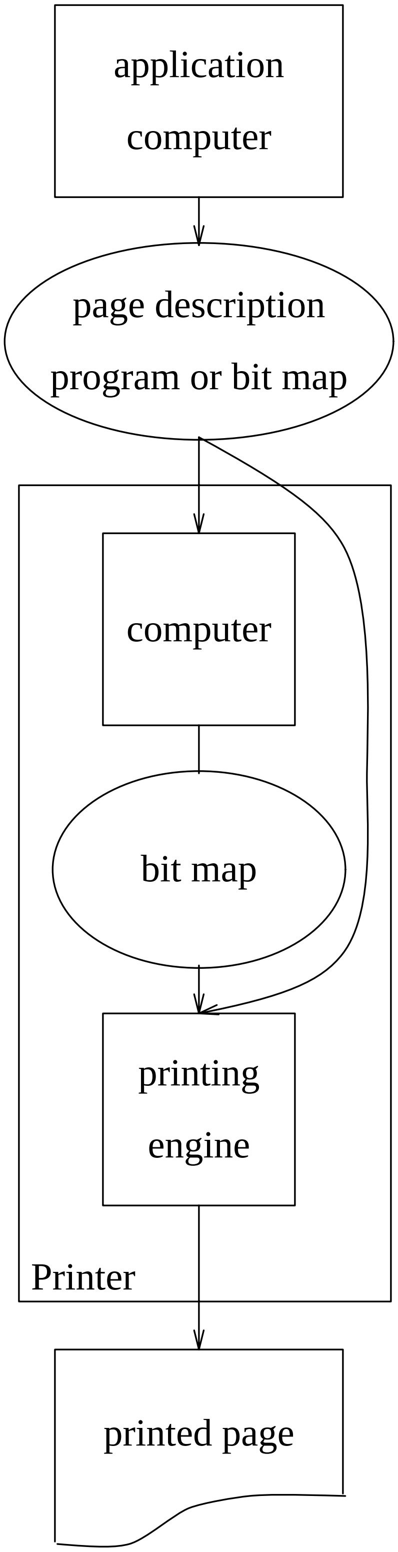
FONTS

by

Daniel M. Berry

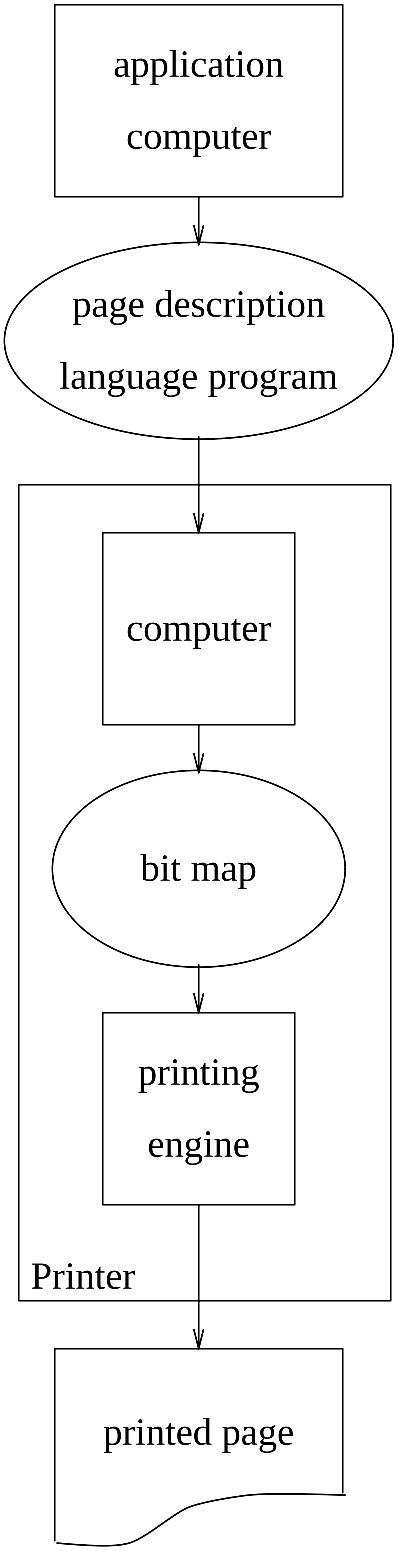
Printer Configuration:



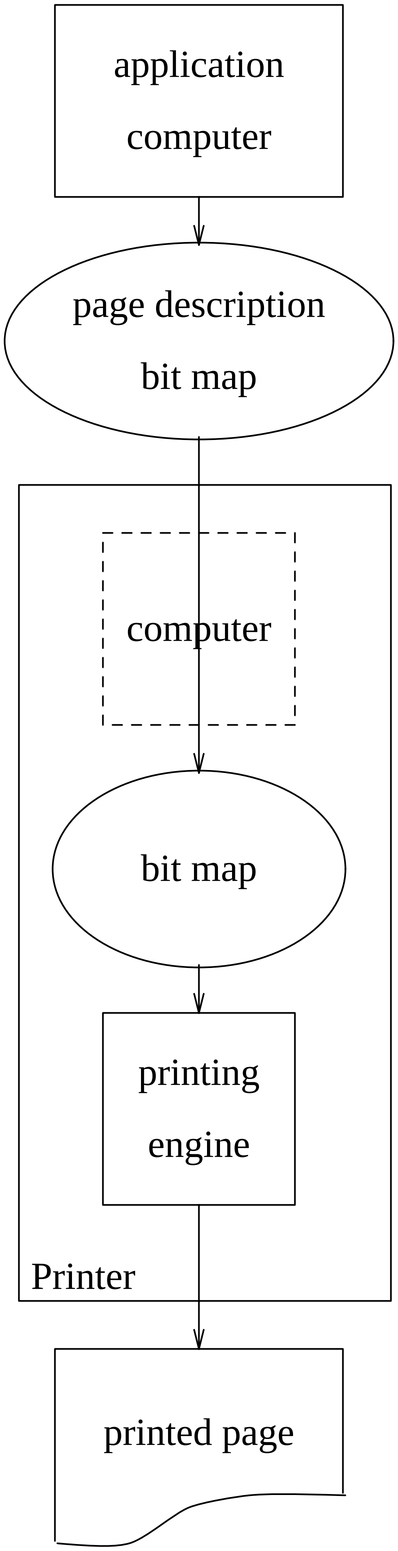
PRINTING CONFIGURATION

Two choices:

- Compilation:
  - Application computer produces bitmaps
  - Example: METAFONT
- Interpretation:
  - Printer computer produces bitmaps
  - Example: POSTSCRIPT printers, e.g., LaserWriter, QMS 800, Linotronic 300

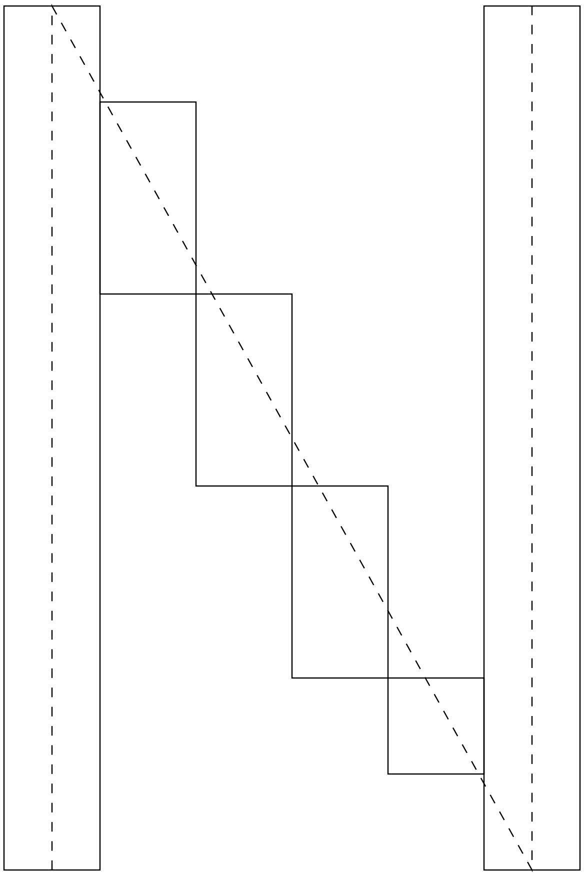


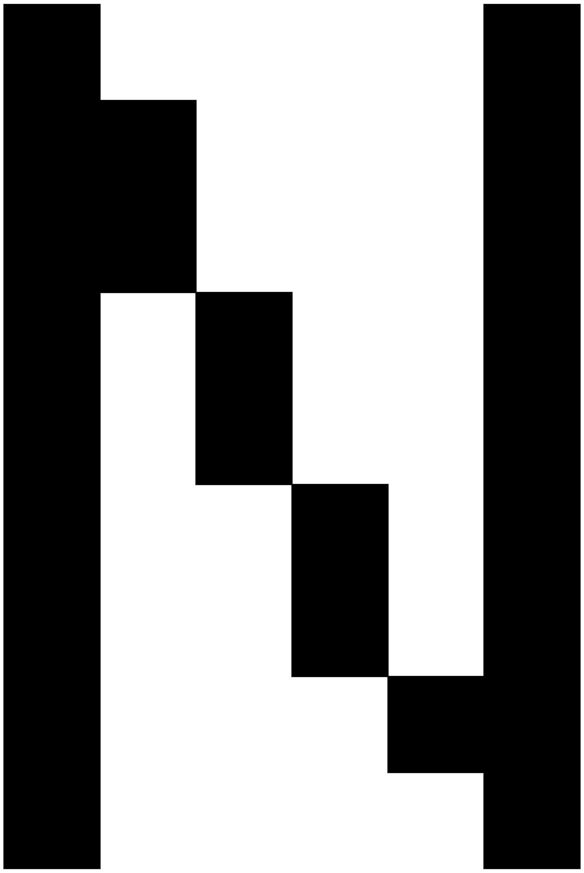
INTERPRETATION



COMPILATION

The way bitmaps work!





Now show this slide at ever smaller sizes!

N

First note that at 300 dpi ~ 118 dpc

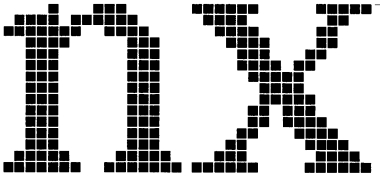
An upper case letter of point size 10
(really 8 or 9 points high) is only
38 dots high

A lower case letter of point size 10
(really 5, 6, or 7 points high) is only
29 dots high

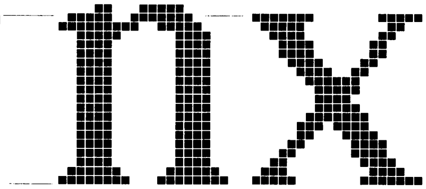
and stems are often only 2 or 3 dots wide!

For point size 5, half of that!

Adobe:

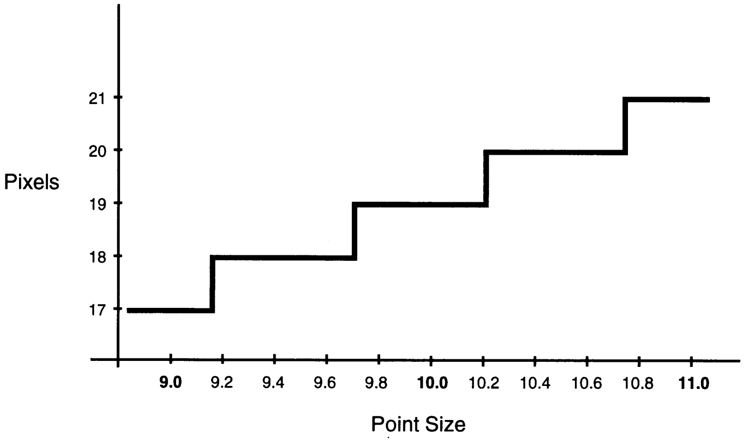


10 point, 240 dpi characters



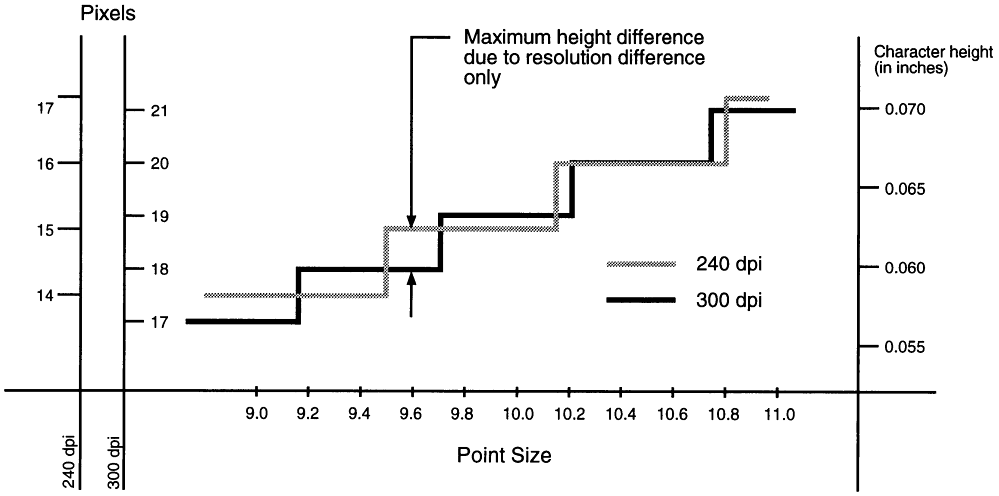
10 point, 300 dpi characters

Adobe:



300 dpi

Adobe:


Pixels
17
16
15
14
21
20
19
18
17
Maximum height difference due to resolution difference only
Character height (in inches)
0.070
0.065
0.060
0.055
240 dpi
300 dpi
9.0
9.2
9.4
9.6
9.8
10.0
10.2
10.4
10.6
10.8
11.0
Point Size
240 dpi
300 dpi

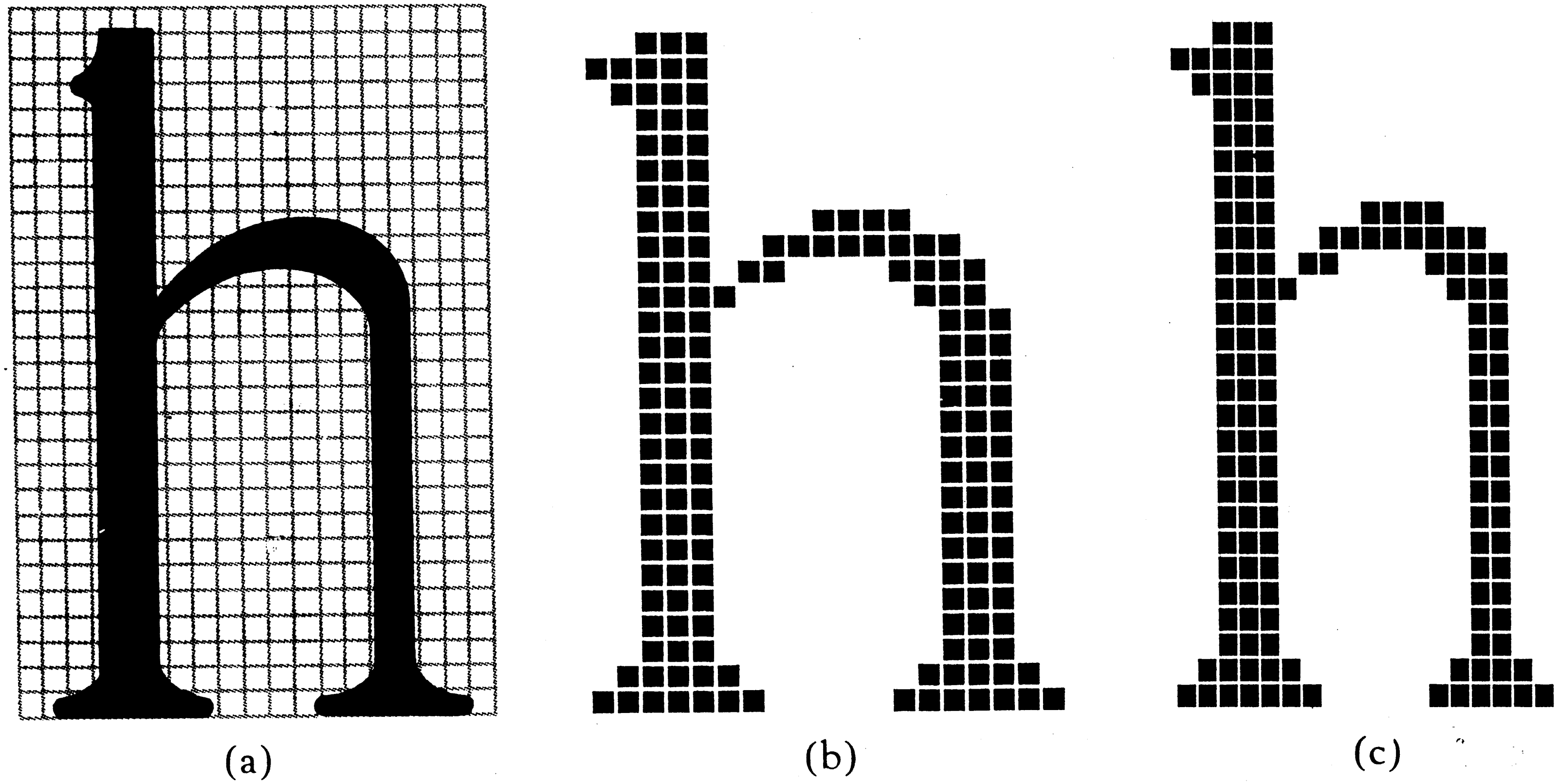
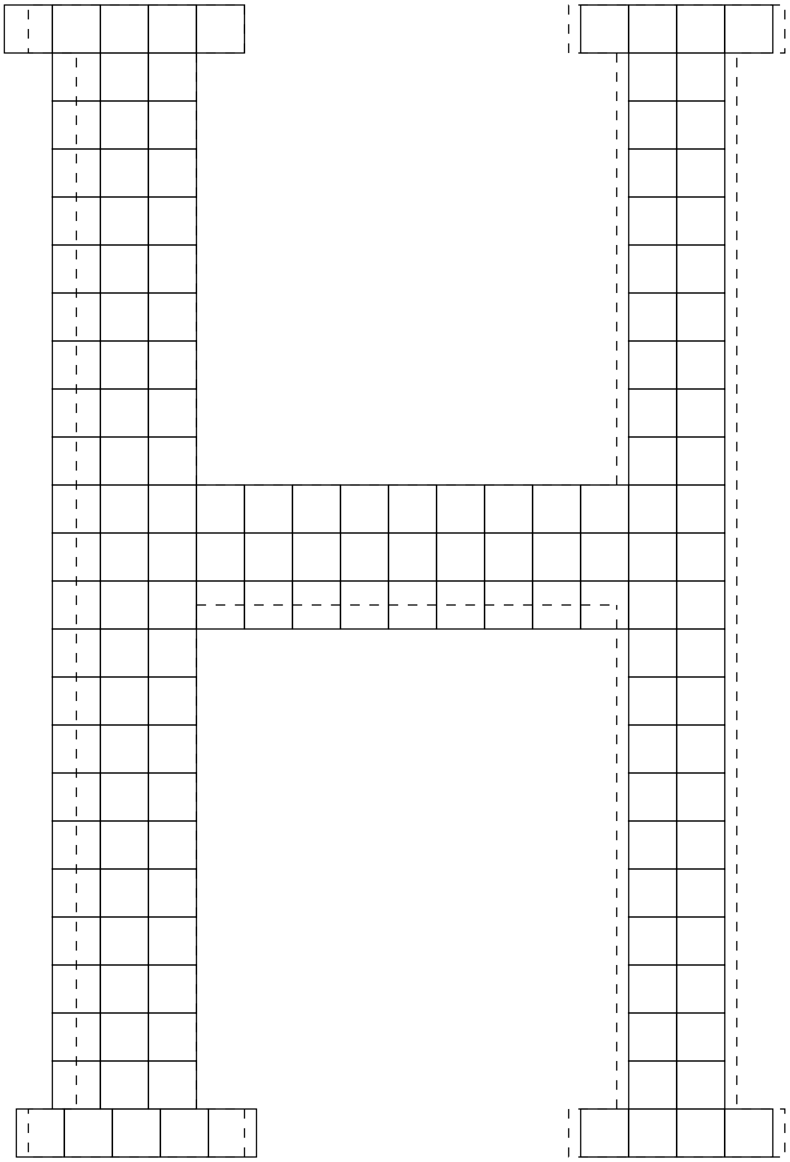
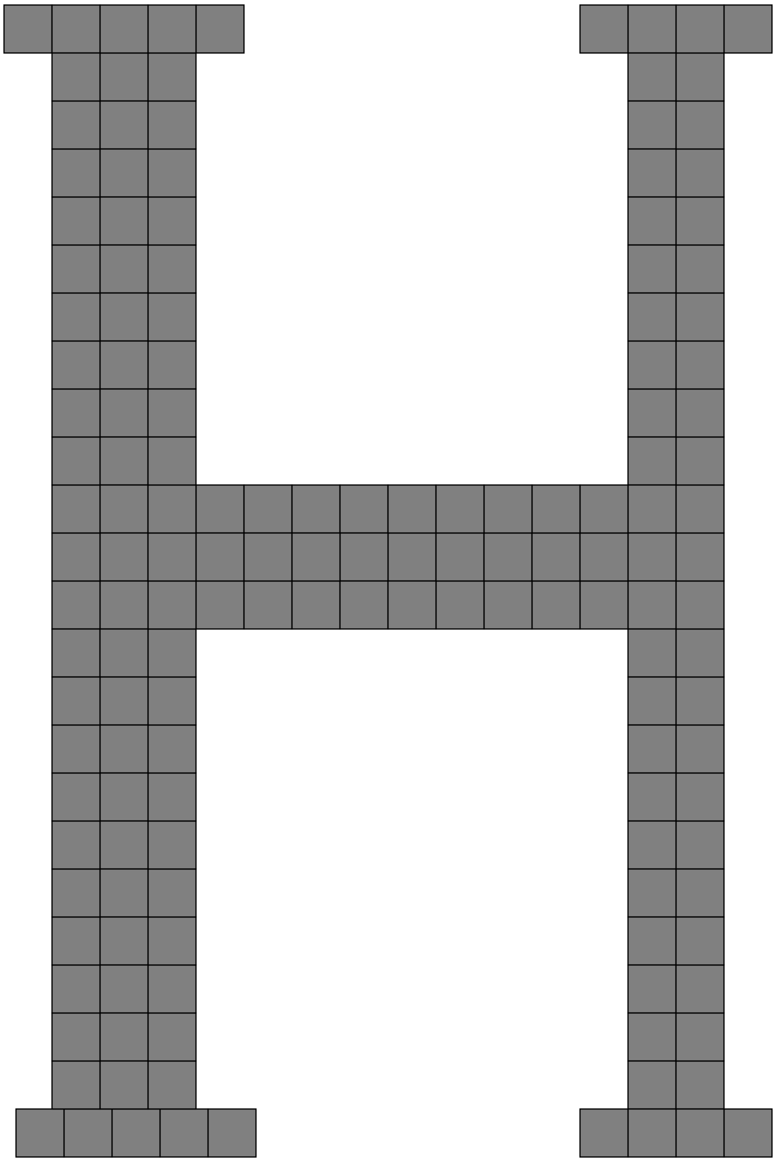


FIGURE 4.11 Possible ways to digitize a letter. The 'h' in (a) does not have features that are integral multiples of the grid. Depending on the intent, various interpretations of the design are possible, such as those in (b) and (c)

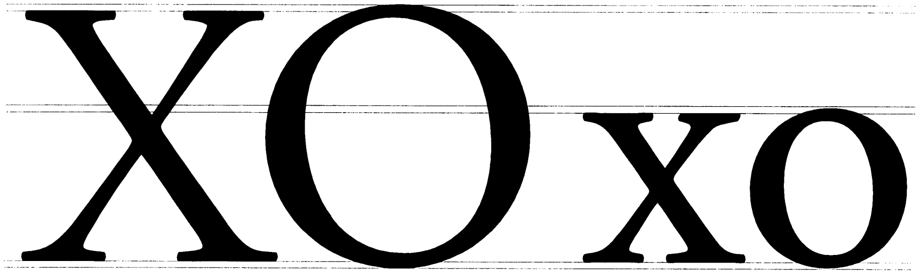
H



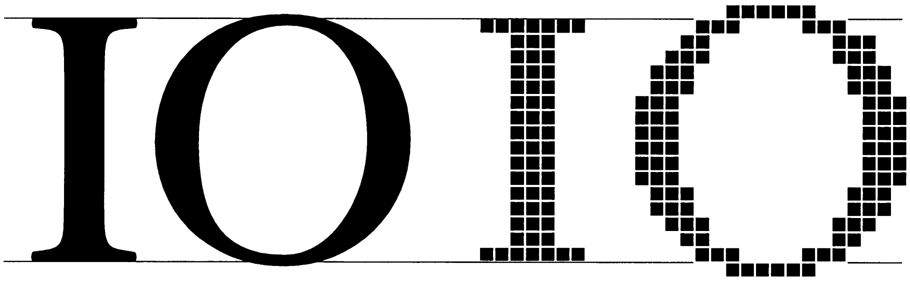


H

Adobe

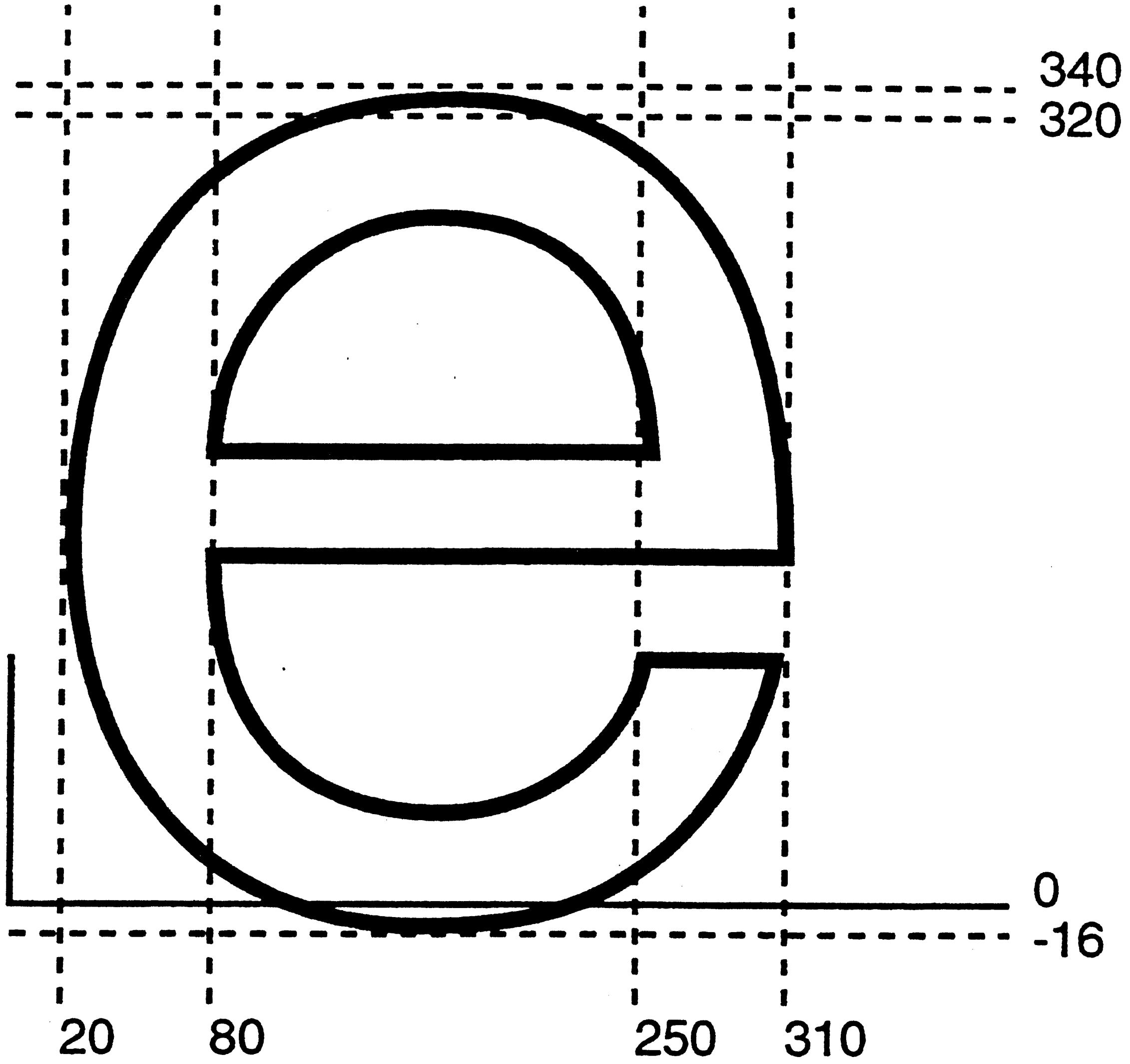


Adobe:



High Resolution Image

Low Resolution Bit Map



Hints to aid in guaranteeing preservation of geometric properties of the character

eg equal stem width
over hang if possible w/o exceeding maximum

abcdefgh
ijklmnopqrstuvwxyz
1234567890

ABCDEFGH
IJKLMNOPQRSTUV
WXYZ&

FIGURE 4.12 Optima, a typeface that is difficult to represent digitally, even at moderately high resolutions. Designed by Hermann Zapf, Optima is characterized by subtle changes in line width and near-vertical edges.

I

In the last analysis, need bitmaps

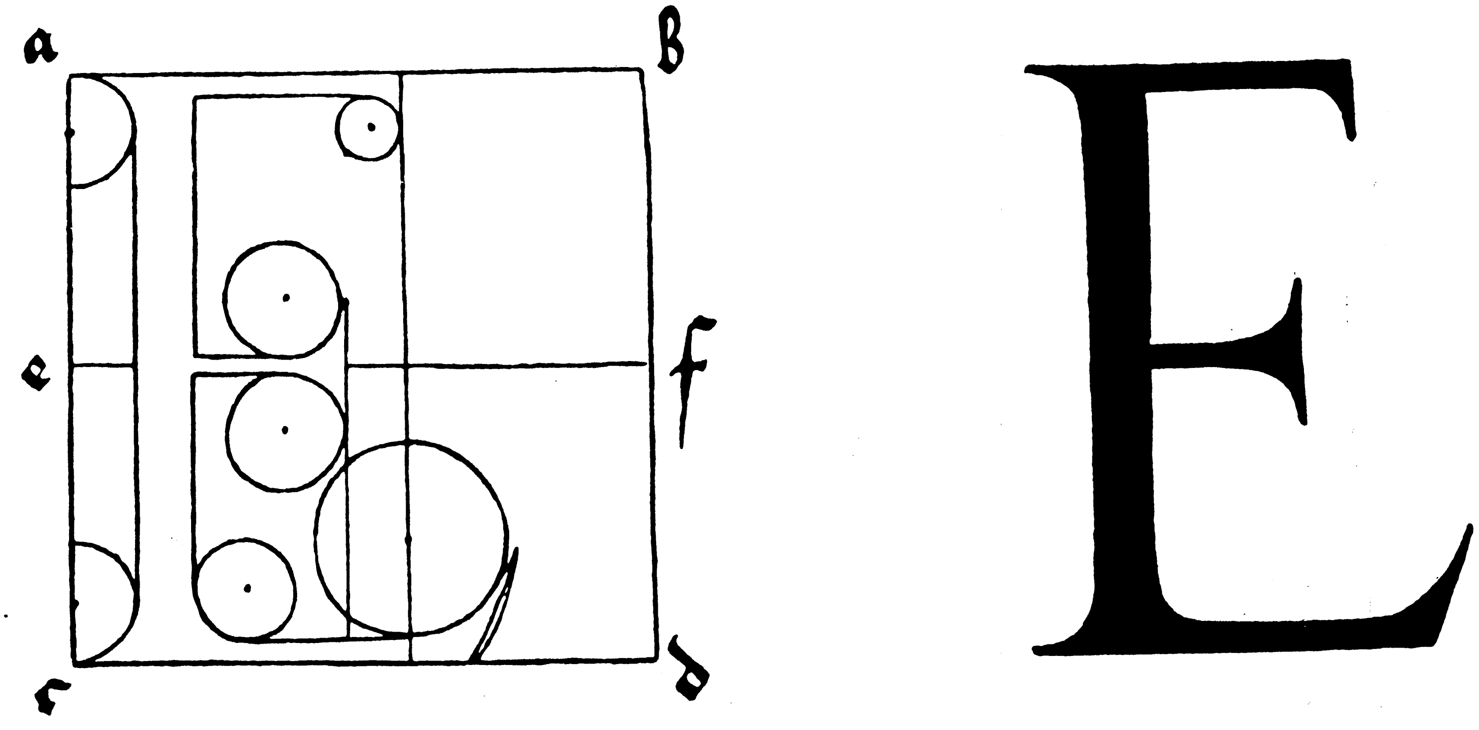
But given compilation and interpretation,

there are several representations of
fonts that are in use

interpretation
- bitmapped
- outline
- stroked

compilation
- ABOVE +
- equational

Examine closely
to see problems
and trade-offs



THE LETTER E.

FIGURE 4.29 Letterform shape can be analyzed and stored as straight lines and circular arcs. This is an old idea, as demonstrated by this 1535 illustration by Albrecht Dürer.

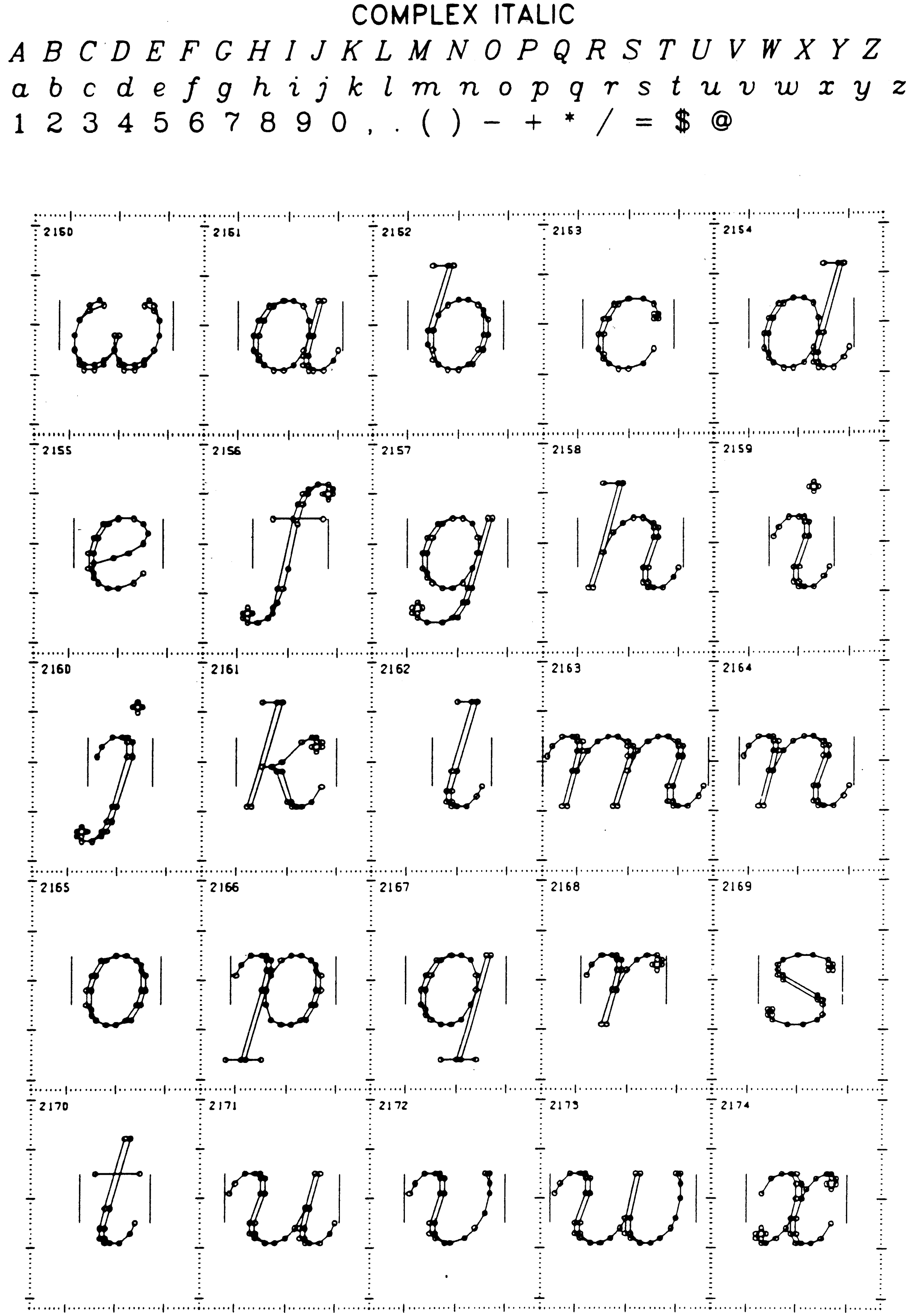


FIGURE 4.27 Hershey fonts are stored as overlapping strokes. The pen path and the resulting inked letterform are shown. (Reproduced from [Wolcott 1976].)

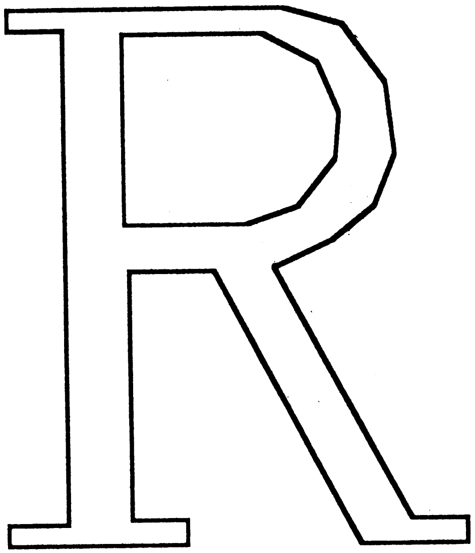


FIGURE 4.28 Character shape stored as a polygonal outline (shown enlarged).

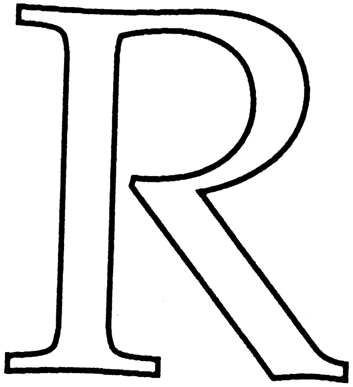


FIGURE 4.8 A letterform design expressed as an outline.

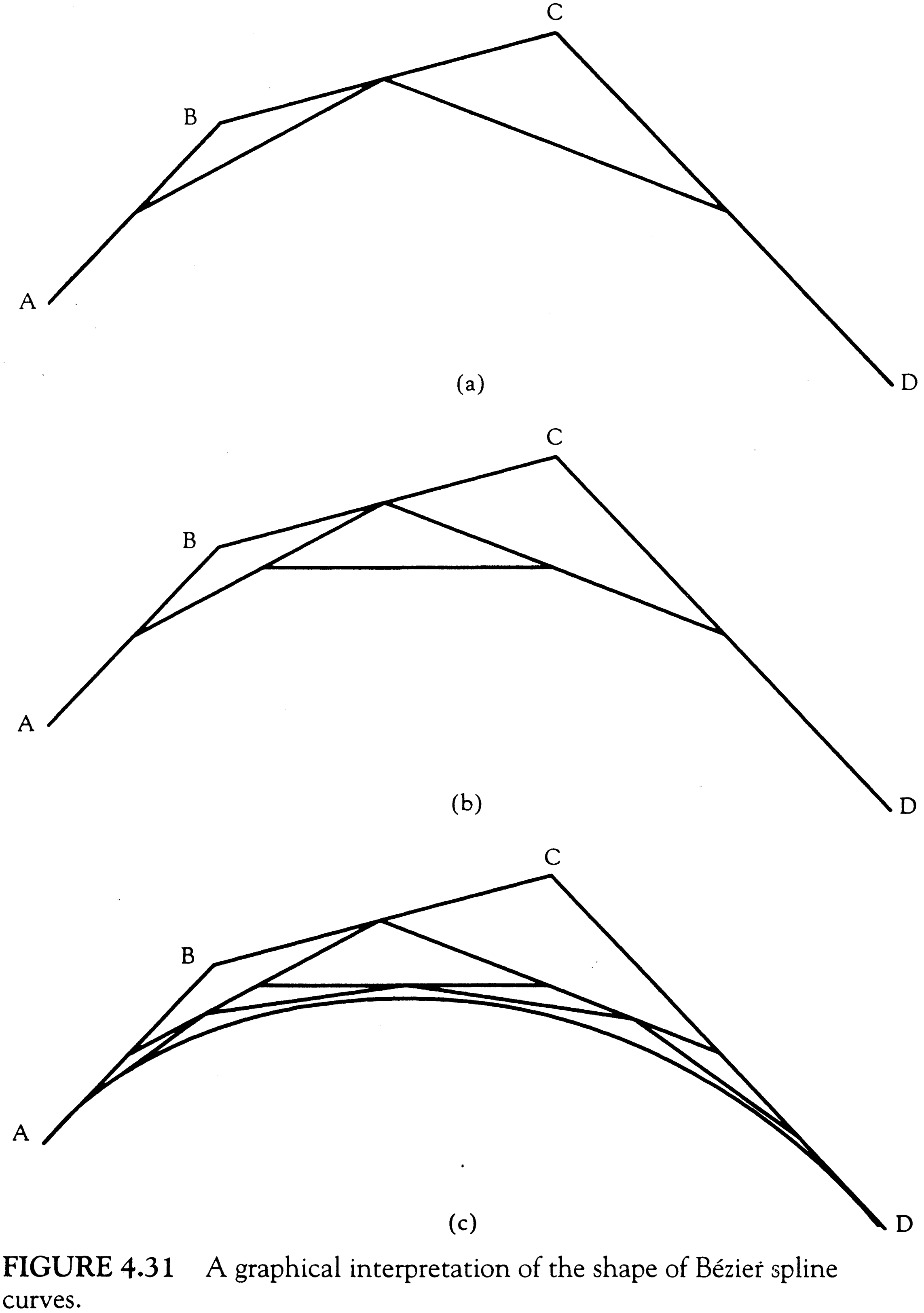
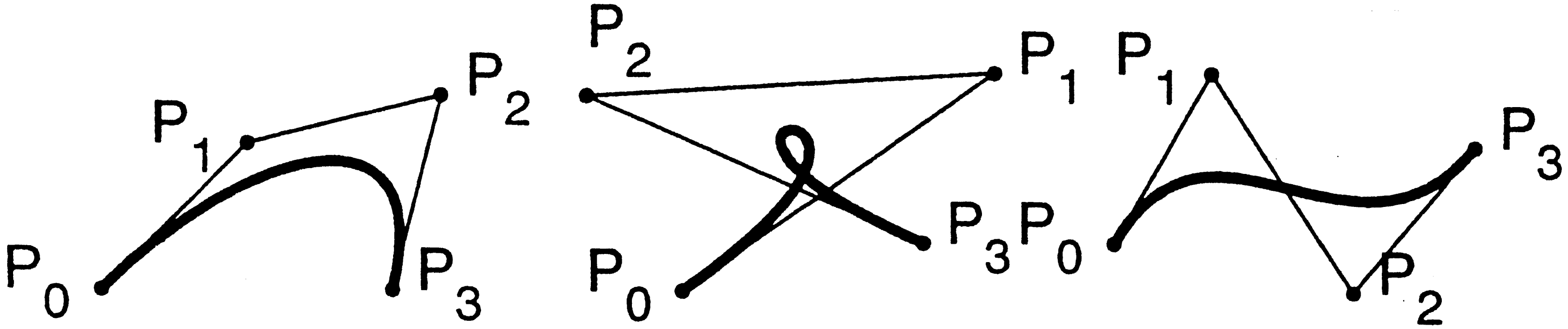


FIGURE 4.31 A graphical interpretation of the shape of Bézieŕ spline curves.


P_0
P_1
P_2
P_3
P_2
P_1
P_0
P_3
P_1
P_3
P_0
P_2

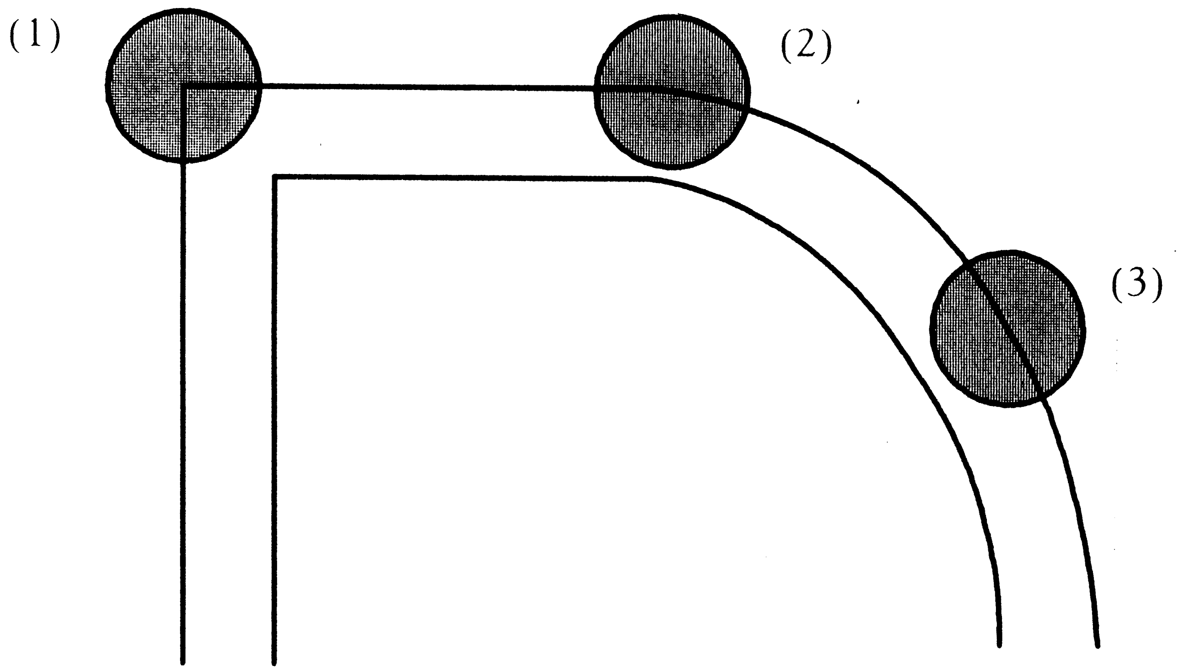


FIGURE 4.37 Different joins useful in outline specification. Case (1) is a vertex where two straight lines join. Case (2) joins a straight segment with a curved one. The tangent may have a discontinuity, depending on the curve. Finally, Case (3) joins to curves with the constraint that the tangent be continuous.

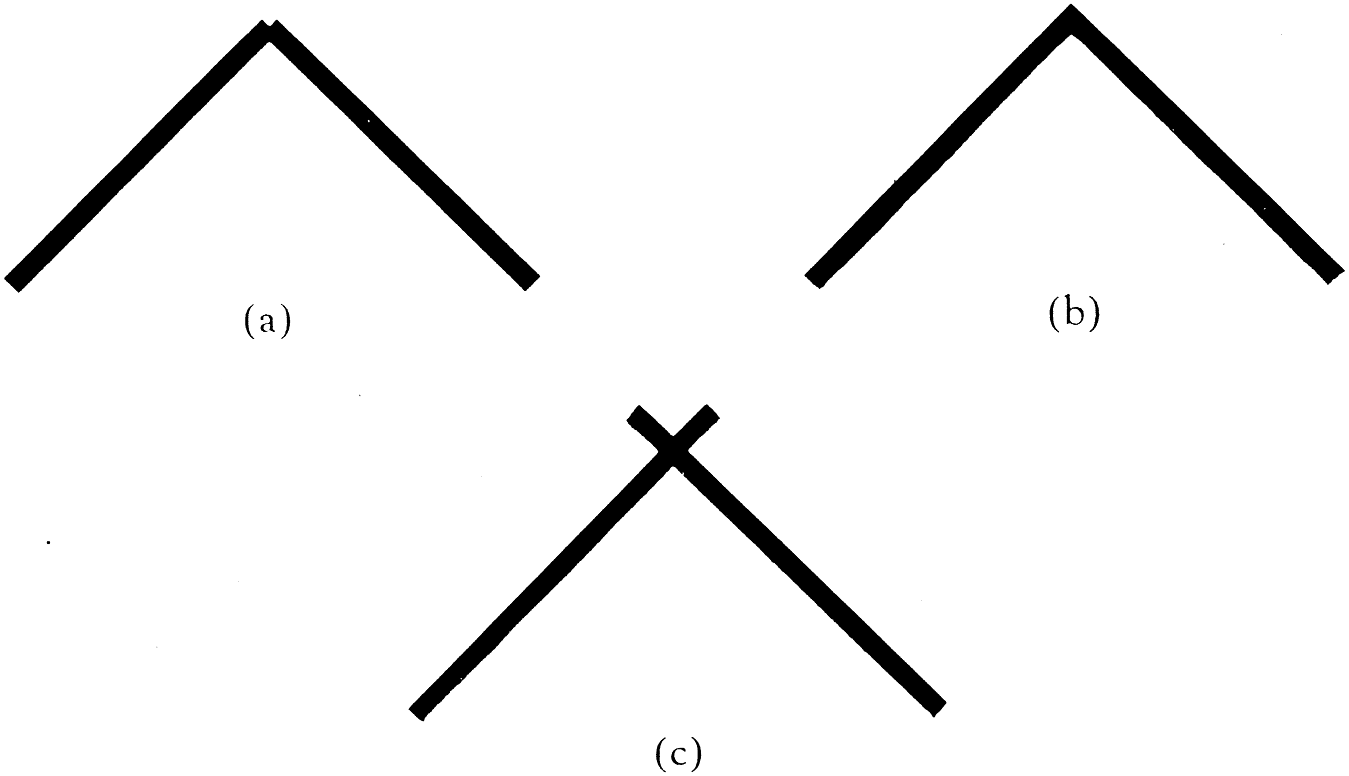


FIGURE 4.38 Three possible relationships between two pen paths. In case (a), the shape of the pen used to draw the lines determines the shape of the join. Alternatively, the lines can be mitered, as shown in (b). Case (c) shows the two strokes not interacting at all.

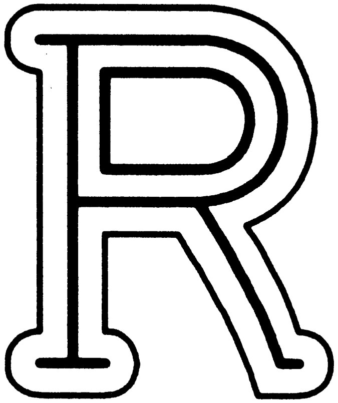
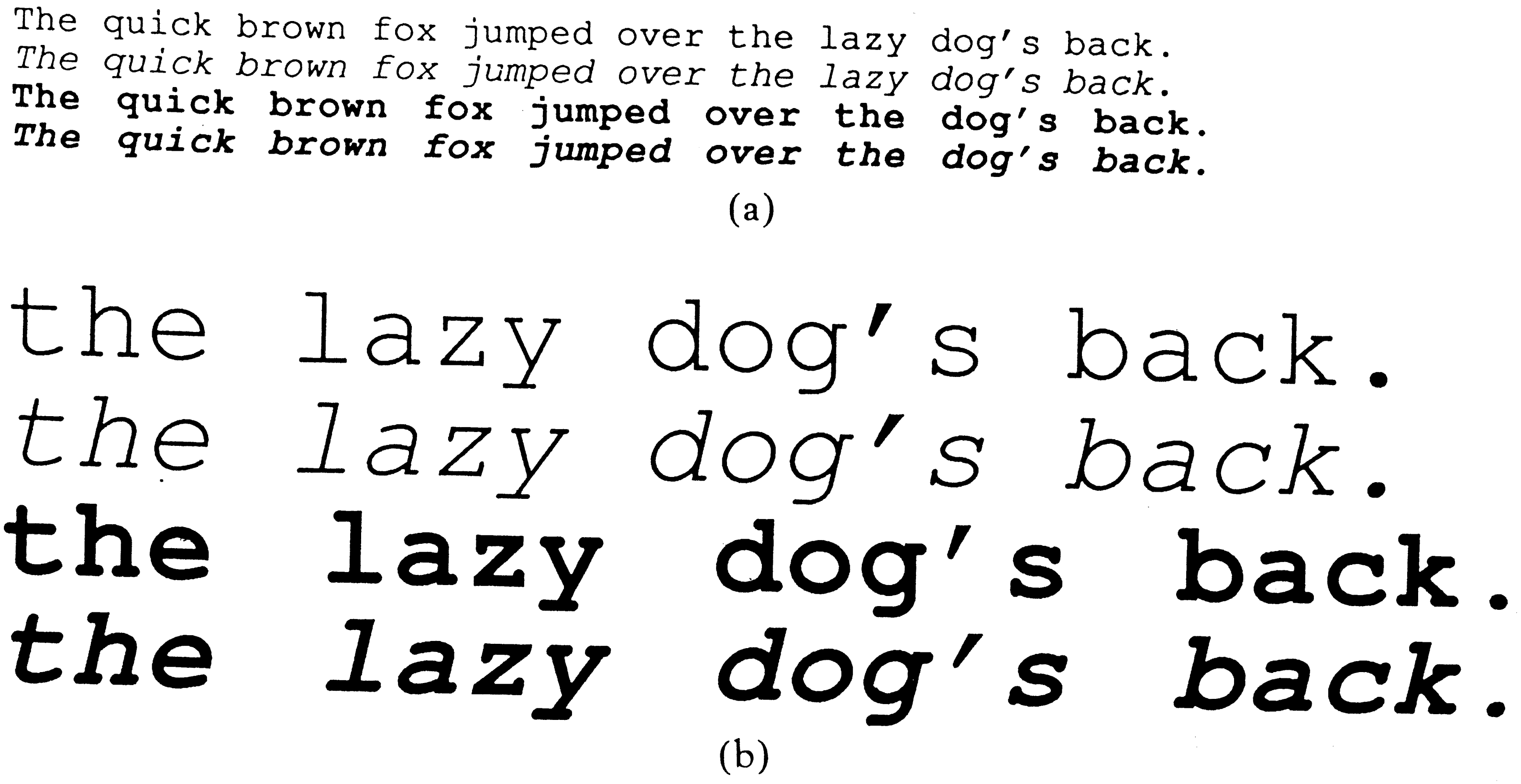


FIGURE 4.33 Inline representation of a character uses a pen path and pen shape to encode the way to draw the character.



The quick brown fox jumped over the lazy dog's back.
The quick brown fox jumped over the lazy dog's back.
The quick brown fox jumped over the dog's back.
The quick brown fox jumped over the dog's back.

(a)

the lazy dog's back.
the lazy dog's back.
the lazy dog's back.
the lazy dog's back.

(b)

FIGURE 4.34 Four fonts generated from one inline representation (a), enlarged in (b).

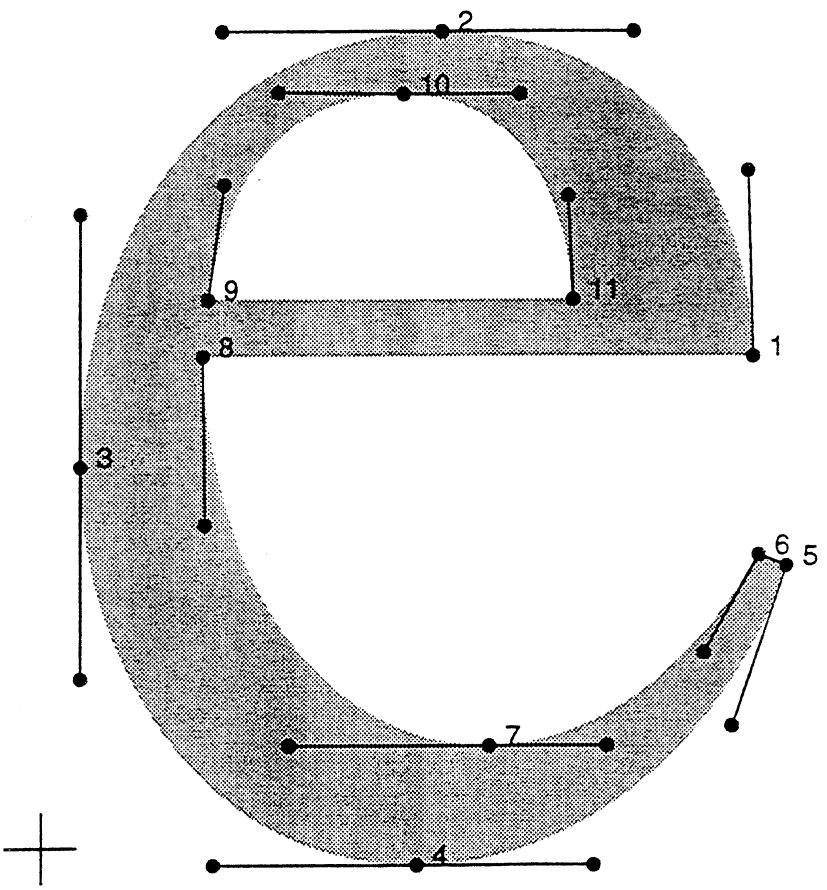


FIG. 3.12 - *Un « e » est défini par quelques points de contrôle et tangentes. Voir les figures 3.13 et 3.4*

```
\e{
402 276 moveto                    % départ point 1
399 380 334 458 226 458 curveto   % courbe de 1 à 2
102 458 22 356 22 214 curveto     % courbe de 2 à 3
22 95 97 -10 212 -10 curveto      % courbe de 3 à 4
312 -10 390 68 421 158 curveto    % courbe de 4 à 5
405 164 lineto                    % droite de 5 à 6
374 109 319 57 253 57 curveto     % courbe de 6 à 7
140 57 92 181 91 276 curveto      % courbe de 7 à 8
closepath                         % fin contour extérieur
94 308 moveto                     % départ au point 9
103 372 134 424 204 423 curveto   % courbe de 9 à 10
270 423 297 366 300 308 curveto   % courbe de 10 à 11
closepath} def                    % fin contour intérieur
```

FIG. 3.13 - *Programme PostScript donnant la définition du contour du « e » de la figure 3.12*

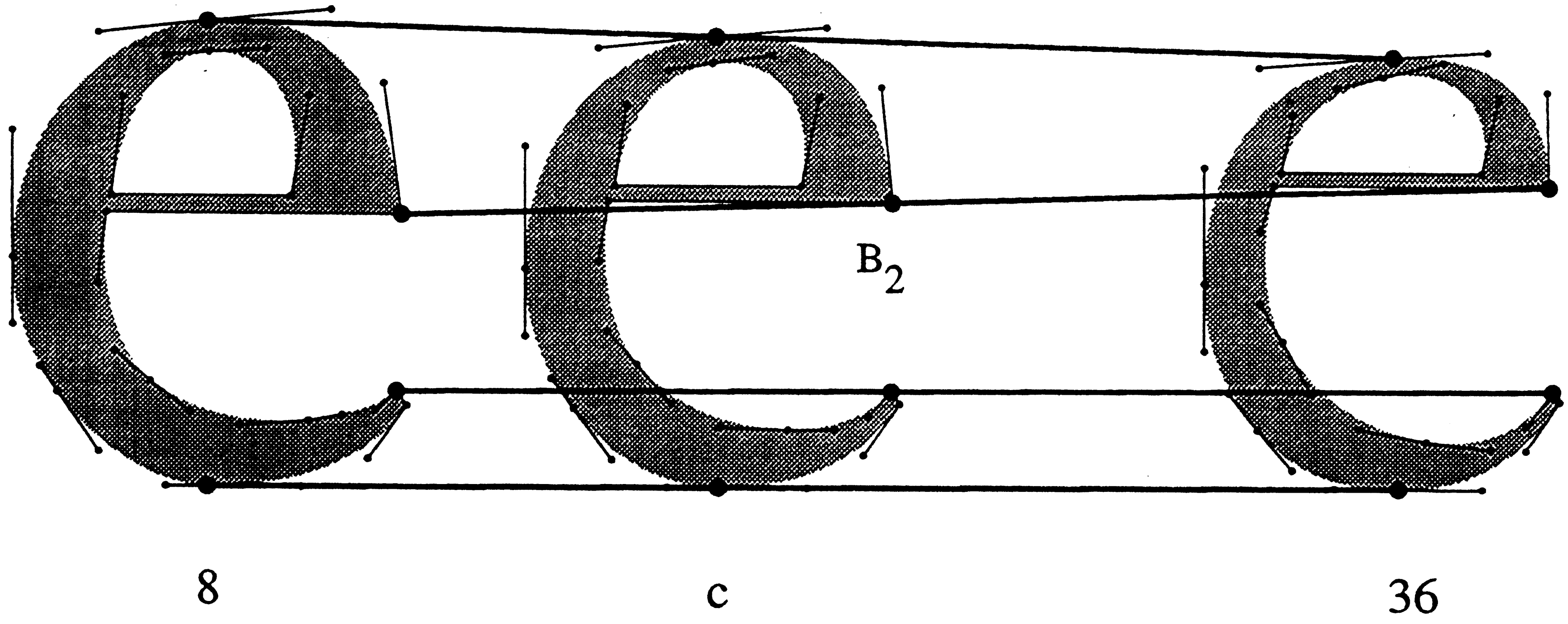


FIG. 4.7 - *Les points de contrôle correspondants des courbes de Bézier sont obtenus par interpolation linéaire*

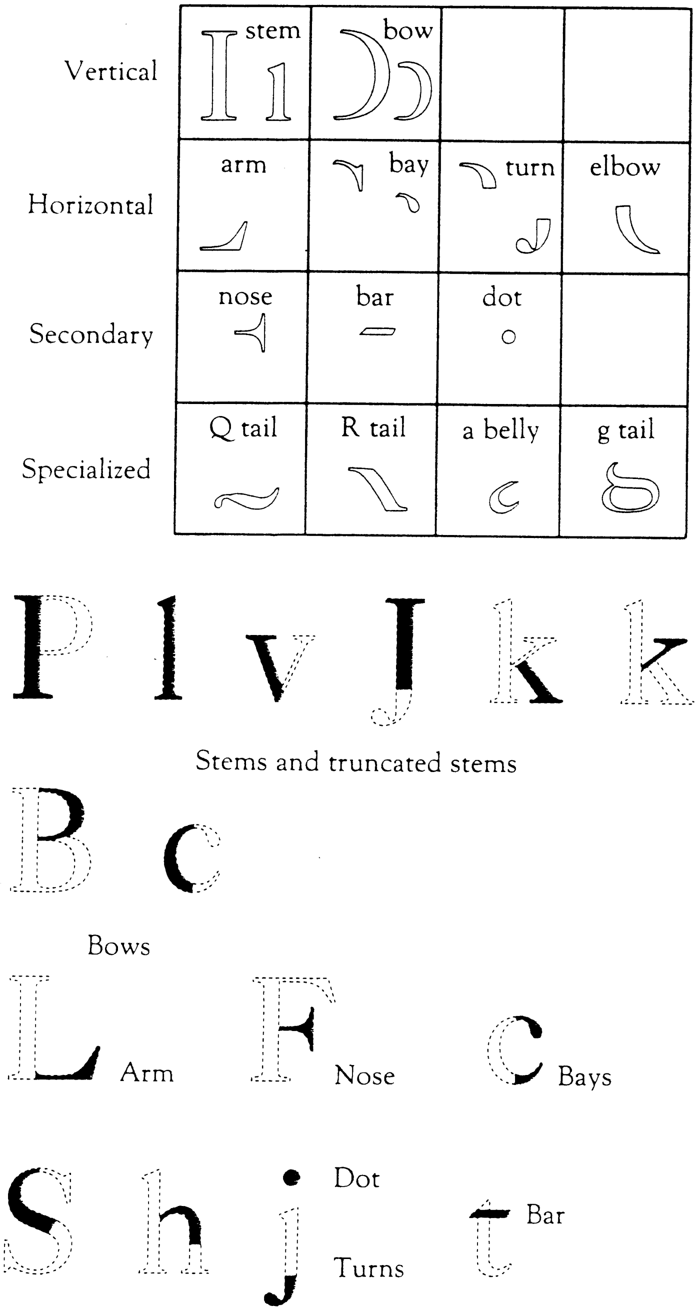
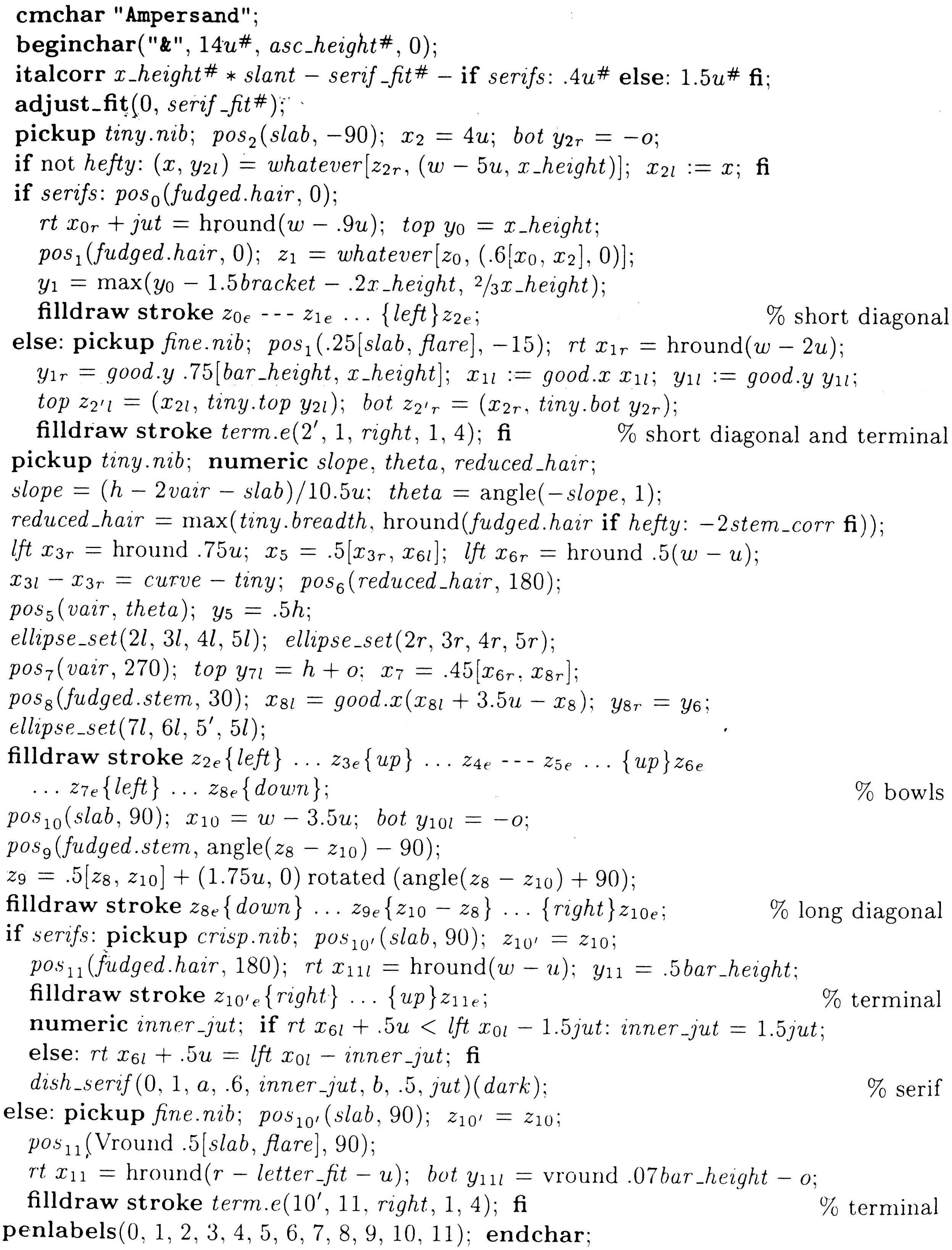


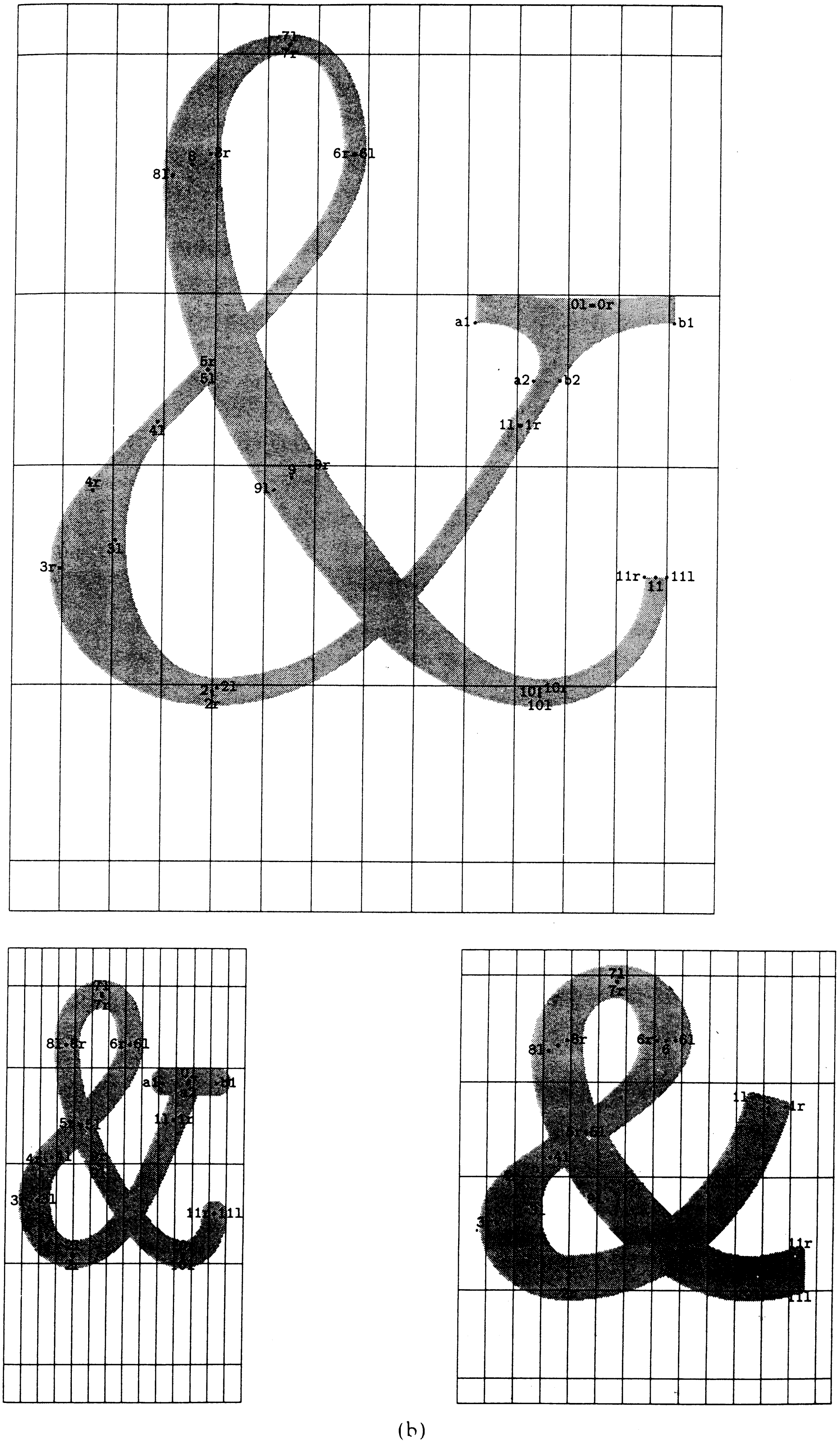
FIGURE 4.42 Primitives in Philippe Coueignoux's Character Simulated Design language. (From Philippe J. M. Coueignoux, "Generation of Roman Printed Fonts." Ph.D. Thesis, June 1975. Reprinted with permission of the Massachusetts Institute of Technology.)



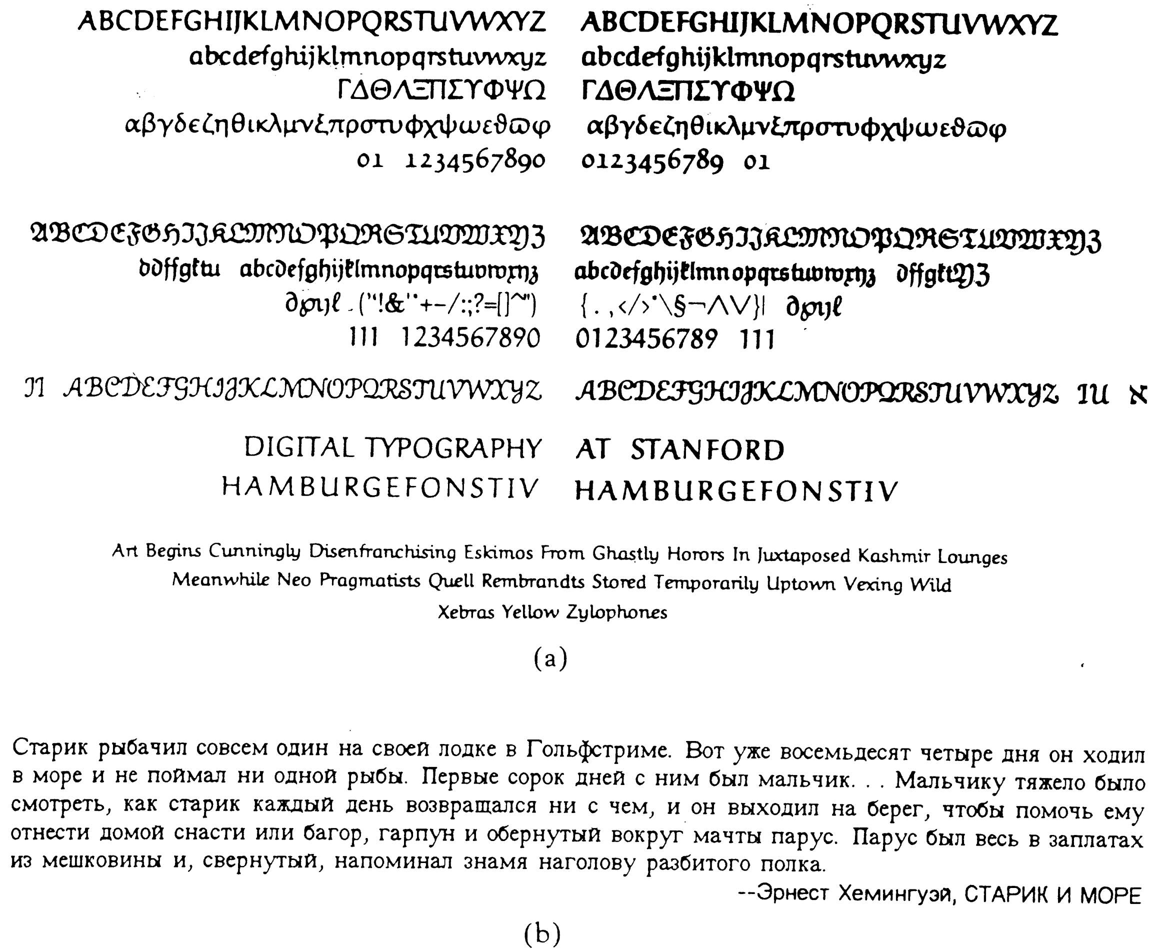
```
cmchar "Ampersand";
beginchar("&", 14u#, asc_height#, 0);
italcorr x_height# * slant - serif_fit# - if serifs: .4u# else: 1.5u# fi;
adjust_fit(0, serif_fit#);
pickup tiny.nib; pos2(slab, -90); x2 = 4u; bot y2r = -o;
if not hefty: (x, y2l) = whatever[z2r, (w - 5u, x_height)]; x2l := x; fi
if serifs: pos0(fudged.hair, 0);
  rt x0r + jut = hround(w - .9u); top y0 = x_height;
  pos1(fudged.hair, 0); z1 = whatever[z0, (.6[x0, x2], 0)];
  y1 = max(y0 - 1.5bracket - .2x_height, 2/3x_height);
  filldraw stroke z0e --- z1e ... {left}z2e;                % short diagonal
else: pickup fine.nib; pos1(.25[slab, flare], -15); rt x1r = hround(w - 2u);
  y1r = good.y .75[bar_height, x_height]; x1l := good.x x1l; y1l := good.y y1l;
  top z2'l = (x2l, tiny.top y2l); bot z2'r = (x2r, tiny.bot y2r);
  filldraw stroke term.e(2', 1, right, 1, 4); fi     % short diagonal and terminal
pickup tiny.nib; numeric slope, theta, reduced_hair;
slope = (h - 2vair - slab)/10.5u; theta = angle(-slope, 1);
reduced_hair = max(tiny.breadth, hround(fudged.hair if hefty: -2stem_corr fi));
lft x3r = hround .75u; x5 = .5[x3r, x6l]; lft x6r = hround .5(w - u);
x3l - x3r = curve - tiny; pos6(reduced_hair, 180);
pos5(vair, theta); y5 = .5h;
ellipse_set(2l, 3l, 4l, 5l); ellipse_set(2r, 3r, 4r, 5r);
pos7(vair, 270); top y7l = h + o; x7 = .45[x6r, x8r];
pos8(fudged.stem, 30); x8l = good.x(x8l + 3.5u - x8); y8r = y6;
ellipse_set(7l, 6l, 5', 5l);
filldraw stroke z2e{left} ... z3e{up} ... z4e --- z5e ... {up}z6e
  ... z7e{left} ... z8e{down};                                        % bowls
pos10(slab, 90); x10 = w - 3.5u; bot y10l = -o;
pos9(fudged.stem, angle(z8 - z10) - 90);
z9 = .5[z8, z10] + (1.75u, 0) rotated (angle(z8 - z10) + 90);
filldraw stroke z8e{down} ... z9e{z10 - z8} ... {right}z10e;      % long diagonal
if serifs: pickup crisp.nib; pos10'(slab, 90); z10' = z10;
  pos11(fudged.hair, 180); rt x11l = hround(w - u); y11 = .5bar_height;
  filldraw stroke z10'e{right} ... {up}z11e;                          % terminal
  numeric inner_jut; if rt x6l + .5u < lft x0l - 1.5jut: inner_jut = 1.5jut;
  else: rt x6l + .5u = lft x0l - inner_jut; fi
  dish_serif(0, 1, a, .6, inner_jut, b, .5, jut)(dark);                  % serif
else: pickup fine.nib; pos10'(slab, 90); z10' = z10;
  pos11(Vround .5[slab, flare], 90);
  rt x11 = hround(r - letter_fit - u); bot y11l = vround .07bar_height - o;
  filldraw stroke term.e(10', 11, right, 1, 4); fi                     % terminal
penlabels(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11); endchar;
```

(a)

FIGURE 4.43 A METAFONT program that generates a continuum of ampersands, depending on the settings of parameters (a). Three possible outputs are shown in (b). (Donald E. Knuth, *Computers and Typesetting* Vol. E, © 1985, Addison-Wesley Publishing Company, Inc., Reading, MA. Pps. 362 & 363. Reprinted with permission.)


7l
7r
8
8r
8l
6r
6l
01
0r
a1
b1
a2
b2
5r
5l
4l
1l
1r
4r
9
9r
9l
3r
3l
11r
11
11l
2
2l
2r
10l
10r
10l
7l
7r
8l
8r
6r
6l
a1
b1
5l
1l
1r
4r
4l
3r
3l
11r
11l
7l
7r
8l
8r
6r
6
6l
1l
1r
5r
5l
4l
3
11r
11l

(b)



ABCDEFGHIJKLMNOPQRSTUVWXYZ ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz abcdefghijklmnopqrstuvwxyz
ΓΔΘΛΞΠΣΥΦΨΩ ΓΔΘΛΞΠΣΥΦΨΩ
αβγδεζηθικλμνξπρστυφχψωεϑϖφ αβγδεζηθικλμνξπρστυφχψωεϑϖφ
01 1234567890 0123456789 01

ABCDEFGHIJKLMNOPQRSTUVWXYZ ABCDEFGHIJKLMNOPQRSTUVWXYZ
bdffgktu abcdefghijklmnopqrstuvwxyz abcdefghijklmnopqrstuvwxyz dffgktYZ
∂pıȷℓ . ("!&''+-/:;?=[]^") {. ,</>'\§¬∧∨}| ∂pıȷℓ
111 1234567890 0123456789 111

JI ABCDEFGHIJKLMNOPQRSTUVWXYZ ABCDEFGHIJKLMNOPQRSTUVWXYZ IU א

DIGITAL TYPOGRAPHY AT STANFORD
HAMBURGEFONSTIV HAMBURGEFONSTIV

Art Begins Cunningly Disenfranchising Eskimos From Ghastly Horors In Juxtaposed Kashmir Lounges
Meanwhile Neo Pragmatists Quell Rembrandts Stored Temporarily Uptown Vexing Wild
Xebras Yellow Zylophones

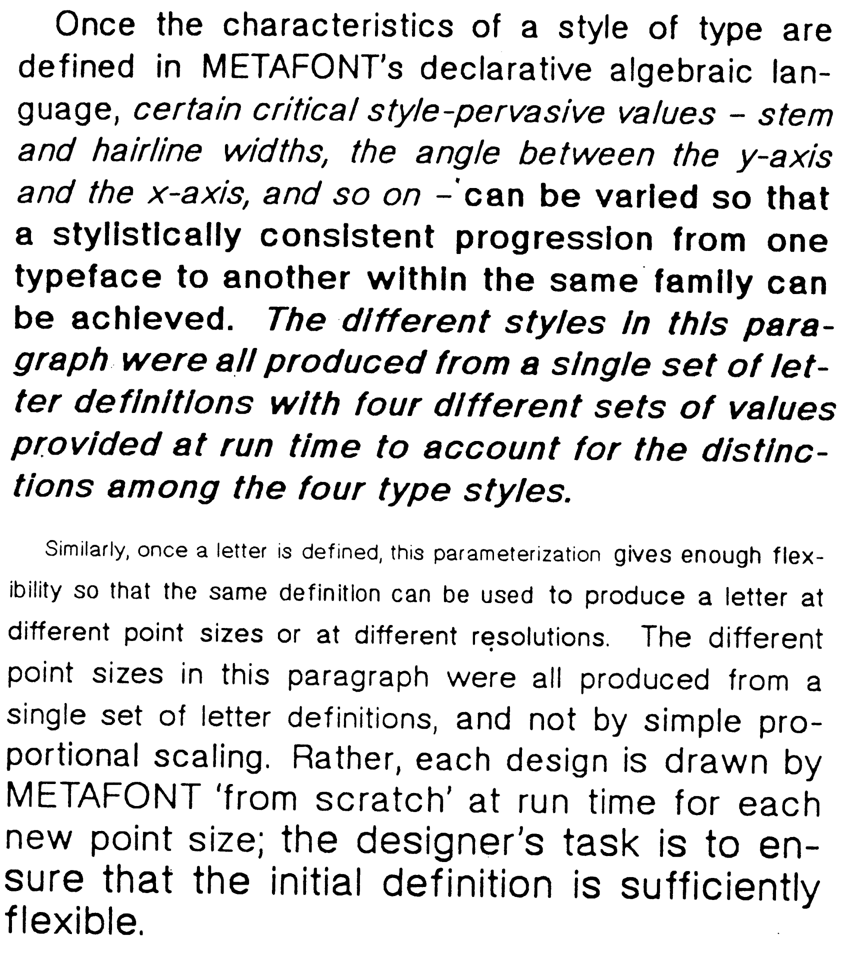
(a)

Старик рыбачил совсем один на своей лодке в Гольфстриме. Вот уже восемьдесят четыре дня он ходил в море и не поймал ни одной рыбы. Первые сорок дней с ним был мальчик. . . Мальчику тяжело было смотреть, как старик каждый день возвращался ни с чем, и он выходил на берег, чтобы помочь ему отнести домой снасти или багор, гарпун и обернутый вокруг мачты парус. Парус был весь в заплатах из мешковины и, свернутый, напоминал знамя наголову разбитого полка.

--Эрнест Хемингуэй, СТАРИК И МОРЕ

(b)

FIGURE 4.44 Some typeface designs created using METAFONT. Euler (a), a Cyrillic face (b), and a version of Helvetica (c). (Samples courtesy of David R. Siegel (a), and Georgia K. M. Tobin, The Metafoundry™ (b) and (c)).

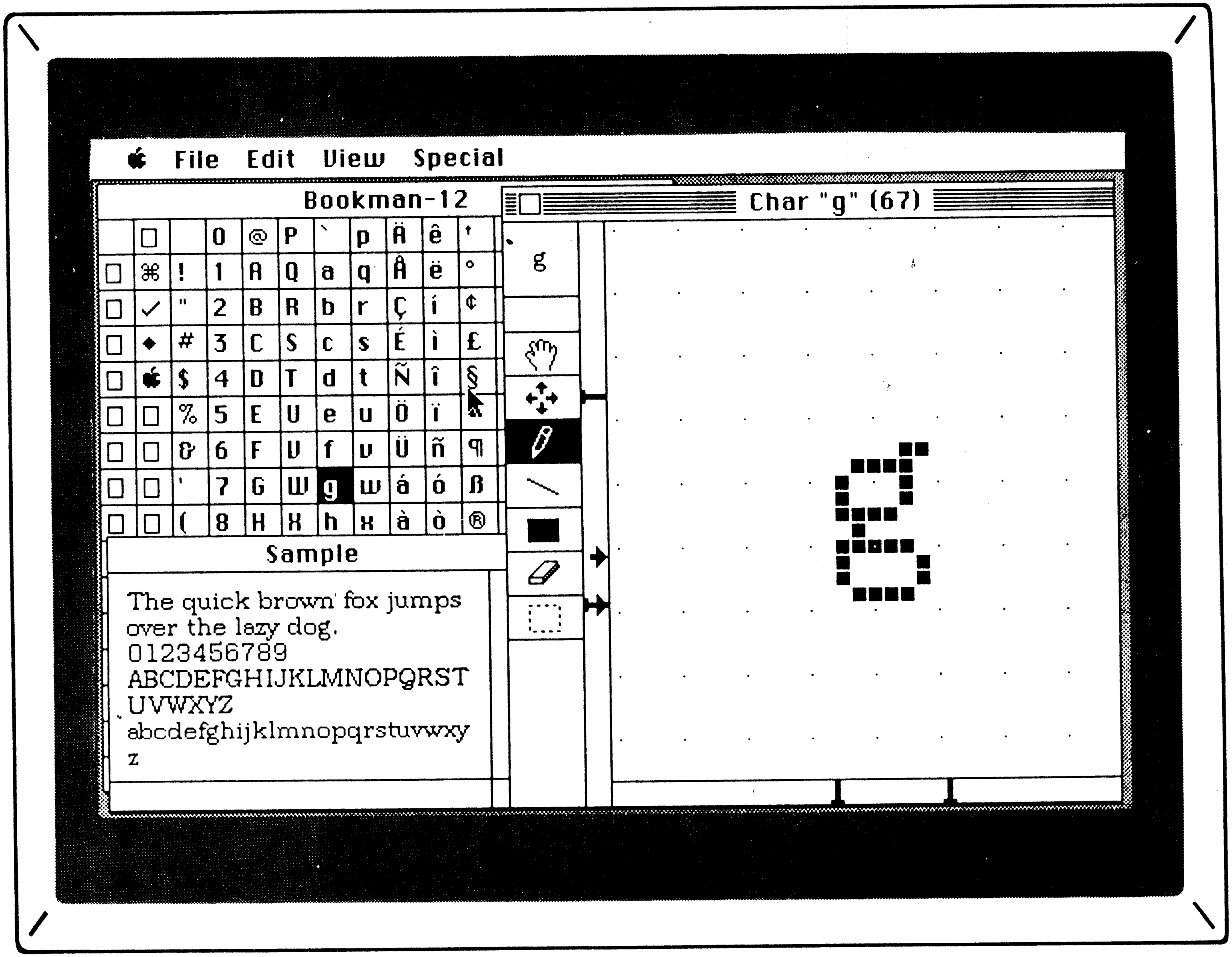


Once the characteristics of a style of type are defined in METAFONT's declarative algebraic language, *certain critical style-pervasive values - stem and hairline widths, the angle between the y-axis and the x-axis, and so on -* **can be varied so that a stylistically consistent progression from one typeface to another within the same family can be achieved.** ***The different styles in this paragraph were all produced from a single set of letter definitions with four different sets of values provided at run time to account for the distinctions among the four type styles.***

Similarly, once a letter is defined, this parameterization gives enough flexibility so that the same definition can be used to produce a letter at different point sizes or at different resolutions. The different point sizes in this paragraph were all produced from a single set of letter definitions, and not by simple proportional scaling. Rather, each design is drawn by METAFONT 'from scratch' at run time for each new point size; the designer's task is to ensure that the initial definition is sufficiently flexible.

	User Representation		
	Bitmap	Outline	Program
Extensional	Fontastic	Fontographer	METAFONT
Intentional	———	Ikarus Typefounder	METAFONT

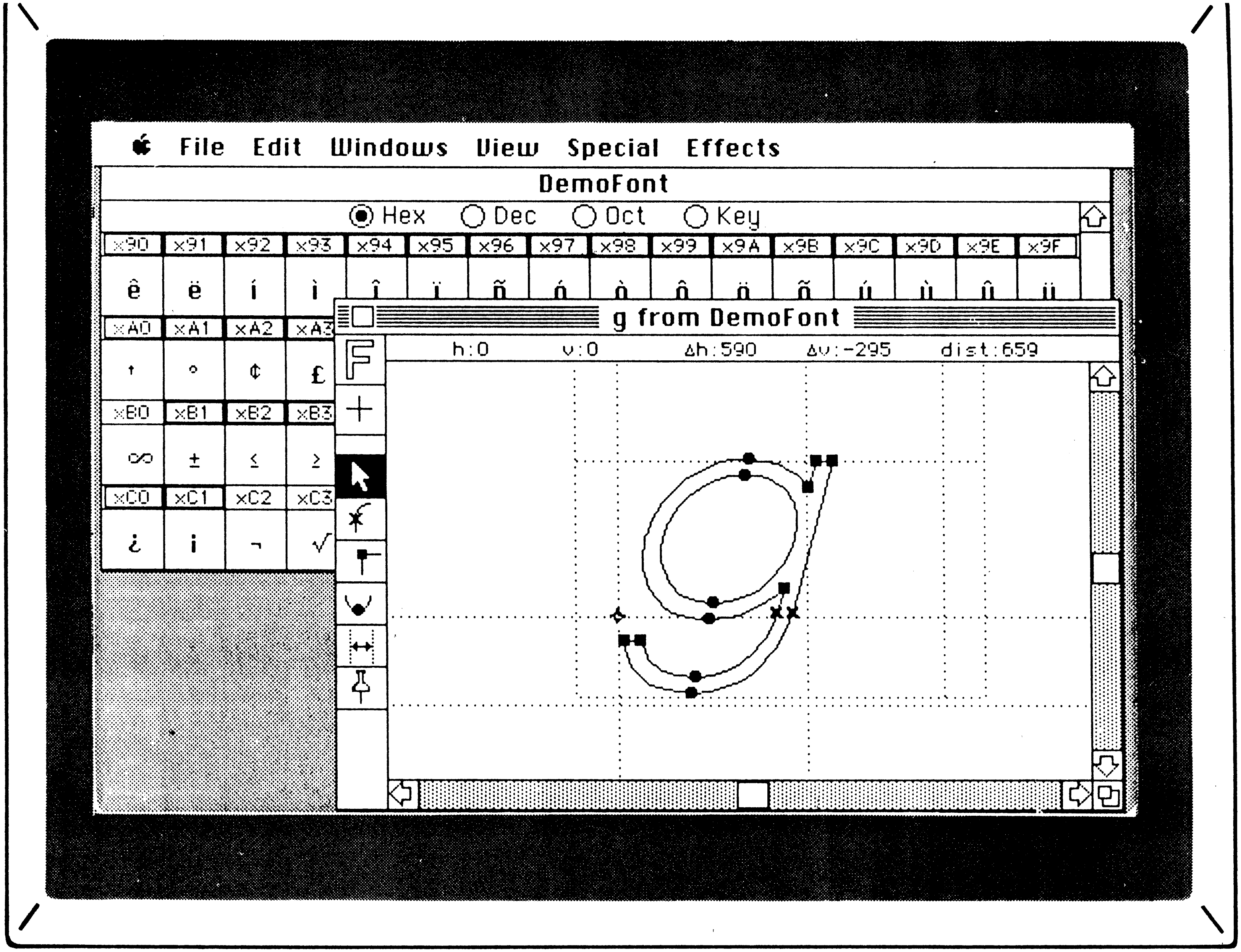
FIGURE 4.35 Some examples of font-design tools, categorized by intentionality and representation. METAFONT appears twice, indicating that it can be used both intentionally and extensionally.



	□		0	@	P	`	p	Ä	ê	†
□	⌘	!	1	A	Q	a	q	Å	ë	°
□	✓	"	2	B	R	b	r	Ç	í	¢
□	◆	#	3	C	S	c	s	É	ì	£
□		$	4	D	T	d	t	Ñ	î	§
□	□	%	5	E	U	e	u	Ö	ï	
□	□	&	6	F	V	f	v	Ü	ñ	¶
□	□	'	7	G	W	g	w	á	ó	ß
□	□	(	8	H	X	h	x	à	ò	®

The quick brown fox jumps
over the lazy dog.
0123456789
ABCDEFGHIJKLMNOPQRST
UVWXYZ
abcdefghijklmnopqrstuvwxy
z

FIGURE 4.36 A bitmap editor screen (Fontastic on the Macintosh computer).



x90	x91	x92	x93	x94	x95	x96	x97	x98	x99	x9A	x9B	x9C	x9D	x9E	x9F
ê	ë	í	ì	î	ï	ñ	ó	ò	ô	ö	õ	ú	ù	û	ü
xA0	xA1	xA2	xA3												
†	°	¢	£												
xB0	xB1	xB2	xB3												
∞	±	≤	≥												
xC0	xC1	xC2	xC3												
¿	¡	¬	√												

h:0 v:0 Δh:590 Δv:-295 dist:659

FIGURE 4.40 An outline-editor screen setup (Fontographer on the Macintosh computer).

Size:	Sample
2	DanielM. Berry
5	Daniel M. Berry
6	Daniel M. Berry
7	Daniel M. Berry
8	Daniel M. Berry
9	Daniel M. Berry
10	Daniel M. Berry
12	Daniel M. Berry
16	Daniel M. Berry
20	Daniel M. Berry
24	Daniel M. Berry
30	Daniel M. Berry
36	Daniel M. Berry
40	Daniel M. Berry
60	Daniel M. Berry
72	Daniel M. Berry

Zoom in on the smallest!

Outline

Size:	*Sample*
2	*DanielM. Berry*
5	*Daniel M. Berry*
6	*Daniel M. Berry*
7	*Daniel M. Berry*
8	*Daniel M. Berry*
9	*Daniel M. Berry*
10	*Daniel M. Berry*
12	*Daniel M. Berry*
16	*Daniel M. Berry*
20	*Daniel M. Berry*
24	*Daniel M. Berry*
30	*Daniel M. Berry*
36	*Daniel M. Berry*
40	*Daniel M. Berry*
60	*Daniel M. Berry*
72	*Daniel M. Berry*

Outline

Size:	Sample
2	DanielM. Berry
5	Daniel M. Berry
6	Daniel M. Berry
7	Daniel M. Berry
8	Daniel M. Berry
9	Daniel M. Berry
10	Daniel M. Berry
12	Daniel M. Berry
16	Daniel M. Berry
20	Daniel M. Berry
24	Daniel M. Berry
30	Daniel M. Berry
36	Daniel M. Berry
40	Daniel M. Berry
60	Daniel M. Berry
72	Daniel M. Berry

Outline

Size:	Sample
2	Daniel M. Berry
5	Daniel M. Berry
6	Daniel M. Berry
7	Daniel M. Berry
8	Daniel M. Berry
9	Daniel M. Berry
10	Daniel M. Berry
12	Daniel M. Berry
16	Daniel M. Berry
20	Daniel M. Berry
24	Daniel M. Berry
30	Daniel M. Berry
36	Daniel M. Berry
40	Daniel M. Berry
60	Daniel M. Berry
72	Daniel M. Ber

Used to be Stroke, but now Outline

Size:	Sample
2	דניאל ברי
5	דניאל ברי
6	דניאל ברי
7	דניאל ברי
8	דניאל ברי
9	דניאל ברי
10	דניאל ברי
12	דניאל ברי
16	דניאל ברי
20	דניאל ברי
24	דניאל ברי
30	דניאל ברי
36	דניאל ברי
40	דניאל ברי
60	דניאל ברי
72	דניאל ברי

Outline, no hinting

Size:	Sample
2	דניאל ברי
5	דניאל ברי
6	דניאל ברי
7	דניאל ברי
8	דניאל ברי
9	דניאל ברי
10	דניאל ברי
12	דניאל ברי
16	דניאל ברי
20	דניאל ברי
24	דניאל ברי
30	דניאל ברי
36	דניאל ברי
40	דניאל ברי
60	דניאל ברי
72	דניאל ברי

Outline, no hinting

Size:	Sample
2	דניאל ברי
5	דניאל ברי
6	דניאל ברי
7	דניאל ברי
8	דניאל ברי
9	דניאל ברי
10	דניאל ברי
12	דניאל ברי
16	דניאל ברי
20	דניאל ברי
24	דניאל ברי
30	דניאל ברי
36	דניאל ברי
40	דניאל ברי
60	דניאל ברי
72	דניאל ברי

Outline, no hinting

Size:	Sample
2	דניאל ברי
5	דניאל ברי
6	דניאל ברי
7	דניאל ברי
8	דניאל ברי
9	דניאל ברי
10	דניאל ברי
12	דניאל ברי
16	דניאל ברי
20	דניאל ברי
24	דניאל ברי
30	דניאל ברי
36	דניאל ברי
40	דניאל ברי
60	דניאל ברי
72	דניאל ברי

Outline, no hinting

Size:	Sample
2	דניאל ברי
5	דניאל ברי
6	דניאל ברי
7	דניאל ברי
8	דניאל ברי
9	דניאל ברי
10	דניאל ברי
12	דניאל ברי
16	דניאל ברי
20	דניאל ברי
24	דניאל ברי
30	דניאל ברי
36	דניאל ברי
40	דניאל ברי
60	דניאל ברי
72	דניאל ברי

Stroke

Size:	Sample
2	ダニエル・ベリ
5	ダニエル・ベリ
6	ダニエル・ベリ
7	ダニエル・ベリ
8	ダニエル・ベリ
9	ダニエル・ベリ
10	ダニエル・ベリ
12	ダニエル・ベリ
16	ダニエル・ベリ
20	ダニエル・ベリ
24	ダニエル・ベリ
30	ダニエル・ベリ
36	ダニエル・ベリ
40	ダニエル・ベリ
60	ダニエル・ベリ
72	ダニエル・ベリ

Bitmapped

Size:	Sample
2	丹尼儿北利
5	丹尼儿北利
6	丹尼儿北利
7	丹尼儿北利
8	丹尼儿北利
9	丹尼儿北利
10	丹尼儿北利
12	丹尼儿北利
16	丹尼儿北利
20	丹尼儿北利
24	丹尼儿北利
30	丹尼儿北利
36	丹尼儿北利
40	丹尼儿北利
60	丹尼儿北利
72	丹尼儿北利

Bitmapped

Daniel M. Berry

Daniel M. Berry

Daniel M. Berry

Daniel M. Berry

Daniel M. Berry

Daniel M. Berry

Daniel M. Berry

Daniel M. Berry

Daniel M. Berry

Daniel M. Berry

Outline and designed at each size

Daniel M. Berry

Daniel M. Berry

Daniel M. Berry

Daniel M. Berry

Daniel M. Berry

Daniel M. Berry

Daniel M. Berry

Daniel M. Berry

Daniel M. Berry

Daniel M. Berry

Outline and designed at each size

Daniel M. Berry

Daniel M. Berry

Daniel M. Berry

Daniel M. Berry

Daniel M. Berry

Daniel M. Berry

Daniel M. Berry

Daniel M. Berry

Daniel M. Berry

Daniel M. Berry

Outline and designed at each size

Daniel M. Berry

Daniel M. Berry

Daniel M. Berry

Daniel M. Berry

Daniel M. Berry

Daniel M. Berry

Daniel M. Berry

Daniel M. Berry

Daniel M. Berry

Daniel M. Berry

Outline and designed at each size

Daniel M. Berry

Daniel M. Berry

Daniel M. Berry

Daniel M. Berry

Daniel M. Berry

Daniel M. Berry

Daniel M. Berry

Daniel M. Berry

Daniel M. Berry

Daniel M. Berry

Outline and scaled to each size

Daniel M. Berry

Daniel M. Berry

Daniel M. Berry

Daniel M. Berry

Daniel M. Berry

Daniel M. Berry

Daniel M. Berry

Daniel M. Berry

Daniel M. Berry

Daniel M. Berry

Outline and scaled to each size

Daniel M. Berry

Daniel M. Berry

Daniel M. Berry

Daniel M. Berry

Daniel M. Berry

Daniel M. Berry

Daniel M. Berry

Daniel M. Berry

Daniel M. Berry

Daniel M. Berry

Outline and scaled to each size

Daniel M. Berry

Daniel M. Berry

Daniel M. Berry

Daniel M. Berry

Daniel M. Berry

Daniel M. Berry

Daniel M. Berry

Daniel M. Berry

Daniel M. Berry

Daniel M. Berry

Outline and scaled to each size

```
/ChinFontc3Dict 8 dict def
ChinFontc3Dict begin
  /FontType 3 def
  /FontMatrix [.8 0 0 .8 0 0] def
  /FontBBox [0 0 1 1] def
  /Encoding /encoding load def
  /Metrics /metrics load def
  /BuildChar /buildchar load def

  /CharacterDefs 95 dict def
```

PS definition of a bitmapped font

```
...

  CharacterDefs /CHb0
    { 24 24 true [24 0 0 −24 0 24]
      { <
        04006007fff00600
        6006006006406006
        3060061860061c60
        060c600600600600
        60060066ffffff06
        0060060060060060
        0c00600c00600c00
        6018006018006030
        00606003e08000c0
      > } imagemask
    } put
...
end

/ChinFontc3 ChinFontc3Dict definefont pop
```

Size:	Sample
2	丹尼儿北利
5	丹尼儿北利
6	丹尼儿北利
7	丹尼儿北利
8	丹尼儿北利
9	丹尼儿北利
10	丹尼儿北利
12	丹尼儿北利
16	丹尼儿北利
20	丹尼儿北利
24	丹尼儿北利
30	丹尼儿北利
36	丹尼儿北利
40	丹尼儿北利
60	丹尼儿北利
72	丹尼儿北利

Bitmapped

```
%! Hebrew−Frank−Ruehl font
% Copyright 1986 Amiram+Omri
/Hebrew−Frank−RuehlFont 8 dict def Hebrew−Frank−RuehlFont begin
/FontType 3 def /FontMatrix [.008333 0 0 .008333 0 0] def
/FontBBox [−2 −39 65 91] def /CharStrings 112 dict def

CharStrings begin

...

/alef{ 31 35 moveto
11 58 lineto 9 60 10 62 9 62 curveto 6 62 2 55 2 52 curveto
2 49 4 48 6 46 curveto 14.5 36 lineto 10 32 4 26 4 20 curveto
4 14 10 10 10 6 curveto 10 3 4 6 4 0 curveto 20 0 lineto
21 2 22 4 22 6 curveto 22 14 13 16 13 26 curveto 13 28 14 31 17 33 curveto
40 4 lineto 42 2 41 0 43 0.2 curveto 46 0 50 5 50 8 curveto
49.8 11 49 12 46 16 curveto 38 26 lineto 34 32 40 40 44 45
curveto 46 42 3.6 123.7 33.7 arcn 50 54 lineto
44 54 6 0 90 arc 38 60 lineto 38 62 2 −90 180 arcn
36 64 34 65 33 62 curveto 30 56 30 47.4 34 47 curveto
40 46 lineto 31 35 lineto }def
```

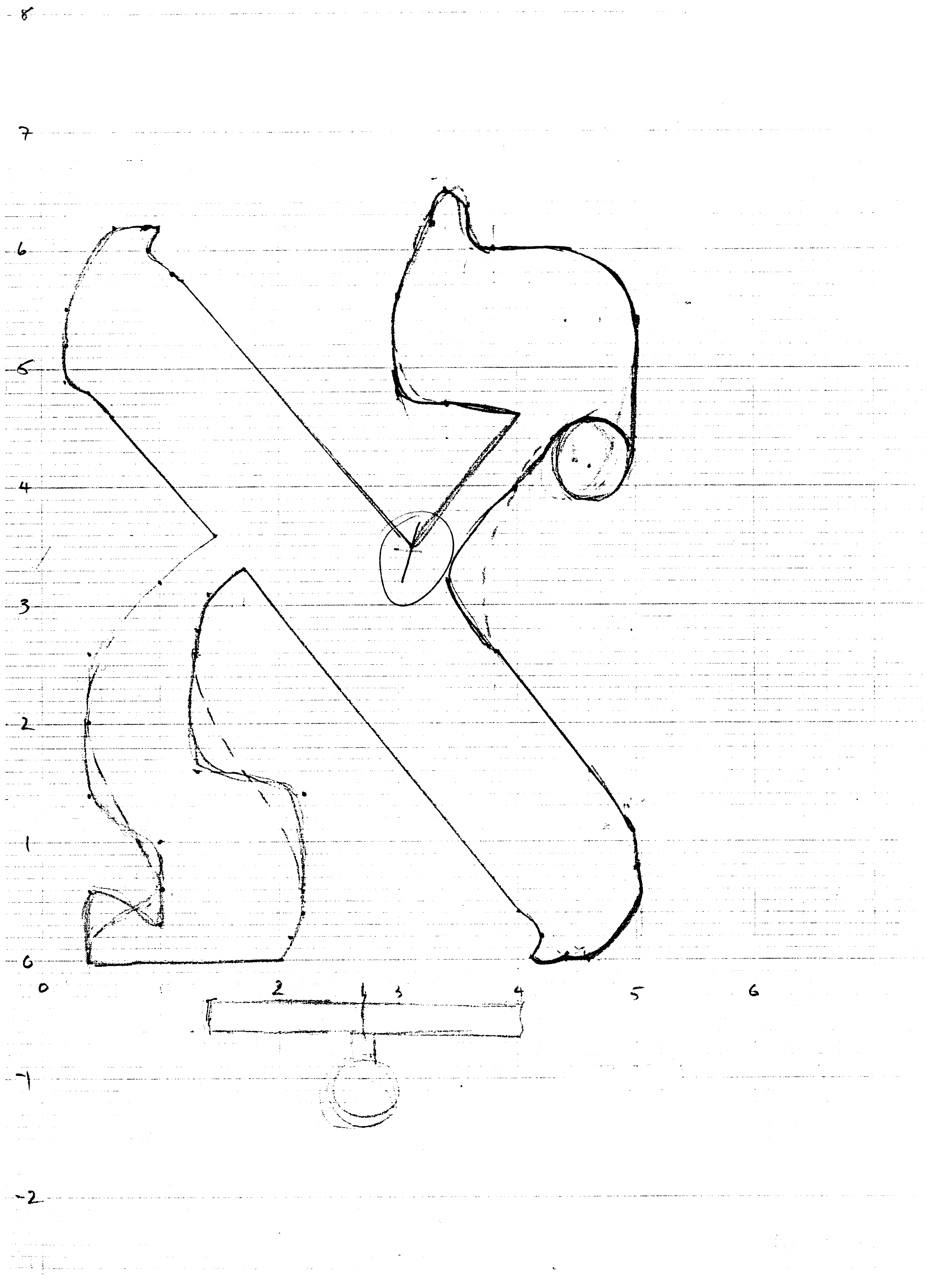
Type 3 PS definition of an outline font

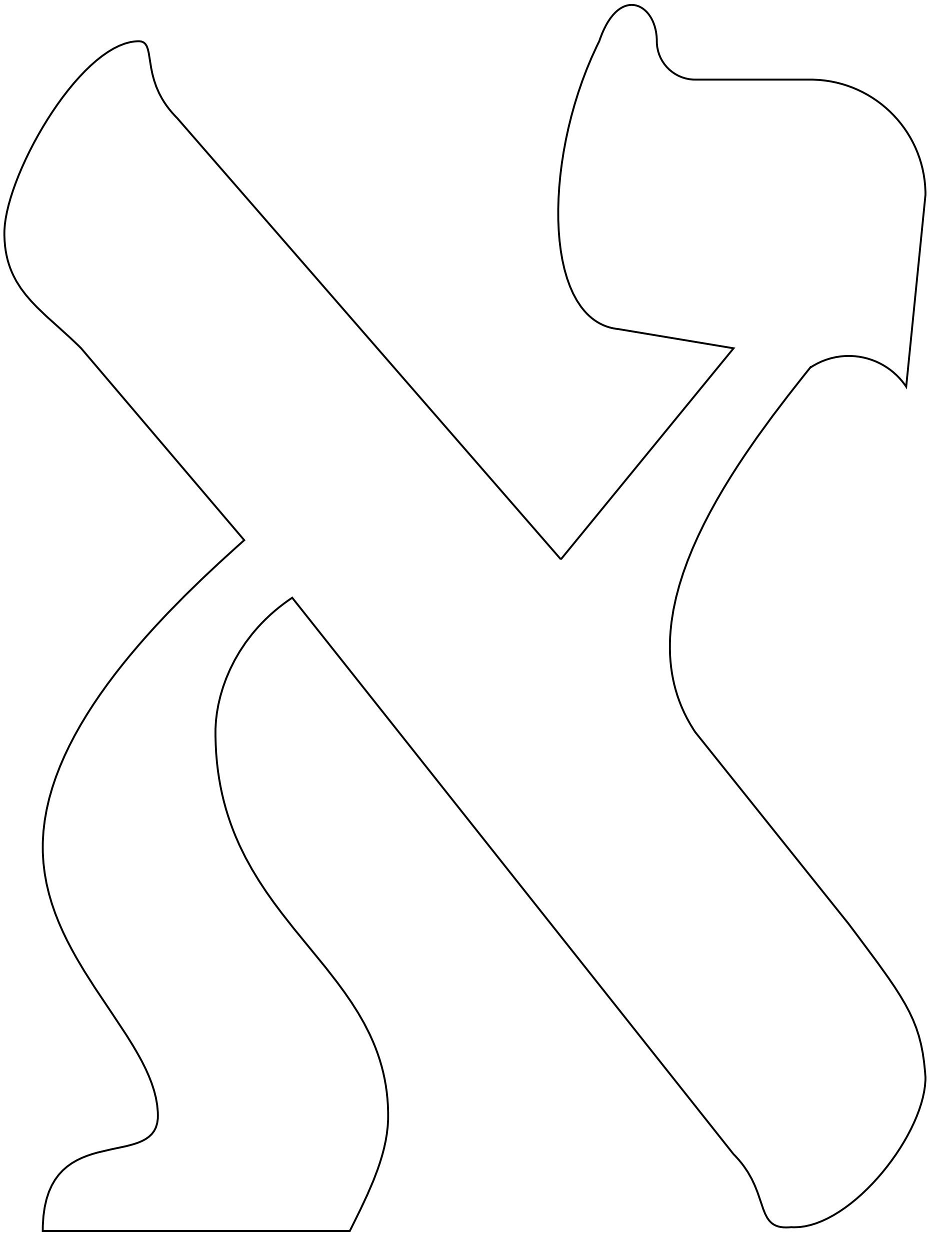
```
/bet{ 10 62 moveto
10 64 8 65 7.2 62 curveto 4 56 4 46 8 46 curveto 30 46 lineto
38 46 40 14 26 14 curveto 6 14 lineto 2 0 lineto 44 0 lineto 48 14 lineto
34 14 lineto 43 14 48 60 34 60 curveto 12 60 lineto 12 62 2 −90 180 arcn
}def
```

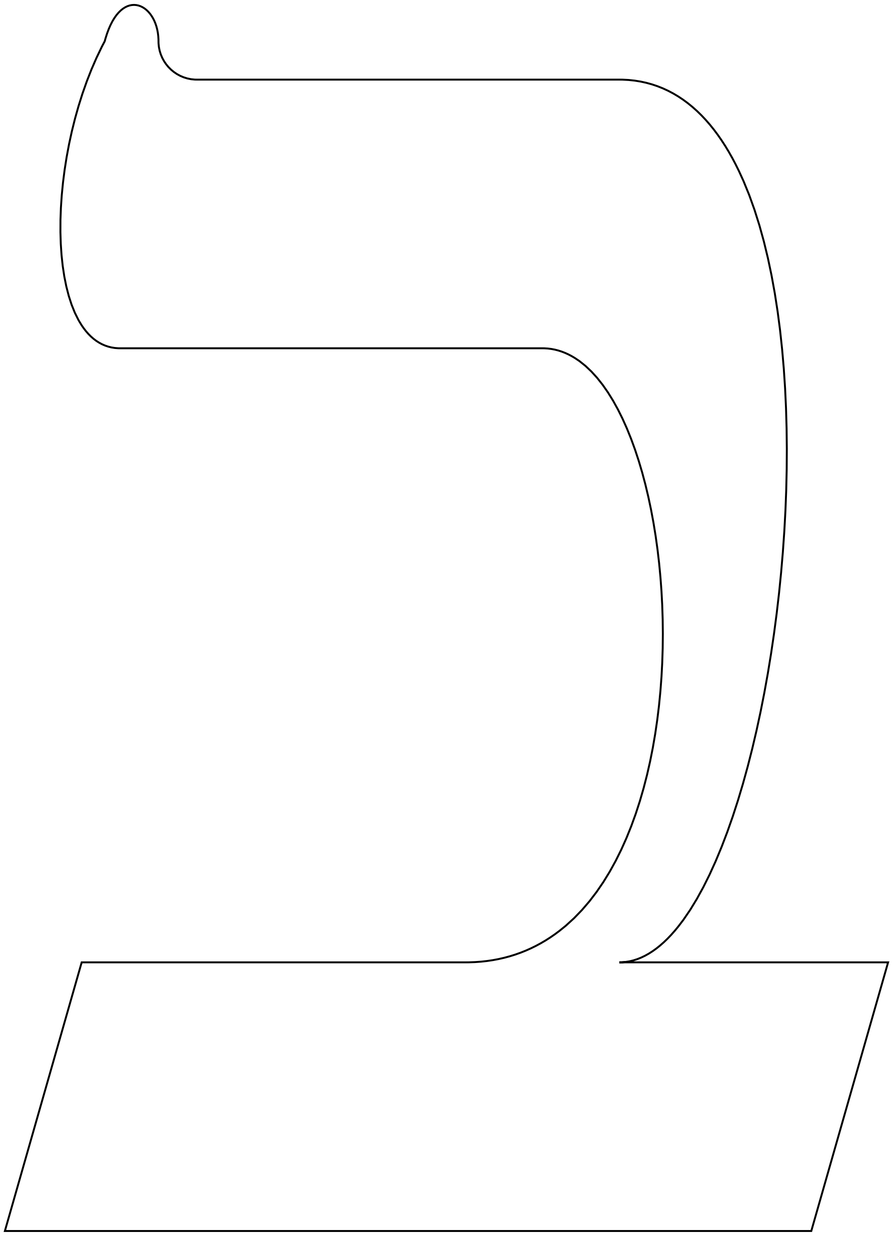
```
/alefqamatz{ 31 35 moveto
11 58 lineto 9 60 10 62 9 62 curveto 6 62 2 55 2 52 curveto
2 49 4 48 6 46 curveto 14.5 36 lineto 10 32 4 26 4 20 curveto
4 14 10 10 10 6 curveto 10 3 4 6 4 0 curveto 20 0 lineto
21 2 22 4 22 6 curveto 22 14 13 16 13 26 curveto 13 28 14 31 17 33 curveto
40 4 lineto 42 2 41 0 43 0.2 curveto 46 0 50 5 50 8 curveto
49.8 11 49 12 46 16 curveto 38 26 lineto 34 32 40 40 44 45
curveto 46 42 3.6 123.7 33.7 arcn 50 54 lineto
44 54 6 0 90 arc 38 60 lineto 38 62 2 −90 180 arcn
36 64 34 65 33 62 curveto 30 56 30 47.4 34 47 curveto
40 46 lineto 31 35 lineto
10 −9 moveto
40 −9 lineto 40 −15 lineto
28 −15 lineto 28 −24 lineto
28 −25 29 −26 29.5 −29 curveto
29 −31 27 −33 25 −33 curveto
23 −33 21 −31 20.5 −29 curveto
21 −26 22 −25 22 −24 curveto
22 −15 lineto 10 −15 lineto
10 −9 lineto }def
```

```
...
end
/Encoding 256 array def
0 1 255{Encoding exch /.notdef put}for
...
dup 81 /alefqamatz put
...
dup 96 /alef put
dup 97 /bet put
...

/Metrics 112 dict def Metrics begin
...
/alefqamatz 52 5 add def
...
/alef 52 5 add def
/bet 50 5 add def
...
end
```

```
/BuildChar{ /CharCode exch def
begin Metrics Encoding CharCode get get 0 FontBBox aload pop setcachedevice
%CharStrings Encoding CharCode get get end exec fill }def end
CharStrings Encoding CharCode get get end exec stroke }def end

/Hebrew−Frank−Ruehl Hebrew−Frank−RuehlFont definefont pop
/Hebrew−Frank−Ruehl findfont 867 scalefont setfont

150 300 moveto
(‘)show                                  showpage

150 300 moveto
(a)show                                  showpage

150 300 moveto
(Q)show                                  showpage

/Hebrew−Frank−Ruehl findfont 20 scalefont setfont
150 300 moveto
(‘Q)show                                   showpage
```


8
7
6
6
4
3
2
1
0
-1
-2
0
2
3
4
5
6





אָ

אָא

א

ב

אָ

אָא

```
%!
/Times−Roman findfont 300 scalefont setfont
25 210 moveto (Daniel M. Berry) show
25 420 moveto (Daniel M. Berry) false charpath .5 setlinewidth stroke
showpage
```

Dani

Dani

Dani

Dani

Dani

Dani

ברי

ברי

אל ברי

אל ברי

ל ברי

ל ברי

ל ברי

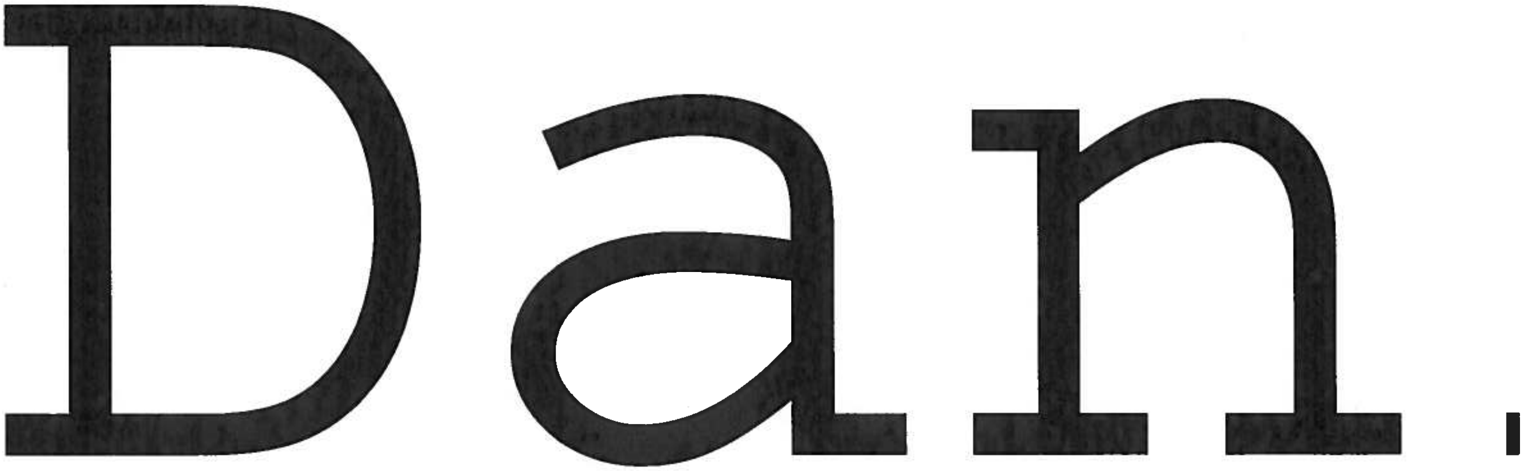
ל ברי

Dan

Stroked Courier, scanned in because current printers do not have it any more

Dan:

Same file run later, now with outline font



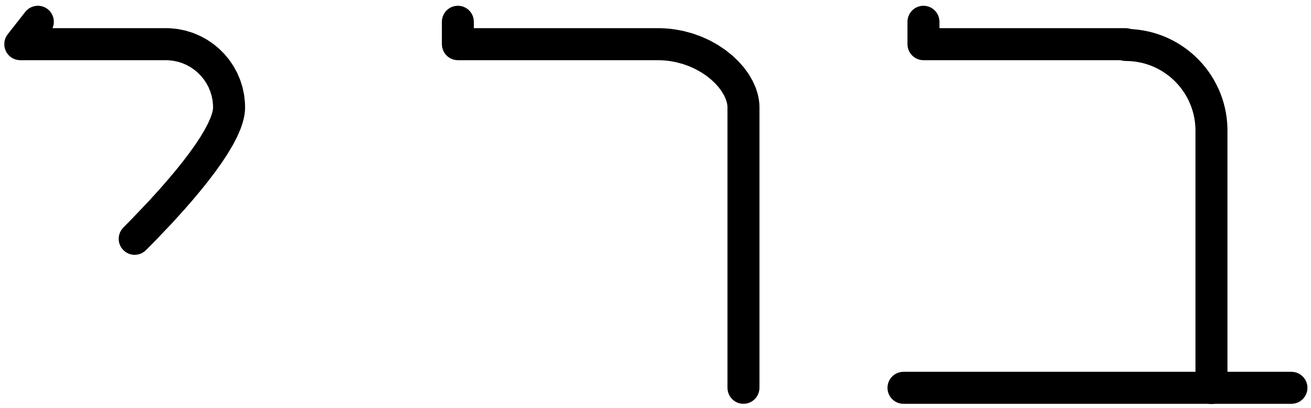
Dan.

Another style of Courier (I don't know where and when I did this, but it was in North America after 1998, because it's on Letter paper. Scanned in because it's not in printer now.

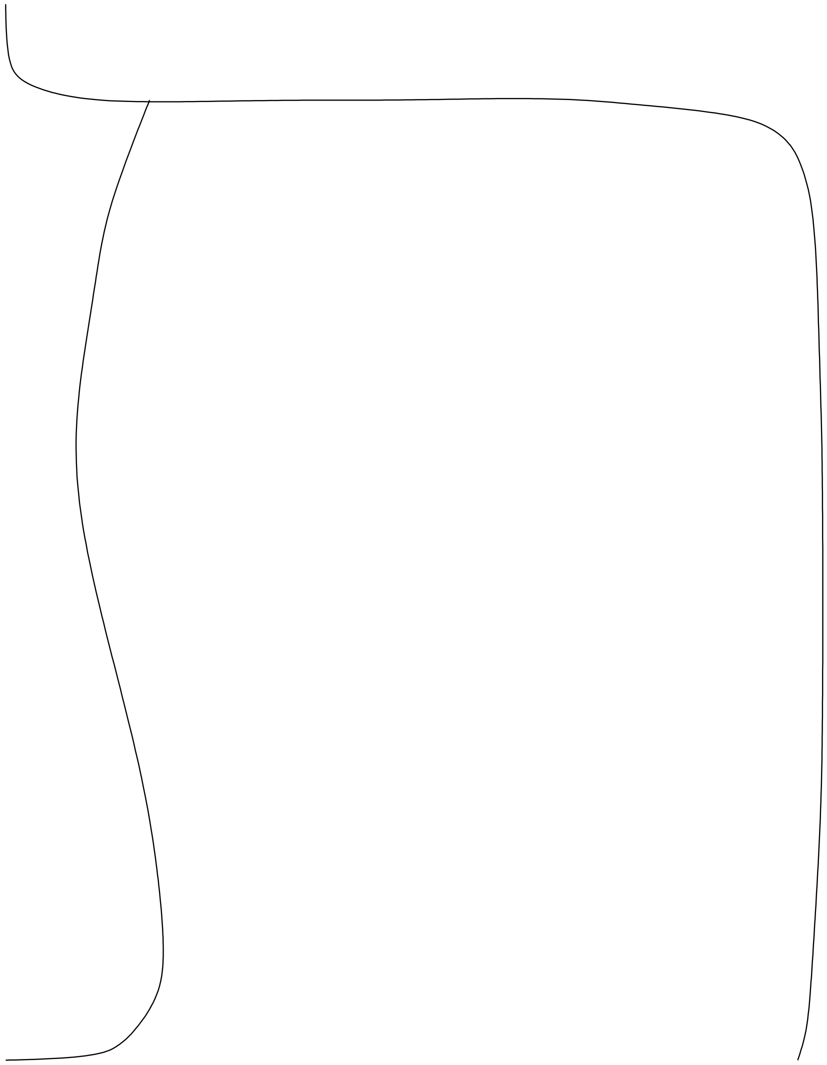
Dan

Dan

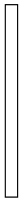
Outline Courier with outline



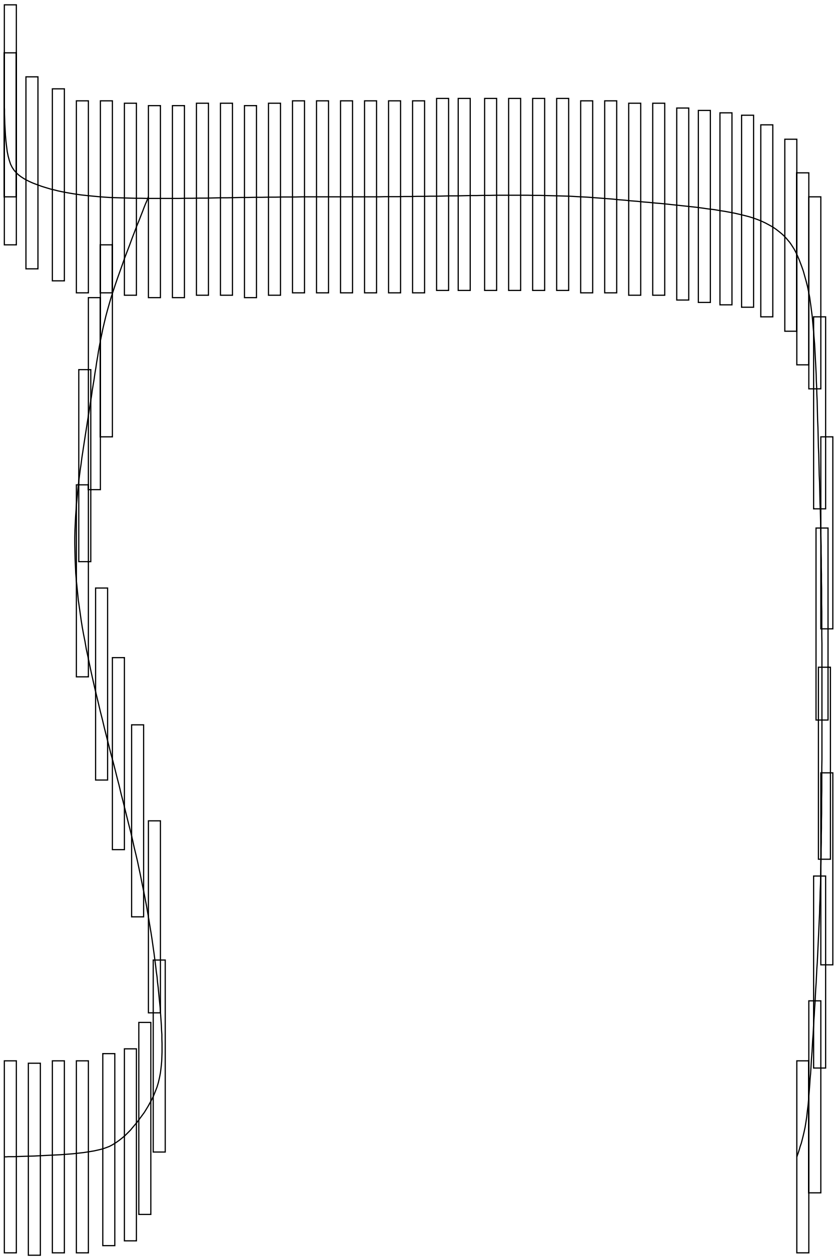
Stroke font



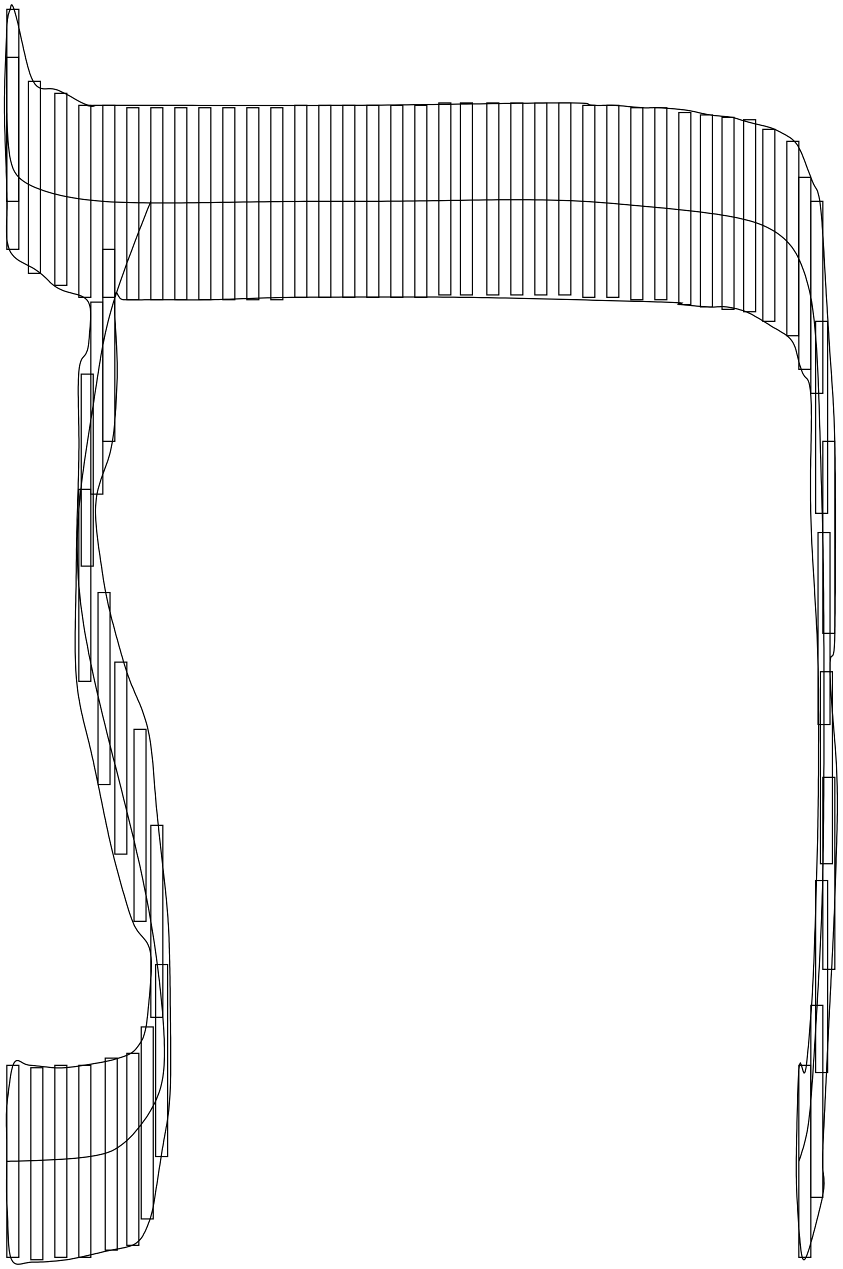
Center Line Stroke for TAV



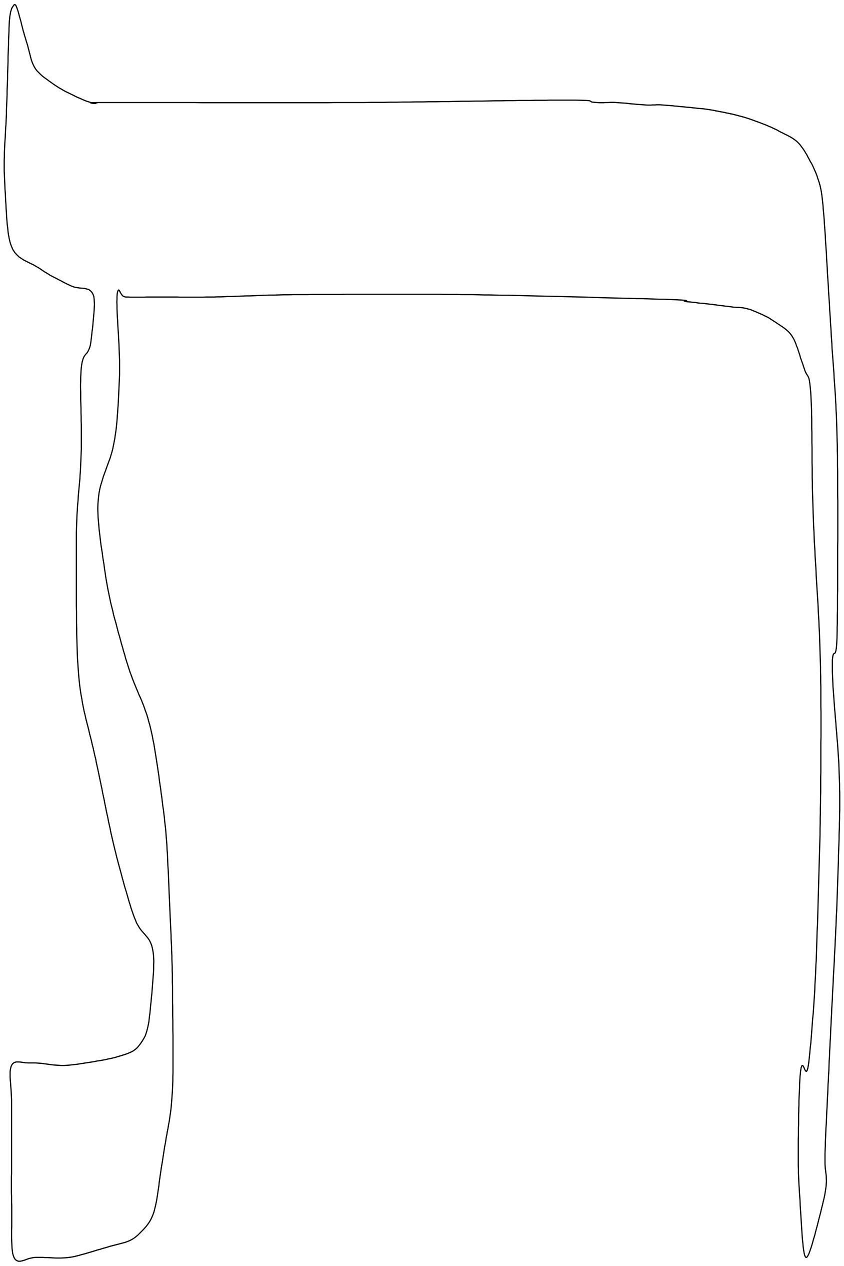
Pen (rectangular)



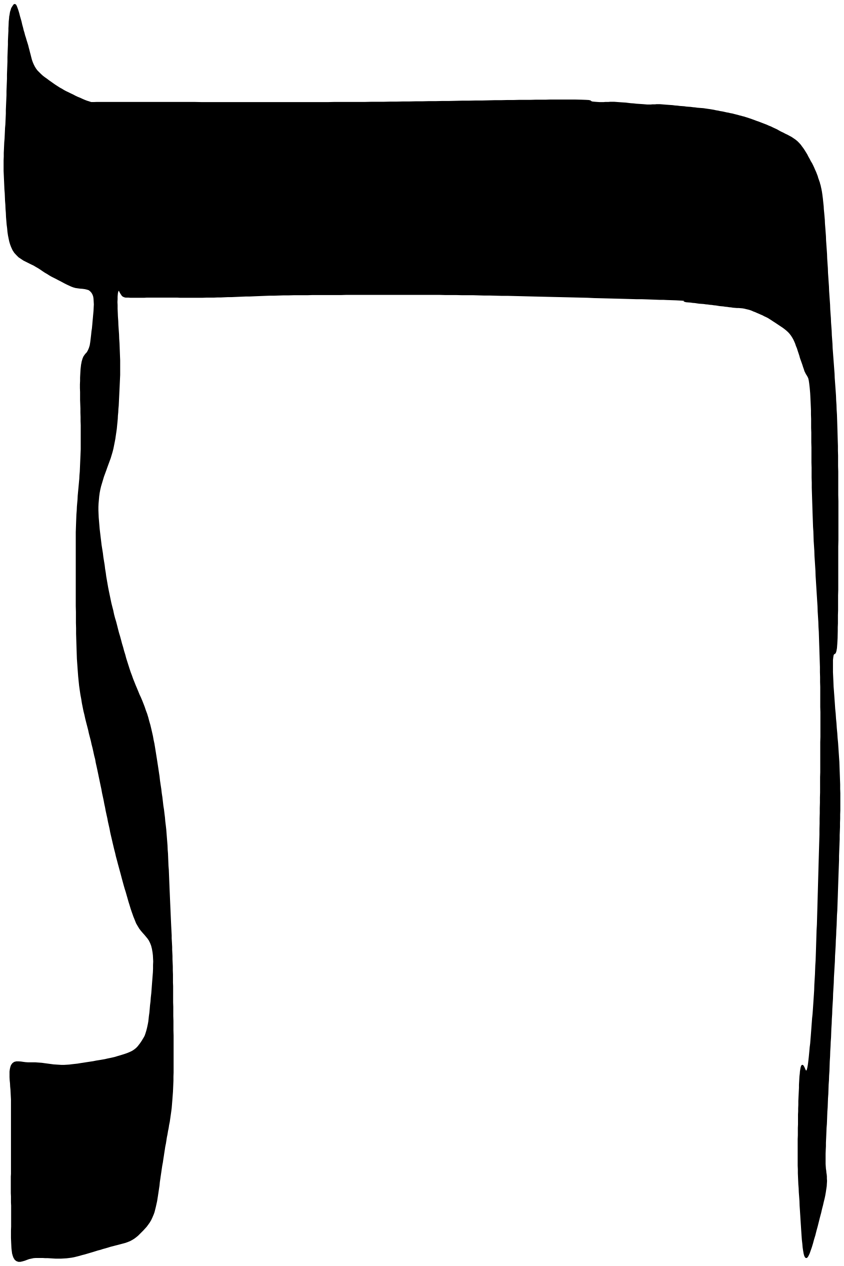
Penned Path



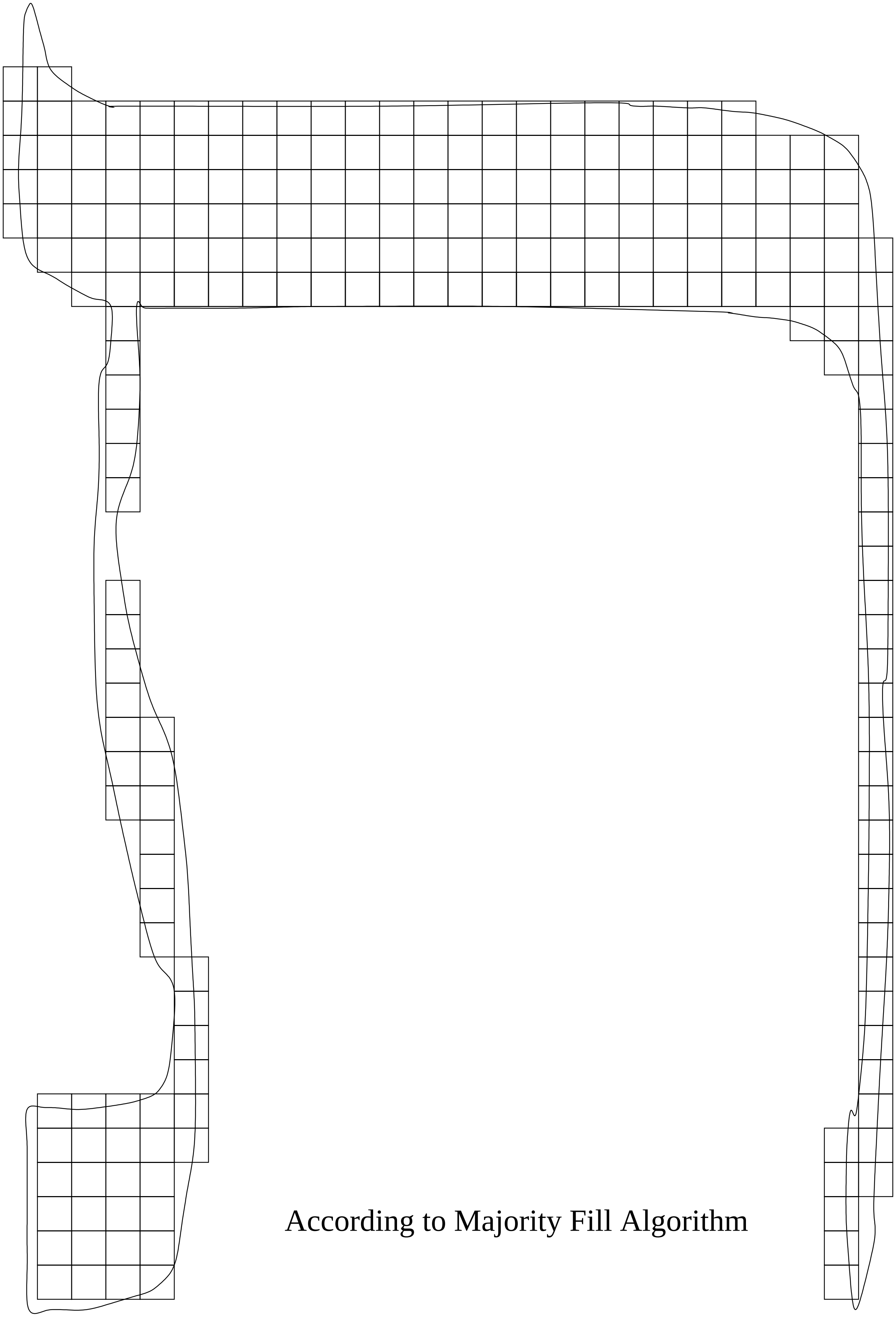
Envelope Generated by Penning Path



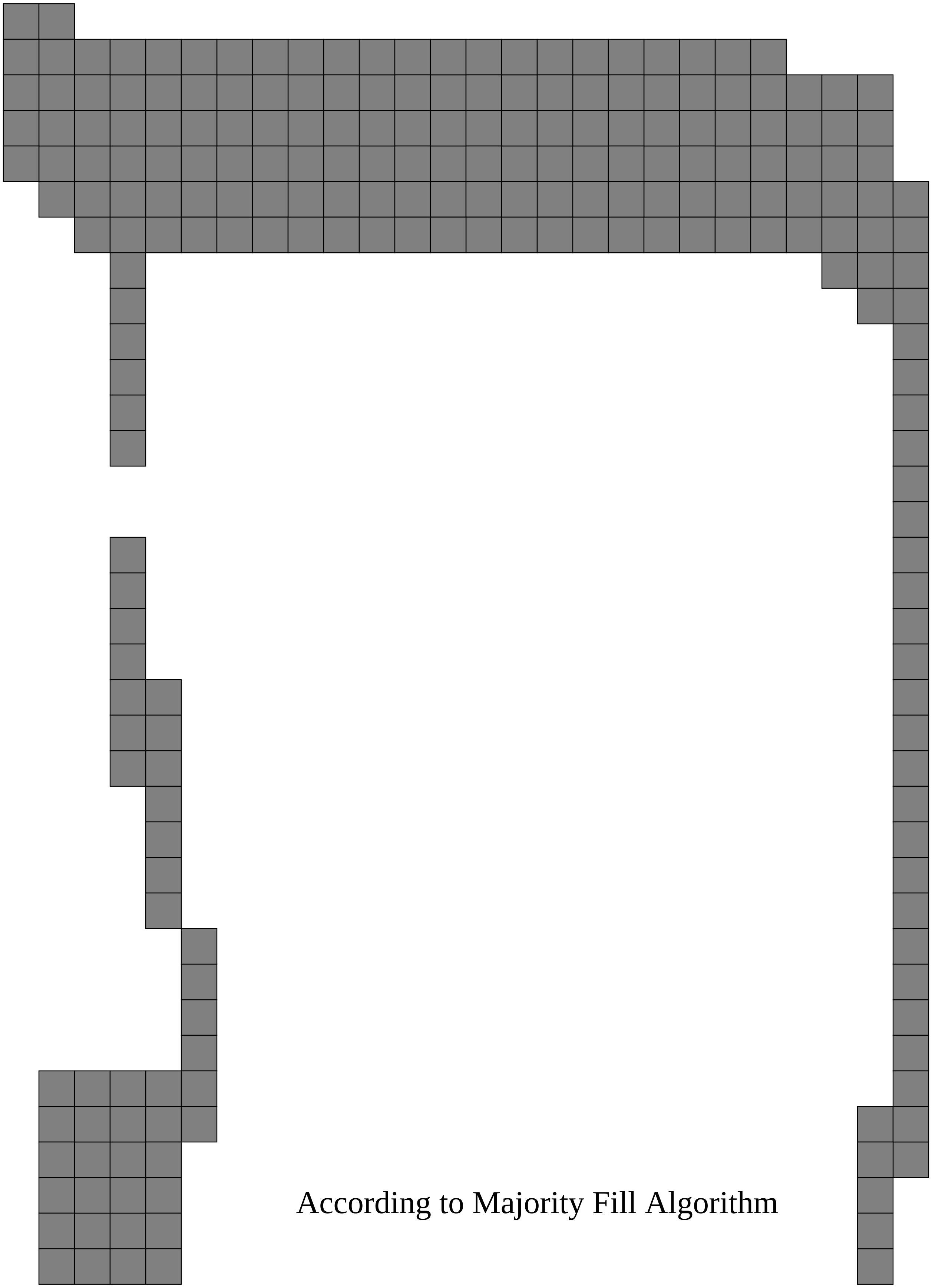
Generated Envelope Only



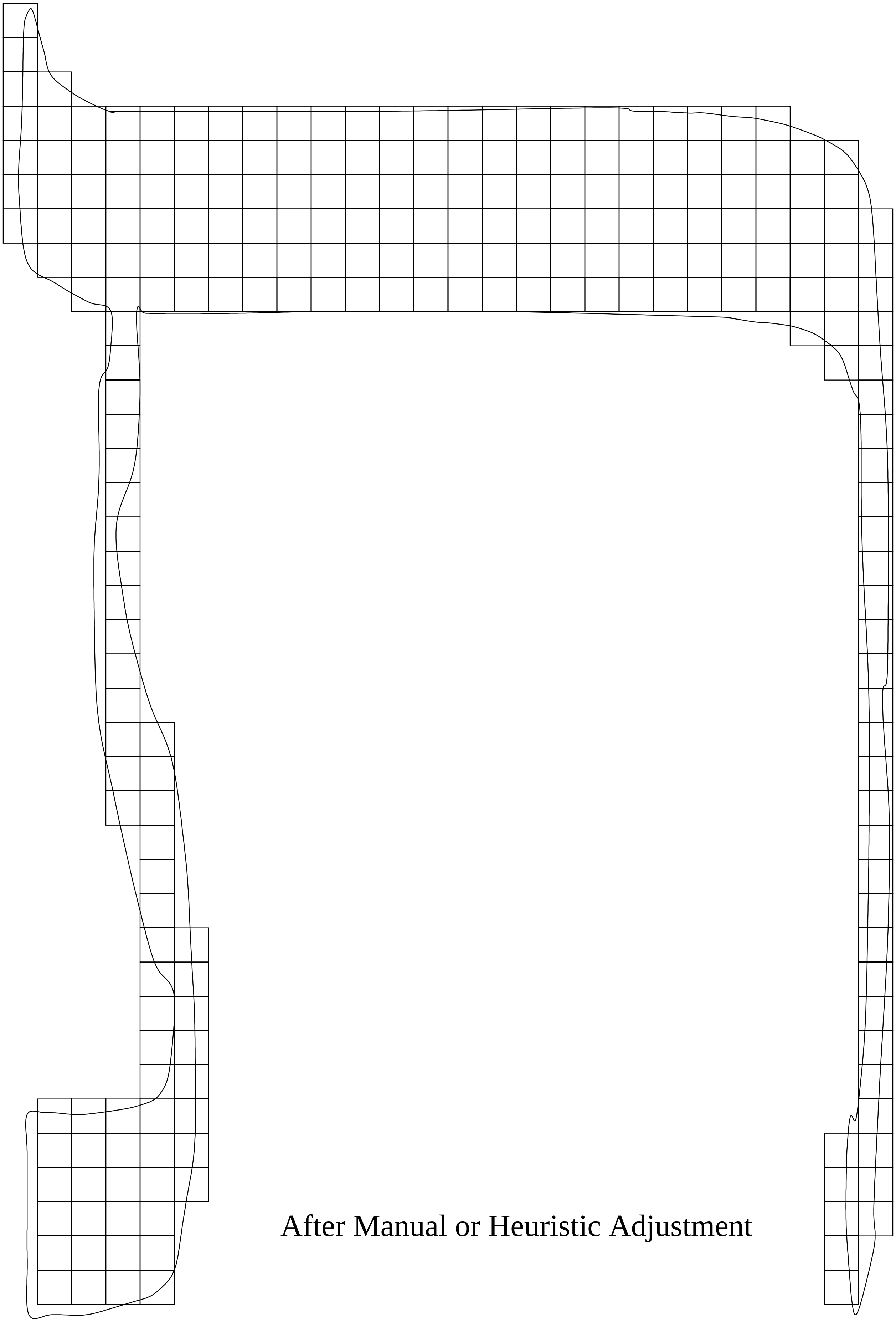
Generated Envelope Only



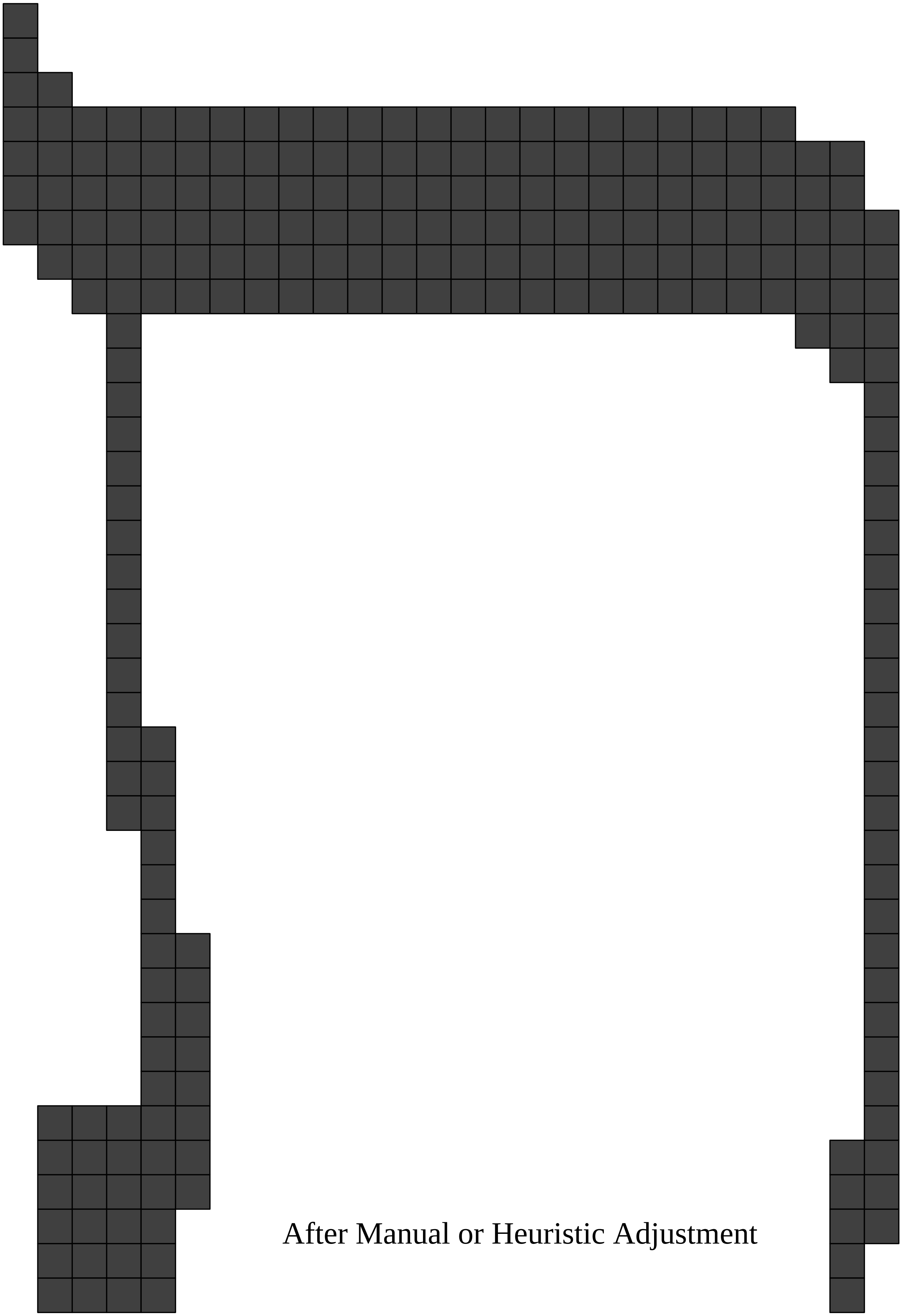
According to Majority Fill Algorithm



According to Majority Fill Algorithm

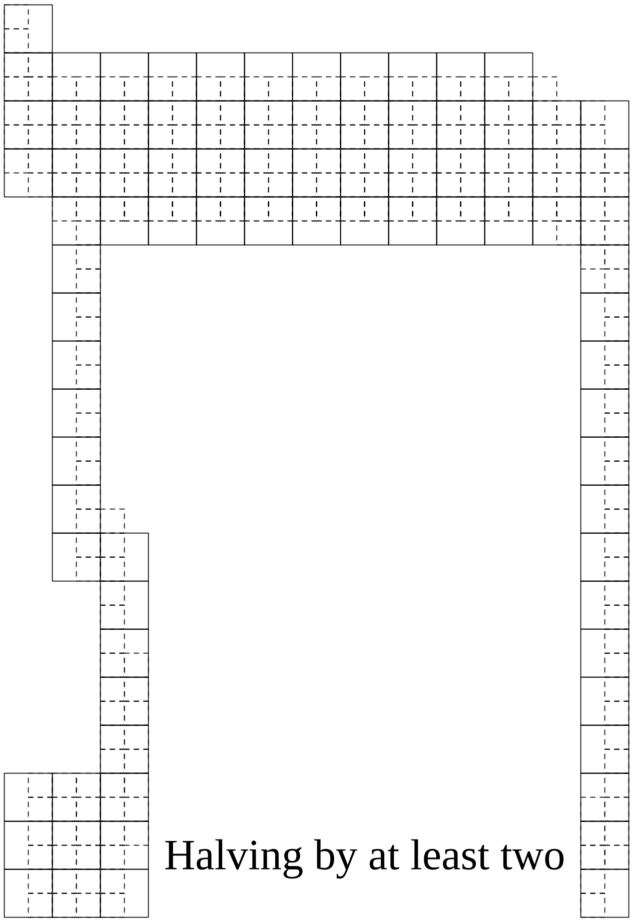


After Manual or Heuristic Adjustment

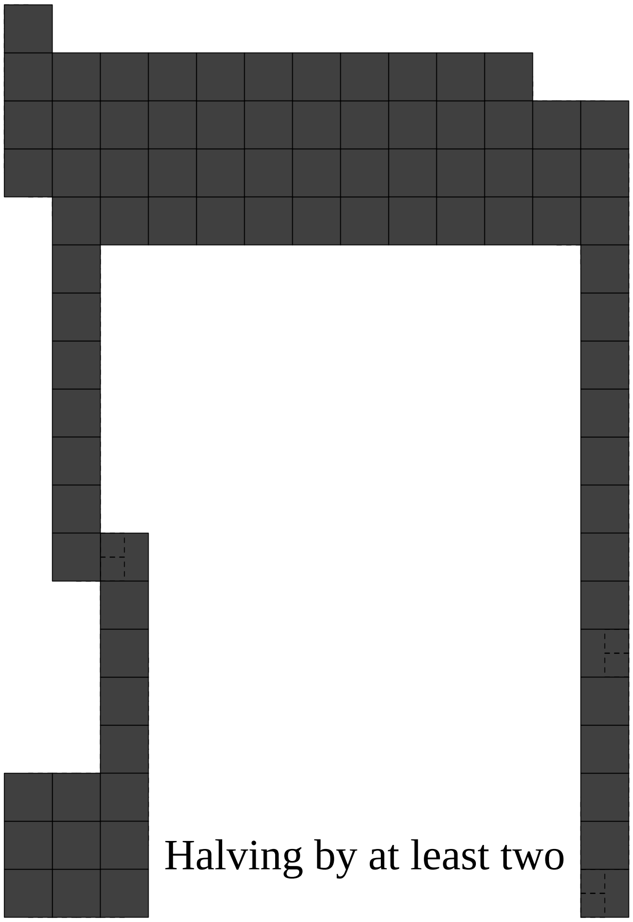


After Manual or Heuristic Adjustment

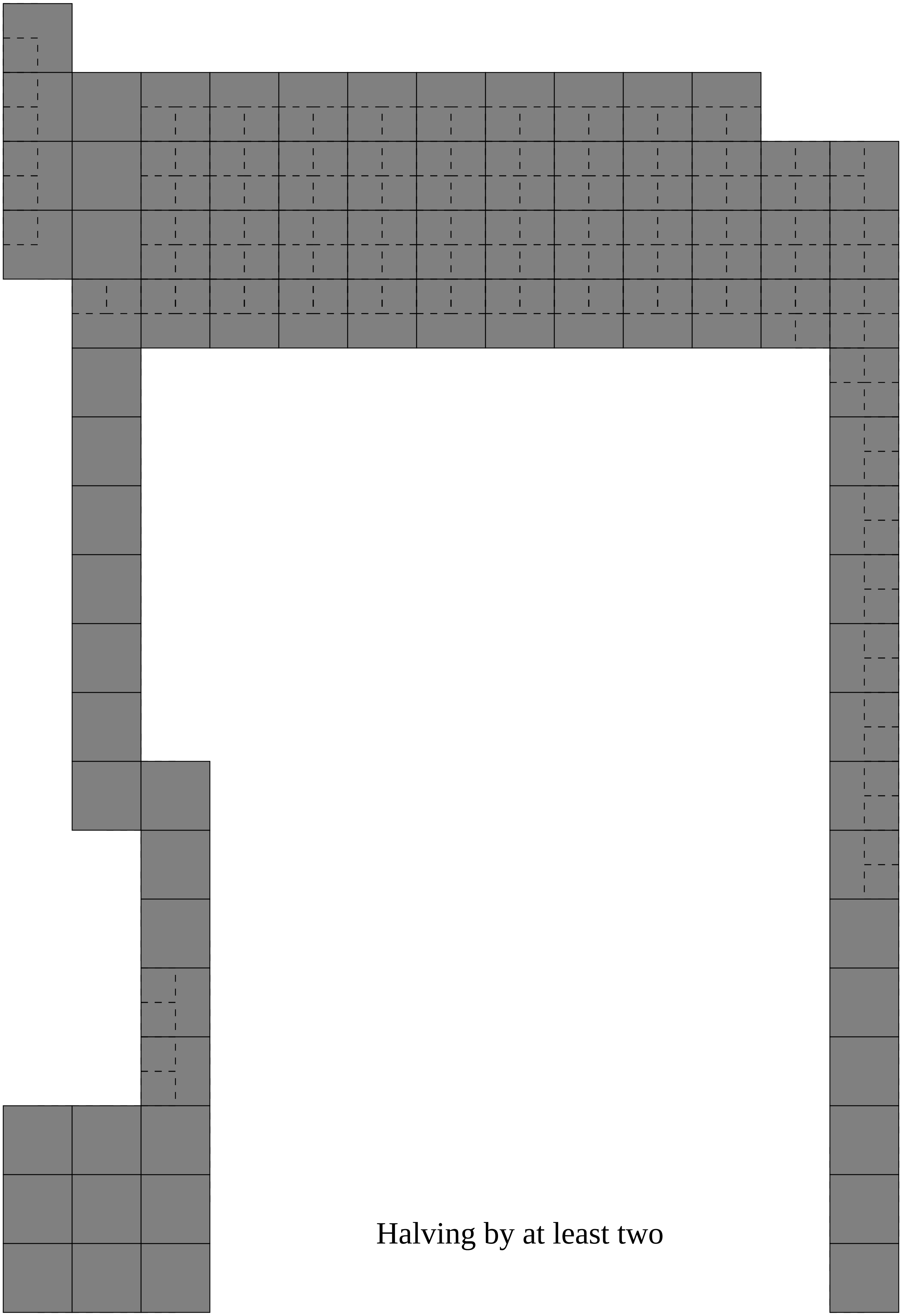
ת



Halving by at least two

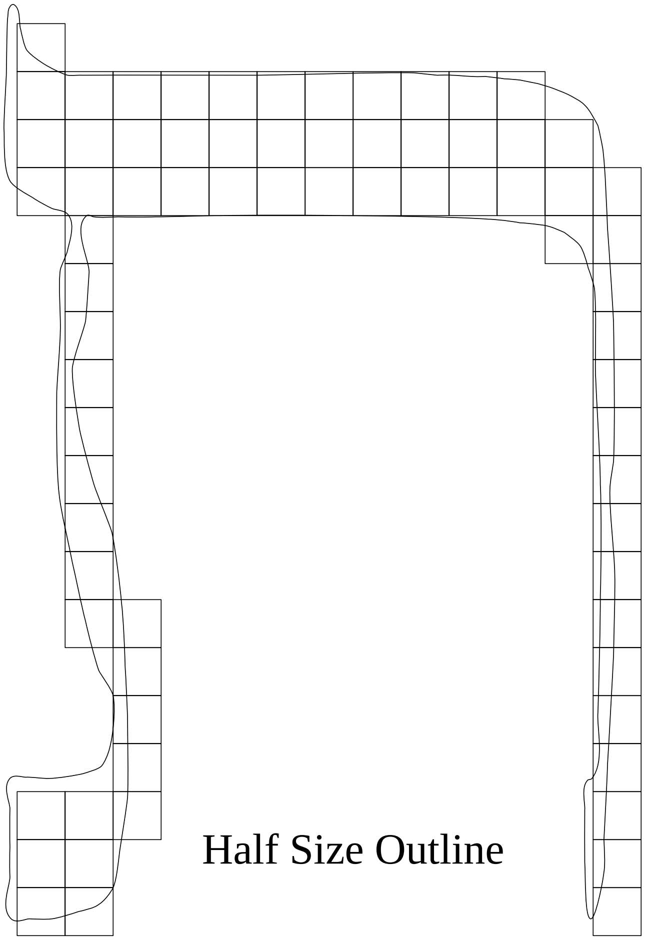


Halving by at least two

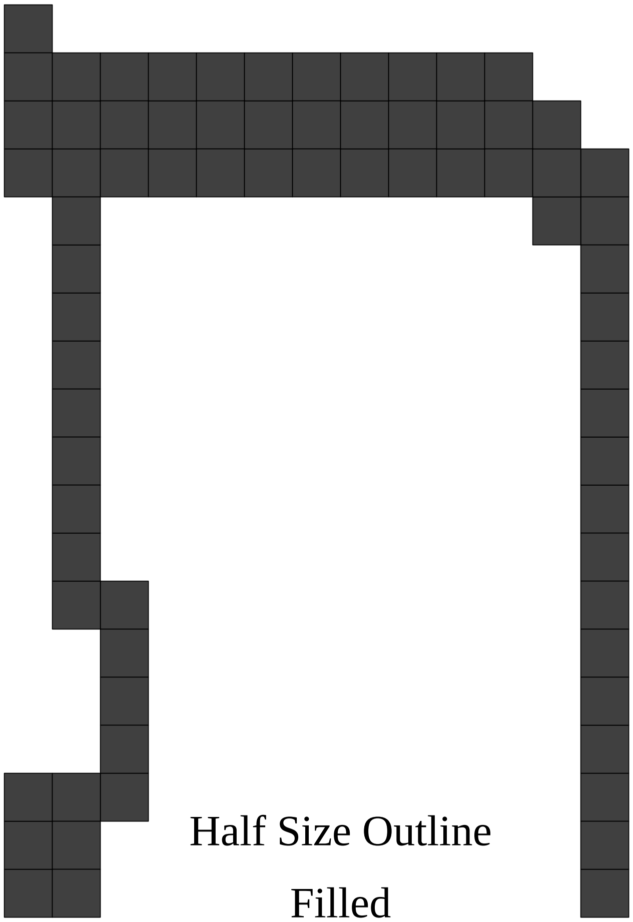


Halving by at least two

ת



Half Size Outline



Half Size Outline

Filled