inline:Nn \g__codepoint_data_ior
  {
    \str_if_eq:eeTF { \tl_head:w #1 \c_hash_str \q_stop } { \c_hash_str }
      {
        \str_if_eq:eeT
          {#1}
          { \c_hash_str \c_space_tl Conditional~Mappings }
          { \ior_map_break: }
        }
        { \__codepoint_data_auxi:w #1 \q_stop }
    }
  \ior_close:N \g__codepoint_data_ior
\group_end:
\cs_new:Npn \__kernel_codepoint_case:nn #1#2
  {
    \exp_args:Ne \__codepoint_case:nnn
      { \codepoint_str_generate:n {#2} } {#1} {#2}
  }
\cs_new:Npn \__codepoint_case:nnn #1#2#3
  {
    \cs_if_exist:cTF { c__codepoint_ #2 _ #1 _tl }
      {
        \tl_use:c
          { c__codepoint_ #2 _ #1 _tl }
      }
      { \use:c { __codepoint_ #2 :n } {#3} }
  }
\cs_new:Npn \__codepoint_uppercase:n { \__codepoint_case:nn { uppercase } }
\cs_new:Npn \__codepoint_lowercase:n { \__codepoint_case:nn { lowercase } }
\cs_new:Npn \__codepoint_titlecase:n { \__codepoint_case:nn { uppercase } }
\cs_new:Npn \__codepoint_casefold:n  { \__codepoint_case:nn { lowercase } }
\cs_new:Npn \__codepoint_case:nn #1#2
  {
    { \int_eval:n { \__kernel_codepoint_data:nn {#1} {#2} + #2 } }
    { }
    { }
  }
\cs_new:Npn \__kernel_codepoint_to_grapheme_class:n #1
  {
    \cs:w
      c__codepoint_grapheme_
      \tex_romannumeral:D
        \__kernel_codepoint_data:nn { grapheme } {#1}
      _str
    \cs_end:
  }
\cs_new:Npn \__kernel_codepoint_to_wordbreak_class:n #1
  {
    \cs:w
      c__codepoint_wordbreak_
      \tex_romannumeral:D
        \__kernel_codepoint_data:nn { wordbreak } {#1}
      _str
    \cs_end:
  }
\cs_new:Npn \__codepoint_nfd:n #1
  { \exp_args:Ne \__codepoint_nfd:nn { \codepoint_str_generate:n {#1} } {#1} }
\cs_new:Npn \__codepoint_nfd:nn #1#2
  {
    \tl_if_exist:cTF { c__codepoint_nfd_ #1 _tl }
      { \tl_use:c { c__codepoint_nfd_ #1 _tl } }
      { {#2} { } }
  }
%% File: l3text.dtx
\cs_generate_variant:Nn \tl_if_head_eq_meaning_p:nN { o }
\scan_new:N \s__text_stop
\quark_new:N \q__text_nil
\__kernel_quark_new_conditional:Nn \__text_quark_if_nil:n { TF }
\quark_new:N \q__text_recursion_tail
\quark_new:N \q__text_recursion_stop
\cs_new:Npn \__text_use_i_delimit_by_q_recursion_stop:nw
  #1 #2 \q__text_recursion_stop {#1}
\__kernel_quark_new_test:N \__text_if_q_recursion_tail_stop_do:Nn
\__kernel_quark_new_test:N \__text_if_q_recursion_tail_stop_do:nn
\scan_new:N \s__text_recursion_tail
\scan_new:N \s__text_recursion_stop
\cs_new:Npn \__text_use_i_delimit_by_s_recursion_stop:nw
  #1 #2 \s__text_recursion_stop {#1}
\cs_new:Npn \__text_if_s_recursion_tail_stop_do:Nn #1
  {
    \bool_lazy_and:nnTF
      { \cs_if_eq_p:NN \s__text_recursion_tail #1 }
      { \str_if_eq_p:nn { \s__text_recursion_tail } {#1} }
      { \__text_use_i_delimit_by_s_recursion_stop:nw }
      { \use_none:n }
  }
\cs_new_eq:NN \__text_sep: \__kernel_int_sep:
\group_begin:
  \char_set_catcode_active:n { 0 }
  \cs_new:Npn \__text_token_to_explicit:N #1
    {
      \if_catcode:w \exp_not:N #1
        \if_catcode:w \scan_stop: \exp_not:N #1
          \scan_stop:
        \else:
          \exp_not:N ^^@
        \fi:
        \exp_after:wN \__text_token_to_explicit_cs:N
      \else:
        \exp_after:wN \__text_token_to_explicit_char:N
      \fi:
      #1
    }
\group_end:
\cs_new:Npn \__text_token_to_explicit_cs:N #1
  {
    \exp_after:wN \if_meaning:w \exp_not:N #1 #1
      \exp_after:wN \use:nn \exp_after:wN
        \__text_token_to_explicit_cs_aux:N
    \else:
      \exp_after:wN \exp_not:n
    \fi:
      {#1}
  }
\cs_new:Npn \__text_token_to_explicit_cs_aux:N #1
  {
    \bool_lazy_or:nnTF
      { \token_if_chardef_p:N #1 }
      { \token_if_mathchardef_p:N #1 }
      {
        \char_generate:nn {#1}
          {
            \if_int_compare:w \char_value_catcode:n {#1} = 10 \exp_stop_f:
              10
            \else:
              12
            \fi:
          }
      }
      {#1}
  }
\cs_new:Npn \__text_token_to_explicit_char:N #1
  {
    \if:w
      \if_catcode:w ^ \exp_args:No \str_tail:n { \token_to_str:N #1 } ^
        \token_to_str:N #1 #1
        \else:
        AB
      \fi:
      \exp_after:wN \exp_not:n
    \else:
      \exp_after:wN \__text_token_to_explicit:n
    \fi:
      {#1}
  }
\cs_new:Npn \__text_token_to_explicit:n #1
```

```
  {
    \exp_after:wN \__text_token_to_explicit_auxi:w
      \int_value:w
        \if_catcode:w \c_group_begin_token #1 1 \else:
        \if_catcode:w \c_group_end_token #1 2 \else:
        \if_catcode:w \c_math_toggle_token #1 3 \else:
        \if_catcode:w ## #1 6 \else:
        \if_catcode:w ^ #1 7 \else:
        \if_catcode:w \c_math_subscript_token #1 8 \else:
        \if_catcode:w \c_space_token #1 10 \else:
        \if_catcode:w A #1 11 \else:
        \if_catcode:w + #1 12 \else:
        4 \fi: \fi: \fi: \fi: \fi: \fi: \fi: \fi: \fi:
    \exp_after:wN \__text_sep:
    \token_to_meaning:N #1 \s__text_stop
  }
\cs_new:Npn \__text_token_to_explicit_auxi:w #1 \__text_sep: #2 \s__text_stop
  {
    \char_generate:nn
      {
        \if_int_compare:w #1 < 9 \exp_stop_f:
          \exp_after:wN \__text_token_to_explicit_auxii:w
        \else:
          \exp_after:wN \__text_token_to_explicit_auxiii:w
        \fi:
        #2
      }
      {#1}
  }
\exp_last_unbraced:NNNNo \cs_new:Npn \__text_token_to_explicit_auxii:w
  #1 { \tl_to_str:n { character ~ } } { ` }
\cs_new:Npn \__text_token_to_explicit_auxiii:w #1 ~ #2 ~ { ` }
\cs_new:Npn \__text_char_catcode:N #1
  {
    \if_catcode:w \exp_not:N #1 \c_math_toggle_token
      3
    \else:
      \if_catcode:w \exp_not:N #1 \c_alignment_token
        4
      \else:
        \if_catcode:w \exp_not:N #1 \c_math_superscript_token
          7
        \else:
          \if_catcode:w \exp_not:N #1 \c_math_subscript_token
            8
          \else:
            \if_catcode:w \exp_not:N #1 \c_space_token
              10
            \else:
              \if_catcode:w \exp_not:N #1 \c_catcode_letter_token
                11
              \else:
                \if_catcode:w \exp_not:N #1 \c_catcode_other_token
                  12
                \else:
                  13
                \fi:
              \fi:
            \fi:
          \fi:
        \fi:
      \fi:
    \fi:
  }
\prg_new_conditional:Npnn \__text_if_expandable:N #1 { T , F , TF }
  {
    \token_if_expandable:NTF #1
      {
        \bool_lazy_any:nTF
          {
            { \token_if_protected_macro_p:N      #1 }
            { \token_if_protected_long_macro_p:N #1 }
            { \token_if_eq_meaning_p:NN \q__text_recursion_tail #1 }
          }
          { \prg_return_false: }
          { \prg_return_true: }
      }
      { \prg_return_false: }
  }
\sys_if_engine_opentype:TF
  {
    \cs_new:Npn \__text_codepoint_process:nN #1#2 { #1 {#2} }
  }
  {
    \cs_new:Npe \__text_codepoint_process:nN #1#2
      {
        \exp_not:N \int_compare:nNnTF {`#2} > { "80 }
          {
            \sys_if_engine_pdftex:TF
              { \exp_not:N \__text_codepoint_process_aux:nN }
              {
                \exp_not:N \int_compare:nNnTF {`#2} > { "FF }
```

```
                  { \exp_not:N \use:n }
                  { \exp_not:N \__text_codepoint_process_aux:nN }
              }
          }
          { \exp_not:N \use:n }
            {#1} #2
      }
    \cs_new:Npn \__text_codepoint_process_aux:nN #1#2
      {
        \int_compare:nNnTF { `#2 } < { "E0 }
          { \__text_codepoint_process:nNN }
          {
            \int_compare:nNnTF { `#2 } < { "F0 }
              { \__text_codepoint_process:nNNN }
              { \__text_codepoint_process:nNNNN }
          }
            {#1} #2
        }
    \cs_new:Npn \__text_codepoint_process:nNN #1#2#3
      { #1 {#2#3} }
    \cs_new:Npn \__text_codepoint_process:nNNN #1#2#3#4
      { #1 {#2#3#4} }
    \cs_new:Npn \__text_codepoint_process:nNNNN #1#2#3#4#5
      { #1 {#2#3#4#5} }
  }
\sys_if_engine_opentype:TF
  {
    \prg_new_conditional:Npnn
      \__text_codepoint_compare:nNn #1#2#3 { TF , p }
      {
        \int_compare:nNnTF {`#1} #2 {#3}
          \prg_return_true: \prg_return_false:
      }
    \cs_new:Npn \__text_codepoint_from_chars:Nw #1 {`#1}
  }
  {
    \prg_new_conditional:Npnn
      \__text_codepoint_compare:nNn #1#2#3 { TF , p }
      {
        \int_compare:nNnTF { \__text_codepoint_from_chars:Nw #1 }
            #2 {#3}
          \prg_return_true: \prg_return_false:
      }
    \cs_new:Npe \__text_codepoint_from_chars:Nw #1
      {
        \exp_not:N \if_int_compare:w `#1 > "80 \exp_not:N \exp_stop_f:
```

```
   \sys_if_engine_pdftex:TF
    {
     \exp_not:N \exp_after:wN
      \exp_not:N \__text_codepoint_from_chars_aux:Nw
    }
    {
     \exp_not:N \if_int_compare:w `#1 > "FF \exp_not:N \exp_stop_f:
      \exp_not:N \exp_after:wN \exp_not:N \exp_after:wN
       \exp_not:N \exp_after:wN
       \exp_not:N \__text_codepoint_from_chars:N
     \exp_not:N \else:
      \exp_not:N \exp_after:wN \exp_not:N \exp_after:wN
       \exp_not:N \exp_after:wN
       \exp_not:N \__text_codepoint_from_chars_aux:Nw
     \exp_not:N \fi:
    }
  \exp_not:N \else:
   \exp_not:N \exp_after:wN \exp_not:N \__text_codepoint_from_chars:N
  \exp_not:N \fi:
   #1
 }
\cs_new:Npn \__text_codepoint_from_chars_aux:Nw #1
 {
  \if_int_compare:w `#1 < "E0 \exp_stop_f:
   \exp_after:wN \__text_codepoint_from_chars:NN
  \else:
   \if_int_compare:w `#1 < "F0 \exp_stop_f:
    \exp_after:wN \exp_after:wN \exp_after:wN
     \__text_codepoint_from_chars:NNN
   \else:
    \exp_after:wN \exp_after:wN \exp_after:wN
     \__text_codepoint_from_chars:NNNN
   \fi:
  \fi:
   #1
 }
\cs_new:Npn \__text_codepoint_from_chars:N #1 {`#1}
\cs_new:Npn \__text_codepoint_from_chars:NN #1#2
 { (`#1 - "C0) * "40 + `#2 - "80 }
\cs_new:Npn \__text_codepoint_from_chars:NNN #1#2#3
 { (`#1 - "E0) * "1000 + (`#2 - "80) * "40 + `#3 - "80 }
\cs_new:Npn \__text_codepoint_from_chars:NNNN #1#2#3#4
 {
   (`#1 - "F0) * "40000
  + (`#2 - "80) * "1000
  + (`#3 - "80) * "40
```

```
        + `#4 - "80
      }
  }
\tl_new:N \l_text_accents_tl
\tl_new:N \l_text_letterlike_tl
\tl_new:N \l_text_case_exclude_arg_tl
\tl_set:Ne \l_text_case_exclude_arg_tl
  {
    \exp_not:n { \begin \cite \end \label \ref }
    \exp_not:c { cite ~ }
    \exp_not:n { \babelshorthand }
  }
\tl_new:N \l_text_math_arg_tl
\tl_set:Nn \l_text_math_arg_tl { \ensuremath }
\tl_new:N \l_text_math_delims_tl
\tl_set:Nn \l_text_math_delims_tl { $ $ \( \) }
\tl_new:N \l_text_expand_exclude_tl
\tl_set:Nn \l_text_expand_exclude_tl
  { \begin \cite \end \label \ref }
\bool_lazy_and:nnT
  { \str_if_eq_p:Vn \fmtname { LaTeX2e } }
  { \tl_if_exist_p:N \@expl@finalise@setup@@ }
  {
    \tl_gput_right:Nn \@expl@finalise@setup@@
      {
        \tl_gput_right:Nn \@kernel@after@begindocument
          {
            \group_begin:
              \cs_set_protected:Npn \__text_tmp:w #1
                {
                  \tl_clear:N \l_text_expand_exclude_tl
                  \tl_map_inline:nn {#1}
                    {
                      \bool_lazy_any:nF
                        {
                          { \token_if_protected_macro_p:N ##1 }
                          { \token_if_protected_long_macro_p:N ##1 }
                          {
                            \str_if_eq_p:ee
                              { \cs_replacement_spec:N ##1 }
                              { \exp_not:n { \protect ##1 } \c_space_tl }
                          }
                        }
                        { \tl_put_right:Nn \l_text_expand_exclude_tl {##1} }
                    }
                }
              \exp_args:NV \__text_tmp:w \l_text_expand_exclude_tl
            \exp_args:NNNV \group_end:
            \tl_set:Nn \l_text_expand_exclude_tl \l_text_expand_exclude_tl
          }
      }
  }
\tl_new:N \l__text_math_mode_tl
\tex_global:D \tex_chardef:D \c__text_chardef_space_token = `\␣ %
\tex_global:D \tex_mathchardef:D \c__text_mathchardef_space_token = `\␣ %
\tex_global:D \tex_chardef:D \c__text_chardef_group_begin_token = `\{ % `\}
\tex_global:D \tex_mathchardef:D \c__text_mathchardef_group_begin_token = `\{ % `\} `\{
\tex_global:D \tex_chardef:D \c__text_chardef_group_end_token = `\} % `\{
\tex_global:D \tex_mathchardef:D \c__text_mathchardef_group_end_token = `\} %
\cs_new:Npn \text_expand:n #1
  {
    \__kernel_exp_not:w \exp_after:wN
      {
        \exp:w
        \__text_expand:n {#1}
      }
  }
\cs_new:Npn \__text_expand:n #1
  {
    \group_align_safe_begin:
    \__text_expand_loop:w #1
      \s__text_recursion_tail \s__text_recursion_stop
    \__text_expand_result:n { }
  }
\cs_new:Npn \__text_expand_store:n #1
  { \__text_expand_store:nw {#1} }
\cs_generate_variant:Nn \__text_expand_store:n { o }
\cs_new:Npn \__text_expand_store:nw #1#2 \__text_expand_result:n #3
  { #2 \__text_expand_result:n { #3 #1 } }
\cs_new:Npn \__text_expand_end:w #1 \__text_expand_result:n #2
  {
    \group_align_safe_end:
    \exp_end:
    #2
  }
\cs_new:Npn \__text_expand_loop:w #1 \s__text_recursion_stop
  {
    \tl_if_head_is_N_type:nTF {#1}
      { \__text_expand_N_type:N }
      {
        \tl_if_head_is_group:nTF {#1}
          { \__text_expand_group:n }
          { \__text_expand_space:w }
      }
    #1 \s__text_recursion_stop
  }
\cs_new:Npn \__text_expand_group:n #1
  {
    \__text_expand_store:o
      {
        \exp_after:wN
          {
            \exp:w
            \__text_expand:n {#1}
          }
      }
    \__text_expand_loop:w
  }
\exp_last_unbraced:NNo \cs_new:Npn \__text_expand_space:w \c_space_tl
  {
    \__text_expand_store:n { ~ }
    \__text_expand_loop:w
  }
\cs_new:Npn \__text_expand_N_type:N #1
  {
    \__text_if_s_recursion_tail_stop_do:Nn #1
      { \__text_expand_end:w }
    \exp_after:wN \__text_expand_math_search:NNN
      \exp_after:wN #1 \l_text_math_delims_tl
      \q__text_recursion_tail \q__text_recursion_tail
      \q__text_recursion_stop
  }
\cs_new:Npn \__text_expand_math_search:NNN #1#2#3
  {
    \__text_if_q_recursion_tail_stop_do:Nn #2
      { \__text_expand_explicit:N #1 }
    \token_if_eq_meaning:NNTF #1 #2
      {
        \__text_use_i_delimit_by_q_recursion_stop:nw
          {
            \__text_expand_store:n {#1}
            \__text_expand_math_loop:Nw #3
          }
      }
      { \__text_expand_math_search:NNN #1 }
  }
\cs_new:Npn \__text_expand_math_loop:Nw #1#2 \s__text_recursion_stop
  {
    \tl_if_head_is_N_type:nTF {#2}
      { \__text_expand_math_N_type:NN }
      {
        \tl_if_head_is_group:nTF {#2}
          { \__text_expand_math_group:Nn }
          { \__text_expand_math_space:Nw }
      }
    #1#2 \s__text_recursion_stop
  }
\cs_new:Npn \__text_expand_math_N_type:NN #1#2
  {
    \__text_if_s_recursion_tail_stop_do:Nn #2
      { \__text_expand_end:w }
    \token_if_eq_meaning:NNF #2 \exp_not:N
      { \__text_expand_store:n {#2} }
    \token_if_eq_meaning:NNTF #2 #1
      { \__text_expand_loop:w }
      { \__text_expand_math_loop:Nw #1 }
  }
\cs_new:Npn \__text_expand_math_group:Nn #1#2
  {
    \__text_expand_store:n { {#2} }
    \__text_expand_math_loop:Nw #1
  }
\exp_after:wN \cs_new:Npn \exp_after:wN \__text_expand_math_space:Nw
  \exp_after:wN # \exp_after:wN 1 \c_space_tl
  {
    \__text_expand_store:n { ~ }
    \__text_expand_math_loop:Nw #1
  }
\cs_new:Npn \__text_expand_explicit:N #1
  {
    \token_if_cs:NTF #1
      { \__text_expand_exclude:N #1 }
      {
        \bool_lazy_and:nnTF
          { \token_if_active_p:N #1 }
          {
            ! \bool_lazy_any_p:n
              {
                { \token_if_protected_macro_p:N #1 }
                { \token_if_protected_long_macro_p:N #1 }
                { \tl_if_head_eq_meaning_p:oN {#1} \UTFviii@two@octets }
                { \tl_if_head_eq_meaning_p:oN {#1} \UTFviii@three@octets }
                { \tl_if_head_eq_meaning_p:oN {#1} \UTFviii@four@octets }
                { \tl_if_head_eq_meaning_p:oN {#1} \active@prefix }
              }
          }
          { \exp_after:wN \__text_expand_loop:w #1 }
          {
            \__text_expand_store:n {#1}
            \__text_expand_loop:w
          }
      }
  }
\cs_new:Npn \__text_expand_exclude:N #1
  {
    \cs_if_eq:NNTF #1 \text_case_switch:nnnn
      { \__text_expand_exclude_switch:Nnnnn #1 }
      {
        \exp_args:Ne \__text_expand_exclude:nN
          {
            \exp_not:V \l_text_math_arg_tl
            \exp_not:V \l_text_expand_exclude_tl
            \exp_not:V \l_text_case_exclude_arg_tl
          }
        #1
      }
  }
\cs_new:Npn \__text_expand_exclude_switch:Nnnnn #1#2#3#4#5
  {
    \__text_expand_store:n { #1 {#2} {#3} {#4} {#5} }
    \__text_expand_loop:w
  }
\cs_new:Npn \__text_expand_exclude:nN #1#2
  {
    \__text_expand_exclude:NN #2 #1
      \q__text_recursion_tail \q__text_recursion_stop
  }
\cs_new:Npn \__text_expand_exclude:NN #1#2
  {
    \__text_if_q_recursion_tail_stop_do:Nn #2
      { \__text_expand_accent:N #1 }
    \str_if_eq:nnTF {#1} {#2}
      {
        \__text_use_i_delimit_by_q_recursion_stop:nw
          { \__text_expand_exclude:Nw #1 }
      }
      { \__text_expand_exclude:NN #1 }
  }
\cs_new:Npn \__text_expand_exclude:Nw #1#2#
  { \__text_expand_exclude:Nnn #1 {#2} }
```

```
\cs_new:Npn \__text_expand_exclude:Nnn #1#2#3
  {
    \__text_expand_store:n { #1#2 {#3} }
    \__text_expand_loop:w
  }
\cs_new:Npn \__text_expand_accent:N #1
  {
    \exp_after:wN \__text_expand_accent:NN \exp_after:wN
      #1 \l_text_accents_tl
      \q__text_recursion_tail \q__text_recursion_stop
  }
\cs_new:Npn \__text_expand_accent:NN #1#2
  {
    \__text_if_q_recursion_tail_stop_do:Nn #2
      { \__text_expand_letterlike:N #1 }
    \cs_if_eq:NNTF #2 #1
      {
        \__text_use_i_delimit_by_q_recursion_stop:nw
          {
            \__text_expand_store:n {#1}
            \__text_expand_loop:w
          }
      }
      { \__text_expand_accent:NN #1 }
  }
\cs_new:Npn \__text_expand_letterlike:N #1
  {
    \exp_after:wN \__text_expand_letterlike:NN \exp_after:wN
      #1 \l_text_letterlike_tl
      \q__text_recursion_tail \q__text_recursion_stop
  }
\cs_new:Npn \__text_expand_letterlike:NN #1#2
  {
    \__text_if_q_recursion_tail_stop_do:Nn #2
      { \__text_expand_cs:N #1 }
    \cs_if_eq:NNTF #2 #1
      {
        \__text_use_i_delimit_by_q_recursion_stop:nw
          {
            \__text_expand_store:n {#1}
            \__text_expand_loop:w
          }
      }
      { \__text_expand_letterlike:NN #1 }
  }
\cs_new:Npe \__text_expand_cs:N #1
```

```
  {
    \exp_not:N \str_if_eq:nnTF {#1} { \exp_not:N \protect }
      { \exp_not:N \__text_expand_protect:w }
      {
        \bool_lazy_and:nnTF
          { \cs_if_exist_p:N \fmtname }
          { \str_if_eq_p:Vn \fmtname { LaTeX2e } }
          { \exp_not:N \__text_expand_testopt:N #1 }
          { \exp_not:N \__text_expand_replace:N #1 }
      }
  }
\cs_new:Npn \__text_expand_protect:w #1 \s__text_recursion_stop
  {
    \tl_if_head_is_N_type:nTF {#1}
      { \__text_expand_protect:N }
      {
        \__text_expand_store:n { \protect }
        \__text_expand_loop:w
      }
        #1 \s__text_recursion_stop
  }
\cs_new:Npn \__text_expand_protect:N #1
  {
    \__text_if_s_recursion_tail_stop_do:Nn #1
      {
        \__text_expand_store:n { \protect }
        \__text_expand_end:w
      }
    \exp_args:Ne \__text_expand_protect:nN
      { \cs_to_str:N #1 } #1
  }
\cs_new:Npn \__text_expand_protect:nN #1#2
  { \__text_expand_protect:Nw #2 #1 \q__text_nil #1 ~ \q__text_nil \q__text_nil \s__text_stop }
\cs_new:Npn \__text_expand_protect:Nw #1 #2 ~ \q__text_nil #3 \q__text_nil #4 \s__text_stop
  {
    \__text_quark_if_nil:nTF {#4}
      {
        \cs_if_exist:cTF {#2}
          { \exp_args:Ne \__text_expand_store:n { \exp_not:c {#2} } }
          { \__text_expand_store:n { \protect #1 } }
      }
      { \__text_expand_store:n { \protect #1 } }
    \__text_expand_loop:w
  }
```

```
\cs_new:Npn \__text_expand_testopt:N #1
  {
    \token_if_eq_meaning:NNTF #1 \@protected@testopt
      { \__text_expand_testopt:NNn }
      { \__text_expand_encoding:N #1 }
  }
\cs_new:Npn \__text_expand_testopt:NNn #1#2#3
  {
    \__text_expand_store:n {#1}
    \__text_expand_loop:w
  }
\cs_new:Npn \__text_expand_encoding:N #1
  {
    \bool_lazy_or:nnTF
      { \cs_if_eq_p:NN  #1 \@current@cmd }
      { \cs_if_eq_p:NN  #1 \@changed@cmd }
      { \exp_after:wN \__text_expand_loop:w \__text_expand_encoding_escape:NN }
      { \__text_expand_replace:N #1 }
  }
\cs_new:Npn \__text_expand_encoding_escape:NN #1#2 { \exp_not:n {#1} }
\cs_new:Npn \__text_expand_replace:N #1
  {
    \bool_lazy_and:nnTF
      { \cs_if_exist_p:c { l__text_expand_ \token_to_str:N #1 _tl } }
      {
        \bool_lazy_or_p:nn
          { \token_if_cs_p:N #1 }
          { \token_if_active_p:N #1 }
      }
      {
        \exp_args:Nv \__text_expand_replace:n
          { l__text_expand_ \token_to_str:N #1 _tl }
      }
      { \__text_expand_cs_expand:N #1 }
  }
\cs_new:Npn \__text_expand_replace:n #1 { \__text_expand_loop:w #1 }
\cs_new:Npn \__text_expand_cs_expand:N #1
  {
    \__text_if_expandable:NTF #1
      {
        \token_if_eq_meaning:NNTF #1 \exp_not:n
          { \__text_expand_unexpanded:w }
          { \exp_after:wN \__text_expand_loop:w #1 }
      }
      {
        \__text_expand_store:n {#1}
        \__text_expand_loop:w
      }
  }
\cs_new:Npn \__text_expand_unexpanded:w
  {
    \exp_after:wN \__text_expand_unexpanded_test:w
    \exp:w \exp_end_continue_f:w
  }
\cs_new:Npn \__text_expand_unexpanded_test:w #1 \s__text_recursion_stop
  {
    \tl_if_head_is_group:nTF {#1}
      { \__text_expand_unexpanded:n }
      {
        \__text_expand_unexpanded:w
        \tl_if_head_is_N_type:nT {#1} { \__text_expand_unexpanded:N }
      }
    #1 \s__text_recursion_stop
  }
\cs_new:Npn \__text_expand_unexpanded:N #1
  {
    \exp_after:wN \if_meaning:w \exp_not:N #1 #1
    \else:
      \exp_after:wN #1
    \fi:
  }
\cs_new:Npn \__text_expand_unexpanded:n #1
  {
    \__text_expand_store:n {#1}
    \__text_expand_loop:w
  }
\cs_new_protected:Npn \text_declare_expand_equivalent:Nn #1#2
  {
    \tl_clear_new:c { l__text_expand_ \token_to_str:N #1 _tl }
    \tl_set:cn { l__text_expand_ \token_to_str:N #1 _tl } {#2}
  }
\cs_generate_variant:Nn \text_declare_expand_equivalent:Nn { c }
\tl_map_inline:nn
  { \` \' \^ \~ \= \u \. \" \r \H \v \d \c \k \b \t }
  { \text_declare_expand_equivalent:Nn #1 { \exp_not:n {#1} } }
\tl_map_inline:nn
  {
    \AA \aa
    \AE \ae
    \DH \dh
    \DJ \dj
    \IJ \ij
    \L  \l
    \NG \ng
    \O  \o
    \OE \oe
    \SS \ss
    \TH \th
  }
  { \text_declare_expand_equivalent:Nn #1 { \exp_not:n {#1} } }
%% File: l3text-case.dtx
\bool_new:N \l_text_titlecase_check_letter_bool
\bool_set_true:N \l_text_titlecase_check_letter_bool
\cs_new:Npn \text_lowercase:n #1
  { \__text_change_case:nnn { lower } { } {#1} }
\cs_new:Npn \text_uppercase:n #1
  { \__text_change_case:nnn { upper } { } {#1} }
\cs_new:Npn \text_titlecase_all:n #1
  { \__text_change_case:nnn { title } { } {#1} }
\cs_new:Npn \text_titlecase_first:n #1
  { \__text_change_case:nnnn { title } { break } { } {#1} }
\cs_new:Npn \text_lowercase:nn #1#2
  { \__text_change_case:nnn { lower } {#1} {#2} }
\cs_new:Npn \text_uppercase:nn #1#2
  { \__text_change_case:nnn { upper } {#1} {#2} }
\cs_new:Npn \text_titlecase_all:nn #1#2
  { \__text_change_case:nnn { title } {#1} {#2} }
\cs_new:Npn \text_titlecase_first:nn #1#2
  { \__text_change_case:nnnn { title } { break } {#1} {#2} }
\cs_new:Npn \__text_change_case:nnn #1#2#3
  { \__text_change_case:nnnn {#1} {#1} {#2} {#3} }
\cs_new:Npn \__text_change_case:nnnn #1#2#3#4
  {
    \__kernel_exp_not:w \exp_after:wN
      {
        \exp:w
        \exp_args:Ne \__text_change_case_auxi:nnnn
          { \text_expand:n {#4} }
          {#1} {#2} {#3}
      }
  }
\cs_new:Npn \__text_change_case_auxi:nnnn #1#2#3#4
  {
    \exp_args:No \__text_change_case_BCP:nnnn
      { \tl_to_str:n {#4} } {#1} {#2} {#3}
  }
\cs_new:Npe \__text_change_case_BCP:nnnn #1#2#3#4
  {
    \exp_not:N \__text_change_case_BCP:nnnw
      {#2} {#3} {#4} #1 \tl_to_str:n { -x- -x- } \exp_not:N \q__text_stop
  }
\use:e
  {
    \cs_new:Npn \exp_not:N \__text_change_case_BCP:nnnw
      #1#2#3#4 \tl_to_str:n { -x- } #5 \tl_to_str:n { -x- } #6
      \exp_not:N \q__text_stop
  }
  { \__text_change_case_BCP:nnnnnw {#1} {#2} {#3} {#5} {#4} #4 - \q__text_stop }
\cs_new:Npn \__text_change_case_BCP:nnnnnw #1#2#3#4#5#6 - #7 \q__text_stop
  {
    \bool_lazy_or:nnTF
      { \cs_if_exist_p:c { __text_change_case_ #2 _ #6 -x- #4 :nnnnn } }
      { \tl_if_exist_p:c { l__text_ #2 case_special_ #6 -x- #4 _tl } }
      { \__text_change_case_auxii:nnnn {#1} {#2} {#3} { #6 -x- #4 } }
      {
        \cs_if_exist:cTF { __text_change_case_ #2 _ #6 :nnnnn }
          { \__text_change_case_auxii:nnnn {#1} {#2} {#3} {#6} }
          { \__text_change_case_auxii:nnnn {#1} {#2} {#3} {#5} }
      }
  }
\cs_new:Npn \__text_change_case_auxii:nnnn #1#2#3#4
  {
    \group_align_safe_begin:
    \cs_if_exist_use:c { __text_change_case_boundary_ #2 _ #4 :Nnnnw }
    \__text_change_case_loop:nnnw {#2} {#3} {#4} #1
      \q__text_recursion_tail \q__text_recursion_stop
    \__text_change_case_result:n { }
  }
\cs_new:Npn \__text_change_case_store:n #1
  { \__text_change_case_store:nw {#1} }
\cs_generate_variant:Nn \__text_change_case_store:n { o , e , V , v }
\cs_new:Npn \__text_change_case_store:nw #1#2 \__text_change_case_result:n #3
  { #2 \__text_change_case_result:n { #3 #1 } }
\cs_new:Npn \__text_change_case_end:w #1 \__text_change_case_result:n #2
  {
    \group_align_safe_end:
    \exp_end:
    #2
  }
\cs_new:Npn \__text_change_case_loop:nnnw #1#2#3#4 \q__text_recursion_stop
  {
    \tl_if_head_is_N_type:nTF {#4}
      { \__text_change_case_N_type:nnnN }
      {
        \tl_if_head_is_group:nTF {#4}
          { \use:c { __text_change_case_group_ #1 :nnnn } }
          { \__text_change_case_space:nnnw }
      }
    {#1} {#2} {#3} #4 \q__text_recursion_stop
  }
\cs_new:Npn \__text_change_case_break:w
  { \__text_change_case_break_aux:w \prg_do_nothing: }
\cs_new:Npn \__text_change_case_break_aux:w
  #1 \q__text_recursion_tail \q__text_recursion_stop
  {
    \__text_change_case_store:o {#1}
    \__text_change_case_end:w
  }
\cs_new:Npn \__text_change_case_group_lower:nnnn #1#2#3#4
  {
    \__text_change_case_store:o
      {
        \exp_after:wN
          {
            \exp:w
            \__text_change_case_auxii:nnnn {#4} {#1} {#2} {#3}
          }
      }
    \__text_change_case_loop:nnnw {#1} {#2} {#3}
  }
\cs_new_eq:NN \__text_change_case_group_upper:nnnn
  \__text_change_case_group_lower:nnnn
\cs_new:Npn \__text_change_case_group_title:nnnn #1#2#3#4
  {
    \__text_change_case_store:o
      {
        \exp_after:wN
          {
            \exp:w
            \__text_change_case_auxii:nnnn {#4} {#1} {#2} {#3}
          }
      }
    \__text_change_case_skip:nnw {#2} {#3}
  }
\use:e
  {
    \cs_new:Npn \exp_not:N \__text_change_case_space:nnnw #1#2#3 \c_space_tl
  }
  {
    \__text_change_case_store:n { ~ }
    \cs_if_exist_use:cF { __text_change_case_space_ #2 :nnn }
      {
        \cs_if_exist_use:c { __text_change_case_boundary_ #1 _ #3 :Nnnnw }
        \__text_change_case_loop:nnnw
      }
        {#2} {#2} {#3}
  }
\cs_new:Npn \__text_change_case_space_break:nnn #1#2#3
  { \__text_change_case_break:w }
\cs_new:Npn \__text_change_case_N_type:nnnN #1#2#3#4
  {
    \__text_if_q_recursion_tail_stop_do:Nn #4
      { \__text_change_case_end:w }
    \__text_change_case_N_type_aux:nnnN {#1} {#2} {#3} #4
  }
\cs_new:Npn \__text_change_case_N_type_aux:nnnN #1#2#3#4
  {
    \exp_args:NV \__text_change_case_N_type:nnnnN
      \l_text_math_delims_tl {#1} {#2} {#3} #4
  }
\cs_new:Npn \__text_change_case_N_type:nnnnN #1#2#3#4#5
  {
    \__text_change_case_math_search:nnnNNN {#2} {#3} {#4} #5 #1
      \q__text_recursion_tail \q__text_recursion_tail
      \q__text_recursion_stop
  }
\cs_new:Npn \__text_change_case_math_search:nnnNNN #1#2#3#4#5#6
  {
    \__text_if_q_recursion_tail_stop_do:Nn #5
      { \__text_change_case_cs_check:nnnN {#1} {#2} {#3} #4 }
    \token_if_eq_meaning:NNTF #4 #5
      {
        \__text_use_i_delimit_by_q_recursion_stop:nw
          {
            \__text_change_case_store:n {#4}
            \__text_change_case_math_loop:nnnNw {#1} {#2} {#3} #6
          }
      }
      { \__text_change_case_math_search:nnnNNN {#1} {#2} {#3} #4 }
  }
\cs_new:Npn \__text_change_case_math_loop:nnnNw #1#2#3#4#5 \q__text_recursion_stop
  {
    \tl_if_head_is_N_type:nTF {#5}
      { \__text_change_case_math_N_type:nnnNN }
      {
        \tl_if_head_is_group:nTF {#5}
```

```
          { \__text_change_case_math_group:nnnNn }
          { \__text_change_case_math_space:nnnNw }
      }
    {#1} {#2} {#3} #4 #5 \q__text_recursion_stop
  }
\cs_new:Npn \__text_change_case_math_N_type:nnnNN #1#2#3#4#5
  {
    \__text_if_q_recursion_tail_stop_do:Nn #5
      { \__text_change_case_end:w }
    \__text_change_case_store:n {#5}
    \token_if_eq_meaning:NNTF #5 #4
      { \__text_change_case_loop:nnnw {#1} {#2} {#3} }
      { \__text_change_case_math_loop:nnnNw {#1} {#2} {#3} #4 }
  }
\cs_new:Npn \__text_change_case_math_group:nnnNn #1#2#3#4#5
  {
    \__text_change_case_store:n { {#5} }
    \__text_change_case_math_loop:nnnNw {#1} {#2} {#3} #4
  }
\use:e
  {
    \cs_new:Npn \exp_not:N \__text_change_case_math_space:nnnNw #1#2#3#4
      \c_space_tl
  }
  {
    \__text_change_case_store:n { ~ }
    \__text_change_case_math_loop:nnnNw {#1} {#2} {#3} #4
  }
\cs_new:Npn \__text_change_case_cs_check:nnnN #1#2#3#4
  {
    \token_if_cs:NTF #4
      { \__text_change_case_exclude:nnnN {#1} {#2} {#3} }
      {
        \__text_codepoint_process:nN
          { \use:c { __text_change_case_custom_ #1 :nnnn } {#1} {#2} {#3} }
      }
        #4
  }
\cs_new:Npn \__text_change_case_exclude:nnnN #1#2#3#4
  {
    \exp_args:Ne \__text_change_case_exclude:nnnnN
      {
        \exp_not:V \l_text_math_arg_tl
        \exp_not:V \l_text_case_exclude_arg_tl
      }
      {#1} {#2} {#3} #4
  }
\cs_new:Npn \__text_change_case_exclude:nnnnN #1#2#3#4#5
  {
    \__text_change_case_exclude:nnnNN {#2} {#3} {#4} #5 #1
      \q__text_recursion_tail \q__text_recursion_stop
  }
\cs_new:Npn \__text_change_case_exclude:nnnNN #1#2#3#4#5
  {
    \__text_if_q_recursion_tail_stop_do:Nn #5
      { \__text_change_case_replace:nnnN {#1} {#2} {#3} #4 }
    \str_if_eq:nnTF {#4} {#5}
      {
        \__text_use_i_delimit_by_q_recursion_stop:nw
          { \__text_change_case_exclude:nnnNw {#1} {#2} {#3} #4 }
      }
      { \__text_change_case_exclude:nnnNN {#1} {#2} {#3} #4 }
  }
\cs_new:Npn \__text_change_case_exclude:nnnNw #1#2#3#4#5#
  { \__text_change_case_exclude:nnnNnn {#1} {#2} {#3} {#4} {#5} }
\cs_new:Npn \__text_change_case_exclude:nnnNnn #1#2#3#4#5#6
  {
    \tl_if_blank:nTF {#5}
      { \__text_change_case_store:n { #4 {#6} } }
      {
        \__text_change_case_store:o
          {
            \exp_after:wN #4
              \exp:w \__text_change_case_auxii:nnnn {#5} {#1} {#2} {#3}
              {#6}
          }
      }
    \__text_change_case_loop:nnnw {#1} {#2} {#3}
  }
\cs_new:Npn \__text_change_case_replace:nnnN #1#2#3#4
  {
    \cs_if_exist:cTF { l__text_case_ \token_to_str:N #4 _tl }
      {
        \__text_change_case_replace:vnnn
          { l__text_case_ \token_to_str:N #4 _tl } {#1} {#2} {#3}
      }
      { \__text_change_case_switch:nnnN {#1} {#2} {#3} #4 }
  }
\cs_new:Npn \__text_change_case_replace:nnnn #1#2#3#4
  { \__text_change_case_loop:nnnw {#2} {#3} {#4} #1 }
\cs_generate_variant:Nn \__text_change_case_replace:nnnn { v }
\cs_new:Npn \__text_change_case_switch:nnnN #1#2#3#4
```

```
  {
    \cs_if_eq:NNTF #4 \text_case_switch:nnnn
      { \use:c { __text_change_case_switch_ #1 :nnnNnnnn  } }
      { \use:c { __text_change_case_letterlike_ #1 :nnnN } }
        {#1} {#2} {#3} #4
  }
\cs_new:Npn \__text_change_case_switch_lower:nnnNnnnn #1#2#3#4#5#6#7#8
  {
    \__text_change_case_store:n {#7}
    \__text_change_case_loop:nnnw {#1} {#2} {#3}
  }
\cs_new:Npn \__text_change_case_switch_upper:nnnNnnnn #1#2#3#4#5#6#7#8
  {
    \__text_change_case_store:n {#6}
    \__text_change_case_loop:nnnw {#1} {#2} {#3}
  }
\cs_new:Npn \__text_change_case_switch_title:nnnNnnnn #1#2#3#4#5#6#7#8
  {
    \__text_change_case_store:n {#8}
    \__text_change_case_skip:nnw {#2} {#3}
  }
\cs_new:Npn \__text_change_case_skip:nnw #1#2#3 \q__text_recursion_stop
  {
    \tl_if_head_is_N_type:nTF {#3}
      { \__text_change_case_skip_N_type:nnN }
      {
        \tl_if_head_is_group:nTF {#3}
          { \__text_change_case_skip_group:nnn }
          { \__text_change_case_skip_space:nnw }
      }
        {#1} {#2} #3 \q__text_recursion_stop
  }
\cs_new:Npn \__text_change_case_skip_N_type:nnN #1#2#3
  {
    \__text_if_q_recursion_tail_stop_do:Nn #3
      { \__text_change_case_end:w }
    \__text_change_case_store:n {#3}
    \__text_change_case_skip:nnw {#1} {#2}
  }
\cs_new:Npn \__text_change_case_skip_group:nnn #1#2#3
  {
    \__text_change_case_store:n { {#3} }
    \__text_change_case_skip:nnw {#1} {#2}
  }
\cs_new:Npn \__text_change_case_skip_space:nnw #1#2
  { \__text_change_case_space:nnnw {#1} {#1} {#2} }
```

```
\cs_new:Npn \__text_change_case_letterlike_lower:nnnN #1#2#3#4
  { \__text_change_case_letterlike:nnnnnN {#1} {#1} {#1} {#2} {#3} #4 }
\cs_new_eq:NN \__text_change_case_letterlike_upper:nnnN
  \__text_change_case_letterlike_lower:nnnN
\cs_new:Npn \__text_change_case_letterlike_title:nnnN #1#2#3#4
  { \__text_change_case_letterlike:nnnnnN { upper } { end } {#1} {#2} {#3} #4 }
\cs_new:Npn \__text_change_case_letterlike:nnnnnN #1#2#3#4#5#6
  {
    \cs_if_exist:cTF { c__text_ #1 case_ \token_to_str:N #6 _tl }
      {
        \__text_change_case_store:v
          { c__text_ #1 case_ \token_to_str:N #6 _tl }
        \use:c { __text_change_case_next_ #2 :nnn } {#2} {#4} {#5}
      }
      {
        \__text_change_case_store:n {#6}
        \cs_if_exist:cTF
          {
            c__text_
            \str_if_eq:nnTF {#1} { lower } { upper } { lower }
            case_ \token_to_str:N #6 _tl
          }
          { \use:c { __text_change_case_next_ #2 :nnn } {#2} {#4} {#5} }
          { \__text_change_case_loop:nnnw {#3} {#4} {#5} }
      }
  }
\cs_new:Npn \__text_change_case_custom_lower:nnnn #1#2#3#4
  {
    \__text_change_case_custom:nnnnnn {#1} {#1} {#2} {#3} {#4}
      { \use:c { __text_change_case_codepoint_ #1 :nnnn } {#1} {#2} {#3} {#4} }
  }
\cs_new_eq:NN \__text_change_case_custom_upper:nnnn
  \__text_change_case_custom_lower:nnnn
\cs_new:Npn \__text_change_case_custom_title:nnnn #1#2#3#4
  {
    \__text_change_case_custom:nnnnnn { title } {#1} {#2} {#3} {#4}
      {
        \__text_change_case_custom:nnnnnn { upper } {#1} {#2} {#3} {#4}
          { \use:c { __text_change_case_codepoint_ #1 :nnnn } {#1} {#2} {#3} {#4} }
      }
  }
\cs_new:Npn \__text_change_case_custom:nnnnnn #1#2#3#4#5#6
  {
    \tl_if_exist:cTF { l__text_ #1 case _ \tl_to_str:n {#5} _ #4 _tl }
      {
        \__text_change_case_replace:vnnn
```

```
          { l__text_ #1 case _ \tl_to_str:n {#5} _ #4 _tl } {#2} {#3} {#4}
      }
      {
        \tl_if_exist:cTF { l__text_ #1 case _ \tl_to_str:n {#5} _tl }
          {
            \__text_change_case_replace:vnnn
              { l__text_ #1 case _ \tl_to_str:n {#5} _tl } {#2} {#3} {#4}
          }
          {#6}
      }
  }
\cs_new:Npn \__text_change_case_codepoint_lower:nnnn #1#2#3#4
  {
    \cs_if_exist_use:cF { __text_change_case_lower_ #3 :nnnnn }
      { \__text_change_case_lower_sigma:nnnnn }
        {#1} {#1} {#2} {#3} {#4}
  }
\cs_new:Npn \__text_change_case_codepoint_upper:nnnn #1#2#3#4
  {
    \cs_if_exist_use:cF { __text_change_case_upper_ #3 :nnnnn }
      { \__text_change_case_codepoint:nnnnn }
        {#1} {#1} {#2} {#3} {#4}
  }
\cs_new:Npn \__text_change_case_lower_sigma:nnnnn #1#2#3#4#5
  {
    \__text_codepoint_compare:nNnTF {#5} = { "03A3 }
      { \__text_change_case_lower_sigma:nnnnw {#2} }
      { \__text_change_case_codepoint:nnnnn {#1} {#2} }
        {#3} {#4} {#5}
  }
\cs_new:Npn \__text_change_case_lower_sigma:nnnnw #1#2#3#4#5 \q__text_recursion_stop
  {
    \tl_if_head_is_N_type:nTF {#5}
      { \__text_change_case_lower_sigma:nnnnN {#4} }
      {
        \__text_change_case_store:e
          { \codepoint_generate:nn { "03C2 } { \__text_char_catcode:N #4 } }
        \__text_change_case_loop:nnnw
      }
        {#1} {#2} {#3} #5 \q__text_recursion_stop
  }
\cs_new:Npn \__text_change_case_lower_sigma:nnnnN #1#2#3#4#5
  {
    \__text_change_case_store:e
      {
        \bool_lazy_or:nnTF
```

```
          { \token_if_letter_p:N #5 }
          {
            \bool_lazy_and_p:nn
              { \token_if_active_p:N #5 }
              { \int_compare_p:nNn {`#5} > { "80 } }
          }
          { \codepoint_generate:nn { "03C3 } { \__text_char_catcode:N #1 } }
          { \codepoint_generate:nn { "03C2 } { \__text_char_catcode:N #1 } }
      }
    \__text_change_case_loop:nnnw {#2} {#3} {#4} #5
  }
\cs_new:Npn \__text_change_case_codepoint_title:nnnn #1#2#3#4
  {
    \bool_if:NTF \l_text_titlecase_check_letter_bool
      {
        \exp_args:Ne \__text_change_case_codepoint_title_auxi:nnnn
          {
            \codepoint_to_category:n
              { \__text_codepoint_from_chars:Nw #4 }
          }
      }
      { \__text_change_case_codepoint_title:nnn }
        {#2} {#3} {#4}
  }
\cs_new:Npn \__text_change_case_codepoint_title_auxi:nnnn #1#2#3#4
  {
    \tl_if_head_eq_charcode:nNTF {#1} { L }
      { \__text_change_case_codepoint_title:nnn }
      { \__text_change_case_codepoint_title_auxii:nnnn { title } }
        {#2} {#3} {#4}
  }
\cs_new:Npn \__text_change_case_codepoint_title:nnn #1#2#3
  { \__text_change_case_codepoint_title_auxii:nnnn { end } {#1} {#2} {#3} }
\cs_new:Npn \__text_change_case_codepoint_title_auxii:nnnn #1#2#3#4
  {
    \cs_if_exist_use:cF { __text_change_case_title_ #3 :nnnnn }
      {
        \cs_if_exist_use:cF { __text_change_case_upper_ #3 :nnnnn }
          { \__text_change_case_codepoint:nnnnn }
      }
        { title } {#1} {#2} {#3} {#4}
  }
\cs_new:Npn \__text_change_case_codepoint:nnnnn #1#2#3#4#5
  {
    \bool_lazy_and:nnTF
      { \tl_if_single_p:n {#5} }
      { \token_if_active_p:N #5 }
      { \__text_change_case_store:n {#5} }
      {
        \__text_change_case_store:e
          { \__text_change_case_codepoint:nn {#1} {#5} }
      }
    \use:c { __text_change_case_next_ #2 :nnn } {#2} {#3} {#4}
  }
\cs_new:Npn \__text_change_case_codepoint:nn #1#2
  {
    \__text_change_case_codepoint:fnn
      { \int_eval:n { \__text_codepoint_from_chars:Nw #2 } } {#1} {#2}
  }
\cs_new:Npn \__text_change_case_codepoint:nnn #1#2#3
  {
    \exp_args:Ne \__text_change_case_codepoint_aux:nn
      { \__kernel_codepoint_case:nn { #2 case } {#1} } {#3}
  }
\cs_generate_variant:Nn \__text_change_case_codepoint:nnn { f }
\sys_if_engine_ptex:T
  {
    \cs_new_eq:NN \__text_change_case_codepoint_aux:nnn
      \__text_change_case_codepoint:nnn
    \cs_gset:Npn \__text_change_case_codepoint:nnn #1#2#3
      {
        \int_compare:nNnTF {#1} = { -1 }
          { \exp_not:n {#3} }
          { \__text_change_case_codepoint_aux:nnn {#1} {#2} {#3} }
      }
  }
\cs_new:Npn \__text_change_case_codepoint_aux:nn #1#2
  {
    \use:e { \__text_change_case_codepoint_aux:nnnn #1 {#2} }
  }
\cs_new:Npn \__text_change_case_codepoint_aux:nnnn #1#2#3#4
  {
    \__text_codepoint_compare:nNnTF {#4} = {#1}
      { \exp_not:n {#4} }
      {
        \codepoint_generate:nn {#1}
          { \__text_change_case_catcode:nn {#4} {#1} }
        \tl_if_blank:nF {#2}
          {
            \codepoint_generate:nn {#2}
              { \char_value_catcode:n {#2} }
            \tl_if_blank:nF {#3}
```

```
              {
                \codepoint_generate:nn {#3}
                  { \char_value_catcode:n {#3} }
              }
          }
      }
  }
\sys_if_engine_opentype:TF
  {
    \cs_new:Npn \__text_change_case_catcode:nn #1#2
      { \__text_char_catcode:N #1 }
  }
  {
    \cs_new:Npn \__text_change_case_catcode:nn #1#2
      {
        \__text_codepoint_compare:nNnTF {#1} < { "80 }
          { \__text_char_catcode:N #1 }
          {
            \int_compare:nNnTF {#2} < { "80 }
              { \char_value_catcode:n {#2} }
              { 13 }
          }
      }
  }
\cs_new:Npn \__text_change_case_next_lower:nnn #1#2#3
  { \__text_change_case_loop:nnnw {#1} {#2} {#3} }
\cs_new_eq:NN \__text_change_case_next_upper:nnn
  \__text_change_case_next_lower:nnn
\cs_new_eq:NN \__text_change_case_next_title:nnn
  \__text_change_case_next_lower:nnn
\cs_new:Npn \__text_change_case_next_end:nnn #1#2#3
  { \__text_change_case_skip:nnw {#2} {#3} }
\cs_new_protected:Npn \text_declare_case_equivalent:Nn #1#2
  {
    \tl_clear_new:c { l__text_case_ \token_to_str:N #1 _tl }
    \tl_set:cn { l__text_case_ \token_to_str:N #1 _tl } {#2}
  }
\cs_new_protected:Npn \text_declare_lowercase_mapping:nn #1#2
  { \__text_declare_case_mapping:nnn { lower } {#1} {#2} }
\cs_new_protected:Npn \text_declare_titlecase_mapping:nn #1#2
  { \__text_declare_case_mapping:nnn { title } {#1} {#2} }
\cs_new_protected:Npn \text_declare_uppercase_mapping:nn #1#2
  { \__text_declare_case_mapping:nnn { upper } {#1} {#2} }
\cs_new_protected:Npn \__text_declare_case_mapping:nnn #1#2#3
  {
    \exp_args:Ne \__text_declare_case_mapping_aux:nnn
```

```
      { \codepoint_str_generate:n {#2} } {#1} {#3}
  }
\cs_new_protected:Npn \__text_declare_case_mapping_aux:nnn #1#2#3
  {
    \tl_clear_new:c { l__text_ #2 case _ #1 _tl }
    \tl_set:cn { l__text_ #2 case _ #1 _ tl } {#3}
  }
\cs_new_protected:Npn \text_declare_lowercase_mapping:nnn #1#2#3
  { \__text_declare_case_mapping:nnnn { lower } {#1} {#2} {#3} }
\cs_new_protected:Npn \text_declare_titlecase_mapping:nnn #1#2#3
  { \__text_declare_case_mapping:nnnn { title } {#1} {#2} {#3} }
\cs_new_protected:Npn \text_declare_uppercase_mapping:nnn #1#2#3
  { \__text_declare_case_mapping:nnnn { upper } {#1} {#2} {#3} }
\cs_new_protected:Npn \__text_declare_case_mapping:nnnn #1#2#3#4
  {
    \exp_args:Ne \__text_declare_case_mapping_aux:nnnn
      { \codepoint_str_generate:n {#3} } {#1} {#2} {#4}
  }
\cs_new_protected:Npn \__text_declare_case_mapping_aux:nnnn #1#2#3#4
  {
    \tl_clear_new:c { l__text_ #2 case _ #1 _ #3 _tl }
    \tl_set:cn { l__text_ #2 case _ #1 _ #3 _ tl } {#4}
    \tl_clear_new:c { l__text_ #2 case_special_ #3 _tl }
  }
\cs_new:Npn \text_case_switch:nnnn #1#2#3#4
  {
    \__text_case_switch_marker:
    #1
  }
\cs_new:Npn \__text_case_switch_marker: { }
\cs_new:Npn \__text_change_case_generate:n #1
  { \codepoint_generate:nn {#1} { \char_value_catcode:n {#1} } }
\cs_new:cpn { __text_change_case_upper_de-x-eszett:nnnnn } #1#2#3#4#5
  {
    \__text_codepoint_compare:nNnTF {#5} = { "00DF }
      {
        \__text_change_case_store:e
          {
            \codepoint_generate:nn { "1E9E }
              { \__text_change_case_catcode:nn {#5} { "1E9E } }
          }
        \use:c { __text_change_case_next_ #2 :nnn }
          {#2} {#3} {#4}
      }
      { \__text_change_case_codepoint:nnnnn {#1} {#2} {#3} {#4} {#5} }
  }
```

```
\cs_new_eq:cc { __text_change_case_upper_de-alt:nnnnn }
  { __text_change_case_upper_de-x-eszett:nnnnn }
\cs_new:Npn \__text_change_case_upper_el:nnnnn #1#2#3#4#5
  {
    \bool_lazy_and:nnTF
      { \__text_change_case_if_greek_p:n {#5} }
      {
        ! \bool_lazy_or_p:nn
          { \__text_codepoint_compare_p:nNn {#5} = { "0374 } }
          { \__text_codepoint_compare_p:nNn {#5} = { "037E } }
      }
      {
        \__text_change_case_if_greek_spacing_diacritic:nTF {#5}
          {
            \__text_change_case_store:n {#5}
            \__text_change_case_loop:nnnw
          }
          {
            \exp_args:Ne \__text_change_case_upper_el:nnnn
              {
                \codepoint_to_nfd:n
                  { \__text_codepoint_from_chars:Nw #5 }
              }
          }
            {#2} {#3} {#4}
      }
      {
        \__text_codepoint_compare:nNnTF {#5} = { "0345 }
          {
            \__text_change_case_store:e
              {
                \codepoint_generate:nn { "0399 }
                  { \char_value_catcode:n { "0399 } }
              }
            \__text_change_case_loop:nnnw {#2} {#3} {#4}
          }
          { \__text_change_case_codepoint:nnnnn {#1} {#2} {#3} {#4} {#5} }
      }
  }
\cs_new_eq:cN { __text_change_case_upper_el-x-iota:nnnnn }
  \__text_change_case_upper_el:nnnnn
\cs_new:Npn \__text_change_case_upper_el:nnnn #1#2#3#4
  {
    \__text_codepoint_process:nN
      { \__text_change_case_upper_el:nnnnw {#2} {#3} {#4} } #1
  }
```

```
\cs_new:Npn \__text_change_case_upper_el:nnnnw #1#2#3#4#5 \q__text_recursion_stop
  {
    \tl_if_head_is_N_type:nTF {#5}
      { \__text_change_case_upper_el:nnnnN {#4} }
      {
        \__text_change_case_store:e
          { \__text_change_case_codepoint:nn { upper } {#4} }
        \__text_change_case_loop:nnnw
      }
        {#1} {#2} {#3} #5 \q__text_recursion_stop
  }
\cs_new:Npn \__text_change_case_upper_el:nnnnN #1#2#3#4#5
  {
    \token_if_cs:NTF #5
      {
        \__text_change_case_store:e
          { \__text_change_case_codepoint:nn { upper } {#1} }
        \__text_change_case_loop:nnnw {#2} {#3} {#4} #5
      }
      {
        \__text_change_case_if_takes_ypogegrammeni:nTF {#1}
          {
            \__text_change_case_upper_el_ypogegrammeni:nnnnnnw
              {#1} {#2} {#3} {#4} { } { } #5
          }
          { \__text_change_case_upper_el_aux:nnnnN {#1} {#2} {#3} {#4} #5 }
      }
  }
\cs_new:Npn \__text_change_case_upper_el_ypogegrammeni:nnnnnnw
  #1#2#3#4#5#6#7 \q__text_recursion_stop
  {
    \tl_if_head_is_N_type:nTF {#7}
      {
        \__text_change_case_upper_el_ypogegrammeni:nnnnnnN
          {#1} {#2} {#3} {#4} {#5} {#6}
      }
      { \__text_change_case_upper_el_aux:nnnnN {#1} {#2} {#3} {#4} #5#6 }
        #7 \q__text_recursion_stop
  }
\cs_new:Npn \__text_change_case_upper_el_ypogegrammeni:nnnnnnN #1#2#3#4#5#6#7
  {
    \token_if_cs:NTF #7
      { \__text_change_case_upper_el_aux:nnnnN {#1} {#2} {#3} {#4} #5#6 }
      {
        \__text_codepoint_process:nN
          {
            \__text_change_case_upper_el_ypogegrammeni:nnnnnnn
              {#1} {#2} {#3} {#4} {#5} {#6}
          }
      }
        #7
  }
\cs_new:Npn \__text_change_case_upper_el_ypogegrammeni:nnnnnnn #1#2#3#4#5#6#7
  {
    \__text_codepoint_compare:nNnTF {#7} = { "0345 }
      {
        \__text_change_case_upper_el_ypogegrammeni:nnnnnnw
          {#1} {#2} {#3} {#4} {#5} {#7}
      }
      {
        \bool_lazy_or:nnTF
          { \__text_change_case_if_greek_accent_p:n {#7} }
          { \__text_change_case_if_greek_breathing_p:n {#7} }
          {
            \__text_change_case_upper_el_ypogegrammeni:nnnnnnw
              {#1} {#2} {#3} {#4} {#5#7} {#6}
          }
          { \__text_change_case_upper_el_aux:nnnnN {#1} {#2} {#3} {#4} #5#6 #7 }
      }
  }
\cs_new:Npn \__text_change_case_upper_el_aux:nnnnN #1#2#3#4#5
  {
    \__text_codepoint_process:nN
      { \__text_change_case_upper_el_aux:nnnnn {#1} {#2} {#3} {#4} } #5
  }
\cs_new:Npn \__text_change_case_upper_el_aux:nnnnn #1#2#3#4#5
  {
    \__text_codepoint_compare:nNnTF {#5} = { "0308 }
      { \__text_change_case_upper_el_dialytika:nnnn {#2} {#3} {#4} {#1} }
      {
        \__text_change_case_if_greek_accent:nTF {#5}
          { \__text_change_case_upper_el_hiatus:nnnnw {#2} {#3} {#4} {#1} }
          {
            \__text_change_case_if_greek_breathing:nTF {#5}
              { \__text_change_case_upper_el:nnnn {#1} {#2} {#3} {#4} }
              {
                \__text_codepoint_compare:nNnTF {#5} = { "0345 }
                  {
                    \__text_change_case_store:e
                      { \use:c { __text_change_case_upper_ #4 _ypogegrammeni:n } {#1} }
                    \__text_change_case_loop:nnnw {#2} {#3} {#4}
                  }
                  {
                    \__text_change_case_if_greek_stress:nTF {#5}
                      {
                        \__text_change_case_store:e
                          { \__text_change_case_upper_el_stress:nn {#1} {#5} }
                        \__text_change_case_loop:nnnw {#2} {#3} {#4}
                      }
                      {
                        \__text_change_case_store:e
                          { \__text_change_case_codepoint:nn { upper } {#1} }
                        \__text_change_case_loop:nnnw {#2} {#3} {#4} #5
                      }
                  }
              }
          }
      }
  }
\cs_new:Npn \__text_change_case_upper_el_dialytika:nnnn #1#2#3#4
  {
    \__text_change_case_if_takes_dialytika:nTF {#4}
      { \__text_change_case_upper_el_dialytika:n {#4} }
      {
        \__text_change_case_store:e
          { \__text_change_case_codepoint:nn { upper } {#4} }
      }
    \__text_change_case_upper_el_gobble:nnnw {#1} {#2} {#3}
  }
\cs_new:Npn \__text_change_case_upper_el_dialytika:n #1
  {
    \__text_change_case_store:e
      {
        \bool_lazy_or:nnTF
          { \__text_codepoint_compare_p:nNn {#1} = { "0399 } }
          { \__text_codepoint_compare_p:nNn {#1} = { "03B9 } }
          {
            \codepoint_generate:nn { "03AA }
              { \__text_change_case_catcode:nn {#1} { "03AA } }
          }
          {
            \codepoint_generate:nn { "03AB }
              { \__text_change_case_catcode:nn {#1} { "03AB } }
          }
      }
  }
\cs_new:Npn \__text_change_case_upper_el_hiatus:nnnnw
  #1#2#3#4#5 \q__text_recursion_stop
  {
    \tl_if_head_is_N_type:nTF {#5}
      { \__text_change_case_upper_el_hiatus:nnnnN {#4} }
      {
        \__text_change_case_store:e
          { \__text_change_case_codepoint:nn { upper } {#4} }
        \__text_change_case_loop:nnnw
      }
        {#1} {#2} {#3} #5 \q__text_recursion_stop
  }
\cs_new:Npn \__text_change_case_upper_el_hiatus:nnnnN #1#2#3#4#5
  {
    \token_if_cs:NTF #5
      {
        \__text_change_case_store:e
          { \__text_change_case_codepoint:nn { upper } {#1} }
        \__text_change_case_loop:nnnw {#2} {#3} {#4} #5
      }
      {
        \__text_codepoint_process:nN
          { \__text_change_case_upper_el_hiatus:nnnnn {#1} {#2} {#3} {#4} } #5
      }
  }
\cs_new:Npn \__text_change_case_upper_el_hiatus:nnnnn #1#2#3#4#5
  {
    \__text_change_case_if_takes_dialytika:nTF {#5}
      {
        \__text_change_case_store:e
          { \__text_change_case_codepoint:nn { upper } {#1} }
        \__text_change_case_upper_el_dialytika:n {#5}
        \__text_change_case_upper_el_gobble:nnnw {#2} {#3} {#4}
      }
      { \__text_change_case_upper_el:nnnn {#1} {#2} {#3} {#4} #5 }
  }
\cs_new:Npn \__text_change_case_upper_el_ypogegrammeni:n #1
  {
    \exp_args:Ne \__text_change_case_generate:n
      {
        \int_case:nn
          { \__text_codepoint_from_chars:Nw #1 }
          {
            { "0391 } { "1FBC }
            { "03B1 } { "1FBC }
            { "0397 } { "1FCC }
            { "03B7 } { "1FCC }
            { "03A9 } { "1FFC }
            { "03C9 } { "1FFC }
          }
      }
  }
\cs_new:cpn { __text_change_case_upper_el-x-iota_ypogegrammeni:n } #1
  {
    \__text_change_case_codepoint:nn { upper } {#1}
    \codepoint_generate:nn { "0399 }
      { \char_value_catcode:n { "0399 } }
  }
\cs_new:Npn \__text_change_case_upper_el_stress:nn #1#2
  {
    \exp_args:Ne \__text_change_case_generate:n
      {
        \int_case:nn
          { \__text_codepoint_from_chars:Nw #2 }
          {
            { "0304 }
              {
                \int_case:nn { \__text_codepoint_from_chars:Nw #1 }
                  {
                    { "0391 } { "1FB9 }
                    { "03B1 } { "1FB9 }
                    { "0399 } { "1FD9 }
                    { "03B9 } { "1FD9 }
                    { "03A5 } { "1FE9 }
                    { "03C5 } { "1FE9 }
                  }
              }
            { "0306 }
              {
                \int_case:nn { \__text_codepoint_from_chars:Nw #1 }
                  {
                    { "0391 } { "1FB8 }
                    { "03B1 } { "1FB8 }
                    { "0399 } { "1FD8 }
                    { "03B9 } { "1FD8 }
                    { "03A5 } { "1FE8 }
                    { "03C5 } { "1FE8 }
                  }
              }
          }
      }
  }
\cs_new:Npn \__text_change_case_upper_el_gobble:nnnw
  #1#2#3#4 \q__text_recursion_stop
  {
    \tl_if_head_is_N_type:nTF {#4}
      { \__text_change_case_upper_el_gobble:nnnN }
      { \__text_change_case_loop:nnnw }
        {#1} {#2} {#3} #4 \q__text_recursion_stop
  }
\cs_new:Npn \__text_change_case_upper_el_gobble:nnnN #1#2#3#4
  {
    \token_if_cs:NTF #4
      { \__text_change_case_loop:nnnw {#1} {#2} {#3} }
      {
        \__text_codepoint_process:nN
          { \__text_change_case_upper_el_gobble:nnnn {#1} {#2} {#3} }
      }
        #4
  }
\cs_new:Npn \__text_change_case_upper_el_gobble:nnnn #1#2#3#4
  {
    \bool_lazy_or:nnTF
      { \__text_change_case_if_greek_accent_p:n {#4} }
      { \__text_change_case_if_greek_breathing_p:n {#4} }
      { \__text_change_case_upper_el_gobble:nnnw {#1} {#2} {#3} }
      { \__text_change_case_loop:nnnw {#1} {#2} {#3} #4 }
  }
\prg_new_conditional:Npnn \__text_change_case_if_greek:n #1 { p , TF }
  {
    \exp_args:Nf \__text_change_case_if_greek:n
      { \int_eval:n { \__text_codepoint_from_chars:Nw #1 } }
  }
\cs_new:Npn \__text_change_case_if_greek:n #1
  {
    \if_int_compare:w #1 < "0370 \exp_stop_f:
      \prg_return_false:
    \else:
      \if_int_compare:w #1 > "03FF \exp_stop_f:
        \if_int_compare:w #1 < "1F00 \exp_stop_f:
          \prg_return_false:
        \else:
          \if_int_compare:w #1 > "1FFF \exp_stop_f:
            \if_int_compare:w #1 = "2126 \exp_stop_f:
              \prg_return_true:
            \else:
              \prg_return_false:
            \fi:
          \else:
            \prg_return_true:
```

```
          \fi:
        \fi:
      \else:
        \prg_return_true:
      \fi:
    \fi:
  }
\prg_new_conditional:Npnn \__text_change_case_if_greek_accent:n #1 { TF , p }
  {
    \exp_args:Nf \__text_change_case_if_greek_accent:n
      { \int_eval:n { \__text_codepoint_from_chars:Nw #1 } }
  }
\cs_new:Npn \__text_change_case_if_greek_accent:n #1
  {
    \if_int_compare:w #1 = "0300 \exp_stop_f:
      \prg_return_true:
    \else:
      \if_int_compare:w #1 = "0301 \exp_stop_f:
        \prg_return_true:
      \else:
        \if_int_compare:w #1 = "0342 \exp_stop_f:
          \prg_return_true:
        \else:
          \if_int_compare:w #1 = "0302 \exp_stop_f:
            \prg_return_true:
          \else:
            \if_int_compare:w #1 = "0303 \exp_stop_f:
              \prg_return_true:
            \else:
              \if_int_compare:w #1 = "0311 \exp_stop_f:
                \prg_return_true:
              \else:
                \prg_return_false:
              \fi:
            \fi:
          \fi:
        \fi:
      \fi:
    \fi:
  }
\prg_new_conditional:Npnn \__text_change_case_if_greek_spacing_diacritic:n
  #1 { TF }
  {
    \exp_args:Nf \__text_change_case_if_greek_spacing_diacritic:n
      { \int_eval:n { \__text_codepoint_from_chars:Nw #1 } }
  }
```

```
\cs_new:Npn \__text_change_case_if_greek_spacing_diacritic:n #1
  {
    \if_int_compare:w #1 < "1FBD \exp_stop_f:
      \if_int_compare:w #1 = "037A \exp_stop_f:
        \prg_return_true:
      \else:
        \prg_return_false:
      \fi:
    \else:
      \if_int_compare:w #1 = "1FBD \exp_stop_f:
        \prg_return_true:
      \else:
        \if_int_compare:w #1 = "1FBF \exp_stop_f:
          \prg_return_true:
        \else:
          \if_int_compare:w #1 = "1FC0 \exp_stop_f:
            \prg_return_true:
          \else:
            \if_int_compare:w #1 = "1FC1 \exp_stop_f:
              \prg_return_true:
            \else:
              \if_int_compare:w #1 = "1FCD \exp_stop_f:
                \prg_return_true:
              \else:
                \if_int_compare:w #1 = "1FCE \exp_stop_f:
                  \prg_return_true:
                \else:
                  \if_int_compare:w #1 = "1FCF \exp_stop_f:
                    \prg_return_true:
                  \else:
                    \if_int_compare:w #1 = "1FDD \exp_stop_f:
                      \prg_return_true:
                    \else:
                      \if_int_compare:w #1 = "1FDE \exp_stop_f:
                        \prg_return_true:
                      \else:
                        \if_int_compare:w #1 = "1FDF \exp_stop_f:
                          \prg_return_true:
                        \else:
                          \if_int_compare:w #1 = "1FED \exp_stop_f:
                            \prg_return_true:
                          \else:
                            \if_int_compare:w #1 = "1FEE \exp_stop_f:
                              \prg_return_true:
                            \else:
                              \if_int_compare:w #1 = "1FEF \exp_stop_f:
                                \prg_return_true:
                              \else:
                                \if_int_compare:w #1 = "1FFD \exp_stop_f:
                                  \prg_return_true:
                                \else:
                                  \if_int_compare:w #1 = "1FFE \exp_stop_f:
                                    \prg_return_true:
                                  \else:
                                    \prg_return_false:
                                  \fi:
                                \fi:
                              \fi:
                            \fi:
                          \fi:
                        \fi:
                      \fi:
                    \fi:
                  \fi:
                \fi:
              \fi:
            \fi:
          \fi:
        \fi:
      \fi:
    \fi:
  }
\prg_new_conditional:Npnn \__text_change_case_if_greek_breathing:n
  #1 { TF , p }
  {
    \exp_args:Nf \__text_change_case_if_greek_breathing:n
      { \int_eval:n { \__text_codepoint_from_chars:Nw #1 } }
  }
\cs_new:Npn \__text_change_case_if_greek_breathing:n #1
  {
    \if_int_compare:w #1 = "0313 \exp_stop_f:
      \prg_return_true:
    \else:
      \if_int_compare:w #1 = "0314 \exp_stop_f:
        \prg_return_true:
      \else:
        \prg_return_false:
      \fi:
    \fi:
  }
\prg_new_conditional:Npnn \__text_change_case_if_greek_stress:n
  #1 { TF , p }
  {
    \exp_args:Nf \__text_change_case_if_greek_stress:n
      { \int_eval:n { \__text_codepoint_from_chars:Nw #1 } }
  }
\cs_new:Npn \__text_change_case_if_greek_stress:n #1
  {
    \if_int_compare:w #1 = "0304 \exp_stop_f:
      \prg_return_true:
    \else:
      \if_int_compare:w #1 = "0306 \exp_stop_f:
        \prg_return_true:
      \else:
        \prg_return_false:
      \fi:
    \fi:
  }
\prg_new_conditional:Npnn \__text_change_case_if_takes_dialytika:n #1 { TF }
  {
    \exp_args:Nf \__text_change_case_if_takes_dialytika:n
      { \int_eval:n { \__text_codepoint_from_chars:Nw #1 } }
  }
\cs_new:Npn \__text_change_case_if_takes_dialytika:n #1
  {
    \if_int_compare:w #1 = "0399 \exp_stop_f:
      \prg_return_true:
    \else:
      \if_int_compare:w #1 = "03B9 \exp_stop_f:
        \prg_return_true:
      \else:
        \if_int_compare:w #1 = "03A5 \exp_stop_f:
          \prg_return_true:
        \else:
          \if_int_compare:w #1 = "03C5 \exp_stop_f:
            \prg_return_true:
          \else:
            \prg_return_false:
          \fi:
        \fi:
      \fi:
    \fi:
  }
\prg_new_conditional:Npnn \__text_change_case_if_takes_ypogegrammeni:n #1 { TF }
  {
    \exp_args:Nf \__text_change_case_if_takes_ypogegrammeni:n
      { \int_eval:n { \__text_codepoint_from_chars:Nw #1 } }
  }
```

```
\cs_new:Npn \__text_change_case_if_takes_ypogegrammeni:n #1
  {
    \if_int_compare:w #1 = "03B1 \exp_stop_f:
      \prg_return_true:
    \else:
      \if_int_compare:w #1 = "03B7 \exp_stop_f:
        \prg_return_true:
      \else:
        \if_int_compare:w #1 = "03C9 \exp_stop_f:
          \prg_return_true:
        \else:
          \prg_return_false:
        \fi:
      \fi:
    \fi:
  }
\cs_new:Npn \__text_change_case_boundary_upper_el:Nnnnw
  #1#2#3#4#5 \q__text_recursion_stop
  {
    \tl_if_head_is_N_type:nTF {#5}
      { \__text_change_case_boundary_upper_el:nnnN }
      { \__text_change_case_loop:nnnw }
        {#2} {#3} {#4} #5 \q__text_recursion_stop
  }
\cs_new_eq:cN { __text_change_case_boundary_upper_el-x-iota:Nnnnw }
  \__text_change_case_boundary_upper_el:Nnnnw
\cs_new:Npn \__text_change_case_boundary_upper_el:nnnN #1#2#3#4
  {
    \token_if_cs:NTF #4
      { \__text_change_case_loop:nnnw {#1} {#2} {#3} }
      {
        \__text_codepoint_process:nN
          { \__text_change_case_boundary_upper_el:nnnn {#1} {#2} {#3} }
      }
        #4
  }
\cs_new:Npn \__text_change_case_boundary_upper_el:nnnn #1#2#3#4
  {
    \bool_lazy_any:nTF
      {
        { \__text_codepoint_compare_p:nNn {#4} = { "0389 } }
        { \__text_codepoint_compare_p:nNn {#4} = { "03AE } }
        { \__text_codepoint_compare_p:nNn {#4} = { "1F22 } }
        { \__text_codepoint_compare_p:nNn {#4} = { "1F2A } }
      }
      { \__text_change_case_boundary_upper_el:nnnnw {#1} {#2} {#3} {#4} }
      { \__text_change_case_breathing:nnnn {#1} {#2} {#3} {#4} }
  }
\cs_new:Npn \__text_change_case_boundary_upper_el:nnnnw
  #1#2#3#4#5 \q__text_recursion_stop
  {
    \tl_if_head_is_N_type:nTF {#5}
      { \__text_change_case_loop:nnnw {#1} {#2} {#3} #4 }
      {
        \__text_change_case_store:e
          {
            \codepoint_generate:nn { "0389 }
              { \__text_change_case_catcode:nn {#4} { "0389 } }
          }
        \__text_change_case_loop:nnnw {#1} {#2} {#3}
      }
        #5 \q__text_recursion_stop
  }
\cs_new:Npn \__text_change_case_breathing:nnnn #1#2#3#4
  {
    \__text_change_case_if_greek:nTF {#4}
      {
        \exp_args:Ne \__text_change_case_breathing:nnnnn
          {
            \codepoint_to_nfd:n
              { \__text_codepoint_from_chars:Nw #4 }
          }
            {#1} {#2} {#3} {#4}
      }
      { \__text_change_case_loop:nnnw {#1} {#2} {#3} #4 }
  }
\cs_new:Npn \__text_change_case_breathing:nnnnn #1#2#3#4#5
  {
    \__text_codepoint_process:nN
      { \__text_change_case_breathing:nnnnnw {#2} {#3} {#4} {#5} }
        #1 \q_mark
  }
\cs_new:Npn \__text_change_case_breathing:nnnnnw #1#2#3#4#5#6 \q_mark
  {
    \tl_if_blank:nTF {#6}
      { \__text_change_case_loop:nnnw {#1} {#2} {#3} #4 }
      {
        \__text_codepoint_process:nN
          { \__text_change_case_breathing:nnnnnnw {#1} {#2} {#3} {#4} {#5} }
            #6 \q_mark
      }
  }
```

```
\cs_new:Npn \__text_change_case_breathing:nnnnnnw #1#2#3#4#5#6#7 \q_mark
  {
    \tl_if_blank:nTF {#7}
      {
        \__text_change_case_breathing_aux:nnnnnn
          {#1} {#2} {#3} {#4} {#5} {#6}
      }
      {
        \__text_codepoint_process:nN
          { \__text_change_case_breathing:nnnnnnw {#1} {#2} {#3} {#4} {#5} }
          #7 \q_mark
      }
  }
\cs_new:Npn \__text_change_case_breathing_aux:nnnnnn #1#2#3#4#5#6
  {
    \bool_lazy_or:nnTF
      { \__text_codepoint_compare_p:nNn {#6} = { "0313 } }
      { \__text_codepoint_compare_p:nNn {#6} = { "0314 } }
      { \__text_change_case_breathing_aux:nnnnw {#1} {#2} {#3} {#5} }
      { \__text_change_case_loop:nnnw {#1} {#2} {#3} #4 }
  }
\cs_new:Npn \__text_change_case_breathing_aux:nnnnw #1#2#3#4#5
  \q__text_recursion_stop
  {
    \__text_change_case_store:e
      { \__text_change_case_codepoint:nn { upper } {#4} }
    \tl_if_head_is_N_type:nTF {#5}
      { \__text_change_case_breathing_aux:nnnN }
      { \__text_change_case_loop:nnnw }
        {#1} {#2} {#3} #5 \q__text_recursion_stop
  }
\cs_new:Npn \__text_change_case_breathing_aux:nnnN #1#2#3#4
  {
    \__text_codepoint_process:nN
      { \__text_change_case_breathing_dialytika:nnnn {#1} {#2} {#3} } #4
  }
\cs_new:Npn \__text_change_case_breathing_dialytika:nnnn #1#2#3#4
  {
    \__text_change_case_if_takes_dialytika:nTF {#4}
      {
        \__text_change_case_upper_el_dialytika:n {#4}
        \__text_change_case_loop:nnnw {#1} {#2} {#3}
      }
      { \__text_change_case_loop:nnnw {#1} {#2} {#3} #4 }
  }
\cs_new:Npn \__text_change_case_title_el:nnnnn #1#2#3#4#5
```

```
  { \__text_change_case_codepoint:nnnnn {#1} {#2} {#3} {#4} {#5} }
\cs_new:Npn \__text_change_case_upper_hy:nnnnn #1#2#3#4#5
  {
    \__text_codepoint_compare:nNnTF {#5} = { "0587 }
      {
        \__text_change_case_store:e
          {
            \codepoint_generate:nn { "0535 }
              { \__text_change_case_catcode:nn {#5} { "0535 } }
            \codepoint_generate:nn { "054E }
              { \__text_change_case_catcode:nn {#5} { "054E } }
          }
        \use:c { __text_change_case_next_ #2 :nnn }
          {#2} {#3} {#4}
      }
      { \__text_change_case_codepoint:nnnnn {#1} {#2} {#3} {#4} {#5} }
  }
\cs_new:Npn \__text_change_case_title_hy:nnnnn #1#2#3#4#5
  {
    \__text_codepoint_compare:nNnTF {#5} = { "0587 }
      {
        \__text_change_case_store:e
          {
            \codepoint_generate:nn { "0535 }
              { \__text_change_case_catcode:nn {#5} { "0535 } }
            \codepoint_generate:nn { "057E }
              { \__text_change_case_catcode:nn {#5} { "057E } }
          }
        \use:c { __text_change_case_next_ #2 :nnn }
          {#2} {#3} {#4}
      }
      { \__text_change_case_codepoint:nnnnn {#1} {#2} {#3} {#4} {#5} }
  }
\cs_new:cpn { __text_change_case_upper_hy-x-yiwn:nnnnn } #1#2#3#4#5
  { \__text_change_case_codepoint:nnnnn {#1} {#2} {#3} {#4} {#5} }
\cs_new_eq:cc { __text_change_case_title_hy-x-yiwn:nnnnn }
  { __text_change_case_upper_hy-x-yiwn:nnnnn }
\cs_new:cpn { __text_change_case_lower_la-x-medieval:nnnnn } #1#2#3#4#5
  {
    \__text_codepoint_compare:nNnTF {#5} = { "0056 }
      {
        \__text_change_case_store:e
          { \char_generate:nn { "0075 } { \__text_char_catcode:N #5 } }
        \use:c { __text_change_case_next_ #2 :nnn }
          {#2} {#3} {#4}
      }
      { \__text_change_case_codepoint:nnnnn {#1} {#2} {#3} {#4} {#5} }
  }
\cs_new:cpn { __text_change_case_upper_la-x-medieval:nnnnn } #1#2#3#4#5
  {
    \__text_codepoint_compare:nNnTF {#5} = { "0075 }
      {
        \__text_change_case_store:e
          { \char_generate:nn { "0056 } { \__text_char_catcode:N #5 } }
        \use:c { __text_change_case_next_ #2 :nnn }
          {#2} {#3} {#4}
      }
      { \__text_change_case_codepoint:nnnnn {#1} {#2} {#3} {#4} {#5} }
  }
\cs_new:Npn \__text_change_case_lower_lt:nnnnn #1#2#3#4#5
  {
    \exp_args:Ne \__text_change_case_lower_lt_auxi:nnnnn
      {
        \int_case:nn { \__text_codepoint_from_chars:Nw #5 }
          {
            { "00CC } { "0300 }
            { "00CD } { "0301 }
            { "0128 } { "0303 }
          }
      }
        {#2} {#3} {#4} {#5}
  }
\cs_new:Npn \__text_change_case_lower_lt_auxi:nnnnn #1#2#3#4#5
  {
    \tl_if_blank:nTF {#1}
      {
        \exp_args:Ne \__text_change_case_lower_lt_auxii:nnnnn
          {
            \int_case:nn { \__text_codepoint_from_chars:Nw #5 }
              {
                { "0049 } { "0069 }
                { "004A } { "006A }
                { "012E } { "012F }
              }
          }
            {#2} {#3} {#4} {#5}
      }
      {
        \__text_change_case_store:e
          {
            \codepoint_generate:nn { "0069 }
              { \__text_change_case_catcode:nn {#5} { "0069 } }
            \codepoint_generate:nn { "0307 }
              { \__text_change_case_catcode:nn {#5} { "0307 } }
            \codepoint_generate:nn {#1}
              { \__text_change_case_catcode:nn {#5} {#1} }
          }
        \__text_change_case_loop:nnnw {#2} {#3} {#4}
      }
  }
\cs_new:Npn \__text_change_case_lower_lt_auxii:nnnnn #1#2#3#4#5
  {
    \tl_if_blank:nTF {#1}
      { \__text_change_case_codepoint:nnnnn {#2} {#2} {#3} {#4} {#5} }
      {
        \__text_change_case_store:e
          {
            \codepoint_generate:nn {#1}
              { \__text_change_case_catcode:nn {#5} {#1} }
          }
        \__text_change_case_lower_lt:nnnw {#2} {#3} {#4}
      }
  }
\cs_new:Npn \__text_change_case_lower_lt:nnnw #1#2#3#4 \q__text_recursion_stop
  {
    \tl_if_head_is_N_type:nTF {#4}
      { \__text_change_case_lower_lt:nnnN }
      { \__text_change_case_loop:nnnw }
        {#1} {#2} {#3} #4 \q__text_recursion_stop
  }
\cs_new:Npn \__text_change_case_lower_lt:nnnN #1#2#3#4
  {
    \__text_codepoint_process:nN
      { \__text_change_case_lower_lt:nnnn {#1} {#2} {#3} } #4
  }
\cs_new:Npn \__text_change_case_lower_lt:nnnn #1#2#3#4
  {
    \bool_lazy_and:nnT
      {
        \bool_lazy_or_p:nn
          { ! \tl_if_single_p:n {#4} }
          { ! \token_if_cs_p:N #4 }
      }
      {
        \bool_lazy_any_p:n
          {
            { \__text_codepoint_compare_p:nNn {#4} = { "0300 } }
            { \__text_codepoint_compare_p:nNn {#4} = { "0301 } }
            { \__text_codepoint_compare_p:nNn {#4} = { "0303 } }
          }
      }
      {
        \__text_change_case_store:e
          {
            \codepoint_generate:nn { "0307 }
              { \__text_change_case_catcode:nn {#4} { "0307 } }
          }
      }
    \__text_change_case_loop:nnnw {#1} {#2} {#3} #4
  }
\cs_new:Npn \__text_change_case_upper_lt:nnnnn #1#2#3#4#5
  {
    \exp_args:Ne \__text_change_case_upper_lt_aux:nnnnn
      {
        \int_case:nn { \__text_codepoint_from_chars:Nw #5 }
          {
            { "0069 } { "0049 }
            { "006A } { "004A }
            { "012F } { "012E }
          }
      }
      {#2} {#3} {#4} {#5}
  }
\cs_new:Npn \__text_change_case_upper_lt_aux:nnnnn #1#2#3#4#5
  {
    \tl_if_blank:nTF {#1}
      { \__text_change_case_codepoint:nnnnn { upper } {#2} {#3} {#4} {#5} }
      {
        \__text_change_case_store:e
          {
            \codepoint_generate:nn {#1}
              { \__text_change_case_catcode:nn {#5} {#1} }
          }
        \__text_change_case_upper_lt:nnnw {#2} {#3} {#4}
      }
  }
\cs_new:Npn \__text_change_case_upper_lt:nnnw #1#2#3#4 \q__text_recursion_stop
  {
    \tl_if_head_is_N_type:nTF {#4}
      { \__text_change_case_upper_lt:nnnN }
      { \use:c { __text_change_case_next_ #1 :nnn } }
        {#1} {#2} {#3} #4 \q__text_recursion_stop
  }
\cs_new:Npn \__text_change_case_upper_lt:nnnN #1#2#3#4
```

```
  {
    \__text_codepoint_process:nN
      { \__text_change_case_upper_lt:nnnn {#1} {#2} {#3} } #4
  }
\cs_new:Npn \__text_change_case_upper_lt:nnnn #1#2#3#4
  {
    \bool_lazy_and:nnTF
      {
        \bool_lazy_or_p:nn
          { ! \tl_if_single_p:n {#4} }
          { ! \token_if_cs_p:N #4 }
      }
      { \__text_codepoint_compare_p:nNn {#4} = { "0307 } }
      { \use:c { __text_change_case_next_ #1 :nnn } {#1} {#2} {#3} }
      { \use:c { __text_change_case_next_ #1 :nnn } {#1} {#2} {#3} #4 }
  }
\cs_new:Npn \__text_change_case_title_nl:nnnnn #1#2#3#4#5
  {
    \tl_if_single:nTF {#5}
      { \__text_change_case_title_nl_aux:nnnnn }
      { \__text_change_case_codepoint:nnnnn }
        {#1} {#2} {#3} {#4}  {#5}
  }
\cs_new:Npn \__text_change_case_title_nl_aux:nnnnn #1#2#3#4#5
  {
    \bool_lazy_or:nnTF
      { \int_compare_p:nNn {`#5} = { "0049 } }
      { \int_compare_p:nNn {`#5} = { "0069 } }
      {
        \__text_change_case_store:e
          { \char_generate:nn { "0049 } { \__text_char_catcode:N #5 } }
        \__text_change_case_title_nl:nnnw {#2} {#3} {#4}
      }
      { \__text_change_case_codepoint:nnnnn {#1} {#2} {#3} {#4} {#5} }
  }
\cs_new:Npn \__text_change_case_title_nl:nnnw #1#2#3#4 \q__text_recursion_stop
  {
    \tl_if_head_is_N_type:nTF {#4}
      { \__text_change_case_title_nl:nnnN }
      { \use:c { __text_change_case_next_ #1 :nnn } }
        {#1} {#2} {#3} #4 \q__text_recursion_stop
  }
\cs_new:Npn \__text_change_case_title_nl:nnnN #1#2#3#4
  {
    \bool_lazy_and:nnTF
      { ! \token_if_cs_p:N #4 }
      {
        \bool_lazy_or_p:nn
          { \int_compare_p:nNn {`#4} = { "004A } }
          { \int_compare_p:nNn {`#4} = { "006A } }
      }
      {
        \__text_change_case_store:e
          { \char_generate:nn { "004A } { \__text_char_catcode:N #4 } }
        \use:c { __text_change_case_next_ #1 :nnn } {#1} {#2} {#3}
      }
      { \use:c { __text_change_case_next_ #1 :nnn } {#1} {#2} {#3} #4 }
  }
\cs_new:Npn \__text_change_case_lower_tr:nnnnn #1#2#3#4#5
  {
    \__text_codepoint_compare:nNnTF {#5} = { "0049 }
      { \__text_change_case_lower_tr:nnnNw {#1} {#3} {#4} #5 }
      {
        \__text_codepoint_compare:nNnTF {#5} = { "0130 }
          {
            \__text_change_case_store:e
              {
                \codepoint_generate:nn { "0069 }
                  { \__text_change_case_catcode:nn {#5} { "0069 } }
              }
            \__text_change_case_loop:nnnw {#1} {#3} {#4}
          }
          { \__text_change_case_codepoint:nnnnn {#1} {#2} {#3} {#4} {#5} }
      }
  }
\cs_new:Npn \__text_change_case_lower_tr:nnnNw #1#2#3#4#5 \q__text_recursion_stop
  {
    \tl_if_head_is_N_type:nTF {#5}
      { \__text_change_case_lower_tr:NnnnN  #4 {#1} {#2} {#3} }
      {
        \__text_change_case_store:e
          {
            \codepoint_generate:nn { "0131 }
              { \__text_change_case_catcode:nn {#4} { "0131 } }
          }
        \__text_change_case_loop:nnnw {#1} {#2} {#3}
      }
        #5 \q__text_recursion_stop
  }
\cs_new:Npn \__text_change_case_lower_tr:NnnnN #1#2#3#4#5
  {
    \__text_codepoint_process:nN
```

```
      { \__text_change_case_lower_tr:Nnnnn #1 {#2} {#3} {#4} } #5
  }
\cs_new:Npn \__text_change_case_lower_tr:Nnnnn #1#2#3#4#5
  {
    \bool_lazy_or:nnTF
      {
        \bool_lazy_and_p:nn
          { \tl_if_single_p:n {#5} }
          { \token_if_cs_p:N #5 }
      }
      { ! \__text_codepoint_compare_p:nNn {#5} = { "0307 } }
      {
        \__text_change_case_store:e
          {
            \codepoint_generate:nn { "0131 }
              { \__text_change_case_catcode:nn {#1} { "0131 } }
          }
        \__text_change_case_loop:nnnw {#2} {#3} {#4} #5
      }
      {
        \__text_change_case_store:e
          {
            \codepoint_generate:nn { "0069 }
              { \__text_change_case_catcode:nn {#1} { "0069 } }
          }
        \__text_change_case_loop:nnnw {#2} {#3} {#4}
      }
  }
\cs_new:Npn \__text_change_case_upper_tr:nnnnn #1#2#3#4#5
  {
    \__text_codepoint_compare:nNnTF {#5} = { "0069 }
      {
        \__text_change_case_store:e
          {
            \codepoint_generate:nn { "0130 }
              { \__text_change_case_catcode:nn {#5} { "0130 } }
          }
        \use:c { __text_change_case_next_ #2 :nnn } {#2} {#3} {#4}
      }
      { \__text_change_case_codepoint:nnnnn {#1} {#2} {#3} {#4} {#5} }
  }
\cs_new_eq:NN \__text_change_case_lower_az:nnnnn
  \__text_change_case_lower_tr:nnnnn
\cs_new_eq:NN \__text_change_case_upper_az:nnnnn
  \__text_change_case_upper_tr:nnnnn
\group_begin:
  \cs_set_protected:Npn \__text_change_case_setup:NN #1#2
    {
      \quark_if_recursion_tail_stop:N #1
      \tl_const:cn { c__text_lowercase_ \token_to_str:N #1 _tl }
        { #2 }
      \tl_const:cn { c__text_uppercase_ \token_to_str:N #2 _tl }
        { #1 }
      \__text_change_case_setup:NN
    }
  \__text_change_case_setup:NN
  \AA \aa
  \AE \ae
  \DH \dh
  \DJ \dj
  \IJ \ij
  \L  \l
  \NG \ng
  \O  \o
  \OE \oe
  \SS \ss
  \TH \th
  \q_recursion_tail ?
  \q_recursion_stop
  \tl_const:cn { c__text_uppercase_ \token_to_str:N \i _tl } { I }
  \tl_const:cn { c__text_uppercase_ \token_to_str:N \j _tl } { J }
\group_end:
\tl_if_exist:NT \@expl@finalise@setup@@
  {
    \tl_gput_right:Nn \@expl@finalise@setup@@
      {
        \tl_gput_right:Nn \@kernel@after@begindocument
          {
            \group_begin:
              \cs_set_protected:Npn \__text_change_case_setup:Nn #1#2
                {
                  \quark_if_recursion_tail_stop:N #1
                  \tl_if_single_token:nT {#2}
                    {
                      \cs_if_exist:cF
                        { c__text_uppercase_ \token_to_str:N #1 _tl }
                        {
                          \tl_const:cn
                            { c__text_uppercase_ \token_to_str:N #1 _tl }
                            { #2 }
                        }
                      \cs_if_exist:cF
            { c__text_lowercase_ \token_to_str:N #2 _tl }
            {
             \tl_const:cn
              { c__text_lowercase_ \token_to_str:N #2 _tl }
              { #1 }
            }
          }
         \__text_change_case_setup:Nn
        }
       \exp_after:wN \__text_change_case_setup:Nn \@uclclist
       \q_recursion_tail ?
       \q_recursion_stop
      \group_end:
     }
   }
 }
\sys_if_engine_opentype:F
 {
  \text_declare_uppercase_mapping:nn { "01F0 } { \v { J } }
 }
%% File: l3text-map.dtx
\cs_new:Npn \__text_map_function:nnN #1#2#3
 {
  \__text_map_loop:Nnnw #3 {#2} { } #1
   \q__text_recursion_tail \q__text_recursion_stop
  \prg_break_point:Nn \text_map_break: { }
 }
\cs_generate_variant:Nn \__text_map_function:nnN { e }
\cs_new:Npn \__text_map_loop:Nnnw #1#2#3#4 \q__text_recursion_stop
 {
  \tl_if_head_is_N_type:nTF {#4}
   { \__text_map_N_type:NnnN }
   {
    \tl_if_head_is_group:nTF {#4}
     { \__text_map_group:Nnnn }
     { \__text_map_space:Nnnw }
   }
  #1 {#2} {#3} #4 \q__text_recursion_stop
 }
\cs_new:Npn \__text_map_group:Nnnn #1#2#3#4
 {
  \__text_map_output:Nn #1 {#3}
  {
   \__text_map_loop:Nnnw #1 {#2} { } #4
    \q__text_recursion_tail \q__text_recursion_stop
   \prg_break_point:Nn \text_map_break: { }
```

```
    }
    \__text_map_loop:Nnnw #1 {#2} { }
  }
\use:e
  { \cs_new:Npn \exp_not:N \__text_map_space:Nnnw #1#2#3 \c_space_tl }
  {
    \__text_map_output:Nn #1 {#3}
    #1 { ~ }
    \__text_map_loop:Nnnw #1 {#2} { }
  }
\cs_new:Npn \__text_map_N_type:NnnN #1#2#3#4
  {
    \__text_if_q_recursion_tail_stop_do:Nn #4
      {
        \__text_map_output:Nn #1 {#3}
        \text_map_break:
      }
    \token_if_cs:NTF #4
      {
        \__text_map_output:Nn #1 {#3}
        #1 {#4}
        \__text_map_loop:Nnnw #1 {#2} { }
      }
      {
        \__text_codepoint_process:nN
          { \__text_map_codepoint:Nnnn #1 {#2} {#3} } #4
      }
  }
\cs_new:Npn \__text_map_codepoint:Nnnn #1#2#3#4
  {
    \__text_codepoint_compare:nNnTF {#4} = { "000D }
      {
        \__text_map_output:Nn #1 {#3}
        \__text_map_CR:Nnnw #1 {#2} {#4}
      }
      {
        \__text_codepoint_compare:nNnTF {#4} = { "200D }
          { \__text_map_loop:Nnnw #1 {#2} {#3#4} }
          { \__text_map_class:Nnnn #1 {#2} {#3} {#4} }
      }
  }
\cs_new:Npn \__text_map_CR:Nnnw #1#2#3#4 \q__text_recursion_stop
  {
    \tl_if_head_is_N_type:nTF {#4}
      { \__text_map_CR:NnnN #1 {#2} {#3} }
      {
        #1 {#3}
        \__text_map_loop:Nnnw #1 {#2} { }
      }
        #4 \q__text_recursion_stop
  }
\cs_new:Npn \__text_map_CR:NnnN #1#2#3#4
  {
    \__text_if_q_recursion_tail_stop_do:Nn #4
      {
        #1 {#3}
        \text_map_break:
      }
    \bool_lazy_and:nnTF
      { ! \token_if_cs_p:N #4 }
      { \int_compare_p:nNn { `#4 } = { "000A } }
      {
        \__text_map_output:Nn #1 {#3#4}
        \__text_map_loop:Nnnw #1 {#2} { }
      }
      { \__text_map_loop:Nnnw #1 {#2} { } #3 }
  }
\cs_new:Npn \__text_map_class:Nnnn #1#2#3#4
  {
    \exp_args:Nnnne \__text_map_class:Nnnnn #1 {#2} {#3} {#4}
      {
        \use:c { __kernel_codepoint_to_ #2 _class:n }
          { \__text_codepoint_from_chars:Nw #4 }
      }
  }
\cs_new:Npn \__text_map_class:Nnnnn #1#2#3#4#5
  {
    \cs_if_exist_use:cF { __text_map_ #5 :Nnnn }
      { \__text_map_Other:Nnnn }
        #1 {#2} {#3} {#4}
  }
\cs_new:Npn \__text_map_lookahead:Nnnnnw #1#2#3#4#5#6 \q__text_recursion_stop
  {
    \tl_if_head_is_N_type:nTF {#6}
      { \__text_map_lookahead:NnnnnN #1 {#2} {#3} {#4} {#5} }
      { \__text_map_loop:Nnnw #1 {#2} {#3} #4 }
    #6 \q__text_recursion_stop
  }
\cs_new:Npn \__text_map_lookahead:NnnnnN #1#2#3#4#5#6
  {
    \__text_if_q_recursion_tail_stop_do:Nn #6
      {
        #1 {#3}
        \tl_if_blank:nF {#4} { #1 {#4} }
      }
    \token_if_cs:NTF #6
      {
        #1 {#3}
        \__text_map_loop:Nnnw #1 {#2} { } #4
      }
      { \__text_codepoint_process:nN { #5 #1 {#2} {#3} {#4} } }
        #6
  }
\prg_new_conditional:Npnn \__text_map_if_ignorable:n #1 { TF }
  {
    \str_case:nnTF {#1}
      {
        { Extend }       { }
        { Format }       { }
        { ZWJ }          { }
      }
      \prg_return_true:
      \prg_return_false:
  }
\cs_new:Npn \__text_map_output:Nn #1#2
  { \tl_if_blank:nF {#2} { #1 {#2} } }
\cs_new:Npn \text_map_break:
  { \prg_map_break:Nn \text_map_break: { } }
\cs_new:Npn \text_map_break:n
  { \prg_map_break:Nn \text_map_break: }
\cs_new:Npn \__text_map_Control:Nnnn #1#2#3#4
  {
    \__text_map_output:Nn #1 {#3}
    \__text_map_output:Nn #1 {#4}
    \__text_map_loop:Nnnw #1 {#2} { }
  }
\cs_new_eq:NN \__text_map_Newline:Nnnn \__text_map_Control:Nnnn
\cs_new:Npn \__text_map_Extend:Nnnn #1#2#3#4
  { \__text_map_loop:Nnnw #1 {#2} {#3#4} }
\cs_new_eq:NN \__text_map_Format:Nnnn \__text_map_Extend:Nnnn
\cs_new_eq:NN \__text_map_SpacingMark:Nnnn \__text_map_Extend:Nnnn
\cs_new:Npn \__text_map_Other:Nnnn #1#2#3#4
  {
    \__text_map_output:Nn #1 {#3}
    \__text_map_loop:Nnnw #1 {#2} {#4}
  }
\cs_new:Npn \__text_map_Regional_Indicator:Nnnn #1#2#3#4
  {
    \__text_map_output:Nn #1 {#3}
    \__text_map_lookahead:Nnnnnw #1 {#2} {#4} { }
      \__text_map_Regional_Indicator_aux:Nnnnn
  }
\cs_new:Npn \__text_map_Regional_Indicator_aux:Nnnnn #1#2#3#4#5
  {
    \bool_lazy_or:nnTF
      { \__text_codepoint_compare_p:nNn {#5} < { "1F1E6 } }
      { \__text_codepoint_compare_p:nNn {#5} > { "1F1FF } }
      {
        \str_if_eq:nnTF {#2} { wordbreak }
          {
            \exp_args:Ne \__text_map_if_ignorable:nTF
              {
                \__kernel_codepoint_to_grapheme_class:n
                  { \__text_codepoint_from_chars:Nw #5 }
              }
              {
                \__text_map_lookahead:Nnnnnw #1 {#2} {#3#5} { }
                  \__text_map_Regional_Indicator_aux:Nnnnn
              }
              { \__text_map_loop:Nnnw #1 {#2} {#3} #5 }
          }
          { \__text_map_loop:Nnnw #1 {#2} {#3} #5 }
      }
      { \__text_map_loop:Nnnw #1 {#2} {#3#5} }
  }
\cs_new:Npn \text_map_function:nN #1#2
  {
    \__text_map_function:enN { \text_expand:n {#1} }
      { grapheme } #2
  }
\cs_new:Npn \__text_map_Prepend:Nnnn #1#2#3#4
  {
    \__text_map_output:Nn #1 {#3}
    \__text_map_lookahead:Nnnnnw #1 { grapheme } {#4} { }
      \__text_map_Prepend_aux:Nnnnn
  }
\cs_new:Npn \__text_map_Prepend_aux:Nnnnn #1#2#3#4#5
  {
    \bool_lazy_or:nnTF
      { \__text_codepoint_compare_p:nNn {#5} = { "000A } }
      { \__text_codepoint_compare_p:nNn {#5} = { "000D } }
      {
        #1 {#3}
        \__text_map_loop:Nnnw #1 { grapheme } {#5}
      }
      { \__text_map_Prepend:Nnn #1 {#3} {#5} }
  }
\cs_new:Npn \__text_map_Prepend:Nnn #1#2#3
  {
    \str_if_eq:eeTF
      { Control }
      {
        \__kernel_codepoint_to_grapheme_class:n
          { \__text_codepoint_from_chars:Nw #3 }
      }
      { \__text_map_loop:Nnnw #1 { grapheme } {#2} #3 }
      { \__text_map_loop:Nnnw #1 { grapheme } {#2#3} }
  }
\cs_new:Npn \__text_map_L:Nnnn #1#2#3#4
  {
    \__text_map_output:Nn #1 {#3}
    \__text_map_hangul:Nnnw
      #1 {#4} { L ; V ; LV ; LVT }
  }
\cs_new:Npn \__text_map_LV:Nnnn #1#2#3#4
  {
    \__text_map_output:Nn #1 {#3}
    \__text_map_hangul:Nnnw
      #1 {#4} { V ; T }
  }
\cs_new_eq:NN \__text_map_V:Nnnn \__text_map_LV:Nnnn
\cs_new:Npn \__text_map_LVT:Nnnn #1#2#3#4
  {
    \__text_map_output:Nn #1 {#3}
    \__text_map_hangul:Nnnw
      #1 {#4} { T }
  }
\cs_new_eq:NN \__text_map_T:Nnnn \__text_map_LVT:Nnnn
\cs_new:Npn \__text_map_hangul:Nnnw #1#2#3#4 \q__text_recursion_stop
  {
    \tl_if_head_is_N_type:nTF {#4}
      { \__text_map_hangul:NnnN #1 {#2} {#3} }
      {
        #1 {#2}
        \__text_map_loop:Nnnw #1 { grapheme } { }
      }
    #4 \q__text_recursion_stop
  }
\cs_new:Npn \__text_map_hangul:NnnN #1#2#3#4
  {
    \__text_if_q_recursion_tail_stop_do:Nn #4
      {
        #1 {#2}
        \text_map_break:
      }
    \token_if_cs:NTF #4
      {
        #1 {#2}
        \__text_map_loop:Nnnw #1 { grapheme } { }
      }
      {
        \__text_codepoint_process:nN
          { \__text_map_hangul:Nnnn #1 {#2} {#3} } #4
      }
  }
\exp_args_generate:n { Nnne }
\cs_new:Npn \__text_map_hangul:Nnnn #1#2#3#4
  {
    \exp_args:NNnne \__text_map_hangul_aux:Nnnnw #1 {#2} {#4}
      {
        \__kernel_codepoint_to_grapheme_class:n
          { \__text_codepoint_from_chars:Nw #4 }
      }
      #3 ; \q_recursion_tail ; \q_recursion_stop
  }
\cs_new:Npn \__text_map_hangul_aux:Nnnnw #1#2#3#4#5 ;
  {
    \quark_if_recursion_tail_stop_do:nn {#5}
      { \__text_map_loop:Nnnw #1 { grapheme } {#2} #3 }
    \__text_map_hangul:Nnnnnw #1 {#2} {#3} {#4} {#5}
  }
\cs_generate_variant:Nn \__text_map_hangul_aux:Nnnnw { Nnne }
\cs_new:Npn \__text_map_hangul:Nnnnnw #1#2#3#4#5#6 \q_recursion_stop
  {
    \str_if_eq:nnTF {#4} {#5}
      { \use:c { __text_map_hangul_ #5 :Nnn } #1 {#2} {#3} }
      { \__text_map_hangul_next:Nnnnn #1 {#2} {#3} {#4} {#6} }
  }
\cs_new:Npn \__text_map_hangul_next:Nnnnn #1#2#3#4#5
  { \__text_map_hangul_aux:Nnnnw #1 {#2} {#3} {#4} #5 \q_recursion_stop }
\cs_new:Npn \__text_map_hangul_end:nw #1#2 \q__text_recursion_stop {#1}
\cs_new:Npn \__text_map_hangul_L:Nnn #1#2#3
  {
    \__text_map_hangul:Nnnw
      #1 {#2#3} { L V { LV } { LVT } }
  }
```

```
\cs_new:Npn \__text_map_hangul_LV:Nnn #1#2#3
  {
    \__text_map_hangul:Nnnw
      #1 {#2#3} { VT }
  }
\cs_new_eq:NN \__text_map_hangul_V:Nnn \__text_map_hangul_LV:Nnn
\cs_new:Npn \__text_map_hangul_LVT:Nnn #1#2#3
  {
    \__text_map_hangul:Nnnw
      #1 {#2#3} { T }
  }
\cs_new_eq:NN \__text_map_hangul_T:Nnn \__text_map_hangul_LVT:Nnn
\cs_new:Npn \text_words_map_function:nN #1#2
  {
    \__text_map_function:enN { \text_expand:n {#1} }
      { wordbreak } #2
  }
\cs_new:Npn \__text_map_collect:Nnnnn #1#2#3#4#5
  {
    \__text_map_lookahead:Nnnnnw #1 { wordbreak } {#2} { }
      { \__text_map_collect_auxi:nnnNnnnn {#3} {#4} {#5} }
  }
\cs_new:Npn \__text_map_collect_auxi:nnnNnnnn #1#2#3#4#5#6#7#8
  {
    \exp_args:Ne \__text_map_collect_auxii:nNnnnnn
      {
        \__kernel_codepoint_to_wordbreak_class:n
          { \__text_codepoint_from_chars:Nw #8 }
      }
  #4 {#6} {#1} {#2} {#3} {#8}
  }
\cs_new:Npn \__text_map_collect_auxii:nNnnnnn #1#2#3#4#5#6#7
  {
    \str_case:neTF {#1}
      {
        \tl_map_function:eN
          {
            #4
            \str_if_eq:nnF {#4} { { WSegSpace } } { { ExtendNumLet } }
          }
          \__text_map_collect_auxiii:n
      }
      {
        \cs_if_exist_use:cF { __text_map_ #1 :Nnnn }
          { \__text_map_Other:Nnnn }
            #2 { wordbreak } { } {#3#7}
      }
      {
        \__text_map_if_ignorable:nTF {#1}
          { \__text_map_collect:Nnnnn #2 {#3#7} {#4} {#5} {#6} }
          {
            \str_case:neTF {#1}
              { \tl_map_function:nN {#5} \__text_map_collect_auxiii:n }
              {
                \__text_map_lookahead:Nnnnnw #2 { wordbreak } {#3} {#7}
                  { \__text_map_collect_auxiv:nnNnnnn {#5} {#6} }
              }
              {
                \__text_map_output:Nn #2 {#3}
                \__text_map_loop:Nnnw #2 { wordbreak } { } #7
              }
          }
      }
  }
\cs_new:Npn \__text_map_collect_auxiii:n #1
  { \exp_not:n { {#1} { } } }
\cs_new:Npn \__text_map_collect_auxiv:nnNnnnn #1#2#3#4#5#6#7
  {
    \exp_args:Ne \__text_map_collect_auxv:nNnnnnn
      {
        \__kernel_codepoint_to_wordbreak_class:n
          { \__text_codepoint_from_chars:Nw #7 }
      }
    #3 {#5} {#6} {#1} {#2} {#7}
  }
\cs_new:Npn \__text_map_collect_auxv:nNnnnnn #1#2#3#4#5#6#7
  {
    \str_case:neTF {#1}
      { \tl_map_function:nN {#6} \__text_map_collect_auxiii:n }
      { \use:c { __text_map_ #1 :Nnnn } #2 { wordbreak } { } {#3#4#7} }
      {
        \__text_map_if_ignorable:nTF {#1}
          {
            \__text_map_lookahead:Nnnnnw #2 { wordbreak } {#3} {#4#7}
              { \__text_map_collect_auxiv:nnNnnnn {#5} {#6} }
          }
          {
            \__text_map_output:Nn #2 {#3}
            \__text_map_loop:Nnnw #2 { wordbreak } { } #4#7
          }
      }
  }
```

```
\cs_new:Npn \__text_map_ALetter:Nnnn #1#2#3#4
  {
    \__text_map_output:Nn #1 {#3}
    \__text_map_collect:Nnnnn #1 {#4}
      { { ALetter } { Hebrew_Letter } { Numeric } }
      { { MidLetter } { MidNumLet } { Single_Quote } }
      { { ALetter } { Hebrew_Letter } }
  }
\cs_new:Npn \__text_map_Hebrew_Letter:Nnnn #1#2#3#4
  {
    \__text_map_output:Nn #1 {#3}
    \__text_map_collect:Nnnnn #1 {#4}
      { { ALetter } { Hebrew_Letter } { Numeric } { Single_Quote } }
      { { MidLetter } { MidNumLet } { Double_Quote } }
      { { Hebrew_Letter } }
  }
\cs_new:Npn \__text_map_Katakana:Nnnn #1#2#3#4
  {
    \__text_map_output:Nn #1 {#3}
    \__text_map_collect:Nnnnn #1 {#4} { { Katakana } } { } { }
  }
\cs_new:Npn \__text_map_Numeric:Nnnn #1#2#3#4
  {
    \__text_map_output:Nn #1 {#3}
    \__text_map_collect:Nnnnn #1 {#4}
      { { ALetter } { Hebrew_Letter } { Numeric } }
      { { MidNum } { MidNumLet } { Single_Quote } }
      { { Numeric } }
  }
\cs_new:Npn \__text_map_WSegSpace:Nnnn #1#2#3#4
  {
    \__text_map_output:Nn #1 {#3}
    \__text_map_collect:Nnnnn #1 {#4} { { WSegSpace } } { } { }
  }
\cs_new:Npn \__text_map_ExtendNumLet:Nnnn #1#2#3#4
  {
    \__text_map_output:Nn #1 {#3}
    \__text_map_lookahead:Nnnnnw #1 { wordbreak } {#4} { }
      \__text_map_ExtendNumLet_auxi:Nnnnn
  }
\cs_new:Npn \__text_map_ExtendNumLet_auxi:Nnnnn #1#2#3#4#5
  {
    \exp_args:Ne \__text_map_ExtendNumLet_auxii:nNnn
      {
        \__kernel_codepoint_to_wordbreak_class:n
          { \__text_codepoint_from_chars:Nw #5 }
      }
      #1 {#3} {#5}
  }
\cs_new:Npn \__text_map_ExtendNumLet_auxii:nNnn #1#2#3#4
  {
    \str_case:nnTF {#1}
      {
        { ALetter }       { }
        { Hebrew_Letter } { }
        { Numeric }       { }
        { Katakana }      { }
        { ExtendNumLet }  { }
      }
      {
        \cs_if_exist_use:cF { __text_map_ #1 :Nnnn } % TEMP?
          { \__text_map_Other:Nnnn }
            #2 { wordbreak } { } {#3#4}
      }
      {
        \__text_map_if_ignorable:nTF {#1}
          {
            \__text_map_lookahead:Nnnnnw #2 { wordbreak } {#3#4} { }
              \__text_map_ExtendNumLet_auxi:Nnnnn
          }
          {
            \__text_map_output:Nn #2 {#3}
            \__text_map_loop:Nnnw #2 { wordbreak } { } #4
          }
      }
  }
\cs_new_protected:Npn \text_map_inline:nn #1#2
  {
    \int_gincr:N \g__kernel_prg_map_int
    \cs_gset_protected:cpn
      { __text_map_ \int_use:N \g__kernel_prg_map_int :w } ##1 {#2}
    \exp_args:Nnc \text_map_function:nN {#1}
      { __text_map_ \int_use:N \g__kernel_prg_map_int :w }
    \prg_break_point:Nn \text_map_break:
      { \int_gdecr:N \g__kernel_prg_map_int }
  }
\cs_new_protected:Npn \text_words_map_inline:nn #1#2
  {
    \int_gincr:N \g__kernel_prg_map_int
    \cs_gset_protected:cpn
      { __text_map_ \int_use:N \g__kernel_prg_map_int :w } ##1 {#2}
    \exp_args:Nnc \text_words_map_function:nN {#1}
```

```
      { __text_map_ \int_use:N \g__kernel_prg_map_int :w }
    \prg_break_point:Nn \text_map_break:
      { \int_gdecr:N \g__kernel_prg_map_int }
  }
%% File: l3text-purify.dtx
\__kernel_quark_new_test:N \__text_if_recursion_tail_stop:N
\cs_new:Npn \text_purify:n #1
  {
    \__kernel_exp_not:w \exp_after:wN
      {
        \exp:w
        \exp_args:Ne \__text_purify:n
          { \text_expand:n {#1} }
      }
  }
\cs_new:Npn \__text_purify:n #1
  {
    \group_align_safe_begin:
    \__text_purify_loop:w #1
      \q__text_recursion_tail \q__text_recursion_stop
    \__text_purify_result:n { }
  }
\cs_new:Npn \__text_purify_store:n #1
  { \__text_purify_store:nw {#1} }
\cs_new:Npn \__text_purify_store:nw #1#2 \__text_purify_result:n #3
  { #2 \__text_purify_result:n { #3 #1 } }
\cs_new:Npn \__text_purify_end:w #1 \__text_purify_result:n #2
  {
    \group_align_safe_end:
    \exp_end:
    #2
  }
\cs_new:Npn \__text_purify_loop:w #1 \q__text_recursion_stop
  {
    \tl_if_head_is_N_type:nTF {#1}
      { \__text_purify_N_type:N }
      {
        \tl_if_head_is_group:nTF {#1}
          { \__text_purify_group:n }
          { \__text_purify_space:w }
      }
    #1 \q__text_recursion_stop
  }
\cs_new:Npn \__text_purify_group:n #1 { \__text_purify_loop:w #1 }
\exp_last_unbraced:NNo \cs_new:Npn \__text_purify_space:w \c_space_tl
  {
    \__text_purify_store:n { ~ }
    \__text_purify_loop:w
  }
\cs_new:Npn \__text_purify_N_type:N #1
  {
    \__text_if_q_recursion_tail_stop_do:Nn #1 { \__text_purify_end:w }
    \__text_purify_N_type_aux:N #1
  }
\cs_new:Npn \__text_purify_N_type_aux:N #1
  {
    \exp_after:wN \__text_purify_math_search:NNN
      \exp_after:wN #1 \l_text_math_delims_tl
      \q__text_recursion_tail ?
      \q__text_recursion_stop
  }
\cs_new:Npn \__text_purify_math_search:NNN #1#2#3
  {
    \__text_if_q_recursion_tail_stop_do:Nn #2
      { \__text_purify_math_cmd:N #1 }
    \token_if_eq_meaning:NNTF #1 #2
      {
        \__text_use_i_delimit_by_q_recursion_stop:nw
          { \__text_purify_math_start:NNw #2 #3 }
      }
      { \__text_purify_math_search:NNN #1 }
  }
\cs_new:Npn \__text_purify_math_start:NNw #1#2#3 \q__text_recursion_stop
  {
    \__text_purify_math_loop:NNw #1#2#3 \q__text_recursion_stop
      \__text_purify_math_result:n { }
  }
\cs_new:Npn \__text_purify_math_store:n #1
  { \__text_purify_math_store:nw {#1} }
\cs_new:Npn \__text_purify_math_store:nw #1#2 \__text_purify_math_result:n #3
  { #2 \__text_purify_math_result:n { #3 #1 } }
\cs_new:Npn \__text_purify_math_end:w #1 \__text_purify_math_result:n #2
  {
    \__text_purify_store:n { $ #2 $ }
    \__text_purify_loop:w #1
  }
\cs_new:Npn \__text_purify_math_stop:Nw #1 \__text_purify_math_result:n #2
  {
    \__text_purify_store:n {#1#2}
    \__text_purify_end:w
  }
\cs_new:Npn \__text_purify_math_loop:NNw #1#2#3 \q__text_recursion_stop
```

```
  {
    \tl_if_head_is_N_type:nTF {#3}
      { \__text_purify_math_N_type:NNN }
      {
        \tl_if_head_is_group:nTF {#3}
          { \__text_purify_math_group:NNn }
          { \__text_purify_math_space:NNw }
      }
        #1#2#3 \q__text_recursion_stop
  }
\cs_new:Npn \__text_purify_math_N_type:NNN #1#2#3
  {
    \__text_if_q_recursion_tail_stop_do:Nn #3
      { \__text_purify_math_stop:Nw #1 }
    \token_if_eq_meaning:NNTF #3 #2
      { \__text_purify_math_end:w }
      {
        \__text_purify_math_store:n {#3}
        \__text_purify_math_loop:NNw #1#2
      }
  }
\cs_new:Npn \__text_purify_math_group:NNn #1#2#3
  {
    \__text_purify_math_store:n { {#3} }
    \__text_purify_math_loop:NNw #1#2
  }
\exp_after:wN \cs_new:Npn \exp_after:wN \__text_purify_math_space:NNw
  \exp_after:wN # \exp_after:wN 1
  \exp_after:wN # \exp_after:wN 2 \c_space_tl
  {
    \__text_purify_math_store:n { ~ }
    \__text_purify_math_loop:NNw #1#2
  }
\cs_new:Npn \__text_purify_math_cmd:N #1
  {
    \exp_after:wN \__text_purify_math_cmd:NN \exp_after:wN #1
      \l_text_math_arg_tl \q__text_recursion_tail \q__text_recursion_stop
  }
\cs_new:Npn \__text_purify_math_cmd:NN #1#2
  {
    \__text_if_q_recursion_tail_stop_do:Nn #2
      { \__text_purify_replace:N #1 }
    \cs_if_eq:NNTF #2 #1
      {
        \__text_use_i_delimit_by_q_recursion_stop:nw
          { \__text_purify_math_cmd:n }
      }
      { \__text_purify_math_cmd:NN #1 }
  }
\cs_new:Npn \__text_purify_math_cmd:n #1
  { \__text_purify_math_end:w \__text_purify_math_result:n {#1} }
\cs_new:Npn \__text_purify_replace:N #1
  {
    \bool_lazy_and:nnTF
      { \cs_if_exist_p:c { l__text_purify_ \token_to_str:N #1 _tl } }
      {
        \bool_lazy_or_p:nn
          { \token_if_cs_p:N #1 }
          { \token_if_active_p:N #1 }
      }
      {
        \exp_args:Nv \__text_purify_replace_auxi:n
          { l__text_purify_ \token_to_str:N #1 _tl }
      }
      {
        \bool_lazy_or:nnTF
          { \token_if_group_begin_p:N #1 }
          { \token_if_group_end_p:N #1 }
          { \__text_purify_loop:w }
          {
            \exp_args:Ne \__text_purify_replace_auxii:n
              { \__text_token_to_explicit:N #1 }
          }
      }
  }
\cs_new:Npn \__text_purify_replace_auxi:n #1 { \__text_purify_loop:w #1 }
\cs_new:Npn \__text_purify_replace_auxii:n #1
  {
    \token_if_cs:NTF #1
      { \__text_purify_expand:N #1 }
      {
        \__text_purify_store:n {#1}
        \__text_purify_loop:w
      }
  }
\cs_new:Npn \__text_purify_expand:N #1
  {
    \str_if_eq:nnTF {#1} { \protect }
      { \__text_purify_protect:N }
      { \__text_purify_encoding:N #1 }
  }
\cs_new:Npn \__text_purify_protect:N #1
```

```
  {
    \__text_if_q_recursion_tail_stop_do:Nn #1 { \__text_purify_end:w }
    \__text_purify_loop:w
  }
\cs_new:Npn \__text_purify_encoding:N #1
  {
    \bool_lazy_or:nnTF
      { \cs_if_eq_p:NN #1 \@current@cmd }
      { \cs_if_eq_p:NN #1 \@changed@cmd }
      { \__text_purify_encoding_escape:NN }
      {
        \__text_if_expandable:NTF #1
          { \exp_after:wN \__text_purify_loop:w #1 }
          { \__text_purify_loop:w }
      }
  }
\cs_new:Npn \__text_purify_encoding_escape:NN #1#2
  {
    \__text_purify_store:n {#1}
    \__text_purify_loop:w
  }
\cs_new_protected:Npn \text_declare_purify_equivalent:Nn #1#2
  {
    \tl_clear_new:c { l__text_purify_ \token_to_str:N #1 _tl }
    \tl_set:cn { l__text_purify_ \token_to_str:N #1 _tl } {#2}
  }
\cs_generate_variant:Nn \text_declare_purify_equivalent:Nn { Ne }
\tl_map_inline:nn
  {
    \fontencoding
    \fontfamily
    \fontseries
    \fontshape
  }
  { \text_declare_purify_equivalent:Nn #1 { \use_none:n } }
\text_declare_purify_equivalent:Nn \fontsize { \use_none:nn }
\text_declare_purify_equivalent:Nn \selectfont { }
\text_declare_purify_equivalent:Nn \usefont { \use_none:nnnn }
\exp_args:Nc \text_declare_purify_equivalent:Nn
  { @protected@testopt } { \use_none:nnn }
\text_declare_purify_equivalent:Nn \begin { \use:c }
\text_declare_purify_equivalent:Nn \end { \__text_end_env:n }
\cs_new:Npn \__text_end_env:n #1 { \cs:w end #1 \cs_end: }
\text_declare_purify_equivalent:Nn \\ { }
\tl_map_inline:nn
  { \{ \} \# \$ \% \_ }
  { \text_declare_purify_equivalent:Ne #1 { \cs_to_str:N #1 } }
\text_declare_purify_equivalent:Nn \label { \use_none:n }
\group_begin:
\char_set_catcode_active:N \~
\use:n
  {
    \group_end:
    \text_declare_purify_equivalent:Ne ~ { \c_space_tl }
  }
\text_declare_purify_equivalent:Nn \nobreakspace { ~ }
\text_declare_purify_equivalent:Nn \␣ { ~ }
\text_declare_purify_equivalent:Nn \, { ~ }
\cs_set_protected:Npn \__text_loop:Nn #1#2
  {
    \quark_if_recursion_tail_stop:N #1
    \text_declare_purify_equivalent:Ne #1
      {
        \codepoint_generate:nn {"#2}
          { \char_value_catcode:n {"#2} }
      }
    \__text_loop:Nn
  }
\__text_loop:Nn
  \AA { 00C5 }
  \AE { 00C6 }
  \DH { 00D0 }
  \DJ { 0110 }
  \IJ { 0132 }
  \L  { 0141 }
  \NG { 014A }
  \O  { 00D8 }
  \OE { 0152 }
  \TH { 00DE }
  \aa { 00E5 }
  \ae { 00E6 }
  \dh { 00F0 }
  \dj { 0111 }
  \i  { 0131 }
  \j  { 0237 }
  \ij { 0132 }
  \l  { 0142 }
  \ng { 014B }
  \o  { 00F8 }
  \oe { 0153 }
  \ss { 00DF }
  \th { 00FE }
  \q_recursion_tail ?
  \q_recursion_stop
\text_declare_purify_equivalent:Nn \SS { SS }
\cs_new:Npn \__text_purify_accent:NN #1#2
  {
    \cs_if_exist:cTF
      { c__text_purify_ \token_to_str:N #1 _ \token_to_str:N #2 _tl }
      {
        \exp_not:v
          { c__text_purify_ \token_to_str:N #1 _ \token_to_str:N #2 _tl }
      }
      {
        \exp_not:n {#2}
        \exp_not:v { c__text_purify_ \token_to_str:N #1 _tl }
      }
  }
\tl_map_inline:nn { \` \' \^ \~ \= \u \. \" \r \H \v \d \c \k \b \t }
  { \text_declare_purify_equivalent:Nn #1 { \__text_purify_accent:NN #1 } }
\group_begin:
  \cs_set_protected:Npn \__text_loop:Nn #1#2
    {
      \quark_if_recursion_tail_stop:N #1
      \tl_const:ce { c__text_purify_ \token_to_str:N #1 _tl }
        { \codepoint_generate:nn {"#2} { \char_value_catcode:n { "#2 } } }
      \__text_loop:Nn
    }
  \__text_loop:Nn
    \` { 0300 }
    \' { 0301 }
    \^ { 0302 }
    \~ { 0303 }
    \= { 0304 }
    \u { 0306 }
    \. { 0307 }
    \" { 0308 }
    \r { 030A }
    \H { 030B }
    \v { 030C }
    \d { 0323 }
    \c { 0327 }
    \k { 0328 }
    \b { 0331 }
    \t { 0361 }
    \q_recursion_tail { }
    \q_recursion_stop
  \cs_set_protected:Npn \__text_loop:NNn #1#2#3
```

```
    {
      \quark_if_recursion_tail_stop:N #1
      \tl_const:ce
        { c__text_purify_ \token_to_str:N #1 _ \token_to_str:N #2 _tl }
        { \codepoint_generate:nn {"#3} { \char_value_catcode:n { "#3 } } }
      \__text_loop:NNn
    }
  \__text_loop:NNn
    \` A   { 00C0 }
    \' A   { 00C1 }
    \^ A   { 00C2 }
    \~ A   { 00C3 }
    \" A   { 00C4 }
    \r A   { 00C5 }
    \c C   { 00C7 }
    \` E   { 00C8 }
    \' E   { 00C9 }
    \^ E   { 00CA }
    \" E   { 00CB }
    \` I   { 00CC }
    \' I   { 00CD }
    \^ I   { 00CE }
    \" I   { 00CF }
    \~ N   { 00D1 }
    \` O   { 00D2 }
    \' O   { 00D3 }
    \^ O   { 00D4 }
    \~ O   { 00D5 }
    \" O   { 00D6 }
    \` U   { 00D9 }
    \' U   { 00DA }
    \^ U   { 00DB }
    \" U   { 00DC }
    \' Y   { 00DD }
    \` a   { 00E0 }
    \' a   { 00E1 }
    \^ a   { 00E2 }
    \~ a   { 00E3 }
    \" a   { 00E4 }
    \r a   { 00E5 }
    \c c   { 00E7 }
    \` e   { 00E8 }
    \' e   { 00E9 }
    \^ e   { 00EA }
    \" e   { 00EB }
    \` i   { 00EC }
    \` \i  { 00EC }
    \' i   { 00ED }
    \' \i  { 00ED }
    \^ i   { 00EE }
    \^ \i  { 00EE }
    \" i   { 00EF }
    \" \i  { 00EF }
    \~ n   { 00F1 }
    \` o   { 00F2 }
    \' o   { 00F3 }
    \^ o   { 00F4 }
    \~ o   { 00F5 }
    \" o   { 00F6 }
    \` u   { 00F9 }
    \' u   { 00FA }
    \^ u   { 00FB }
    \" u   { 00FC }
    \' y   { 00FD }
    \" y   { 00FF }
    \= A   { 0100 }
    \= a   { 0101 }
    \u A   { 0102 }
    \u a   { 0103 }
    \k A   { 0104 }
    \k a   { 0105 }
    \' C   { 0106 }
    \' c   { 0107 }
    \^ C   { 0108 }
    \^ c   { 0109 }
    \. C   { 010A }
    \. c   { 010B }
    \v C   { 010C }
    \v c   { 010D }
    \v D   { 010E }
    \v d   { 010F }
    \= E   { 0112 }
    \= e   { 0113 }
    \u E   { 0114 }
    \u e   { 0115 }
    \. E   { 0116 }
    \. e   { 0117 }
    \k E   { 0118 }
    \k e   { 0119 }
    \v E   { 011A }
    \v e   { 011B }
    \^ G   { 011C }
    \^ g   { 011D }
    \u G   { 011E }
    \u g   { 011F }
    \. G   { 0120 }
    \. g   { 0121 }
    \c G   { 0122 }
    \c g   { 0123 }
    \^ H   { 0124 }
    \^ h   { 0125 }
    \~ I   { 0128 }
    \~ i   { 0129 }
    \~ \i  { 0129 }
    \= I   { 012A }
    \= i   { 012B }
    \= \i  { 012B }
    \u I   { 012C }
    \u i   { 012D }
    \u \i  { 012D }
    \k I   { 012E }
    \k i   { 012F }
    \k \i  { 012F }
    \. I   { 0130 }
    \^ J   { 0134 }
    \^ j   { 0135 }
    \^ \j  { 0135 }
    \c K   { 0136 }
    \c k   { 0137 }
    \' L   { 0139 }
    \' l   { 013A }
    \c L   { 013B }
    \c l   { 013C }
    \v L   { 013D }
    \v l   { 013E }
    \. L   { 013F }
    \. l   { 0140 }
    \' N   { 0143 }
    \' n   { 0144 }
    \c N   { 0145 }
    \c n   { 0146 }
    \v N   { 0147 }
    \v n   { 0148 }
    \= O   { 014C }
    \= o   { 014D }
    \u O   { 014E }
    \u o   { 014F }
    \H O   { 0150 }
    \H o  { 0151 }
    \' R  { 0154 }
    \' r  { 0155 }
    \c R  { 0156 }
    \c r  { 0157 }
    \v R  { 0158 }
    \v r  { 0159 }
    \' S  { 015A }
    \' s  { 015B }
    \^ S  { 015C }
    \^ s  { 015D }
    \c S  { 015E }
    \c s  { 015F }
    \v S  { 0160 }
    \v s  { 0161 }
    \c T  { 0162 }
    \c t  { 0163 }
    \v T  { 0164 }
    \v t  { 0165 }
    \~ U  { 0168 }
    \~ u  { 0169 }
    \= U  { 016A }
    \= u  { 016B }
    \u U  { 016C }
    \u u  { 016D }
    \r U  { 016E }
    \r u  { 016F }
    \H U  { 0170 }
    \H u  { 0171 }
    \k U  { 0172 }
    \k u  { 0173 }
    \^ W  { 0174 }
    \^ w  { 0175 }
    \^ Y  { 0176 }
    \^ y  { 0177 }
    \" Y  { 0178 }
    \' Z  { 0179 }
    \' z  { 017A }
    \. Z  { 017B }
    \. z  { 017C }
    \v Z  { 017D }
    \v z  { 017E }
    \v A  { 01CD }
    \v a  { 01CE }
    \v I  { 01CF }
    \v \i { 01D0 }
    \v i   { 01D0 }
    \v O   { 01D1 }
    \v o   { 01D2 }
    \v U   { 01D3 }
    \v u   { 01D4 }
    \v G   { 01E6 }
    \v g   { 01E7 }
    \v K   { 01E8 }
    \v k   { 01E9 }
    \k O   { 01EA }
    \k o   { 01EB }
    \v \j  { 01F0 }
    \v j   { 01F0 }
    \' G   { 01F4 }
    \' g   { 01F5 }
    \` N   { 01F8 }
    \` n   { 01F9 }
    \' \AE { 01FC }
    \' \ae { 01FD }
    \' \O  { 01FE }
    \' \o  { 01FF }
    \v H   { 021E }
    \v h   { 021F }
    \. A   { 0226 }
    \. a   { 0227 }
    \c E   { 0228 }
    \c e   { 0229 }
    \. O   { 022E }
    \. o   { 022F }
    \= Y   { 0232 }
    \= y   { 0233 }
    \q_recursion_tail ? { }
    \q_recursion_stop
\group_end:
%% File: l3legacy.dtx
\prg_new_conditional:Npnn \legacy_if:n #1 { p , T , F , TF }
  {
    \exp_after:wN \reverse_if:N
      \cs:w if#1 \cs_end:
      \prg_return_false:
    \else:
      \prg_return_true:
    \fi:
  }
\cs_new_protected:Npn \legacy_if_set_true:n #1
  { \cs_set_eq:cN { if#1 } \if_true: }
```

```
\cs_new_protected:Npn \legacy_if_set_false:n #1
  { \cs_set_eq:cN { if#1 } \if_false: }
\cs_new_protected:Npn \legacy_if_gset_true:n #1
  { \cs_gset_eq:cN { if#1 } \if_true: }
\cs_new_protected:Npn \legacy_if_gset_false:n #1
  { \cs_gset_eq:cN { if#1 } \if_false: }
\cs_new_protected:Npn \legacy_if_set:nn #1#2
  {
    \bool_if:nTF {#2} \legacy_if_set_true:n \legacy_if_set_false:n
    {#1}
  }
\cs_new_protected:Npn \legacy_if_gset:nn #1#2
  {
    \bool_if:nTF {#2} \legacy_if_gset_true:n \legacy_if_gset_false:n
    {#1}
  }
%% File: l3deprecation.dtx
\cs_new_protected:Npn \__kernel_patch_deprecation:nnNNpn #1#2#3#4#5#
  { \__deprecation_patch_aux:nnNNnn {#1} {#2} #3 #4 {#5} }
\cs_new_protected:Npn \__deprecation_patch_aux:nnNNnn #1#2#3#4#5#6
  {
    \__kernel_deprecation_code:nn
      {
        \tex_let:D #4 \scan_stop:
        \__kernel_deprecation_error:Nnn #4 {#2} {#1}
      }
      { \tex_let:D #4 \scan_stop: }
    \cs_if_eq:NNTF #3 \cs_gset_protected:Npn
      { \__deprecation_warn_once:nnNnn {#1} {#2} #4 {#5} {#6} }
      { \__deprecation_patch_aux:Nn #3 { #4 #5 {#6} } }
  }
\cs_new_protected:Npn \__deprecation_warn_once:nnNnn #1#2#3#4#5
  {
    \cs_gset_protected:Npe #3
      {
        \__kernel_if_debug:TF
          {
            \exp_not:N \msg_warning:nneee
              { deprecation } { deprecated-command }
              {#1}
              { \token_to_str:N #3 }
              { \tl_to_str:n {#2} }
          }
          { }
        \exp_not:n { \cs_gset_protected:Npn #3 #4 {#5} }
        \exp_not:N #3
      }
    \__kernel_deprecation_code:nn { }
      { \cs_set_protected:Npn #3 #4 {#5} }
  }
\cs_new_protected:Npn \__deprecation_patch_aux:Nn #1#2
  {
    #1 #2
    \cs_if_eq:NNTF #1 \cs_gset_protected:Npn
      { \__kernel_deprecation_code:nn { } { \cs_set_protected:Npn #2 } }
      { \__kernel_deprecation_code:nn { } { \cs_set:Npn #2 } }
  }
\cs_new_protected:Npn \__kernel_deprecation_error:Nnn #1#2#3
  {
    \tex_protected:D \tex_outer:D \tex_edef:D #1
      {
        \exp_not:N \msg_expandable_error:nnnnn
          { deprecation } { deprecated-command }
          { \tl_to_str:n {#3} } { \token_to_str:N #1 } { \tl_to_str:n {#2} }
        \exp_not:N \msg_error:nneee
          { deprecation } { deprecated-command }
          { \tl_to_str:n {#3} } { \token_to_str:N #1 } { \tl_to_str:n {#2} }
      }
  }
\msg_new:nnn { deprecation } { deprecated-command }
  {
    \tl_if_blank:nF {#3} { Use~ \tl_trim_spaces:n {#3} ~not~ }
    #2~deprecated~on~#1.
  }
\cs_new:Npn \cs_argument_spec:N { \cs_parameter_spec:N }
\cs_new_protected:Npn \iow_shipout_x:Nn { \iow_shipout_e:Nn }
\cs_generate_variant:Nn \iow_shipout_x:Nn { Nx , c, cx }
\cs_new_protected:cpn { \c__keys_props_root_str .str_set_x:N } #1
  { \__keys_variable_set:NnnN #1 { str } { } x }
\cs_new_protected:cpn { \c__keys_props_root_str .str_set_x:c } #1
  { \__keys_variable_set:cnnN {#1} { str } { } x }
\cs_new_protected:cpn { \c__keys_props_root_str .str_gset_x:N } #1
  { \__keys_variable_set:NnnN #1 { str } { g } x }
\cs_new_protected:cpn { \c__keys_props_root_str .str_gset_x:c } #1
  { \__keys_variable_set:cnnN {#1} { str } { g } x }
\cs_new_protected:cpn { \c__keys_props_root_str .tl_set_x:N } #1
  { \__keys_variable_set:NnnN #1 { tl } { } x }
\cs_new_protected:cpn { \c__keys_props_root_str .tl_set_x:c } #1
  { \__keys_variable_set:cnnN {#1} { tl } { } x }
\cs_new_protected:cpn { \c__keys_props_root_str .tl_gset_x:N } #1
  { \__keys_variable_set:NnnN #1 { tl } { g } x }
\cs_new_protected:cpn { \c__keys_props_root_str .tl_gset_x:c } #1
```

```
  { \__keys_variable_set:cnnN {#1} { tl } { g } x }
\cs_new_protected:Npn \keys_set_filter:nnn { \keys_set_exclude_groups:nnn }
\cs_generate_variant:Nn \keys_set_filter:nnn { nnV , nnv , nno }
\cs_new_protected:Npn  \keys_set_filter:nnnN { \keys_set_exclude_groups:nnnN }
\cs_generate_variant:Nn \keys_set_filter:nnnN { nnV , nnv , nno }
\cs_new_protected:Npn  \keys_set_filter:nnnnN { \keys_set_exclude_groups:nnnnN }
\cs_generate_variant:Nn \keys_set_filter:nnnnN { nnV , nnv , nno }
\__kernel_patch_deprecation:nnNNpn { 2024-02-13 } { \msg_set:nnnn }
\cs_new_protected:Npn \msg_gset:nnnn { \msg_set:nnnn }
\__kernel_patch_deprecation:nnNNpn { 2024-02-13 } { \msg_set:nnn }
\cs_new_protected:Npn \msg_gset:nnn { \msg_set:nnn }
\prop_new:N \g__pdf_object_prop
\__kernel_patch_deprecation:nnNNpn { 2022-08-30 } { [\pdf_object_new:n] }
\cs_new_protected:Npn \pdf_object_new:nn #1#2
  {
    \prop_gput:Nnn \g__pdf_object_prop {#1} {#2}
    \pdf_object_new:n {#1}
  }
\__kernel_patch_deprecation:nnNNpn { 2022-08-30 } { [\pdf_object_write:n] }
\cs_new_protected:Npn \pdf_object_write:nn #1#2
  {
    \exp_args:Nee \__pdf_backend_object_write:nnn
      { \__pdf_object_retrieve:n {#1} }
      { \prop_item:Nn \g__pdf_object_prop {#1} } {#2}
    \bool_gset_true:N \g__pdf_init_bool
  }
\cs_generate_variant:Nn \pdf_object_write:nn { nx }
\__kernel_patch_deprecation:nnNNpn { 2023-05-03 } { \bool_case:n }
\cs_new:Npn \bool_case_true:n { \bool_case:n }
\__kernel_patch_deprecation:nnNNpn { 2023-05-03 } { \bool_case:nT }
\cs_new:Npn \bool_case_true:nT { \bool_case:nT }
\__kernel_patch_deprecation:nnNNpn { 2023-05-03 } { \bool_case:nF }
\cs_new:Npn \bool_case_true:nF { \bool_case:nF }
\__kernel_patch_deprecation:nnNNpn { 2023-05-03 } { \bool_case:nTF }
\cs_new:Npn \bool_case_true:nTF { \bool_case:nTF }
\__kernel_patch_deprecation:nnNNpn { 2020-01-03 } { \str_lowercase:n }
\cs_new:Npn \str_lower_case:n { \str_lowercase:n }
\__kernel_patch_deprecation:nnNNpn { 2020-01-03 } { \str_lowercase:f }
\cs_new:Npn \str_lower_case:f { \str_lowercase:f }
\__kernel_patch_deprecation:nnNNpn { 2020-01-03 } { \str_uppercase:n }
\cs_new:Npn \str_upper_case:n { \str_uppercase:n }
\__kernel_patch_deprecation:nnNNpn { 2020-01-03 } { \str_uppercase:f }
\cs_new:Npn \str_upper_case:f { \str_uppercase:f }
\__kernel_patch_deprecation:nnNNpn { 2020-01-03 } { \str_casefold:n }
\cs_new:Npn \str_fold_case:n { \str_casefold:n }
\__kernel_patch_deprecation:nnNNpn { 2020-01-03 } { \str_casefold:V }
```

```
\cs_new:Npn \str_fold_case:V { \str_casefold:V }
\__kernel_patch_deprecation:nnNNpn { 2020-10-17 } { \str_casefold:n }
\cs_new:Npn \str_foldcase:n { \str_casefold:n }
\__kernel_patch_deprecation:nnNNpn { 2022-10-17 } { \str_casefold:V }
\cs_new:Npn \str_foldcase:V { \str_casefold:V }
\__kernel_patch_deprecation:nnNNpn { 2020-08-20 } { }
\cs_new_protected:Npn \str_declare_eight_bit_encoding:nnn #1
  { \__str_declare_eight_bit_encoding:nnnn {#1} { 1114112 } }
\__kernel_patch_deprecation:nnNNpn { 2020-06-18 } { \seq_map_indexed_inline:Nn }
\cs_new_protected:Npn \seq_indexed_map_inline:Nn { \seq_map_indexed_inline:Nn }
\__kernel_patch_deprecation:nnNNpn { 2020-06-18 } { \seq_map_indexed_function:NN }
\cs_new:Npn \seq_indexed_map_function:NN { \seq_map_indexed_function:NN }
\__kernel_patch_deprecation:nnNNpn { 2023-05-10 } { \seq_map_pairwise_function:NNN }
\cs_new:Npn \seq_mapthread_function:NNN { \seq_map_pairwise_function:NNN }
\__kernel_patch_deprecation:nnNNpn { 2023-10-26 } { \seq_set_map_e:NNn }
\cs_new_protected:Npn \seq_set_map_x:NNn { \seq_set_map_e:NNn }
\__kernel_patch_deprecation:nnNNpn { 2023-10-26 } { \seq_gset_map_e:NNn }
\cs_new_protected:Npn \seq_gset_map_x:NNn { \seq_gset_map_e:NNn }
\__kernel_patch_deprecation:nnNNpn { 2021-01-11 } { (no~longer~required) }
\cs_new_protected:Npn \sys_load_deprecation: { }
\__kernel_patch_deprecation:nnNNpn { 2025-03-26 } { (no~longer~required) }
\cs_new:Npn \sys_if_timer_exist:T #1 {#1}
\__kernel_patch_deprecation:nnNNpn { 2025-03-26 } { (no~longer~required) }
\cs_new:Npn \sys_if_timer_exist:F #1 { }
\__kernel_patch_deprecation:nnNNpn { 2025-03-26 } { (no~longer~required) }
\cs_new:Npn \sys_if_timer_exist:TF #1#2 {#1}
\__kernel_patch_deprecation:nnNNpn { 2025-03-26 } { (no~longer~required) }
\cs_new:Npn \sys_if_timer_exist_p: { \c_true_bool }
\__kernel_patch_deprecation:nnNNpn { 2023-07-08 } { \text_titlecase_first:n }
\cs_new:Npn \text_titlecase:n #1
  { \text_titlecase_first:n { \text_lowercase:n {#1} } }
\__kernel_patch_deprecation:nnNNpn { 2023-07-08 } { \text_titlecase_first:nn }
\cs_new:Npn \text_titlecase:nn #1#2
  { \text_titlecase_first:nn {#1} { \text_lowercase:n {#2} } }
\__kernel_patch_deprecation:nnNNpn { 2020-01-03 } { \text_lowercase:n }
\cs_new:Npn \tl_lower_case:n #1
  { \text_lowercase:n {#1} }
\__kernel_patch_deprecation:nnNNpn { 2020-01-03 } { \text_lowercase:nn }
\cs_new:Npn \tl_lower_case:nn #1#2
  { \text_lowercase:nn {#1} {#2} }
\__kernel_patch_deprecation:nnNNpn { 2020-01-03 } { \text_uppercase:n }
\cs_new:Npn \tl_upper_case:n #1
  { \text_uppercase:n {#1} }
\__kernel_patch_deprecation:nnNNpn { 2020-01-03 } { \text_uppercase:nn }
\cs_new:Npn \tl_upper_case:nn #1#2
  { \text_uppercase:nn {#1} {#2} }
```

```
\__kernel_patch_deprecation:nnNNpn { 2020-01-03 } { \text_titlecase_first:n }
\cs_new:Npn \tl_mixed_case:n #1
  { \text_titlecase_first:n { \text_lowercase:n {#1} } }
\__kernel_patch_deprecation:nnNNpn { 2020-01-03 } { \text_titlecase_first:nn }
\cs_new:Npn \tl_mixed_case:nn #1#2
  { \text_titlecase_first:nn {#1} { \text_lowercase:n {#2} } }
\__kernel_patch_deprecation:nnNNpn { 2022-05-23 } { \token_case_meaning:Nn }
\cs_new:Npn \tl_case:Nn { \token_case_meaning:Nn }
\__kernel_patch_deprecation:nnNNpn { 2022-05-23 } { \token_case_meaning:NnT }
\cs_new:Npn \tl_case:NnT { \token_case_meaning:NnT }
\__kernel_patch_deprecation:nnNNpn { 2022-05-23 } { \token_case_meaning:NnF }
\cs_new:Npn \tl_case:NnF { \token_case_meaning:NnF }
\__kernel_patch_deprecation:nnNNpn { 2022-05-23 } { \token_case_meaning:NnTF }
\cs_new:Npn \tl_case:NnTF { \token_case_meaning:NnTF }
\cs_generate_variant:Nn \tl_case:Nn   { c }
\prg_generate_conditional_variant:Nnn \tl_case:Nn
  { c } { T , F , TF }
\__kernel_patch_deprecation:nnNNpn { 2023-10-18 } { \tl_build_begin:N }
\cs_new_protected:Npn \tl_build_clear:N { \tl_build_begin:N }
\__kernel_patch_deprecation:nnNNpn { 2023-10-18 } { \tl_build_gbegin:N }
\cs_new_protected:Npn \tl_build_gclear:N { \tl_build_gbegin:N }
\__kernel_patch_deprecation:nnNNpn { 2023-10-25 } { \tl_build_get_intermediate:NN }
\cs_new_protected:Npn \tl_build_get:NN { \tl_build_get_intermediate:NN }
\__kernel_patch_deprecation:nnNNpn { 2022-10-09 } { [ \codepoint_generate:nn ] }
\cs_new:Npn \char_to_utfviii_bytes:n { \__kernel_codepoint_to_bytes:n }
\__kernel_patch_deprecation:nnNNpn { 2022-10-09 } { \codepoint_to_nfd:n }
\cs_new:Npn \char_to_nfd:N #1 { \codepoint_to_nfd:n {`#1} }
\__kernel_patch_deprecation:nnNNpn { 2022-10-09 } { \codepoint_to_nfd:n }
\cs_new:Npn \char_to_nfd:n { \codepoint_to_nfd:n }
\__kernel_patch_deprecation:nnNNpn { 2020-01-03 } { \text_lowercase:n }
\cs_new:Npn \char_lower_case:N { \text_lowercase:n }
\__kernel_patch_deprecation:nnNNpn { 2020-01-03 } { \text_uppercase:n }
\cs_new:Npn \char_upper_case:N { \text_uppercase:n }
\__kernel_patch_deprecation:nnNNpn { 2020-01-03 } { \text_titlecase_first:n }
\cs_new:Npn \char_mixed_case:N { \text_titlecase_first:n }
\__kernel_patch_deprecation:nnNNpn { 2020-01-03 } { \str_casefold:n }
\cs_new:Npn \char_fold_case:N { \str_casefold:n }
\__kernel_patch_deprecation:nnNNpn { 2020-01-03 } { \str_lowercase:n }
\cs_new:Npn \char_str_lower_case:N { \str_lowercase:n }
\__kernel_patch_deprecation:nnNNpn { 2020-01-03 } { \str_uppercase:n }
\cs_new:Npn \char_str_upper_case:N { \str_uppercase:n }
\__kernel_patch_deprecation:nnNNpn { 2020-01-03 } { \str_titlecase:n }
\cs_new:Npn \char_str_mixed_case:N { \str_titlecase:n }
\__kernel_patch_deprecation:nnNNpn { 2020-01-03 } { \str_casefold:n }
\cs_new:Npn \char_str_fold_case:N { \str_casefold:n }
\__kernel_patch_deprecation:nnNNpn { 2022-10-17 } { \text_lowercase:n }
```

```
\cs_new:Npn \char_lowercase:N { \text_lowercase:n }
\__kernel_patch_deprecation:nnNNpn { 2022-10-17 } { \text_uppercase:n }
\cs_new:Npn \char_uppercase:N { \text_uppercase:n }
\__kernel_patch_deprecation:nnNNpn { 2022-10-17 } { \text_titlecase_first:n }
\cs_new:Npn \char_titlecase:N { \text_titlecase_first:n }
\__kernel_patch_deprecation:nnNNpn { 2022-10-17 } { \str_casefold:n }
\cs_new:Npn \char_foldcase:N { \str_casefold:n }
\__kernel_patch_deprecation:nnNNpn { 2022-10-17 } { \str_lowercase:n }
\cs_new:Npn \char_str_lowercase:N { \str_lowercase:n }
\__kernel_patch_deprecation:nnNNpn { 2022-10-17 }
  { \tl_to_str:e { \text_titlecase_first:n } }
\cs_new:Npn \char_str_titlecase:N #1
  { \tl_to_str:e { \text_titlecase_first:n {#1} } }
\__kernel_patch_deprecation:nnNNpn { 2022-10-17 } { \str_uppercase:n }
\cs_new:Npn \char_str_uppercase:N { \str_uppercase:n }
\__kernel_patch_deprecation:nnNNpn { 2022-10-17 } { \str_casefold:n }
\cs_new:Npn \char_str_foldcase:N { \str_casefold:n }
\tl_map_inline:nn
  {
    { catcode } { catcode_remove }
    { charcode } { charcode_remove }
    { meaning } { meaning_remove }
  }
  {
    \use:e
      {
        \__kernel_patch_deprecation:nnNNpn { 2022-01-11 } { \peek_remove_spaces:n }
        \cs_gset_protected:Npn \exp_not:c { peek_ #1 _ignore_spaces:NTF } ##1##2##3
          {
            \peek_remove_spaces:n
              { \exp_not:c { peek_ #1 :NTF } ##1 {##2} {##3} }
          }
        \__kernel_patch_deprecation:nnNNpn { 2022-01-11 } { \peek_remove_spaces:n }
        \cs_gset_protected:Npn \exp_not:c { peek_ #1 _ignore_spaces:NT } ##1##2
          {
            \peek_remove_spaces:n
              { \exp_not:c { peek_ #1 :NT } ##1 {##2} }
          }
        \__kernel_patch_deprecation:nnNNpn { 2022-01-11 } { \peek_remove_spaces:n }
        \cs_gset_protected:Npn \exp_not:c { peek_ #1 _ignore_spaces:NF } ##1##2
          {
            \peek_remove_spaces:n
              { \exp_not:c { peek_ #1 :NF } ##1 {##2} }
          }
      }
  }
```

```
\cs_new_protected:Npn \prop_put_if_new:Nnn { \prop_put_if_not_in:Nnn }
\cs_new_protected:Npn \prop_gput_if_new:Nnn { \prop_gput_if_not_in:Nnn }
\cs_generate_variant:Nn \prop_put_if_new:Nnn
  { NnV , NV , c , cnV , cV }
\cs_generate_variant:Nn \prop_gput_if_new:Nnn
  { NnV , NV , c , cnV , cV }
%%
%%
%% End of file `expl3-code.tex'.
```