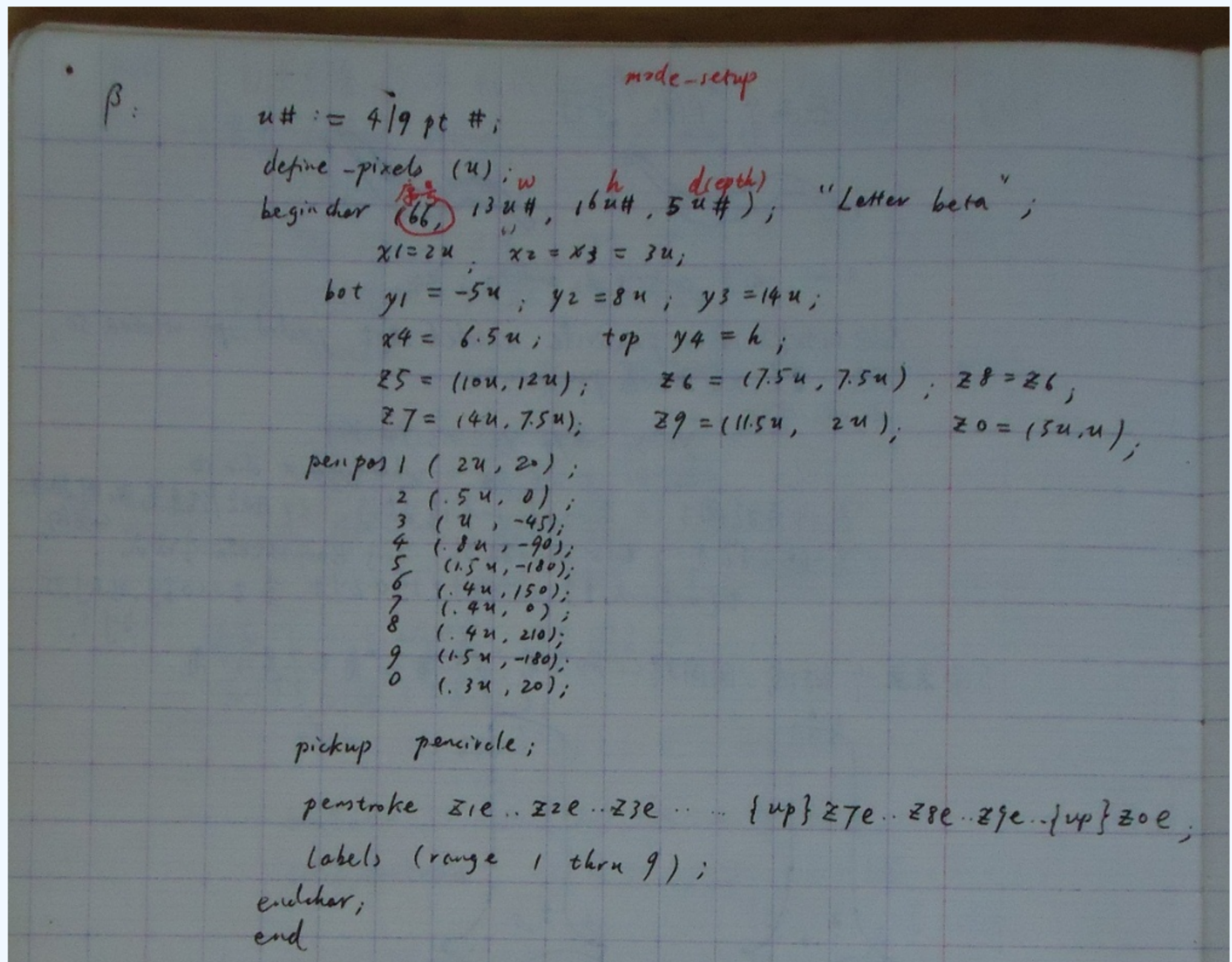
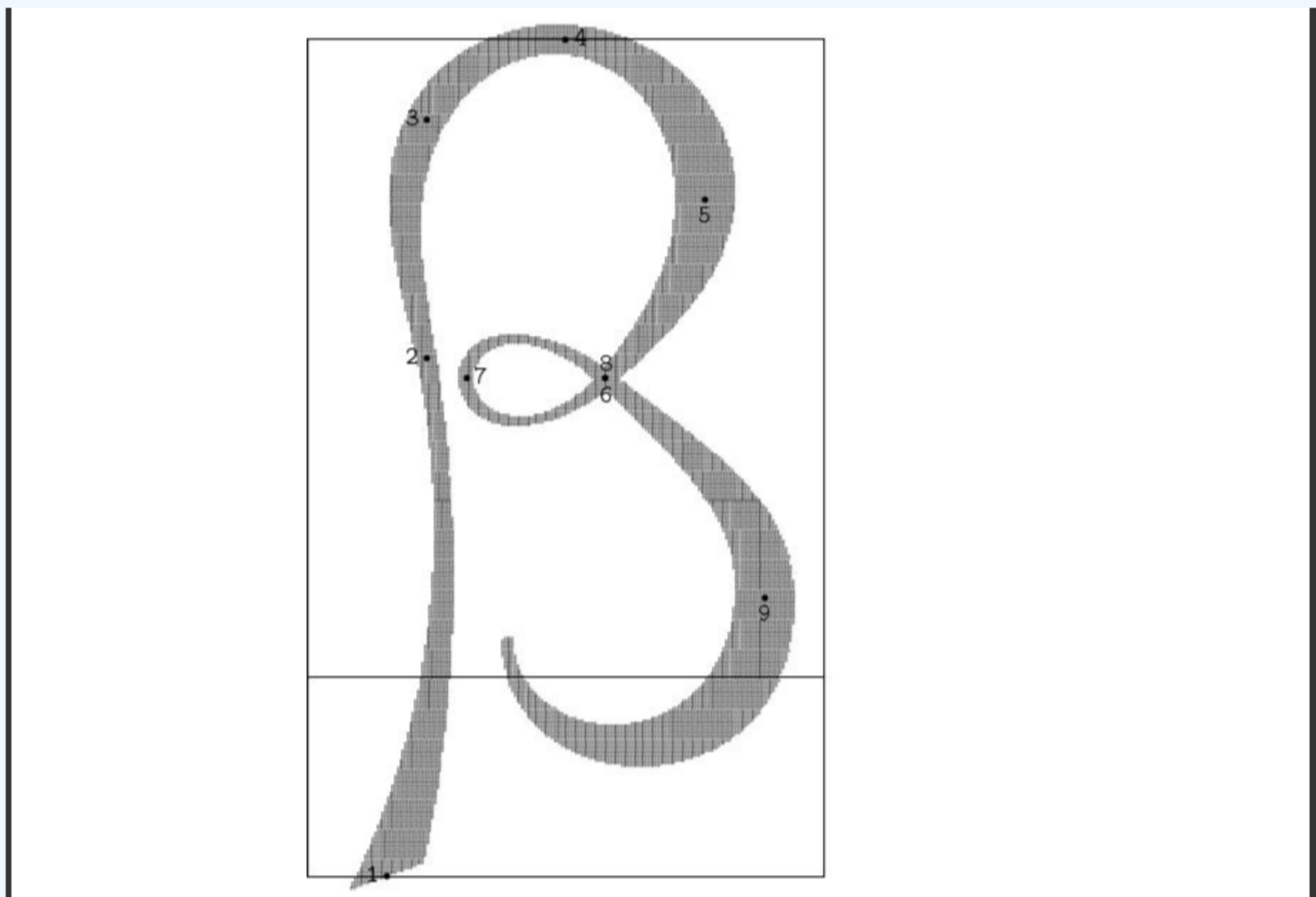


本站Web首页的制作：使用METAFONT

首页链接到中文版和英文版，为了显示各自的特点，指示中文版处使用中文汉字书法图片作背景，指示英文版处又想以一定的字体风格来显示“English”单词的这几个字母，让字母带点汉字隶书的风格。所以制作网页时的主要工作，就是“English”的几个字母的字体设计。这时就用上METAFONT了，边学边做。从Christophe Grandsire的[The METAFONT tutorial](#)(mftut.pdf)（wikipedia Metafont词条中有链接）开始，其中第一个例子是：



这生成一个字母beta:



陆陆续续了解到一些更多的命令，觉得可能用到，罗列如下：



draw ?

whatever $[z_1, z_2]$.

dir 90 angle vector

shifted (x, y) scaled rotated around reflected about

pickup pencircle xscaled 10 yscaled 35 rotated (45)

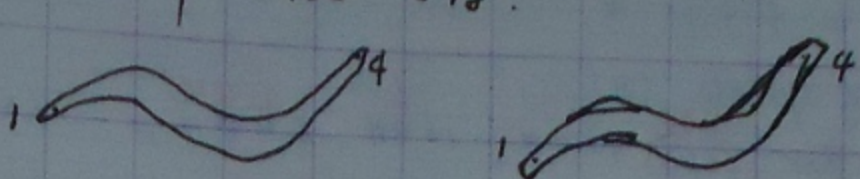
pickup pencircle scaled 15 xscaled Phi rotated angle (Phi, 1);

draw $z_1 \{right\} \dots z_2 \{up\} \dots \{left\} z_3$;

fill
filldraw
drawdot

最能体现隶书风格的笔画在字母E和g上。但套用最多的模式是笔划起始终止处的衬线，所以对衬线的画法做了较详细的了解。摘抄翻译了一些 The METAFONTbook中讲到衬线画法的内容。
书中先是在p152讲到一处画衬线。然后在p162开始较详细地讲了画衬线的方法。
p152处：

p152. 先写两个 "tilde" 字符。



draw $z_1 \dots controls z_2 \text{ and } z_3 \dots z_4$

左边的例子用的 pencircle xscaled .8pt yscaled .2pt rotated 50,
右边一样,只是用 m 是 pensquare.

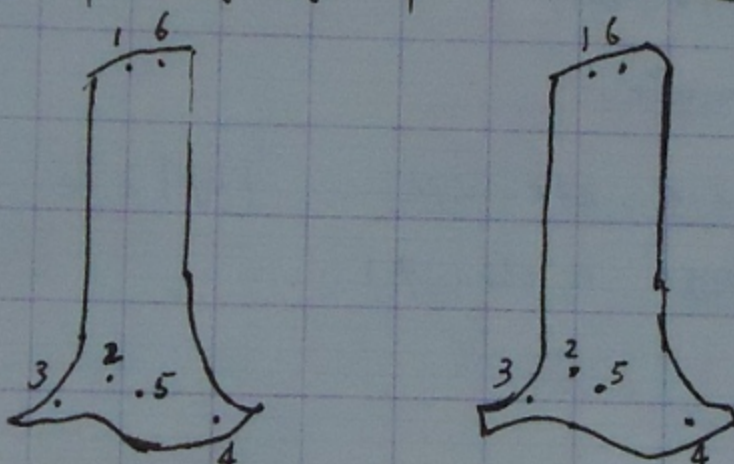
z_2, z_3 定义:

$$y_2 - y_1 = y_4 - y_3 = 3(y_4 - y_1);$$

$$z_2 - z_1 = z_4 - z_3 = \text{whatever} * \text{dir } 50$$

第二个方程用了一个老的美文书法技巧, 即开始/结束笔画用执笔
第一组方程是一个数学上的技巧, 基于 Bernstein 多项式
 $t \in [0, 3, -2, 1]$ 从 0 到 1 到 0 到 1 当 t 从 0 到 .25 到 .75 到 1.

下面来画一个衬线, 用同样的两支笔. 只是用 20° 角而不是 50° 角.



可用 'filldraw' 命令来画:

filldraw $z_1 \dots controls z_2 \dots z_3$

flex (z_1, z_2, z_3)

表示路径
 $z_1 \dots z_2 \{z_3 - z_1\} \dots z_3$

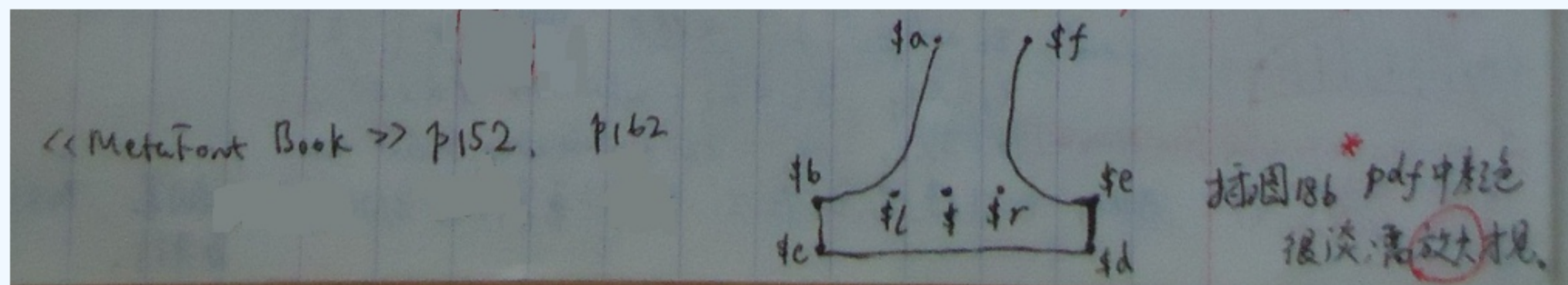
flex (z_1, z_2, z_3, z_4)

表示
 $z_1 \dots z_2 \{z_4 - z_1\} \dots z_3 \{z_4 - z_3\} \dots z_4$

dishing 参数在 z_3 和 z_4 之间产生一个微小的升起. flex 被
用 epsilon 降低来避免 "strange paths" 的危险, 否则可能
会被导致在 z_3 或 z_4 的 "tiny loops". 但该例子最有趣的是
用了双控制点, z_2 和 z_5 . (注意 'controls z_2 ' 等同于
controls z_2 and z_2'). 这些点 回忆 如下决定:
 $x_2 = x_1, z_2 = z_3 + \text{whatever} * \text{dir } 20,$

$z_5 = z_6$, $z_5 = z_4 + \text{whatever} * \text{dir} - 20$;
 因此, 它们使笔画在 z_1, z_6 处垂直, 在 z_3 处平行于钢笔, z_4 处反平行。

再看p162处:
 书中先后讲述了两不同方法。
 插图186给出了一种待画衬线的形状(使用第一种方法画的):



然后开始先后讲述:

p162 现在来看一个画衬线的例子。... 我们考虑两种不同的方法。
 - 一个基于 outline-filling, 一个基于 use of a fixed pen nib. ...
 (图186, 见前)
 点 $z_{fa}, z_{fb}, \dots, z_{ff}$, 3个 "penpos" 点 z_{fl}, z_f, z_{fr} ,
 f 是 serif 宽的一个参数。其他参数是:
 breadth: $z_{fl} \rightarrow z_{fa}, z_{fr} \rightarrow z_{ff}$ 两条平行线间距;
 theta: 上边两条线在 f 方向角。
 left-jut: z_{fl} 到 z_{fb} 的距离;
 right-jut: z_{fr} 到 z_{fe} 的距离;
 (~~serif~~ 衬线通过 jut 参数的是 "juts out")
 还有一个 serif-edge 宏, 产生图186路径。
 例程引用3个变量:
 slab: 从 b 到 e 的垂直距离;
 (z_{fb})
 bracket: 从 a 到 f 到 c 和 d 的垂直距离;
 serif-darkness, 一个分数, 控制三角区域 (a, l, b)
 和 (f, r, e) 填充多少。

```

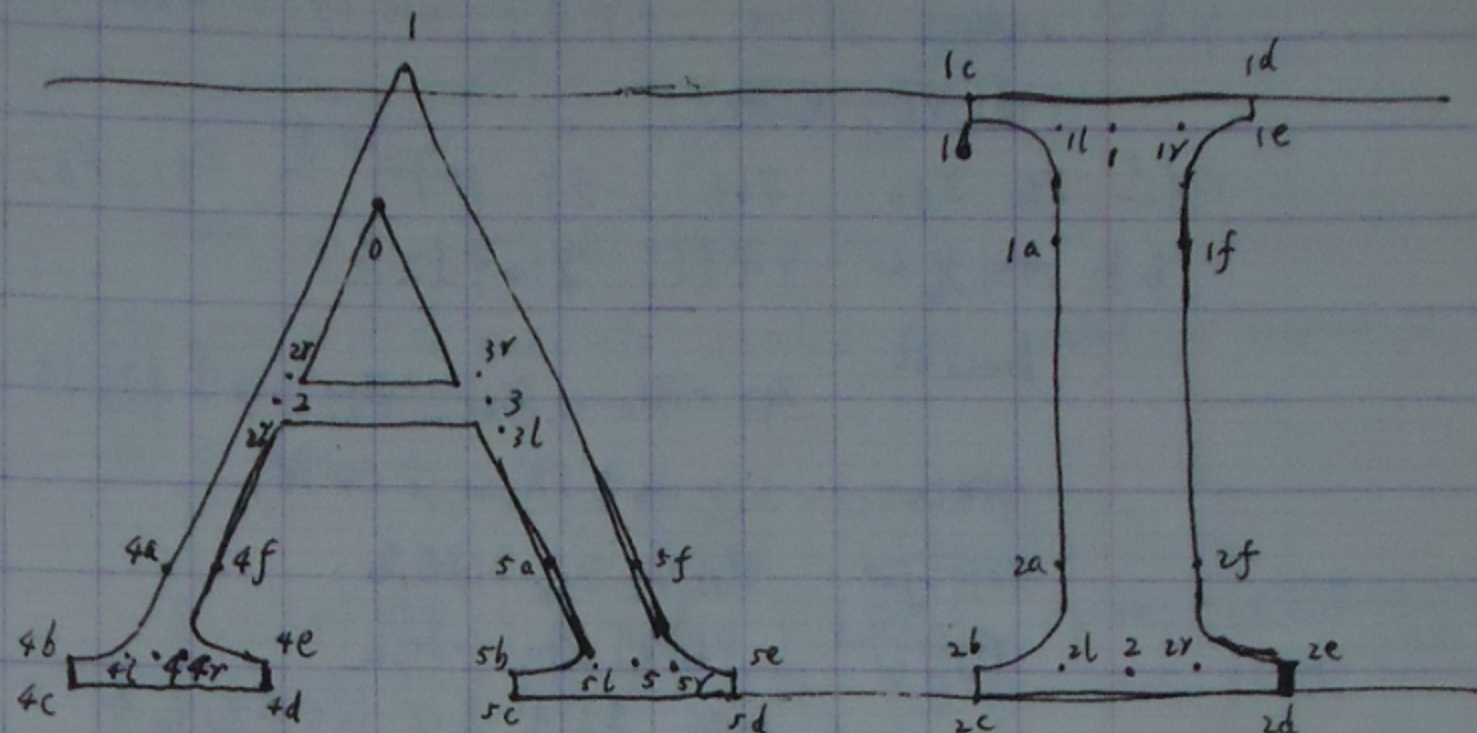
def serif (suffix $) (expr breadth, theta, left-jut, right-jut) =
  penpos $ (breadth / abs sind theta, 0);
   $z_{fa} - z_{fl} = z_{ff} - z_{fr} = (\text{bracket} / \text{abs sind theta}) * \text{dir theta}$ ;
   $y_{fc} = y_{fd}; y_{fb} = y_{fe} = y_f$ ;  $y_{fb} - y_{fc} = \text{if theta} < 0 : -\text{fi slab}$ ;
   $x_{fb} = x_{fc} = x_{fl} - \text{left-jut}$ ;  $x_{fd} = x_{fe} = x_{fr} + \text{right-jut}$ ;
  labels ($a, $b, $c, $d, $e, $f) enddef;

def serif-edge suffix $ =
  (serif-bracket ($a, $l, $b) --  $z_{fc}$ 
   --  $z_{fd}$  -- reverse serif-bracket ($f, $r, $e)) enddef;

def serif-bracket (suffix i, j, k) =
  (  $z.i \{ z.j - z.i \} \dots \text{serif-darkness} [z.j, .5[z.i, z.k]] \{ z.k - z.i \}$ 
     $\dots z.k \{ z.k - z.j \}$  ) enddef;
  
```

用上面的例程，作者画了带该种衬线的A和I:

下面是两个用上面例程的字母:



```
begin char ("A", 13u#, ht#, 0);
    Z1 = (.5w, 1.05h); % top point
    x4l = w - x5r = u; y4l = y5r = slab; % bottom point
    numeric theta[];
    theta4 = angle (Z1, Z4l); % left stroke angle
    theta5 = angle (Z1, Z5r); % right
    serif (4, thin, theta4, .6jut, jut); % left serifs
    serif (5, thick, theta5, jut, .6jut); % right
    Z0 = Z4r + whatever * dir theta4
        = Z5l + whatever * dir theta5; % inside top point
    fill Z1 -- serif-edge4 -- Z0 % the left stroke
        & Z0 -- serif-edge5 -- Z1 & cycle; % right
    penpos2 (whatever, theta4);
    penpos3 (whatever, theta5);
    y2r = y3r = .5 [y4, y0]; % cross bar height
    y2l = y3l = y2r - thin; % thickness
    Z2 = whatever [Z1, Z4r];
    Z3 = whatever [Z1, Z5l];
    penstroke Z2e -- Z3e; % the cross bar
    penlabels (0, 1, 2, 3, 4, 5); endchar;
```

```
begin char ("I", 6u#, ht#, 0);
    x1 = x2 = .5w;
    y1 = h - y2; y2 = slab;
    serif (1, thick, -90, 1.1jut, 1.1jut); % upper serifs
    serif (2, thick, 90, 1.1jut, 1.1jut); % lower
    fill serif-edge1 -- reverse serif-edge1 -- cycle; % the stroke
```

pen labels (1,2); endchar;

插图使用 w thin = .5pt, thick = 1.1pt;
 $u = .6pt$, $ht = 7pt$;
 $slab = .25pt$, $jut = .9pt$;
 $bracket = pt$, $serif-darkness = \frac{1}{3}$.

第二种方法

第二种方法基于 Ch16 末尾的例子 (即 P152 列见前)

在此假设 broad-pen 是一个 'pensquare xscaled px yscaled py rotated phi'
 ($px > py$, phi 是转角) 填充一个较大区域时会用这画笔制作
 较粗的笔画; serif 例程被给出 Z_{fl} 和 Z_{fr} 间距 xx .
 (浓度)

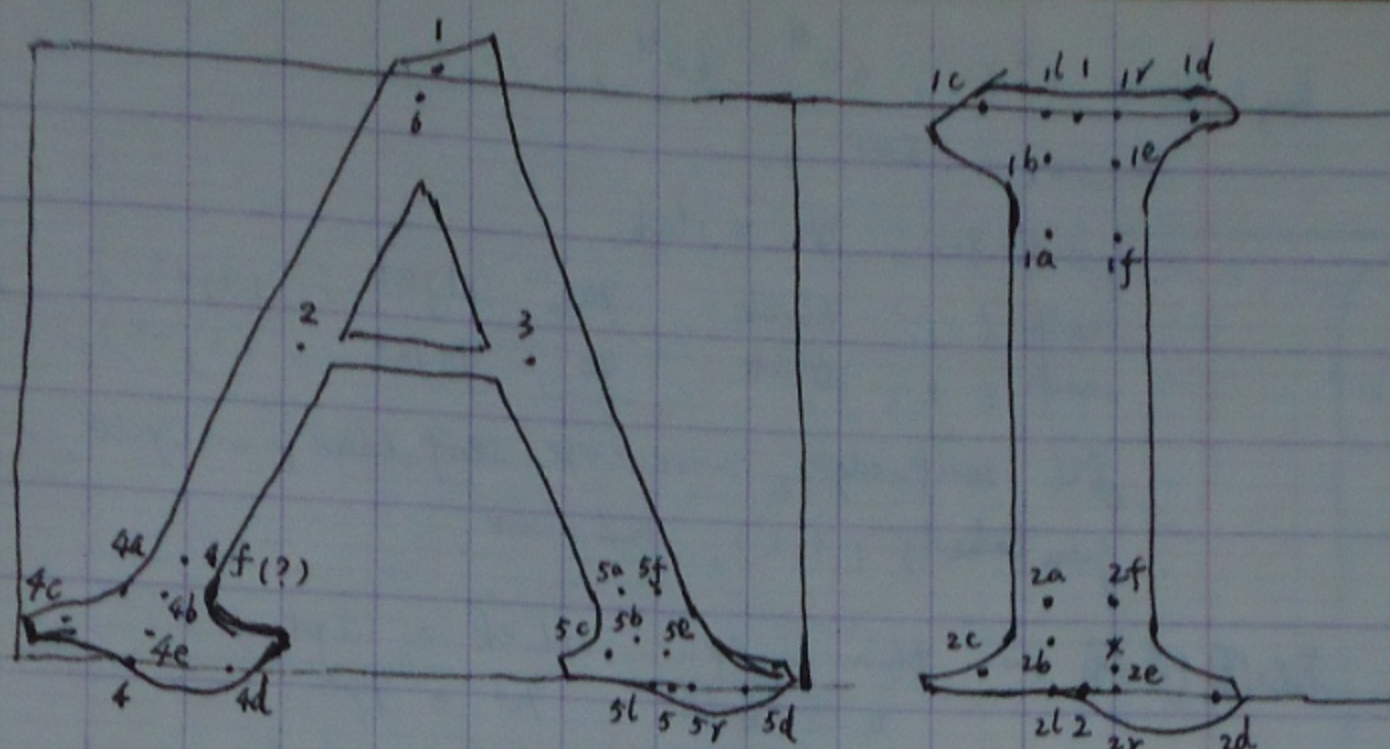
有一个 pair 变量 dishing 控制 Z_{fc} 和 Z_{fd} 间的 curvature.

上、下衬线是类似的, 但最好把它们写成分离的宏.

```
def bot_serif(suffix $) (expr xx, theta, left-jut, right-jut) =
  penpos $ (xx, 0);  $Z_{fa} - Z_{fl} = Z_{ff} - Z_{fr} = (bracket / abs \sin \theta) * \text{dir } \theta$ ;
   $y_{fc} = \text{top } y_{fl}$ ;  $y_{fd} = y_{fr}$ ;  $x_{fc} = x_{fl} - \text{left-jut}$ ;  $x_{fd} = x_{fr} + \text{right-jut}$ ;
   $Z_{fb} = Z_{fl} + \text{whatever} * \text{dir } \theta$  =  $Z_{fc} + \text{whatever} * \text{dir } \phi$ ;
   $Z_{fe} = Z_{fr} + \dots = Z_{fd} + \dots - \phi$ ;
  labels ($a, $b, $c, $d, $e, $f) enddef;

def bot_serif-edge suffix $ =
  (  $Z_{fa} \dots$  controls  $Z_{fb} \dots Z_{fc}$ 
    -- (flex ( $Z_{fc}$ , .5 [ $Z_{fc}$ ,  $Z_{fd}$ ] + dishing,  $Z_{fd}$ )) shifted
      ( $0, -\text{epiln}$ )
    --  $Z_{fd} \dots$  controls  $Z_{fe} \dots Z_{ff}$ ) enddef;
```

并接着开始讲第二种方法的例程 (如上图下半部分), 然后演示了用该例程画的又一种衬线的 A 和 I:



```
beg'char ("A", 13u#, ht#, 0); pickup broad-pen;
 $Z_1 = (.5w, \text{top } h)$ ; lft  $x_{4l} = w - rt \ x_{5r} = 1.2u$ ;  $y_{4l} = y_{5r} = 0$ ;
numeric theta[];  $\theta_4 = \text{angle}(Z_1 - Z_{4l})$ ;  $\theta_5 = \text{angle}(Z_1 - Z_{5r})$ ;
numeric xxx;  $px * \sin(\theta_5 - \phi) + xxx * \sin \theta_5 =$ 
 $px * \cos \phi + xxx$ ;
bot_serif(4, 0,  $\theta_4$ , .8jut, .8jut);
bot_serif(5, xxx,  $\theta_5$ , .6jut, .8jut);
```

```

Z0 = Z4r + whatever * dir theta4 = Z5l + whatever * dir theta5;
filldraw Z1 -- bot_serif-edge4 -- Z0 & Z0 -- bot_serif-edge5
-- Z1 & cycle;
top y2 = top y3 = .45 bot y0; Z2 = whatever[Z1, Z4r];
Z3 = whatever[Z1, Z5l];
draw Z2 -- Z3; penlabels (0, 1, 2, 3, 4, 5); endchar;
beginchar("I", bu#, ht#, 0); pickup broad_pen;
x1 = x2 = .5w; y1 = h; y2 = 0;
top_serif (1, xx, -90, 1.1 jut, 1.1 jut);
bot_serif (2, xx, 90, 1.1 jut, 1.1 jut);
filldraw bot_serif-edge2 -- reverse top_serif-edge1 -- cycle;
penlabels(1, 2); endchar;

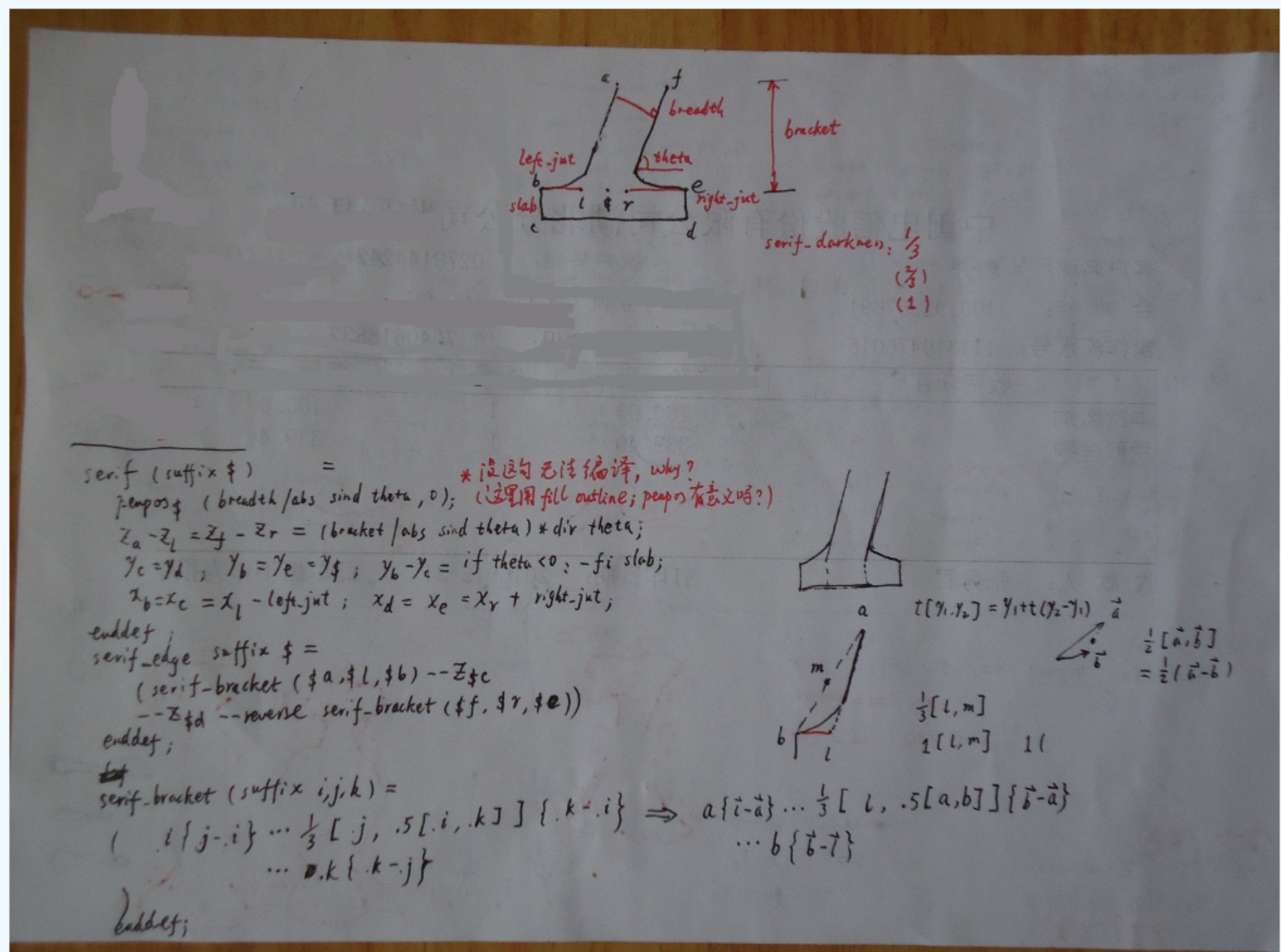
```

插图中, $px = .8pt$ $py = .2pt$ $phi = 20$ $xx = .3pt$
 $u = .6pt$ $ht = 7pt$ $jut = .9pt$ $bracket = pt$
 $dishing = (.25pt, 0)$ rotated 20.

其中说到第二种方法基于第16章末尾的例子，指的就是p152处。

第一种方法是勾勒出轮廓线然后填充（使用fill命令来画），比较易于我理解，所以我用了第一种方法来画衬线。沿用原书中的宏和代码，作了一点改动即可用了。

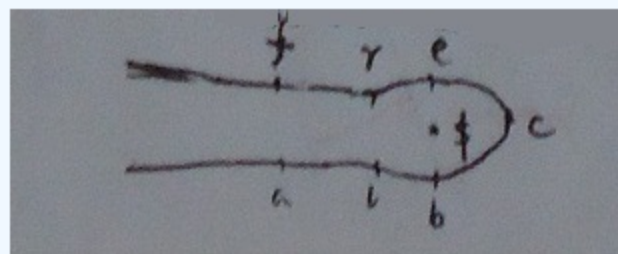
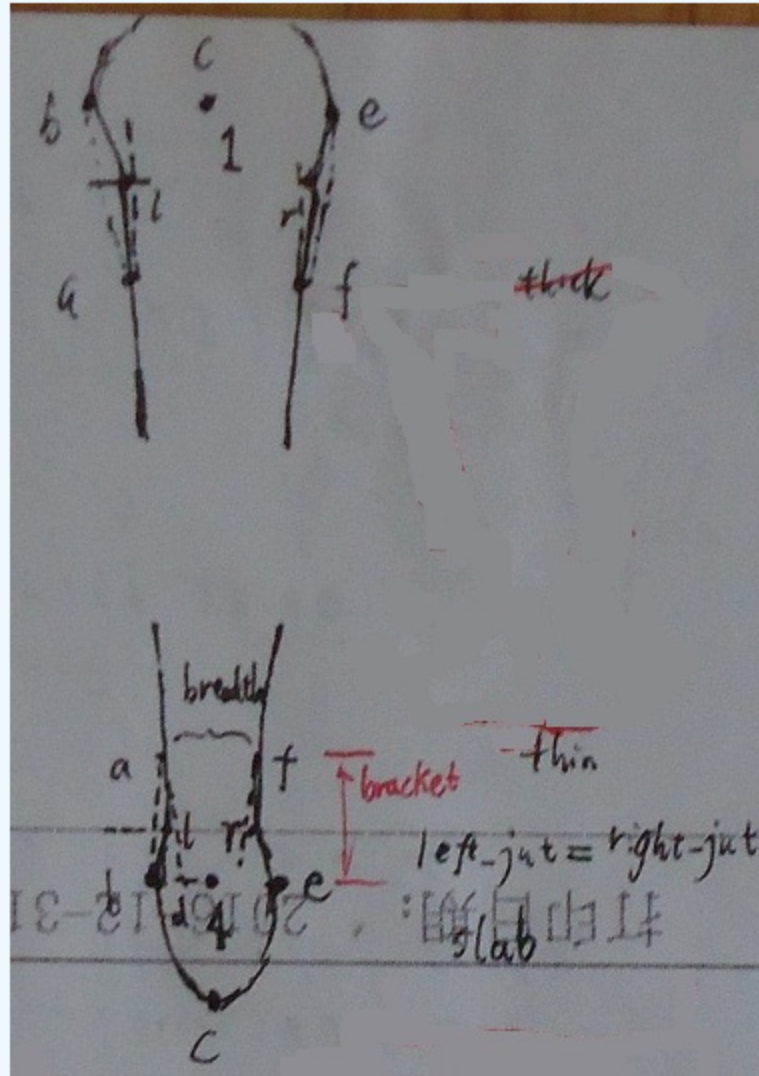
先对作者原书中的例程作了一些理解：



然后我大致想画成这样：

Let's try having a strange

即起止处的笔画稍微粗一些。上面是竖，横也一样。勾勒用点的设置草图：



比原书中点的设置就少一个d点。
实现竖的最好例子莫过于字母l了：

tmp5\lowerL.mf:

```
1 u#:=.6pt#;
2 thin#:=.5pt#;
3 thick#:=1.1pt#;
4 ht#:=7pt#;
5 %slab#:=.25pt#;
6 slab#:=.8pt#;
7 %jut#:=.9pt#;
8 jut#:=.3pt#;
9 bracket#:=pt#;
10 define_pixels(u, ht, slab, jut, bracket);
11 define_blacker_pixels(thin, thick);
12
13 def serif_my(suffix $(expr breadth, breadth_out, theta, left_jut, right_jut) =
14 %penpos$(breadth / abs sind theta, 0);
15 x$a = x$ - 0.5breadth;
16 x$f = x$ + 0.5breadth;
17 y$a = y$f = y$+bracket*(sind theta);
18
19 x$b = x$ - left_jut ;
20 x$e = x$ + right_jut;
21 y$b = y$e = y$;
22
23 x$c = x$;
24 y$c = y$ - slab * sind theta;
25
26 y$l = 0.4[y$b, y$a];
27 y$r = 0.4[y$e, y$f];
28
29 z$m = 0.4[z$b, z$a];
30 y$n - y$b = 0.4(y$a - y$b);
31 x$n-x$a = (0.5*(breadth_out-breadth))*(0.6*(y$a-y$b)) / (y1-y2);
32
33 z$l = 0.5[z$m, z$n];
34 x$r = x$ + x$ - x$l;
35 labels($a,$b,$c,$e,$f,$l,$r, $m, $n)
36 enddef;
37
38
39 def serif_edge_my suffix $ =
40 (z$a{z$l-z$a}..z$l{z$b-z$l}...z$b..z$c..z$e{z$r-z$e}...z$r{z$f-z$r}..z$f)
41 enddef;
42
43
44
45 beginchar("l",13u#,16u#,0);"Letter l";
46
47 x1=x2=.5w;
48 y1=h;y2=0;
49
50 serif_my(1, thick,thin, -90, 2.1jut, 2.1jut);
51 serif_my(2, thin, thick, 90, 0.9jut, 0.9jut);
52
53 fill serif_edge_my2 -- reverse serif_edge_my1 -- cycle;
54
55 penlabels(1,2);
56 endchar;
57 end
```

多增了一个参数breadth_out是因为笔画两端粗细不一样，用来表示对端的宽度（是thick还是thin）。

对于横，我发现字母调用宏例程的代码基本上把角度改一下就可以了，宏中则把x坐标改为相应的y坐标。字母l被设计成下面的竖加上面（点）实现为一

短横， 比较字母i正好多一短横， 拿来作例子， 代码如下：

tmp5\lowerI.mf:

```
1 u#:=.6pt#;
2 thin#:=.5pt#;
3 thick#:=1.1pt#;
4 ht#:=7pt#;
5 %slab#:=.25pt#;
6 slab#:=.8pt#;
7 %jut#:=.9pt#;
8 jut#:=.3pt#;
9 bracket#:=pt#;
10 define_pixels(u, ht, slab, jut, bracket);
11 define_blacker_pixels(thin, thick);
12
13 def serif_my(suffix $(expr breadth, breadth_out, theta, left_jut, right_jut) =
14 %penpos$(breadth / abs sind theta, 0);
15 x$a = x$ - 0.5breadth;
16 x$f = x$ + 0.5breadth;
17 y$a = y$f = y$+bracket*(sind theta);
18
19 x$b = x$ - left_jut ;
20 x$e = x$ + right_jut;
21 y$b = y$e = y$;
22
23 x$c = x$;
24 y$c = y$ - slab * sind theta;
25
26 y$l = 0.4[y$b, y$a];
27 y$r = 0.4[y$e, y$f];
28
29 z$m = 0.4[z$b, z$a];
30 y$n - y$b = 0.4(y$a - y$b);
31 x$n-x$a = (0.5*(breadth_out-breadth))*(0.6*(y$a-y$b)) / (y1-y2);
32
33 z$l = 0.5[z$m, z$n];
34 x$r = x$ + x$ - x$l;
35 labels($a,$b,$c,$e,$f,$l,$r, $m, $n)
36 enddef;
37
38 def serif_my_horizontal(suffix $(expr breadth, breadth_out, theta, left_jut, right_jut) =
39 %penpos$(breadth / abs sind theta, 0);
40 y$a = y$ - 0.5breadth;
41 y$f = y$ + 0.5breadth;
42 x$a = x$f = x$+bracket*(sind theta);
43
44 y$b = y$ - left_jut ;
45 y$e = y$ + right_jut;
46 x$b = x$e = x$;
47
48 y$c = y$;
49 x$c = x$ - slab * sind theta;
50
51 x$l = 0.4[x$b, x$a];
52 x$r = 0.4[x$e, x$f];
53
54 z$m = 0.4[z$b, z$a];
55 x$n - x$b = 0.4(x$a - x$b);
56 y$n-y$a = (0.5*(breadth_out-breadth))*(0.6*(x$a-x$b)) / (x4-x5);
57
58 z$l = 0.5[z$m, z$n];
59 y$r = y$ + y$ - y$l;
60 labels($a,$b,$c,$e,$f,$l,$r, $m, $n)
61 enddef;
62
63 def serif_edge_my suffix $ =
64 (z$a{z$l-z$a}..z$l{z$b-z$l}...z$b..z$c..z$e{z$r-z$e}...z$r{z$f-z$r}..z$f)
65 enddef;
66
67
68
69 beginchar("i",13u#,16u#,0);"Letter i";
70
71 x1=x2=.5w;
72 y1=h/2;y2=0;
73
74
75 serif_my(1, thick,thin, -90, 2.1jut, 2.1jut);
76 serif_my(2, thin, thick, 90, 0.9jut, 0.9jut);
77 fill serif_edge_my2 -- reverse serif_edge_my1 -- cycle;
78
79
80 z3 = z1 + (0, 0.2h);
81 x4 = x3 - 0.2w; x5 = x3+0.2w;
82 y4 = y5 = y3 ;
83 serif_my_horizontal(4, 0.7thick,thin, 90, 2.1jut, 2.1jut);
84 serif_my_horizontal(5, thin, 0.7thick, -90, 0.9jut, 0.9jut);
85 fill serif_edge_my5 -- reverse serif_edge_my4 -- cycle;
86
87
88 penlabels(1,2,3,4,5);
89 endchar;
90 end
```

其中可见serif_my宏完全没有改变， serif_my_horizontal宏相应于serif_my宏的改变则如下图：

<pre>def serif_my(suffix \$(expr breadth, breadth_out, theta, left_jut, right_jut) = %penpos\$(breadth / abs sind theta, 0); x\$a = x\$ - 0.5breadth; x\$f = x\$ + 0.5breadth; y\$a = y\$f = y\$+bracket*(sind theta); x\$b = x\$ - left_jut ; x\$e = x\$ + right_jut; y\$b = y\$e = y\$; x\$c = x\$; y\$c = y\$ - slab * sind theta; y\$l = 0.4[y\$b, y\$a]; y\$r = 0.4[y\$e, y\$f];</pre>	<pre>def serif_my_horizontal(suffix \$(expr breadth, breadth_out, theta, left_jut, right_jut) = %penpos\$(breadth / abs sind theta, 0); y\$a = y\$ - 0.5breadth; y\$f = y\$ + 0.5breadth; x\$a = x\$f = x\$+bracket*(sind theta); y\$b = y\$ - left_jut ; y\$e = y\$ + right_jut; x\$b = x\$e = x\$; y\$c = y\$; x\$c = x\$ - slab * sind theta; x\$l = 0.4[x\$b, x\$a]; x\$r = 0.4[x\$e, x\$f];</pre>
--	---

```

z$m = 0.4[z$b, z$a];
y$n - y$b = 0.4(y$a - y$b);
x$n-x$a = (0.5*(breadth_out-breadth))*(0.6*(y$a-y$b)) / (y1-y2);

z$l = 0.5[z$m, z$n];
x$r = x$ + x$ - x$l;
labels($a,$b,$c,$e,$f,$l,$r, $m, $n)
enddef;

def serif edge my_suffix $ =

```

```

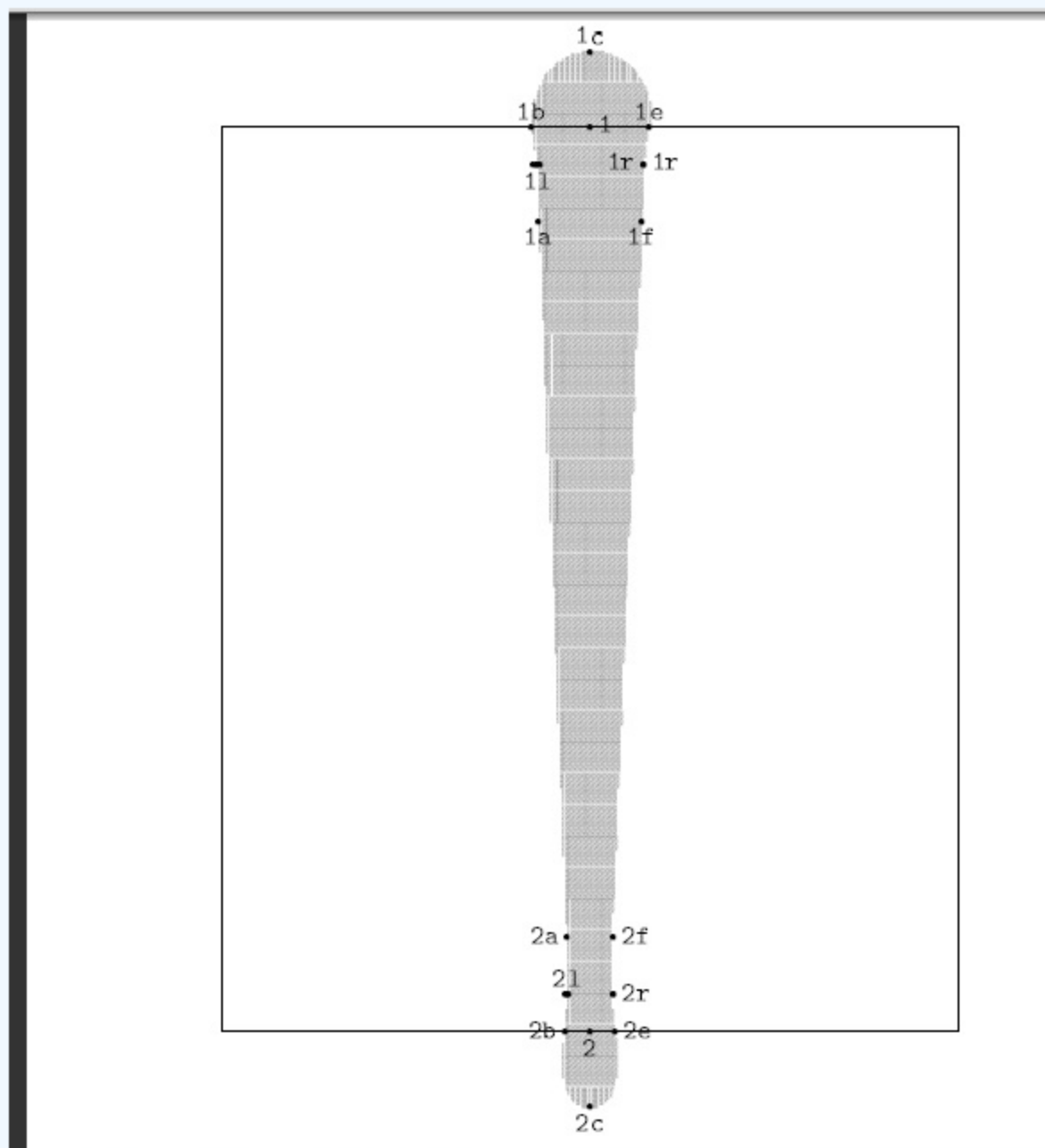
z$m = 0.4[z$b, z$a];
x$n - x$b = 0.4(x$a - x$b);
y$n-y$a = (0.5*(breadth_out-breadth))*(0.6*(x$a-x$b)) / (x4-x5);

z$l = 0.5[z$m, z$n];
y$r = y$ + y$ - y$l;
labels($a,$b,$c,$e,$f,$l,$r, $m, $n)
enddef;

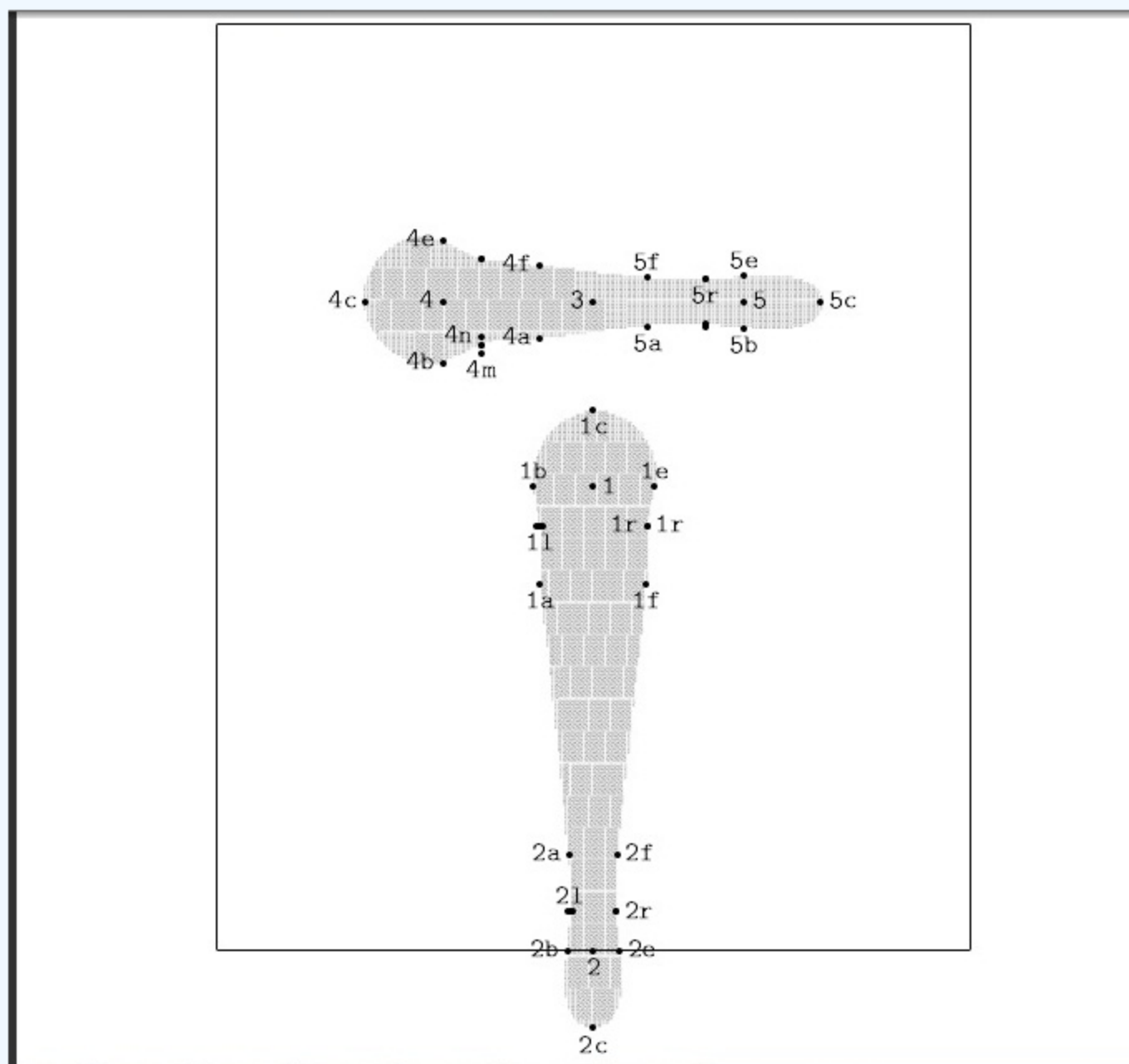
def serif edge my_suffix $ =

```

实际实现i和i的proof mode的图分别为



和



操作过程用到的命令一般如下：

```

mf '\mode=ljfour;mode_setup;input xx.mf'
gftopk xx.600gf xx.600pk

mf '\mode=ljfour; mag=magstep(7); mode_setup;input xx.mf'
mv xx.tfm xx2150.tfm
gftopk xx2150.2150gf xx2150.2150pk
ln -s xx2150.2150pk xx2150.600pk

gftodvi xx.2602qf

```

```
(dvipdf xx.dvi; xpdf xx.pdf)
```

```
tmp9\genH.sh:
```

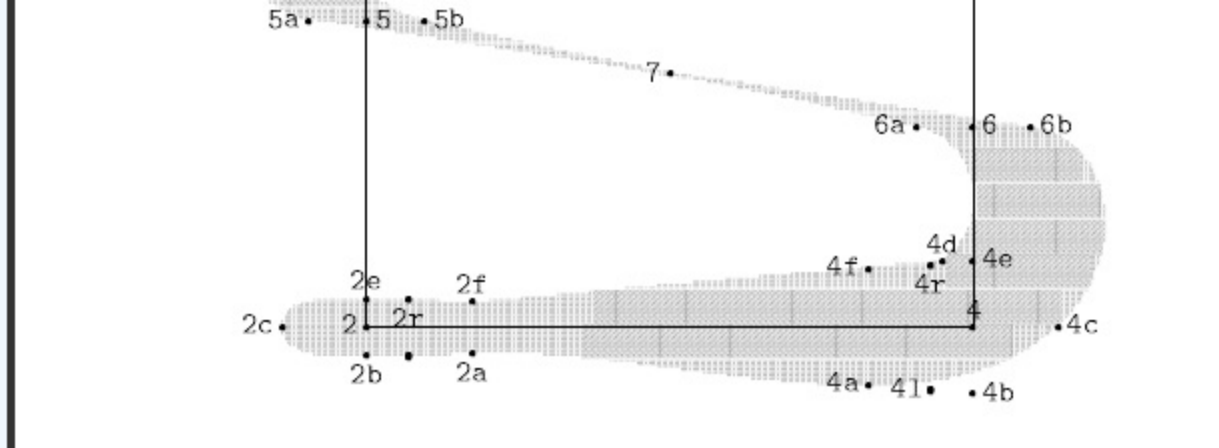
```
15 latex liH 2.tex
```

tmp9\liH 2.tex:

```
12 \end{document}
```

("n", 16u, 8u, 0)

 $x_1 = x_2 = 0; \quad x_3 = x_4 = w;$
 $y_2 = y_4 = 0; \quad y_1 = y_3 = \frac{1}{2}h;$
 serif_my (1, thick, thin, ...)
 12, thin, thick, ...
 13,
 14,
 fill serif_edge.my2 --reverse...
 fill 4 --
 $x_5 = \frac{w}{2}; \quad y_5 = y_1 + \frac{1}{8}h;$
 $z_6 = z_1; \quad z_7 = z_3;$
 penpos6 (2u, 150);
 penpos5 (2u, 90);
 penpos7 (2u, 30);
 pickup pencircle;
 penstroke z6e .. z5e .. z7e;



实现代码如下：

tmp8\liS.mf:

```

1  u#:=.6pt#;
2  thin#:=.5pt#;
3  thick#:=1.1pt#;
4  %thin#:=1pt#;
5  %thick#:=2.2pt#;
6  ht#:=7pt#;
7  %slab#:=.25pt#;
8  slab#:=.8pt#;
9  %jut#:=.9pt#;
10 jut#:=.3pt#;
11 %jut#:=.6pt#;
12 bracket#:=pt#;
13 define_pixels(u, ht, slab, jut, bracket);
14 define_blacker_pixels(thin, thick);
15
16 def serif_my(suffix $(expr breadth, breadth_out, theta, left_jut, right_jut) =
17   %penpos$(breadth / abs sind theta, 0);
18   x$a = x$ - 0.5breadth;
19   x$f = x$ + 0.5breadth;
20   y$a = y$f = y$+bracket*(sind theta);
21
22   x$b = x$ - left_jut ;
23   x$e = x$ + right_jut;
24   y$b = y$e = y$;
25
26   x$c = x$;
27   y$c = y$ - slab * sind theta;
28
29   y$l = 0.4[y$b, y$a];
30   y$r = 0.4[y$e, y$f];
31
32   z$m = 0.4[z$b, z$a];
33   y$n - y$b = 0.4(y$a - y$b);
34   x$n-x$a = (0.5*(breadth_out-breadth))*(0.6*(y$a-y$b)) / (y1-y2);
35
36   z$l = 0.5[z$m, z$n];
37   x$r = x$ + x$ - x$l;
38   labels($a,$b,$c,$e,$f,$l,$r, $m, $n)
39   enddef;
40
41 def serif_my_horizontal(suffix $(expr breadth, breadth_out, theta, left_jut, right_jut) =
42   %penpos$(breadth / abs sind theta, 0);
43   y$a = y$ - 0.5breadth;
44   y$f = y$ + 0.5breadth;
45   x$a = x$f = x$+bracket*(sind theta);
46
47   y$b = y$ - left_jut ;
48   y$e = y$ + right_jut;
49   x$b = x$e = x$;
50
51   y$c = y$;
52   x$c = x$ - slab * sind theta;
53
54   x$l = 0.4[x$b, x$a];
55   x$r = 0.4[x$e, x$f];
56
57   z$m = 0.4[z$b, z$a];
58   x$n - x$b = 0.4(x$a - x$b);
59   y$n-y$a = (0.5*(breadth_out-breadth))*(0.6*(x$a-x$b)) / (x2-x1);
60
61   z$l = 0.5[z$m, z$n];
62   y$r = y$ + y$ - y$l;
63   labels($a,$b,$c,$e,$f,$l,$r, $m, $n)
64   enddef;
65
66
67 def serif_edge_my suffix $ =
68   (z$a{z$l-z$a}...z$l{z$b-z$l}...z$b..z$c..z$e{z$r-z$e}...z$r{z$f-z$r}..z$f)
69   enddef;
70
71
72 beginchar("S",9.6u#,16u#,0);"Letter S";
73
74 x1=w; y1=0.5h;
75 x2=0; y2= 0;
76 x3=0; y3=0.5h;
77 x4=w; y4= 0;
78 x7=0.5w; y7=0.5[y1,y2];
79 x5=0; y5=y7+thin;
80 x6=w; y6=y7-thin;
81 y5a=y5b=y5;
82 y6a=y6b=y6;
83 x5a=x5-0.5thick;
84 x5b=x5+0.5thick;
85 x6a=x6-0.5thick;
86 x6b=x6+0.5thick;
87
88 serif_my_horizontal(3, thick,thin, 90, 2.1jut, 2.1jut);
89 serif_my_horizontal(1, thin, thick, -90, 0.9jut, 0.9jut);
90 serif_my_horizontal(4, thick,thin, -90, 2.1jut, 2.1jut);
91 serif_my_horizontal(2, thin, thick, 90, 0.9jut, 0.9jut);
92
93 x3d=x3c + thick;

```

```

94 x4d=x4c - thick;
95 y3d=y3b;
96 y4d=y4e;
97
98 fill serif_edge_my1 ..z3f..z3c..z5a..controls z7..z6a..z4d--reverse serif_edge_my2..z4a..z4c..z6b..
99 penlabels(1,2,3,3d, 4,4d,5,5a,5b,6,6a,6b,7);
100 endchar;
101 end

```

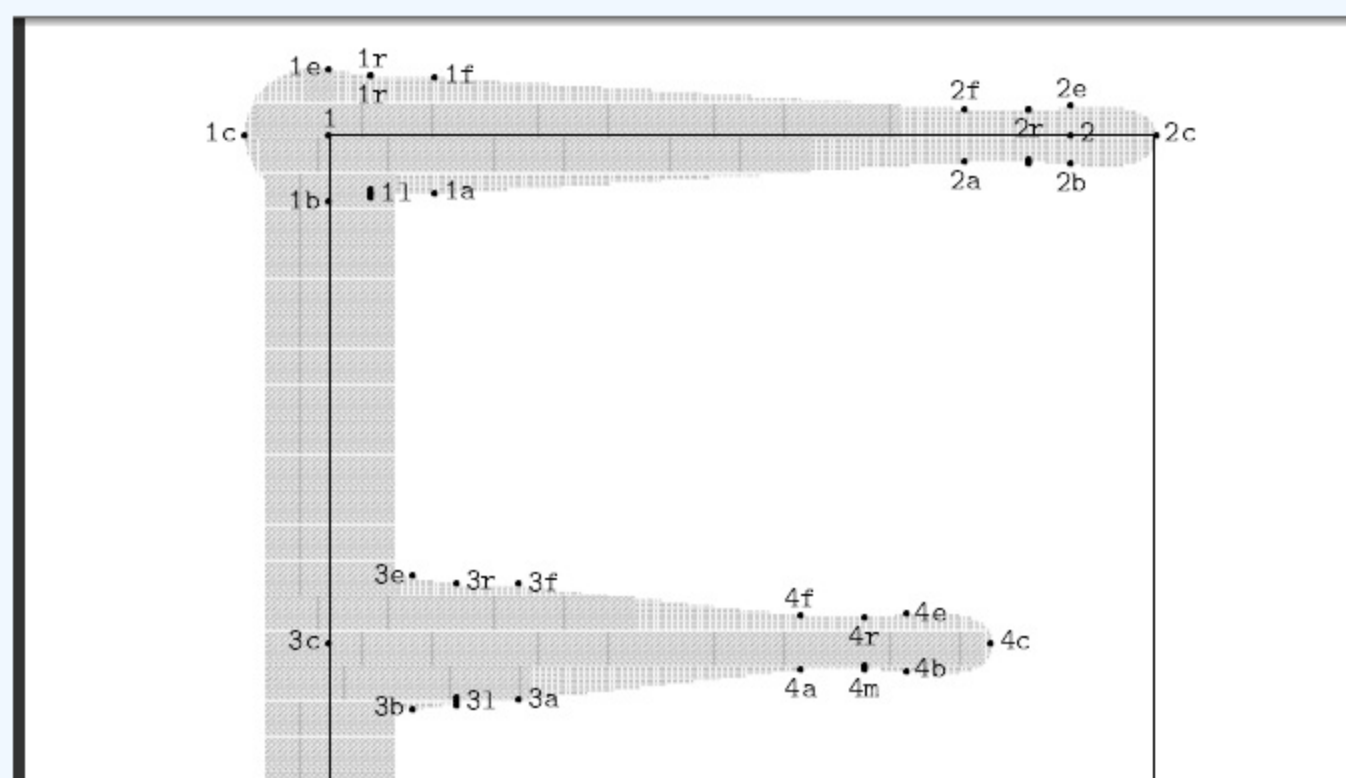
代码可见其中实际并未用到serif_my宏，这里只是从别处拷贝过来，保留未删而已。serif_my_horizontal宏基本保留未动，只是有一处x坐标相减的两点要改，未能完全代码重用，相比上面字母i使用的serif_my_horizontal宏改动如下：

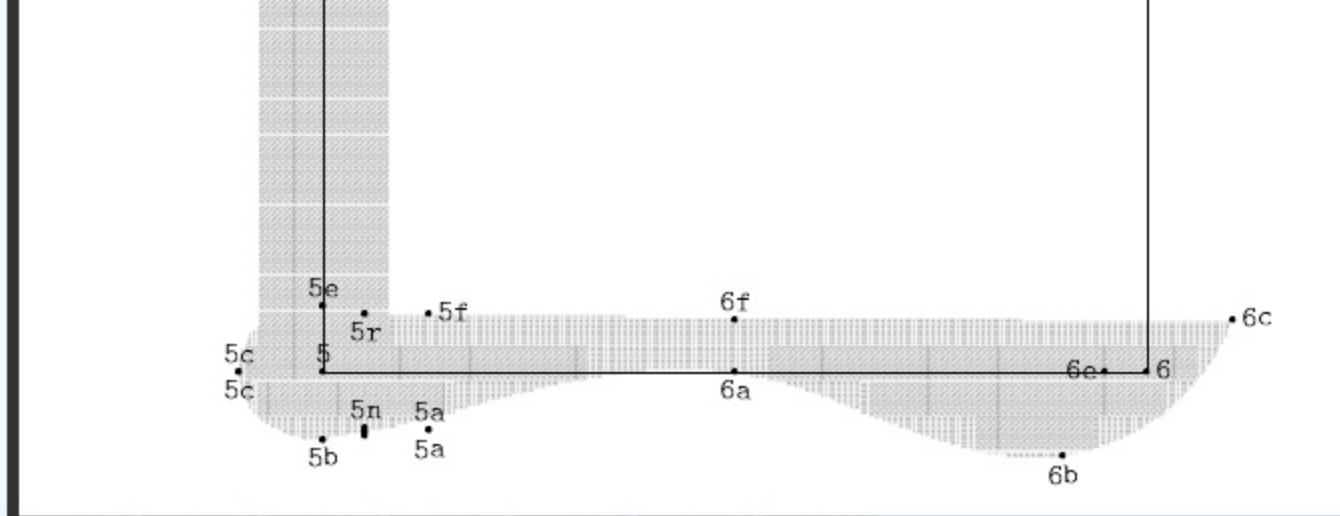
File	Size	Code
\\client_ftp\page_making\LiShu_ENGLISH_english\tmp5\lowerI.mf	2,050 字节	<pre> def serif_my_horizontal(suffix \$(expr breadth, breadth_out, theta, 1 %penpos\$(breadth / abs sind theta, 0); y\$a = y\$ - 0.5breadth; y\$f = y\$ + 0.5breadth; x\$a = x\$f = x\$+bracket*(sind theta); y\$b = y\$ - left_jut ; y\$e = y\$ + right_jut; x\$b = x\$e = x\$; y\$c = y\$; x\$c = x\$ - slab * sind theta; x\$l = 0.4[x\$b, x\$a]; x\$r = 0.4[x\$e, x\$f]; z\$m = 0.4[z\$b, z\$a]; x\$n - x\$b = 0.4(x\$a - x\$b); y\$n-y\$a = (0.5*(breadth_out-breadth))*(0.6*(x\$a-x\$b)) / (x4-x5); z\$l = 0.5[z\$m, z\$n]; y\$r = y\$ + y\$ - y\$l; labels(\$a,\$b,\$c,\$e,\$f,\$l,\$r, \$m, \$n) enddef; </pre>
D:\...\client_ftp\page_making\LiShu_ENGLISH_english\tmp8\liS.mf	2,337 字节	<pre> def serif_my_horizontal(suffix \$(expr breadth, breadth_out, theta, 1 %penpos\$(breadth / abs sind theta, 0); y\$a = y\$ - 0.5breadth; y\$f = y\$ + 0.5breadth; x\$a = x\$f = x\$+bracket*(sind theta); y\$b = y\$ - left_jut ; y\$e = y\$ + right_jut; x\$b = x\$e = x\$; y\$c = y\$; x\$c = x\$ - slab * sind theta; x\$l = 0.4[x\$b, x\$a]; x\$r = 0.4[x\$e, x\$f]; z\$m = 0.4[z\$b, z\$a]; x\$n - x\$b = 0.4(x\$a - x\$b); y\$n-y\$a = (0.5*(breadth_out-breadth))*(0.6*(x\$a-x\$b)) / (x2-x1); z\$l = 0.5[z\$m, z\$n]; y\$r = y\$ + y\$ - y\$l; labels(\$a,\$b,\$c,\$e,\$f,\$l,\$r, \$m, \$n) enddef; </pre>

字母E中也用到serif_my_horizontal宏，相比上面字母中的该名称的宏也是进行类似的微小修改，如下所示将相减的x1、x2 调换一下：

File	Size	Code
\\client_ftp\page_making\LiShu_ENGLISH_english\tmp42\liE2.mf	2,751 字节	<pre> def serif_my_horizontal(suffix \$(expr breadth, breadth_out, theta, 1 %penpos\$(breadth / abs sind theta, 0); y\$a = y\$ - 0.5breadth; y\$f = y\$ + 0.5breadth; x\$a = x\$f = x\$+bracket*(sind theta); y\$b = y\$ - left_jut ; y\$e = y\$ + right_jut; x\$b = x\$e = x\$; y\$c = y\$; x\$c = x\$ - slab * sind theta; x\$l = 0.4[x\$b, x\$a]; x\$r = 0.4[x\$e, x\$f]; z\$m = 0.4[z\$b, z\$a]; x\$n - x\$b = 0.4(x\$a - x\$b); y\$n-y\$a = (0.5*(breadth_out-breadth))*(0.6*(x\$a-x\$b)) / (x1-x2); z\$l = 0.5[z\$m, z\$n]; y\$r = y\$ + y\$ - y\$l; labels(\$a,\$b,\$c,\$e,\$f,\$l,\$r, \$m, \$n) enddef; </pre>
D:\...\client_ftp\page_making\LiShu_ENGLISH_english\tmp8\liS.mf	2,337 字节	<pre> def serif_my_horizontal(suffix \$(expr breadth, breadth_out, theta, 1 %penpos\$(breadth / abs sind theta, 0); y\$a = y\$ - 0.5breadth; y\$f = y\$ + 0.5breadth; x\$a = x\$f = x\$+bracket*(sind theta); y\$b = y\$ - left_jut ; y\$e = y\$ + right_jut; x\$b = x\$e = x\$; y\$c = y\$; x\$c = x\$ - slab * sind theta; x\$l = 0.4[x\$b, x\$a]; x\$r = 0.4[x\$e, x\$f]; z\$m = 0.4[z\$b, z\$a]; x\$n - x\$b = 0.4(x\$a - x\$b); y\$n-y\$a = (0.5*(breadth_out-breadth))*(0.6*(x\$a-x\$b)) / (x2-x1); z\$l = 0.5[z\$m, z\$n]; y\$r = y\$ + y\$ - y\$l; labels(\$a,\$b,\$c,\$e,\$f,\$l,\$r, \$m, \$n) enddef; </pre>

字母E实际实现的proof mode的图为



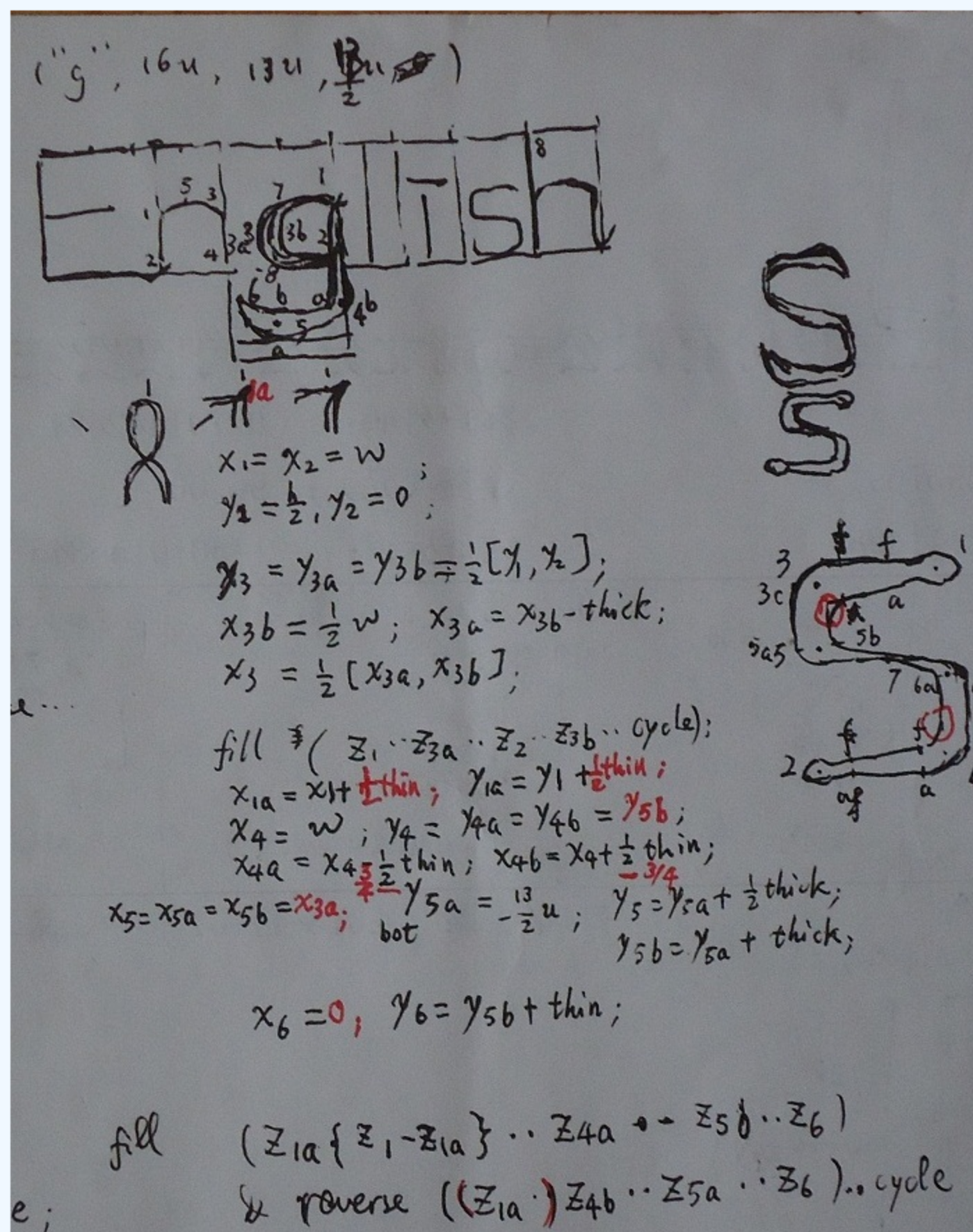


实现代码如下：

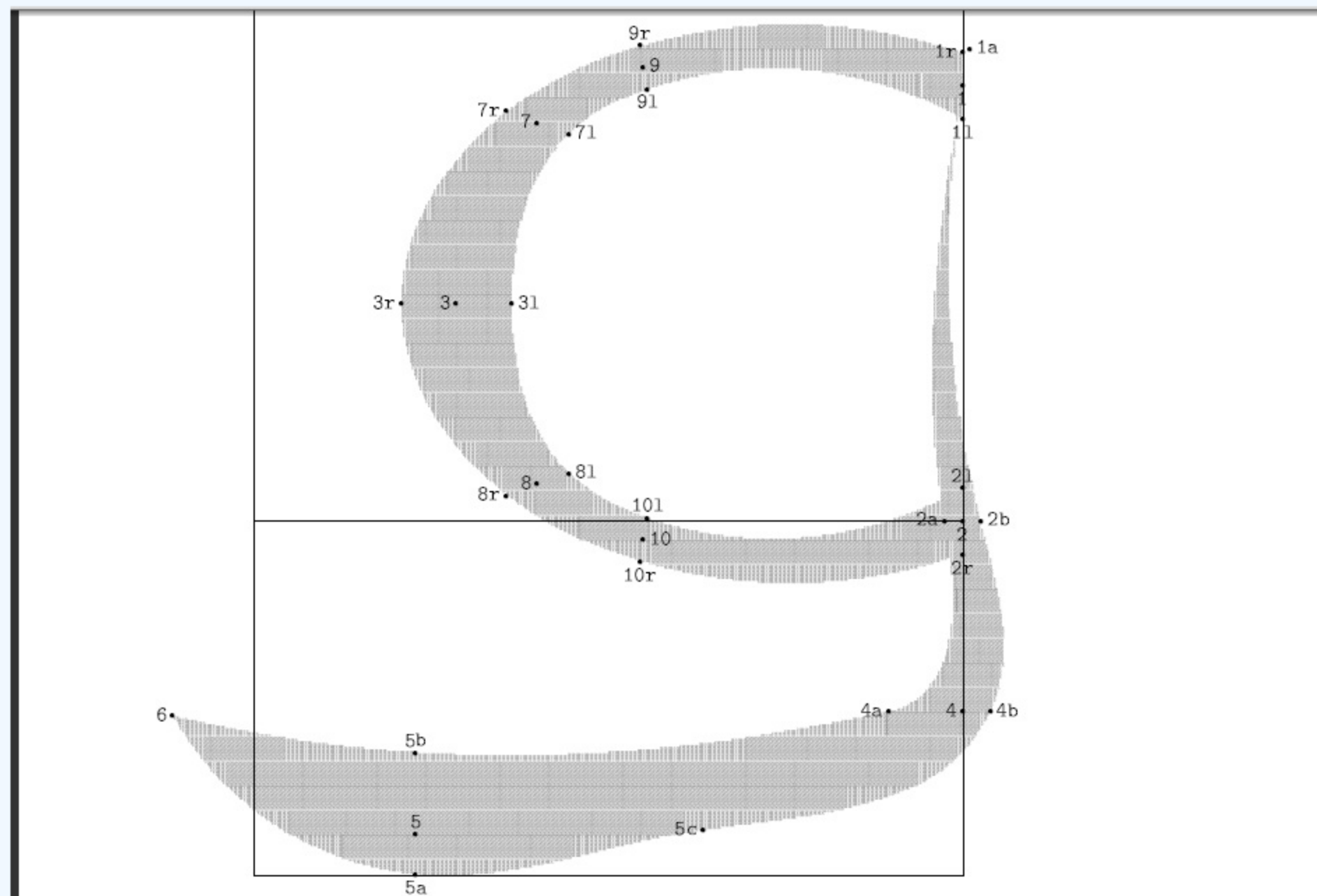
```
tmp42\liE2.mf:
1  u#:=.6pt#;
2  thin#:=.5pt#;
3  thick#:=1.1pt#;
4  %thin#:=1pt#;
5  %thick#:=2.2pt#;
6  ht#:=7pt#;
7  %slab#:=.25pt#;
8  slab#:=.8pt#;
9  %jut#:=.9pt#;
10 jut#:=.3pt#;
11 %jut#:=.6pt#;
12 bracket#:=pt#;
13 define_pixels(u, ht, slab, jut, bracket);
14 define_blacker_pixels(thin, thick);
15
16 def serif_my_horizontal(suffix $(expr breadth, breadth_out, theta, left_jut, right_jut) =
17   %penpos$(breadth / abs sind theta, 0);
18   y$a = y$ - 0.5breadth;
19   y$f = y$ + 0.5breadth;
20   x$a = x$f = x$+bracket*(sind theta);
21
22   y$b = y$ - left_jut ;
23   y$e = y$ + right_jut;
24   x$b = x$e = x$;
25
26   y$c = y$;
27   x$c = x$ - slab * sind theta;
28
29   x$l = 0.4[x$b, x$a];
30   x$r = 0.4[x$e, x$f];
31
32   z$m = 0.4[z$b, z$a];
33   x$n - x$b = 0.4(x$a - x$b);
34   y$n-y$a = (0.5*(breadth_out-breadth))*(0.6*(x$a-x$b)) / (x1-x2);
35
36   z$l = 0.5[z$m, z$n];
37   y$r = y$ + y$ - y$l;
38   labels($a,$b,$c,$e,$f,$l,$r, $m, $n)
39   enddef;
40
41
42 def serif_edge_my suffix $ =
43   (z$a{z$l-z$a}..z$l{z$b-z$l}...z$b..z$c..z$e{z$r-z$e}...z$r{z$f-z$r}..z$f)
44   enddef;
45
46 beginchar("E",13u#,16u#,0);"Letter E";
47
48 x1=0; x2=.9w;
49 y1=h;y2=h;
50
51 x3=0+slab; x4=.7w;
52 y3=h/2;y4=h/2;
53
54 x5=0;
55 y5=0;
56
57 z7 = z1;
58 z8 = z5;
59
60 serif_my_horizontal(1, thick,thin, 90, 2.1jut, 2.1jut);
61 serif_my_horizontal(2, thin, thick, -90, 0.9jut, 0.9jut);
62 fill serif_edge_my2 -- reverse serif_edge_my1 -- cycle;
63
64 serif_my_horizontal(3, thick,thin, 90, 2.1jut, 2.1jut);
65 serif_my_horizontal(4, thin, thick, -90, 0.9jut, 0.9jut);
66 fill serif_edge_my4 -- reverse serif_edge_my3 -- cycle;
67
68 x6 = w;
69 y6 = 0;
70 x6a = x6f = (1/2)[x5,x6];
71 y6a = y5; y6f = y6a + thin;
72 x6b = x6-slab; y6b = y5b - thin/3; %y6b = y6 - thick/2;
73 x6c = x6 + slab; y6c = y6 + thin;
74 x6e = x6 - 0.5*slab; y6e = y6;
75 serif_my_horizontal(5, thick,thin, 90, 2.1jut, 2.1jut);
76 fill reverse ( z5l{z5b-z5l}...z5b..z5c..z5e{z5r-z5e}...z5r{z5f-z5r})..z6a..z6b..z6c -- cycle;
77
78
79 penpos7(2u, 0);
80 penpos8(2u, 0);
81 pickup pencircle;
82 penstroke z7e--z8e;
83
84 penlabels(1,2,5,5a,5b,5c,5d,5e,6, 6a, 6b, 6c, 6e, 6f);
85 endchar;
86 end
87
```

可见在点1、2、3、4、5处都同上面一样使用了 serif my horizontal 例程。只有点6处的勾勒点是单独各个设置的，这里就是E中最能显示隶书风格笔画的地方了。这里没有使用什么一定之规，只是不断调整设置试验然后检查，看效果大致可以就停止改动了。

此外最能显示隶书风格笔画的地方就是字母g了。起始设计草图如下图中：



字母g实际实现的proof mode的图为



字母g的实现代码中没有用到以上衬线的例程。用的是类似前面画beta例子的笔法（penstoke），以及仍然是轮廓线填充的方法。代码如下：

tmp7liG.mf:

```
1 u#:=.6pt#;
2 thin#:=.5pt#;
3 thick#:=1.1pt#;
4 %thin#:=1pt#;
5 %thick#:=2.2pt#;
```

2

```

6 ht#:=7pt#;
7 %slab#:=.25pt#;
8 slab#:=.8pt#;
9 %jut#:=.9pt#;
10 jut#:=.3pt#;
11 %jut#:=.6pt#;
12 bracket#:=pt#;
13 define_pixels(u, ht, slab, jut, bracket);
14 define_blacker_pixels(thin, thick);
15
16
17 beginchar("G",16u#,16u#,8u#);"Letter G";
18
19 x1=x2=w;
20 y1=0.5*h + thick; y2 = 0;
21
22 y3=0.5[y1,y2];
23 x3=0.4w-thick;
24
25 x7=0.4w-0.5thin;y7=y1-thin;
26 x8=x7; y8=y2+thin;
27
28 x9 = 0.25[x7,x1]; x10 = x9;
29 y9=y1+.5thin;
30 y10=y2-.5thin;
31
32 penpos1(1.5u, 90);
33 penpos9(1u, 100);
34 penpos7(1.6u, 160);
35 penpos3(2.5u, 180);
36 penpos8(1.6u, 200);
37 penpos10(1u, 260);
38 penpos2(1.5u, 270);
39 pickup pencircle;
40 penstroke z1e..z9e..z7e..z3e..z8e..z10e..z2e;
41
42
43
44 x1a = x1 + .2thin; y1a = y1 + thin;
45 x4 = w; x4a = x4 - 2thin; x4b = x4 + .75thin;
46 bot y5a = -8u; y5 = y5a + .5thick; y5b = y5 + thick;
47 y4 = y4a = y4b = y5b + .5thick;
48 x5 = x5a = x5b = x3 - .5thick;
49 z5c = 0.5[z5a, z4b] + (0, -thin);
50 x6 = 0-thick; y6 = y5b + thin;
51
52 x2a = x2 - .5thin; x2b = x2 + .5thin; y2a = y2b = y2;
53
54 fill (z1a{z1-z1a}..z2a..z4a{z5-z4}..z5b..z6) & reverse (z1a..z2b..z4b..z5c..z5a..z6) ..cycle;
55
56 penlabels(1,2,3, 4,5, 6, 7, 8, 9, 10, 1a, 2a,2b, 4a,4b, 5a,5b,5c);
57 endchar;
58 end

```

轮廓线勾勒点（含控制点）4、5、6的设置仍然是不断修改试验而成。

至此“English”的几个字母的字体设计基本完成，效果图就如本站首页中所见。

剩下的就是些背景图的制作和叠加了。没有用到象Photoshop或GIMP等强大的软件，也没有用在线工具软件，用Windows画图、ACDSee、Office（PowerPoint和Word）就做成了。PowerPoint实现了背景图的透明化，ACDSee实现了文字的倾斜、晕影特效，从The *TEX*book和The METAFONTbook等书上取得原字形风格文字，中文书法背景图来自某书法比赛的图书中作者的优秀作品，这里要表示感谢。Word制作一些常见字体的文字。叠加和拼接等等诸多处理都用Windows画图完成。

至此首页的主体就告完成，要感谢METAFONT和 TEX 的强大功能，以上使用的 latex 和 mf 程序均为RedHat 9的安装版本。
源文件打包在此：[LiShu ENGLISH english.tgz](#)