

1. Introduction

17. **The character set.** In order to make

21. Since we are assuming that our Pascal system is able to read and write the visible characters of

44. Once a sequence of characters has been appended to *str_pool*, it officially becomes a string when the function *make_string*

47. The initial values of *str*

54. On-line and off-line printing. Messages that are sent to a user's terminal and to the transcript-log file are produced by several *print* procedures. These procedures will direct their output to a variety of places, based on the setting of the global variable *selector*, which has the following possible values:

term

59. An entire string is output by calling *print*

63.

67. Reporting errors. When something anomalous is detected, METAFONT typically does something like this:

```
print_err("Something anomalous has been detected");
help3("This is the first line of my offer to      )
      ("This is  second line. I m/
      ("expl best way for you to
error;
```

71.

83. We allow deletion of up to 99 tokens at a time.

hDelete c "0" tokens and

111. The invariant relations in this case are (i) $b(qf + p) = 2$

116.

118. *h*

127.

132. Here is the routine that calculates 2^8 times the natural logarithm of a *scaled* quantity; it is an integer approximation to $2^{24} \ln(x=2^{16})$, when x is a given positive integer.

The method is based on exercise 1.2.2{25 in *The Art of Computer Programming*: During the main iteration we have $1 \leq 2^{-30}x \leq (1 - 2^{-1-k})$, and the logarithm of $2^{30}x$ remains to be added to an accumulator register called y . Three auxiliary bits of accuracy are retained in y during the calculation, and sixteen auxiliary bits to extend y are kept in z .

139. Given integers x and y

141. *h*Return an appropriate answer based on

144. Conversely, the n_{\sin}

153. Packed data.

METAFONT

173. Just before INIMF writes out the memory, it sorts the doubly linked available space list. The list is probably very short at such times, so a simple insertion sort is used. The smallest a

180. Procedure *check_mem* makes sure that the available space lists of

186. The command codes. Before we can go much further, we need to define symbolic names for the internal code numbers that represent the various commands obeyed by METAFONT. These codes are somewhat arbitrary, but not completely so. For example, some codes have been made adjacent so that **case**

```
de ne internal_quantity = 40 finternal numeric parameter (e.g., pausing) g  
de ne min_su x_token = internal_quantity  
de ne tag_token = 41 f
```


189. Primitive operations that produce values have a secondary identification code in addition to their com-

193. Well, we do have to list the names one more time, for use in symbolic printouts.

*h*Initialize table entries (done by INIMF only) 176 *i* +
 int_name[tracing_titles]

196.

200. The hash table.

213.

228. **Data structures for variables.** The variables of METAFONT

233. The *id*

237. *h* Preface the output with a part specifier; **return** in the case of a capsule 237 *i*
begin case *name_type(p)* **of**
x_

242. The *nd_*

nd

246.

250. **Saving and restoring equivalents.** The nested structure provided by **begingroup** and **endgroup**■
allows *eqtb*

276.

303. Generating discrete moves.

304. We will later reduce the problem of digitizing a complex cubic $z(t) =$

314. *h*Make moves for current subinterval; if bisection is necessary, push the second subinterval onto the stack, and **goto**

323. **Edge structures.** Now we come

330. *h* Insert exactly $n_min(cur_edges)$

345.

349. The entire *sorted* list is returned to available memory in this step; a new list is built starting (temporarily) at *temp_head*

354. The last and most difficult routine for transforming an edge structure| and the most interesting one!| is `xy_`

370. It's not necessary to add the o sets to the x coordinates, because an entire edge structure can be shifted without affecting its total weight. Similarly, we don't need to subtract *zero_eld*.

*h*Add the contribution of node q to the total weight, and set $q \leftarrow link($

373. Just after a new edge weight has been put into the *info* field of node r , in row n

374. One way to put new edge weights into an edge structure is to use the following routine, which simply draws a straight line from $(x0;y0)$ to $(x1;y1)$. More precisely, it introduces weights for the edges of the discrete path $bt[x_0;x_1] + \frac{1}{2} + c; bt[y_0;y_1]$

392. The general bisection method is quite simple when $n = 2n9$ (henceNT

403. The *make_spec* routine has an interesting side effect, namely to set the global variable *turning_number* to the number of times the path winds around the origin. This is done by setting the global variable *turning_number* to the value of the expression $\frac{1}{2\pi} \int_0^1 \frac{d\theta}{dt} dt$, where θ is the angle of the vector from the origin to the point on the path at time t . The expression $\frac{1}{2\pi} \int_0^1 \frac{d\theta}{dt} dt$ is the winding number of the path around the origin.

407.

422. The swapping here doesn't simply interchange x and

425. *h*Subdivide the cubic a second time with respect to x^0 y^0 425 *i*
begin *split*

433. Here, then, is the procedure that rounds x

437. When y has been negated, the *octant* codes are $> \text{negate_}y$. Otherwise these routines are essentially identical to the routines for x coordinates that we have just seen.

*h*If node q is a transition point for y coordinates, compute and save its before-and-after coordinates **437**

442. In octants whose code number is even, x has been negated; we want to round ambiguous cases

445. In node p , the coordinates $(b; d$

451. The *new_boundary*

470.

471.

473. The *print_pen*

479. There should be exactly one node whose octant number is less than its predecessor in the cycle; that

484. Conversely, *make_*

490. **Filling an envelope.** We are about to reac hre

497. Let us first solve a special case of the problem: Suppose we know an index k such that either
 (i) $s(t) \leq s_{k-1}$ for all t and $s(0) < s_k$, or
 (ii) $s(t) \geq s_k$ for all t and $s(0) > s_k$

524. Elliptical pens. To get the envelope of a cyclic path with respect to an ellipse, METAFONT calculates the envelope with respect to a polygonal approximation to the ellipse, using an approach due to John Hobby (Ph.D. thesis, Stanford University, 1985). This has two important advantages over trying to obtain the "exact" envelope:

526.

527. The *make_ellipse* subroutine produces a pointer to a cyclic path whose vertices define a polygon suitable for envelopes. The control points on this path will be ignored; in fact, the fields in knot nodes

541. Since we're interested in the tangent directions, we work with the derivative

$$\frac{1}{3} B^0(x_0; x_1; x_2; x_3; t) = B(x_1 - x_0; x_2 - x_1; x_3 - x_2; t) \text{ instead of } B(x_0; x_1; x_2; x_3; t) \text{ itself.}$$

accuracy.
The given path may turn abruptly
time. Therefore we remember the c
value of *phi* will be unde ned on the
h

547. The quadratic polynomial $B(y ; y ; y ; t)$ begins

550.

The given cubics $\quad ;w ;w ;w \quad)$ and $\quad ;z ;z$


```
570. de ne start_screen
      begin if : screen_started then
        begin screen_OK init_
```

574. Opening a window isn't like opening a file, because you can open it as often as you like, and7(y)28(ou)-4

577. Now here comes METAFONT's most complicated operation related to window display: Given the number k of an open window, the pixels of positive weight in *cur_edges* will be shown as *black* in the window; all other pixels will be shown as *white*.

procedure *disp_edges*(k : *window_number*);

label *done*; *found*;

583. Now m is the column $< right_col[k]$.

h Record a possible position in column m 583 i

if $w = 0$ then

b(then) \rightarrow \rightarrow 0 0 1 217.114 /F1 9. \rightarrow 1.852.636 cm 0 0 m 3.055 0 l 0.398 w SQBT/F37 9.

589. Here is a procedure that prints a dependency list in symbolic form. The second parameter should be
57broutineso foe(prnm)28tingoexpressionsT

pt03210068 0 Td ;1

er0 0 Td [,doto33.5304 0 Td();1) Tj/Fx4t9.9626 Tf 246.23i9 -11.955 Td[8labo

gr009T26;710t002338-0200325T0037Tj4009.962695508680Tdo82.09

592. One of the main operations needed on dependency lists is to add a multiple of one list to the other; we call this *p_plus_fq*, where *p* and *q* point to dependency lists and *f* is a fraction.

Ifcoe cientyendent6 Tfi(v)55(ariabl26 Tfi(b)-28(ecomes)]TJ /F37 9.9626 Tf 231.221 0 Td [(c)51(o)51(ef)]TJ ET
_needing

594. The *p_plus_fq*

596. *h*Contribute a term from q , multiplied by f 596 i
begin if $tt = dependent$ **then** v *take*

602. Here's a simple procedure that reports an error when a variable has just received a known value that's

607.

618. Dynamic nonlinear equations. Variables of numeric type are maintained by the general scheme of independent, dependent, and known values that we have just studied; and the components of pair and transform variables are handled in the same way. But METAFONT also has five other types of values: **boolean**, **string**, **pen**, **path**, and **picture**; what about them?

621. Eventually there might be an equation that assigns values to all of the variables in a ring. The *nonlinear_*

635.

647. Maintaining the input stacks. The following subroutines change the input status in commonly

651. The contents of *cur_cmd*; *cur_mod*; *cur_sym* are placed into an equivalent token bb the *cur*

665. The *runaway* procedure displays the first part of the text that occurred when METAFONT began its special *scanner_status*, if that text has been saved.

h

669. A percent sign appears in *bu er[limit*

674. *h*Get the fraction part f of a numeric token

688. Four commands are intended to be used only within macro texts: **quote**, **#@**, **@**, and **and**

700. *h*Scan the token or variable to be defined; set

706. **Expanding the next token.** Only a few command codes < *min*

726. At this point, the reader will find it advisable to review the explanation of token list format that was presented earlier, paying special attention to the conventions that apply only at the beginning of a macro's token list.

On the other hand, the reader will have to take the expression-parsing aspects of the following program on faith; we will explain *cur_type* and *cur_exp* later. (Several things in this program depend on each other, and it's necessary to jump into the circle somewhere.)

*h*Scan the delimited argument represented by *info(r)* **726** *i*

752. Iterations.

760. The loop text is inserted into

766. File names. It's time now to fret about file names. Besides the fact that different operating systems

774.

780. Operating systems often make it possible to determine the exact name (and possible version number)

783. Initially *job_name*

791. Here's an example of how these file-name-parsing routines work in practice. We shall use the macro *set_output_* *le_name* when it is time to crank up the output file.

de ne

810. *h*Recycle a big node 810

815. *h*

821. A global variable called

834. Later we'll come to procedures that perform actual operations like addition, square root, and so on;

838. *h*Protest division by zero 838

843. *h*Other local variables for

848. Here's a routine that puts the current expression back to34(bac)e read again.

895.

902. *h*Trace the current unary operation 902 *i*

```
begin begin_diagnostic; print_nl("{"); print_op(c); print_char("(");  
print_exp(null
```

```
906. de ne three_sixty_units 23592960 fthat's 360 unity g
    de ne boolean_reset(#)
        if # then cur_exp true_code else cur_exp false_code
h
```


918. `de ne type`

924. *h*Trace the current binary operation 924 *i*

begin *begin_diagnostic; print_nl("{ ("); print_exp(p;0); f*show the operand, but not verbosely *g*
print_char(")"); print_op(c); print

930. The first argument to *add*

932. We prefer *dependent* lists to *proto_dependent* ones, because it is nice to retain the extra accuracy of

936. Let's turn now to the six basic relations of comparison.

*h*Additional cases of binary operators

950. *h*Squeal about division by zero 950 *i*

```
begin exp_err("Di vi si on by zero");  
help2("You re trying to divide the quantity shown above the error")  
("message by zero. I m going to divide it by one instead."); put_get_error;  
end
```

This code is used in section 948.

9501 *h*

963. The next simplest transformation procedure applies to edges. It is simple primarily because

984. *h*Declare binary action procedures 923 *i* +
procedure *set_*

1000. *h*Assign the current expression to the variable *lhs* 1000 *i*
begin *p* *nd_variable(lhs)*;
if *p* \neq *null* then
begin *q* *stash*

xMETAFONT

1018. *h*Put each of METAFONT's primitives into the hash table 192 ¹⁹¹⁸ *h* $\text{m253 0 Td+92primitve(x) Tj /F1 9.9626 Tf0}$

1020. Commands. Let's turn now to statements that are classified as `\commands` because of their imperative nature. We'll begin with simple ones, so that it will be clear how to hook command processing

1025. *h*Cases of *print_cmd_*

1035. The following procedure is careful not to undo the left-hand symbol too soon, lest commands like `\let x=x` have a surprising effect.

*h*Declare action procedures for use by *do_statement* 995 *i* +

procedure *do_let*;

var *l*: *pointer*; *f*hash location of the left-hand symbol *let*

1053. *h*Cases of *print*

1082. *h*Declare action procedures for use by *do_statement* 995 *i* +

1089. The rest of the TFM file may be regarded as a sequence of ten data arrays having the informal specification

```
header : array [0 :: lh - 1] of stu  
char_info : array [bc ::
```


1098. *h*

1103. *h*Declare action procedures for use by *do_*

1107. *h*Store a list of ligature/kern steps **1107** *i*
begin */k_*

1115. *h*Store a list of font dimensions 1115 *i*
repeat if *j* > *max_font_dimen* then

1128. Bytes 5{8 of the header are set to the design size, unless the user has some crazy reason for specifying them differently.

1139.

xxx4 242 k[4] x

1162. Two of the parameters to *gf_boc* are global.
h3 Globalw S Qv

gf_boc

1165. With those preliminaries out of the way, *ship_out* is not especially difficult.

*h*Declare generic font output procedure *ship_out* *i* +

ship_out(c : eight_bits);

lab *done;*

1182. At the end of the program we must finish things off by writing the postamble. The TFM information should have been computed first.

An integer variable k and a *scaled* variable x will be declared for use by this routine.

hFinish th33(use)-333(b)27(y)-33by this routine.

1195. *h*Undump the dynamic memory **1195** *i*
undump(

1204. Now this is really it: METAFONT starts and ends here.

1208. The present section goes directly to the log file instead of using *print* commands, because there's no need for these strings to take up *str_pool* memory when a non-**stat** version of METAFONT is being used.

1209. We get to the *nal_cleanup* routine when **end** or **dump** has been scanned.

*h*Last-minute procedures 1205 *i* +

```

procedure nal_cleanup;
  label exit;
  var c: small_number; f0 for end, 1 for dump g
  begin c cur_mod;
  if job_name = 0 then open_log_ le;
  while input_ptr > 0 do
    if token_state then end_token_list ifend

```


1215. Index. Here is where you can find all uses of each identifier in the program, with underlined

pythag_add:

b_close: [27](#), [1134](#), [1182](#)

y_coord: [255](#), [256](#), [258](#), [265](#), [266](#), [271](#), [281](#), [282](#),

eqtb: 158, 200, 201, 202, 210, 211,

rm: [357](#), [358](#), [359](#)

ampersand: [186](#), [868](#), [869](#), [874](#), [886](#), [887](#), [891](#),
[893](#), [894](#)

Improper type: [1055](#)

An expression... : [868](#)

end_attr: [175](#), [229](#), [239](#),

get_aval: [163](#), [165](#), [235](#), [236](#), [250](#), [335](#), [350](#), [362](#),
[375](#), [376](#), [605](#), [662](#), [694](#), [697](#), [698](#), [704](#), [728](#), [734](#),

yr_packet: [553](#), [558](#), [559](#)

tracing_macros: [190](#), [192](#), [193](#), [720](#), [728](#), [734](#)

print: [54](#), [59](#), [60](#), [62](#), [66](#), [68](#),

rst:

Improper ``openwindow'` : 1073
Improper font parameter: 1115
Improper kern: 1112
Improper location: 1106
Incomplete string token... : 672
end

r_delim: [697](#), [703](#), [720](#), [725](#), [726](#),

233, 239, 242, 244, 246, 250, 253, 254, 585, 587,
589, 590, 591, 594, 595, 596, 597, 598, 599, 600,
601, 603, 604, 605, 607, 608, 609, 610, 611, 612,
615, 616, 617, 619, 620, 621, 622, 651, 678, 685,
686, 694, 698, 700, 704, 705, 752, 755, 760, 765,
798, 799, 800, 801, 803, 806, 809, 812, 814, 816,
817, 818, 819, 827, 829, 830, 845, 853, 855, 857,
858, 872, 873, 899, 903, 904, 907, 910, 915, 919,
928, 929, 930, 931, 933, 935, 936, 938, 939, 940,
942, 943, 944, 946, 948, 949, 951, 955, 956, 957,
958, 959, 966, 967, 968, 969, 970, 971, 972,
973, 974, 975, 976, 977, 978, 982, 983, 984,
988, 1000, 1001, 1005, 1006, 1007, 1008, 1009,
1010,

step primitive: [211](#)

iteration: [186](#), [683](#), [684](#), [685](#), [706](#), [707](#), [758](#)

stop—

980, 985, 1010, 1068,

h Abandon edges command because there's no variable 1060 *i* Used in sections 1059, 1070, 1071, and 1074.
h

h Compute the incoming and outgoing directions 457 *i* Used in section 454.

h Compute the ligature/kern program o set and implant the left boundary label 1137

*h*Display a numeric token [220](#) *i* Used in section [219](#).

*h*Log the sub le sizes of the TFM le 1141 *i* Used in section 1134.

*h*space(amp;[the M])TtoFF 7.9701 T42695229 0 Td 1 0 0 rg89641

*h*Pack the numeric and fraction parts of a numeric token and **return** 675𐌆𐌆𐌆

*h*Read the first line of the new file

h

*h*Skip down

