The Pacioli fonts[*]

Peter Wilson
Catholic University of America
Now at peter.r.wilson@boeing.com

2001/01/02

Abstract

The pacioli package provides a set of uppercase fonts designed by Fra Luca de Pacioli in 1497.

Contents

1 Introduction

Around the beginning of the 16th century several authors published designs for uppercase characters. The more well known among these were Albrecht Dürer who wrote *On the Just Shaping of Letters* which was published in Nuremberg in 1525, and Geofroy Tory who wrote *Champ Fleury* which was published in Paris in 1529. Fra Luca Bartolomeo de Pacioli preceeded both of these by developing

[*] This file has version number v1.0a, last revised 2001/01/02.

his recipe for uppercase characters in 1497 and which was published in *De Divina Proportione* at Venice in 1509. Stanley Morison [Mor94] provides a more accessible source for Pacioli's characters.

This manual is typeset according to the conventions of the LaTeX docstrip utility which enables the automatic extraction of the LaTeX macro source files [GMS94].

Section 2 describes the usage of the package. Commented code for the fonts is in Sections 3 and 4 and source code for the package is in Section 5.

2 The pacioli package

Pacioli gave instructions for most of the uppercase letters in the alphabet, except for J, U, W and Z. All the designs were based on a square within which the letters were fitted. The width of the thickest limbs are 1/9 of the size of the square. Thinner limbs were normally half the width of the thick limbs and occasionaly, as in the cross-bar of the A, were 1/3 of the width.

Albrecht Dürer's characters were similar to Pacioli's, and I have used a modified version of Alan Hoenig's rendition of the Dürer font for the missing characters.[1]

I have also added modified versions of Knuth's Computer Modern punctuation characters so that, perhaps, the Pacioli font could be used as a titling font.

Pacioli's font as provided here consists of the uppercase letters, the punctuation marks !.,;:‘’?, the hyphen together with the en- and em-dash, and the analphabetic characters ()[]&. All the chacters are accessed by typing them as you usually do in LaTeX.

Pacioli's fonts are designed using ruler and compass, but ‘real’ font design is more subtle than this. Not only do the characters have to be well-formed and consistent, but they must also appear to be consistent. Because of optical illusions, certain characters, like a C or an O, may look smaller than they actually are when used in conjunction with other characters, like an A or an M. Therefore the Cs and Os are often enlarged slightly in a vertical direction. I have made these optical adjustments to the characters obtained when using uppercase input. Alternative non-adjusted characters are also provided which can be accessed by inputting lowercase characters. These are the C, G, O and Q characters; typing c, g, o or q will give the non-adjusted uppercase versions.

The Pacioli font, which I have called `cpc` (for Computer PaCioli), is provided in an upright and a slanted version only.

`\cpcfamily` This pacioli package command will switch any following text into the Pacioli font family.

`\textcpc` The pacioli package command `\textcpc{`⟨*text*⟩`}` will typeset its ⟨*text*⟩ argument using the Pacioli font family.

[1] Alan Hoenig's Dürer font is available from CTAN in the `duerer` subdirectory.

3 The Metafont code

3.1 The parameter file

We deal with the two parameter files first, and start by announcing what they are for.

```
⟨*up | sl⟩
⟨up⟩ %%% CPCR10.MF  Computer Pacioli Roman at 10 point design size.
⟨sl⟩ %%% CPCSL10.MF  Computer Pacioli Slanted at 10 point design size.
%
```

Parameters from CMR10.MF are used as much as possible. We also make sure that cmbase is loaded as well as plain Metafont.

```
if unknown cmbase: input cmbase fi
%
⟨up⟩font_identifier:="CPCR"; font_size 10pt#;
⟨sl⟩font_identifier:="CPCSL"; font_size 10pt#;
%
```

The majority of the parameters and values are the same for both fonts.

u
hstretch
vstretch
width-adj
serif-fit
cap-serif-fit
letter-fit

```
u#:=20/36pt#;              % unit width
hstretch:=1;               % horizontal stretching factors; in lieu of u
vstretch:=1;               % vertical stretching factor
width_adj#:=0pt#;          % width adjustment for certain characters
serif_fit#:=0pt#;          % extra sidebar near lowercase serifs
cap_serif_fit#:=5/36pt#;   % extra sidebar near uppercase serifs
letter_fit#:=0pt#;         % extra space added to all sidebars
```

body-height
asc-height
cap-height
fig-height
x-height
math-axis
bar-height
comma-depth
desc-depth
half-height

```
body_height#:=270/36pt#;       % height of tallest characters
asc_height#:=250/36pt#;        % height of lowercase ascenders
cap_height#:=246/36pt#;        % height of caps
fig_height#:=232/36pt#;        % height of numerals
x_height#:=155/36pt#;          % height of lowercase without ascenders
math_axis#:=90/36pt#;          % axis of symmetry for math symbols
bar_height#:=87/36pt#;         % height of crossbar in lowercase e
comma_depth#:=70/36pt#;        % depth of comma below baseline
desc_depth#:=70/36pt#;         % depth of lowercase descenders
half_height#:=1/2cap_height#;  % half the height of caps
%
```

side
thickfudge
mediumfudge
thinfudge
thick
medium
thin

```
side#:=cap_height#;          % Height and width of the square
thickfudge:=1/9; mediumfudge:=1/2; thinfudge:=1/3;
thick#:=thickfudge*side#;    % thickness of all thick limbs in Roman font
medium#:=mediumfudge*thick#; % width of medium limbs
thin#:=thinfudge*thick#;     % thinness of all thin limbs
%
```

eta
serif-darkness
crisp
tiny
fine
rule-thickness

```
eta:=2-sqrt2;            % darkness for serifs; $\eta \approx 0.58$
serif_darkness:=eta;
crisp#:=0pt#;            % diameter of serif corners
tiny#:=0pt#;             % diameter of rounded corners
fine#:=7/36pt#;          % diameter of sharply rounded corners
rule_thickness#:=.4pt#;  % thickness of lines in math symbols
%
```

cap-stem
stem
dish
bracket
jut
cap-jut
beak-jut
slab
dslab

```
cap_stem#:=thick#;     % thickness of majuscule stems
stem#:=thick#;         % thickness of minuscule stems
dish#:=1/36pt#;        % amount erased at top or bottom of serifs
bracket#:=thick#;      % vertical distance from serif base to tangent
jut#:=thick#;          % protrusion of lowercase serifs
cap_jut#:=2/3thick#;   % protrusion of uppercase serifs for I
beak_jut#:=10/36pt#;   % horizontal protrusion of beak serifs
slab#:=eps*1pt#;       % serif and arm thickness
dslab#=0;              %slab value used in Pacioli fonts
%
```

Hrad
Lrad
lrad
srad
trad

Pacioli uses several different radii for serifs, with cap_jut being normal.

```
Hrad#:=2thick#;      % Huge
Lrad#:=3/2thick#;    % Large
lrad#:=thick#;       % large
srad#:=1/2thick#;    % small
trad#:=3/8thick#;    % tiny
%
```

hair
vair

hair is the lowercase hairline breadth, and vair is the vertical diameter of hairlines.

```
hair#:=vair#:=thin#;
%
```

o
apex-o
apex-corr

```
o#:=8/36pt#;        % amount of overshoot for curves
apex_o#:=thin#;     % amount of overshoot for diagonal junctions
apex_corr#=0pt#;    % fine tuning of overshoot
%
```

slant

slant is the amount the font slopes to the right. It is different for the two fonts.

```
⟨up⟩slant:=0;       % tilt ratio $(\Delta x/\Delta y)$
⟨sl⟩slant:=1/6;     % tilt ratio $(\Delta x/\Delta y)$
```

fudge
superness
superpull
beak-darkness

```
%
fudge:=1;       % factor applied to weights of heavy characters
superness:=1/sqrt2;    % parameter for superellipses
superpull:=0;      % extra openness inside bowls
```

```
beak_darkness:=11/30;    % fraction of triangle inside beak serifs
%
```

serifs monospace

We are generating a seriffed, variable-width font.

```
serifs:=true;        % should serifs and bulbs be attached?
monospace:=false;    % should all characters have the same width?
%
```

dot-size square-dots hefty ligs

These parameters are required for the punctation glyphs.

```
dot_size#:=38/36pt#; % size of dots
square_dots:=false;  % should dots be square?
hefty:=false;        % should we try hard not to be overweight?
ligs:=2;             % level of ligatures to be included

```

flare stem-corr curve

These parameters are required for the ampersand and question mark.

```
flare#:=33/36pt#;     % diameter of bulbs or breadth of terminals
stem_corr#:=1/36pt#;  % small refinement of stem breadth
curve#:=30/36pt#;     % lowercase curve breadth

```

whole-font

We are not generating a complete font.

```
boolean whole_font;  % try generating a complete font?
whole_font:=false;

```

Finally, call the driver file for the Pacioli font.

```
generate cpctitle      %% switch to the driver file---Pacioli title

⟨/up | sl⟩
```

3.2 The driver file

To a large extent the driver file is based on Alan Hoenig's dtitle.mf.

```
⟨*mfd⟩
% This is CPCTITLE.MF, adapted from Alan Hoenig's dtitle.mf
% This makes a short font.
% This file also contains special macros universally needed in the
% Pacioli font but either not in the cmbase.mf file or
% modifications of this.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%  MACRO SECTION %%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
```

makebox

This is a new version of the cmbase.mf makebox macro. Essentially, as we are only generating an uppercase font, we don't need all the horizontals for the lowercase characters.

```
% A new version of cmbase.mf makebox macro
def makebox(text rule) =
  for y=0,half_height,cap_height,body_height,x_height,-desc_depth,-body_depth:
    rule((l,y)t_,(r,y)t_); endfor  % horizontals
  for x=l,r: rule((x,-body_depth)t_,(x,body_height)t_); endfor % verticals
  for x=u*(1+floor(1/u)) step u until r-1:
    rule((x,-body_depth)t_,(x,body_height)t_); endfor         % more verticals
  if charic<>0:
    rule((r+charic*pt,h.o_),(r+charic*pt,.5h.o_)); fi  % italic correction
enddef;

```

proofpcbb The macro proofpcbb(S, xoff, yoff) draws crosses at the corners of Pacioli's square, and at the bottom and top midpoints, where the point zS is at position (xoff,yoff) in the square.

```

% A macro to delineate Pacioli's bounding square in proof mode.
% Point z$ is at (xoff,yoff) in the bounding square.
def proofpcbb(suffix $)(expr xoff, yoff) =
if proofing > 0:
  x$bl = x$ - xoff; y$bl = y$ - yoff;         % Bottom Left Hand Corner
  z$br = (x$bl+cap_height, y$bl);             % BRHC
  z$tr = (x$bl+cap_height, y$bl+cap_height);  % TRHC
  z$tl = (x$bl, y$bl+cap_height);             % TLHC
  z$bc = 1/2[z$bl,z$br]; z$tc = 1/2[z$tl,z$tr];  % horizontal mid points
  pair hc, vc;
  hc = (2u,0); vc = (0,2u);
  pickup pensquare scaled 2;
  for s = z$bl, z$br, z$tr, z$tl, z$bc, z$tc:
    draw s-hc--s+hc;
    draw s-vc--s+vc;
  endfor
fi
enddef;

```

cm-arm Hoenig's modification to Knuth's arm macro.

arm

```

let cm_arm=arm;

vardef arm(suffix $,$$,@)(expr darkness, armjut) =  % arm from |z$| to |z$$|
numeric vjut_; vjut_= abs(bot y$l-bot y$$);
x@0=good.x(x$$r-armjut); y@0=y$r;
 if serifs: y@1=y$l; z@1=z$$l+whatever*(z$$r-z@0);
   if x$$>=x$: x@2=x@1 - vjut_; else: x@2=x@1+vjut_; fi y@2=y$l;
  filldraw z$$l{z@1-z$$l}...darkness[z@1,.5[z@2,z$$l]]{z@2-z$$l}...
  z@2if x$$>=x$: {left} else: {right} fi
   ---z$l--z$r--z@0--z$$r..cycle; % arm and beak
 else: filldraw z$l--z$r--z@0--z$$r--cycle; fi  % sans-serif arm
 penlabels(@0,@1,@2); enddef;

```

foot

This is a modified version of Hoenig's `foot` macro to draw the bottom leg of the E and L characters.

```

vardef foot(suffix $, $$, @)(expr darkness, armjut) = % curved beak on E and L
numeric vjut_; vjut_= abs(bot y$-bot y$$);
x@0=good.x(x$$r-armjut); y@0=y$r;
 if serifs:
   y@1=y$l; z@1=z$$l+whatever*(z$$r-z@0);
   x@2=x@1-vjut_; y@2=y$l;
   filldraw z$$l{z@1-z$$l}...darkness[z@1,.5[z@2,z$$l] ]...z@2
      ---z$l--z$r--z@0--z$$r--cycle; % arm and beak
 else:
   filldraw z$l--z$r--(x$$r,y$r)--(x$$r,y$l)--cycle;
 fi
 penlabels(@0,@1,@2); enddef;

```

pulled-super-arc
pulled-arc

Hoenig's rewrite of Knuth's macros.

```

vardef pulled_super_arc.l(suffix $, $$)(expr darkness)=
 pair corner;
 if y$=y$r: corner=(x$l, y$$l);
 else: corner=(x$$l,y$l); fi
 z$l{corner-z$l}...(darkness)[corner, .5[z$l, z$$l]]{z$$l-z$l}
 ...{z$$l-corner}z$$l enddef;

vardef pulled_super_arc.r(suffix $, $$)(expr darkness)=
 pair corner;
 if y$=y$r: corner=(x$r,y$$r);
 else: corner=(x$$r, y$r); fi
 z$r{corner-z$r}...(darkness)[corner, .5[z$r, z$$r]]{z$$r-z$r}
 ...z$$r{z$$r-corner} enddef;

vardef pulled_arc@#(suffix $, $$)=
 pulled_super_arc@#($,$$)(eta) enddef;

```

cm-font-setup
slim-font-setup

Hoenig's font set up.

```

let cm_font_setup=font_setup; % Knuth's original font_setup

def slim_font_setup = % trimmed font_setup for use with Pacioli caps only
 if monospace: let adjust_fit=mono_adjust_fit;
  def mfudged=fudged enddef;
  mono_charic#:=body_height#*slant;
  if mono_charic#<0: mono_charic#:=0; fi
  mono_charwd#:=9u#; define_whole_pixels(mono_charwd);
 else: let adjust_fit=normal_adjust_fit;
  def mfudged= enddef; fi
 define_pixels(u,width_adj,serif_fit,cap_serif_fit,jut,cap_jut,bar_height,
  dish,bracket,beak_jut,apex_o,apex_corr);
 define_whole_pixels(letter_fit,fine,crisp,tiny);
 define_whole_vertical_pixels(body_height,asc_height,
  cap_height,x_height,comma_depth,desc_depth);
 define_whole_blacker_pixels(hair,stem,cap_stem);
 define_whole_vertical_blacker_pixels(vair,dslab,slab);
if slab<eps: slab:=eps fi;
 define_corrected_pixels(o);
 forsuffixes $=hair,stem,cap_stem:
  fudged$.#:=fudge*$.#; fudged$:=hround(fudged$.#*hppp+blacker);
  forever: exitif fudged$>.9fudge*$; fudged$:=fudged$+1; endfor endfor
 rule_thickness:=ceiling(rule_thickness#*hppp);
 heavy_rule_thickness:=ceiling(3rule_thickness#*hppp);
 oo:=vround(.5o#*hppp*o_correction)+eps;
 apex_oo:=vround(.5apex_o#*hppp*o_correction)+eps;
 lowres_fix(stem) 1.3;
 ess:=(ess#/stem#)*stem; cap_ess:=(cap_ess#/cap_stem#)*cap_stem;
 dw:=(curve#-stem#)*hppp; bold:=curve#*hppp+blacker;
 dh#:=.6designsize;
 more_super:=max(superness,sqrt .77superness);
 hein_super:=max(superness,sqrt .81225258superness); % that’s $2^{-.3}$
 clear_pen_memory;
 if fine=0: fine:=1; fi
 forsuffixes $=fine,crisp,tiny:
%%% fine $ %%%% temporary formatting convention for MFT
  if $>fudged.hair: $:=fudged.hair; fi
  $.breadth:=$;
  pickup if $=0: nullpen else: pencircle scaled $; $:=$-eps fi;
  $.nib:=savepen; breadth_[$.nib]:=$;
  forsuffixes $$=lft,rt,top,bot: shiftdef($.$$,$$ 0); endfor endfor
%%% @ $ %%%% restore ordinary formatting for $
 min_Vround:=max(fine.breadth,crisp.breadth,tiny.breadth);
 if min_Vround<vround min_Vround: min_Vround:=vround min_Vround; fi
 pickup pencircle scaled rule_thickness; rule.nib:=savepen;
 currenttransform:=identity slanted slant
  yscaled aspect_ratio scaled granularity;
 if currenttransform=identity: let t_=relax
 else: def t_ = transformed currenttransform enddef fi;
 numeric paren_depth#; .5[body_height#,-paren_depth#]=math_axis#;
 numeric asc_depth#; .5[asc_height#,-asc_depth#]=math_axis#;
 body_depth:=desc_depth+body_height-asc_height;
 shrink_fit:=1+hround(2letter_fit#*hppp)-2letter_fit;
 if not string mode: if mode<=smoke: shrink_fit:=0; fi fi
 enddef;

```

pacioli-font-setup This macro contains additional set up for the Pacioli fonts.

```

def pacioli_font_setup= % contains special stuff for Pacioli fonts
  define_pixels(side,thick,thin,medium,half_height);
  define_pixels(Hrad,Lrad,lrad,srad,trad);
  define_pixels(dot_size,math_axis,flare,stem_corr,curve);
  if not known dslab#: dslab=eps; fi
enddef;


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%  END OF MACROS %%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

Set the mode and font setup.

```

font_coding_scheme:="ASCII caps and punctuation";

mode_setup;

pacioli_font_setup;
if whole_font: font_setup; %slab:=eps;
 define_whole_vertical_blacker_pixels(dslab,slab);
 else: slim_font_setup; fi

```

We generate Pacioli's uppercase letters, together with (matching) Computer Modern Roman punctuation and some analphabetics.

```

input cpcromanu;   % Pacioli upper case (majuscules)
input cpcpunct;    % punctuation
input cpclig;      % the several dashes
input cpcromanp    % ampersand and ?mark

```

Finally, do the uppercase ligtable.

```

font_slant slant; font_x_height x_height#;
if monospace: font_normal_space 9u#; % no stretching or shrinking
 font_quad 18u#;
 font_extra_space 9u#;
else: font_normal_space 6u#+2letter_fit#;
 font_normal_stretch 3u#; font_normal_shrink 2u#;
 font_quad 18u#+4letter_fit#;
 font_extra_space 2u#;
 k#:=-.5u#; kk#:=-1.5u#; kkk#:=-2u#; % three degrees of kerning
 ligtable "P": "T": "Y": "A" kern kk#;
 ligtable "F": "V": "W": "A" kern if serifs: kkk# else: kk#fi,
  "K": "X": "O" kern k#, "C" kern k#, "G" kern k#, "Q" kern k#;
 ligtable "O": "D": "X" kern k#, "W" kern k#, "A" kern k#,
   "V" kern k#, "Y" kern k#;
 ligtable "A": if serifs: "R": fi
  "C" kern k#, "O" kern k#, "G" kern k#, "U" kern k#, "Q" kern k#,
  "L": "T" kern kk#, "Y" kern kkk#, "V" kern kk#, "W" kern kkk#;
ligtable "I": "I" kern -k#; fi % Richard III

bye.

⟨/mfd⟩
```

3.3 The majuscule code

The following code is based on Alan Hoenig's code for the Dürer characters. The geometry, of course, is different for Pacioli's characters.

```
⟨*maj⟩
% CPCROMANU.MF Program files for formal Computer Pacioli fonts.
% Based on Alan Hoenig's dromanu.mf for Durer fonts

def checkthin=
if thin-currentbreadth<eps: thin:=currentbreadth+eps; fi;
if thin-currentbreadth<eps: thin:=currentbreadth+eps; fi enddef;

numeric P_width#; P_width#=cap_jut#+3thick#+1/4side#-thin#;
define_pixels(P_width);

pair xeps, yeps; xeps=(eps,0); yeps=(0,eps);
numeric tail_corr; tail_corr=1/8;
numeric cap_jut_factor; cap_jut_factor=8/7;

numeric pcshiftx#; % x value of LHS of Pacioli box wrt origin of char box
define_pixels(pcshiftx);

```

A The letter A. The right and left legs are thick and medium respectively, while the cross-bar is thin. With serifs, it just fits inside the square. The apex is slightly extended, then an arc is removed.

```

cmchar "Pacioli's letter A";
beginchar("A", 2u#+side#, cap_height#, 0);
adjust_fit(cap_serif_fit#, cap_serif_fit#);
pcshiftx := u;
top y3= top y2=h + 2cap_jut;
  % for left leg
numeric alpha[]; alpha1=(h++(0.5side - cap_jut - 0.5thin))/(side);
penpos1(alpha1*medium,0); penpos2(alpha1*medium,0);
  % and for the right leg
alpha2=(h++(0.5side-0.5thick))/(side);
penpos3(alpha2*thick,0); penpos4(alpha2*thick,0);
```

```
lft x1l=cap_jut_factor*lrad+.5u; rt x4r=pcshiftx + side;
bot y1=bot y4=0 if monospace: +1/2 crisp fi;
x3r=x2l = 0.5side + pcshiftx;
z0=whatever[z1r,z2r]=whatever[z3l,z4l];
fill z0--diag_end(0,4l,1,1,4r,3r)--z3r..z2l--diag_end(2l,1l,1,1,1r,0)
  --z0..cycle;  % diagonals
top y5r=top y6r=hround(h/2); top y5r-bot y5l=top y6r-bot y6l=thin;
penpos5(whatever, angle(z2-z1)); penpos6(whatever, angle(z3-z4));
z5=whatever[z1,z2]; z6=whatever[z3,z4];
penstroke z5e--z6e; % bar line
pickup tiny.nib;
numeric top_bar; top_bar=top y5r-bot y6l;
if monospace: pickup pencircle scaled 0pt;
 pos11(top_bar,90); pos12(top_bar,90);
 top y11r=top y12r=h; w/2=.5[x11,x12]; x12-x11=apex_corr+2cap_jut;
 fill z11l--z12l{right}..tension atleast.95..{left}z12r--z11r{left}..
 tension atleast.95..{right}cycle;
 penlabels(11,12); fi
if serifs: bracket:=10/7 bracket; cap_jut:=cap_jut_factor*cap_jut;
 pickup tiny.nib; numeric jut_factor;
 if monospace: jut_factor:=1; else: jut_factor:=3/4; fi
 serif(1,2,a,.8serif_darkness,-cap_jut);serif(4,3,d,.8serif_darkness, cap_jut);
 bracket:=jut_factor*bracket; if monospace: jut_factor:=.6; fi
 serif(1,2,b,7/8 serif_darkness,jut_factor*cap_jut);
 serif(4,3,c,7/8 serif_darkness,-jut_factor*cap_jut);
 fi
% remove the apex
  x20 = x2l - 0.5lrad; y20 = side+lrad;
  erase filldraw fullcircle scaled 2lrad shifted (x20,y20);
proofpcbb(1, lrad+0.5medium, 0);
penlabels(0,1,2,3,4,5,6,20); endchar;

```

B The letter B.

```

cmchar "Pacioli's letter B";
beginchar("B",u#+cap_jut#+0.64side#, cap_height#, 0);
adjust_fit(cap_serif_fit#,0); pickup tiny.nib;
pcshiftx := 0.5u;
pos1(thick,0); pos2(thick,0); lft x1l=lft x2l=pcshiftx+cap_jut;
 bot y2=0; top y1=h; filldraw stroke z1e--z2e; % vertical stem
pos3(thin,-90); pos5(thick,0); pos7(medium,90);
z3r=z2; y4l=y3l; x4l=x3l+thick; y4r=y3r; x4r=x4l+thick-1/2thin; z4=.5[z4l,z4r];
y7=h/2; x7=x1; y6l=y7l; x6l=x4l; top y6r-bot y6l=medium; x6r=x4r;
z6=.5[z6r,z6l];
x5=hround(x1+0.64side-thick); y5=0.25side;
filldraw stroke z3e--z4e..pulled_arc.e(4,5)&pulled_arc.e(5,6)..z6e--z7e;
 % bottom lobe
pos7'(thin,-90); pos8(thick,0); pos10(medium,90);
z7'=z7; z6'=z6; x6'r=x4r; x6'l=x4l; y6'r=y7'r; y6'l=y7'l;
```

```
z10r=z1; x9r=x6'r; x9l=x6'l; y9r=y10r; top y9r-bot y9l=medium; z9=.5[z9r,z9l];
x8=hround(x1+0.53side-thick); y8=.75side;
filldraw stroke z7'e--z6'e..pulled_arc.e(6',8)&pulled_arc.e(8,9)
 ..z9e--z10e; % upper lobe
if serifs: serif(1,2,a, serif_darkness,-cap_jut);
  serif(2,1,b,serif_darkness,-cap_jut);
 numeric tempa_, tempb_; tempa_=slab; tempb_=bracket;
 slab:=thin; bracket:=2/3cap_jut; serif(2,1,c,serif_darkness,2/3cap_jut);
 slab:=tempa_; bracket:=tempb_; fi
proofpcbb(2, cap_jut+0.5thick, 0);
penlabels(1,2,3,4,5,6,7,8,9,10,11,12);
penlabels(6',7'); endchar;

```

c The letter C. This is Pacioli's original definition.

```

cmchar "Pacioli's original letter C";
beginchar("c",2u#+.95side#, cap_height#, 0);
adjust_fit(0, 0); if monospace: pickup fine.nib; else: pickup tiny.nib;fi
pcshiftx := u;
numeric alpha; alpha=w-u-thick;
pos2(thin, 90); x2=.5[x3r,x0r]; top y2r=h;% topmost pen position
pos3(thick,180); lft z3r=(u,h/2);
pos4(thin, 270); x4=x2; bot y4r=0;  % bottommost pen position
pos0(thick,0); lft z0l=(w+thick-u,h/2);
path p, p', line.u, line.b; line.u=(alpha,1.5h)--(alpha,h/2);
 line.b=(.5[alpha,x0],0)--(.5[alpha,x0],h/2);
p=pulled_arc.r(0,2)&pulled_arc.r(2,3)&pulled_arc.r(3,4)
  &pulled_arc.r(4,0)&cycle;
p'=pulled_arc.l(0,2)&pulled_arc.l(2,3)&pulled_arc.l(3,4)
  &pulled_arc.l(4,0)&cycle;
z'1l=line.u intersectionpoint p'; z'1r=line.u intersectionpoint p;
bot rt z1l=z'1l; top rt z1r=z'1r; z1=.5[z1l,z1r];
z'5l=line.b intersectionpoint p'; z'5r=line.b intersectionpoint p;
z5r = (side+pcshiftx-3/4thick, 3/4thick); z5l = (x5r+o, y5r+1.5thick); z5=.5[z5r,z5l];
if not monospace:
 x1l:=x1l+.6thin; y1l:=y1r-1.0thick-2tiny;
  y5l:=2thick+y5r-2tiny; fi
if monospace: x1:=x1l:=x1r:=x5; y1:=y1-thin; y1l:=y1l-thin; y1r:=y1r-.6thin; fi
 % readjust to have a better opening in CDTT
numeric slope_factor; if monospace:slope_factor=4/3; else:slope_factor=3/4; fi
filldraw
 z1r{-1,slope_factor}...{left}z2r..pulled_arc.r(2,3)&
 pulled_arc.r(3,4)..{right}z4r..tension 0.9..z5r--
 reverse
  (z1l..tension 0.9..{left}z2l..pulled_arc.l(2,3)&pulled_arc.l(3,4)..{right}z4l
  ..tension 0.9..z5l)
 ..(z1l+(eps,0))--(z1r+(eps,0))..cycle;
proofpcbb(3, 0.5thick, 0.5side);
penlabels(0,1,2,3,4,5); endchar;
```

C The letter C. This includes an optical adjustment at the top and bottom of the character.

```
cmchar "Pacioli’s letter C";
beginchar("C",2u#+.95side#, cap_height#, 0);
adjust_fit(0, 0); if monospace: pickup fine.nib; else: pickup tiny.nib;fi
pcshiftx := u;
numeric alpha; alpha=w-u-thick;
pos2(thin, 90); x2=.5[x3r,x0r]; top y2r=h+o;% topmost pen position
pos3(thick,180); lft z3r=(u,h/2);
pos4(thin, 270); x4=x2; bot y4r=-o;  % bottommost pen position
pos0(thick,0); lft z0l=(w+thick-u,h/2);
path p, p’, line.u, line.b; line.u=(alpha,1.5h)--(alpha,h/2);
 line.b=(.5[alpha,x0],0)--(.5[alpha,x0],h/2);
p=pulled_arc.r(0,2)&pulled_arc.r(2,3)&pulled_arc.r(3,4)
  &pulled_arc.r(4,0)&cycle;
p’=pulled_arc.l(0,2)&pulled_arc.l(2,3)&pulled_arc.l(3,4)
  &pulled_arc.l(4,0)&cycle;
z’1l=line.u intersectionpoint p’; z’1r=line.u intersectionpoint p;
bot rt z1l=z’1l; top rt z1r=z’1r; z1=.5[z1l,z1r];
z’5l=line.b intersectionpoint p’; z’5r=line.b intersectionpoint p;
z5r = (side+pcshiftx-3/4thick, 3/4thick); z5l = (x5r+o, y5r+1.5thick); z5=.5[z5r,z5l];
if not monospace:
 x1l:=x1l+.6thin; y1l:=y1r-1.0thick-2tiny;
  y5l:=2thick+y5r-2tiny; fi
if monospace: x1:=x1l:=x1r:=x5; y1:=y1-thin; y1l:=y1l-thin; y1r:=y1r-.6thin; fi
 % readjust to have a better opening in CDTT
numeric slope_factor; if monospace:slope_factor=4/3; else:slope_factor=3/4; fi
filldraw
 z1r{-1,slope_factor}...{left}z2r..pulled_arc.r(2,3)&
 pulled_arc.r(3,4)..{right}z4r..tension 0.9..z5r--
 reverse
  (z1l..tension 0.9..{left}z2l..pulled_arc.l(2,3)&pulled_arc.l(3,4)..{right}z4l
  ..tension 0.9..z5l)
 ..(z1l+(eps,0))--(z1r+(eps,0))..cycle;
proofpcbb(3, 0.5thick, 0.5side);
penlabels(0,1,2,3,4,5); endchar;
```

D The letter D. With serifs, it fills the square.

```
cmchar "Pacioli’s letter D";
beginchar("D", cap_jut#+.95side#+u#, cap_height#, 0);
adjust_fit(cap_serif_fit#,0);
pickup tiny.nib; pos1(cap_stem,0); pos2(cap_stem,0);
lft x1l=.5u+cap_jut; x2=x1; bot y2=0; top y1=h;
filldraw stroke z1e--z2e; % stem
pos3(thin, 90); pos5(thick, 0); pos7(5/4thin, -90);
```

```
x3=x7=x1; top y3r=h; bot y7r=0;
y5=good.y h/2; x5=h-3/2thick+x1;
if not monospace: x4r=x6r=hround(w/2); top y4r=h; bot y6r=0;
  bot y4l=vround(h-thin); top y6l=vround 5/4thin;
  x4l=x6l;  2(lft x5l-x4l)=y4l-y6l;
  z4=.5[z4l, z4r]; z6=.5[z6l, z6r];
  else: pos4(thin,90); pos6(thin,-90); x4=x6=hround w/2; top y4r=h; bot y6r=0;
fi
filldraw stroke z3e{right}...pulled_super_arc.e(4,5,eta)&
  pulled_super_arc.e(5,6,eta)...{left}z7e; % lobe
if serifs: serif(1, 2, a, serif_darkness, -cap_jut);
           serif(2, 1, b, serif_darkness, -cap_jut);
  if not monospace: numeric temp_; temp_=slab; slab:=5/4thin;
    serif(2, 1, c, serif_darkness, cap_jut); slab:=temp_; fi fi
proofpcbb(2, cap_jut+0.5thick, 0);
penlabels(1,2,3,4,5,6,7); endchar;

```

E The letter E. It is half the width of the square, but the bottom arm is slighty longer.

```

cmchar "Pacioli's letter E";
numeric topbar_; topbar_=1/2;          % half the square for Pacioli
numeric middlebar_; middlebar_=10/31;  % as measured
beginchar("E", u#+cap_jut#+5.25thick#, cap_height#,0);
adjust_fit(cap_serif_fit#,0);
pcshiftx := 0.5u;
pickup tiny.nib; pos1(cap_stem,0); pos2(cap_stem,0);
lft x1l=lft x2l=.5u+cap_jut; top y1=h; bot y2=0;
filldraw stroke z1e--z2e; % stem
pickup crisp.nib; % prepare for arms and serifs
pos3(medium,90); pos4(dslab,0);
x1=x3; top y3r=h; rt x4l=x2l+topbar_*side;
 bot y4=vround(h-if monospace: 2 fi vstretch*thick);
arm(3,4,d,serif_darkness,0); % top arm and beak
pos5(thin,-90); pos0(thin,90); pos6(dslab,0); pos7(dslab,0);
x5=x0=x1; y5=y0=hround(h/2); rt x6r=rt x7r= x1r+middlebar_*side;
top y6-bot y7= thin+3/4thick; y5=.5[y6,y7];
arm(5,6,e,eta,0); arm(0,7,f,eta,0);  % middle arms and beaks
% now do the bottom arm
pos8(medium,-90); pos9(dslab,0); x8=x2; bot y8r=0; rt x9r=hround(x2l+5.25thick);
y9=y8r+thick;
foot(8,9,g,serif_darkness,0.25thick); % lower arm
if serifs: serif(1,2,a,serif_darkness,-cap_jut);
           serif(2,1,b,serif_darkness,-cap_jut);
  if not monospace: temp_:=slab; slab:=medium; tmp_:=bracket; bracket:=cap_jut+medium;
    serif(2,1,c,serif_darkness,cap_jut);
    slab:=temp_; bracket:=tmp_; fi fi
proofpcbb(2, cap_jut+0.5thick, 0);
penlabels(0,1,2,3,4,5,6,7,8,9); endchar; % end of "E"
```

F The letter F. This is an E without the bottom arm.

```
cmchar "Pacioli's letter F";
numeric topbar_; topbar_=1/2;   % half the square for Pacioli
numeric middlebar_; middlebar_=10/31;  % as measured
beginchar("F", u#+1/2side#+cap_jut#, cap_height#,0);
adjust_fit(cap_serif_fit#,0);
pcshiftx := 0.5u;
pickup tiny.nib; pos1(cap_stem,0); pos2(cap_stem,0);
lft x1l=lft x2l=.5u+cap_jut; top y1=h; bot y2=0;
filldraw stroke z1e--z2e; % stem
pickup crisp.nib; % prepare for arms and serifs
pos3(medium,90); pos4(dslab,0);
x1=x3; top y3r=h; rt x4l= x2l+topbar_*side;
 bot y4=vround(h-if monospace: 2 fi vstretch*thick);
arm(3,4,d,serif_darkness,0); % top arm and beak
pos5(thin,-90); pos0(thin,90); pos6(dslab,0); pos7(dslab,0);
x5=x0=x1; y5=y0=hround(h/2); rt x6r=rt x7r= x1r+middlebar_*side;
top y6-bot y7= thin + 3/4thick; y5=.5[y6,y7];
arm(5,6,e,eta,0); arm(0,7,f,eta,0);  % middle arms and beaks
if serifs: serif(1,2,a,serif_darkness,-cap_jut);
 serif(2,1,b,serif_darkness,-cap_jut); serif(2,1,c,serif_darkness, cap_jut);
 fi
proofpcbb(2, cap_jut+0.5thick, 0);
penlabels(0,1,2,3,4,5,6,7); endchar; % end of "F"
```

g The letter G. This is based on the C and is Pacioli's original definition.

```
cmchar "Pacioli's original letter G";
beginchar("g",2u#+.925side#, cap_height#, 0);
adjust_fit(0, cap_serif_fit#);
pcshiftx := u;
if monospace: pickup fine.nib; else: pickup tiny.nib;fi
% This code is identical to that for the letter C.
numeric alpha; alpha=w-u-thick;
pos2(thin, 90); x2=.5[x3r,x0r]; top y2r=h;% topmost pen position
pos3(thick,180); lft z3r=(u,h/2);
pos4(thin, 270); x4=x2; bot y4r=0;
pos0(thick,0); lft z0l=(w+thick-u,h/2);
path p, p', line.u, line.b; line.u=(alpha,1.5h)--(alpha,h/2);
 line.b=(.5[alpha,x0],0)--(.5[alpha,x0],h/2);
p=pulled_arc.r(0,2)&pulled_arc.r(2,3)&pulled_arc.r(3,4)
  &pulled_arc.r(4,0)&cycle;
p'=pulled_arc.l(0,2)&pulled_arc.l(2,3)&pulled_arc.l(3,4)
  &pulled_arc.l(4,0)&cycle;
z'1l=line.u intersectionpoint p'; z'1r=line.u intersectionpoint p;
bot rt z1l=z'1l; top rt z1r=z'1r; z1=.5[z1l,z1r];
```

```
z'5l=line.b intersectionpoint p'; z'5r=line.b intersectionpoint p;
z5r = (side+pcshiftx-3/4thick, 3/4thick); z5l = (x5r+o, y5r+1.5thick); z5=.5[z5r, z5l];
if not monospace:
 x1l:=x1l+.6thin; y1l:=y1r-1.0thick-2tiny;
 y5l:=2thick+y5r-2tiny; fi
if monospace: x1:=x1l:=x1r:=x5; y1:=y1-thin; y1l:=y1l-thin; y1r:=y1r-.6thin; fi
 % readjust to have a better opening in CDTT
numeric slope_factor; if monospace:slope_factor=4/3; else:slope_factor=3/4; fi
% Addtional G work (preceding program is the same as that for letter C)!
path p,p',line.o, line.i;
p=z1r{-1,slope_factor}..{left}pulled_arc.r(2,3)&
 pulled_arc.r(3,4){right}..tension 0.9..z5r;
p'= reverse
  (z1l..tension 0.9..{left}pulled_arc.l(2,3)&pulled_arc.l(3,4){right}
  ..tension 0.9..z5l);
pos7(thick,0);
top z7=(x1r,7/2thick);
numeric counter; counter=0;
forever:
 counter:=counter+1;
 line.o:=z7r--(x7r,-2o); line.i:=(z7r-(thick,0))--(x7r-thick,-2o);
 numeric t[],tt[];
 (t1,tt1)=p intersectiontimes line.o; (t2,tt2)=p' intersectiontimes line.i;
 exitif t1*t2>0; forsuffixes $$=l, r: x7$$:=x7$$-.1u; endfor
 x7:=x7-.1u; endfor
x6r=x7r; x6l=x7l; bot y6r=ypart point t1 of p; bot y6l=ypart point t2 of p';
z6=.5[z6r,z6l];
p:= subpath(0,t1) of p..(z6r-xeps+yeps)--(z6l+xeps+yeps)..
 subpath(t2,infinity) of p'..(z1l+xeps)--cycle;
filldraw p;
filldraw z6l--z7l--z7r--z6r--cycle;
if serifs: serif(7,6,a,serif_darkness,-srad);
 serif(7,6,b,serif_darkness, srad); fi
proofpcbb(3, 0.5thick, 0.5side);
penlabels(0,1,2,3,4,5,6,7); endchar;

```

G The letter G. This is based on the C and has optical adjustments.

```

cmchar "Pacioli's letter G";
beginchar("G",2u#+.925side#, cap_height#, 0);
adjust_fit(0, cap_serif_fit#);
pcshiftx := u;
if monospace: pickup fine.nib; else: pickup tiny.nib;fi
% This code is identical to that for the letter C.
numeric alpha; alpha=w-u-thick;
pos2(thin, 90); x2=.5[x3r,x0r]; top y2r=h+o;% topmost pen position
pos3(thick,180); lft z3r=(u,h/2);
pos4(thin, 270); x4=x2; bot y4r=-o;
pos0(thick,0); lft z0l=(w+thick-u,h/2);
```

```
path p, p', line.u, line.b; line.u=(alpha,1.5h)--(alpha,h/2);
 line.b=(.5[alpha,x0],0)--(.5[alpha,x0],h/2);
p=pulled_arc.r(0,2)&pulled_arc.r(2,3)&pulled_arc.r(3,4)
  &pulled_arc.r(4,0)&cycle;
p'=pulled_arc.l(0,2)&pulled_arc.l(2,3)&pulled_arc.l(3,4)
  &pulled_arc.l(4,0)&cycle;
z'1l=line.u intersectionpoint p'; z'1r=line.u intersectionpoint p;
bot rt z1l=z'1l; top rt z1r=z'1r; z1=.5[z1l,z1r];
z'5l=line.b intersectionpoint p'; z'5r=line.b intersectionpoint p;
z5r = (side+pcshiftx-3/4thick, 3/4thick); z5l = (x5r+o, y5r+1.5thick); z5=.5[z5r, z5l];
if not monospace:
 x1l:=x1l+.6thin; y1l:=y1r-1.0thick-2tiny;
 y5l:=2thick+y5r-2tiny; fi
if monospace: x1:=x1l:=x1r:=x5; y1:=y1-thin; y1l:=y1l-thin; y1r:=y1r-.6thin; fi
 % readjust to have a better opening in CDTT
numeric slope_factor; if monospace:slope_factor=4/3; else:slope_factor=3/4; fi
% Addtional G work (preceding program is the same as that for letter C)!
path p,p',line.o, line.i;
p=z1r{-1,slope_factor}..{left}pulled_arc.r(2,3)&
 pulled_arc.r(3,4){right}..tension 0.9..z5r;
p'= reverse
  (z1l..tension 0.9..{left}pulled_arc.l(2,3)&pulled_arc.l(3,4){right}
  ..tension 0.9..z5l);
pos7(thick,0);
top z7=(x1r,7/2thick);
numeric counter; counter=0;
forever:
 counter:=counter+1;
 line.o:=z7r--(x7r,-2o); line.i:=(z7r-(thick,0))--(x7r-thick,-2o);
 numeric t[],tt[];
 (t1,tt1)=p intersectiontimes line.o; (t2,tt2)=p' intersectiontimes line.i;
 exitif t1*t2>0; forsuffixes $$=l, r: x7$$:=x7$$-.1u; endfor
 x7:=x7-.1u; endfor
x6r=x7r; x6l=x7l; bot y6r=ypart point t1 of p; bot y6l=ypart point t2 of p';
z6=.5[z6r,z6l];
p:= subpath(0,t1) of p..(z6r-xeps+yeps)--(z6l+xeps+yeps)..
 subpath(t2,infinity) of p'..(z1l+xeps)--cycle;
filldraw p;
filldraw z6l--z7l--z7r--z6r--cycle;
if serifs: serif(7,6,a,serif_darkness,-srad);
 serif(7,6,b,serif_darkness, srad); fi
proofpcbb(3, 0.5thick, 0.5side);
penlabels(0,1,2,3,4,5,6,7); endchar;

```

H The letter H. With serifs, this fills the square. The two legs are thick and the cross-bar is thin like the A.

```

cmchar "Pacioli's letter H";
beginchar("H", u#+side#, cap_height#,0);
```

```
adjust_fit(cap_serif_fit#, cap_serif_fit#); pickup tiny.nib;
forsuffixes t=1,2,3,4: pos.t(thick,0); endfor
lft x1l=lft x2l=.5u+cap_jut; bot y2=0; top y1=h;
filldraw stroke z1e--z2e; % left vertical stem
x3=x4=w-x1; y3=y1; y4=y2;
filldraw stroke z3e--z4e; % right vertical stem
pos5(thin, 90); pos6(thin,90);
x5=.5[x1r,x1l]; x6=w-x5; top y5r=top y6r=h/2;
filldraw stroke z5e--z6e; % horizontal crossbar
if serifs:
 serif(1,2,a,serif_darkness,-cap_jut); serif(1,2,b,serif_darkness,cap_jut);
 serif(2,1,c,serif_darkness,-cap_jut); serif(2,1,d,serif_darkness,cap_jut);
 serif(3,4,e,serif_darkness,-cap_jut); serif(3,4,f,serif_darkness,cap_jut);
 serif(4,3,g,serif_darkness,-cap_jut); serif(4,3,h,serif_darkness,cap_jut); fi
proofpcbb(2, cap_jut+0.5thick, 0);
labels(1,2,3,4,5,6); endchar;

```

I The letter I.

```

cmchar "Pacioli's letter I";
beginchar("I", .5u#+2cap_jut#+cap_stem#+.5u#, cap_height#,0);
adjust_fit(cap_serif_fit#, cap_serif_fit#);
pickup tiny.nib; checkthin; pos1(cap_stem,0); pos2(cap_stem,0);
x1=x2=hround(w/2); top y1=h; bot y2=0;
filldraw stroke z1e--z2e; % vertical stem
if serifs:
  serif(1, 2, a, serif_darkness, -cap_jut);
  serif(1, 2, b, serif_darkness, cap_jut); % top lft,rt
  serif(2, 1, c, serif_darkness, -cap_jut);
  serif(2, 1, d, serif_darkness, cap_jut); % bot lft,rt
fi
proofpcbb(2, 0.5side, 0);
penlabels(1, 2); endchar;

```

J The letter J. Pacioli didn't specify a J, so we'll use Dürer's J instead (with suitable changes to the serifs).

```

cmchar "Pacioli's letter J";
%% Pacioli didn't have a J, so use Durer's.
beginchar("J",u#+3thick#+cap_jut#,cap_height#, 1/4 cap_height#);
adjust_fit(0, cap_serif_fit#); pickup tiny.nib;
pos1(thick,0); pos2(thick,0); pos3(thin, -90);
top y1=h; rt x1r=w-.5u-cap_jut; x2=x1; bot y2=0; lft x3=.5u; bot y3r=-d;
filldraw stroke z1e--z2e{down}..{left}z3e;
if serifs: serif(1,2,a,serif_darkness,-cap_jut);
 serif(1,2,b,serif_darkness,cap_jut); fi
proofpcbb(2, 0.5side, 0);
penlabels(1,2,3); endchar;
```

K The letter K.

```
cmchar "Pacioli's letter K";
beginchar("K", 3/2u#+side#, cap_height#,0);
adjust_fit(cap_serif_fit#,0); pickup tiny.nib;
pcshiftx := 3/4u;
pair durer[]; % represent Pacioli's corner points
durer1=(pcshiftx,side); durer2=(pcshiftx+side,side); durer3=(xpart durer1,0);
 durer4=(xpart durer2,0);
numeric diag_angle[];
diag_angle1=angle(durer2-durer3);
diag_angle2=angle(durer1-durer4);
numeric alpha;
alpha=((xpart durer2 - xpart durer1)++h)/h;
pos1(thick,0); pos2(thick,0);
lft x1l=lft x2l=pcshiftx+cap_jut;
 top y1=h; bot y2=0; filldraw stroke z1e--z2e; % left stem
pos3(alpha*medium,0); pos4(alpha*medium,0); z4l = (x2,h/2);
top y3=h; z3r-z4r=whatever*(durer2-durer3);
z0=whatever[z1r,z2r]=whatever[z3l,z4l];
filldraw z4l--z3l--z3r--z4r--cycle; % upper diagonal stroke
pos5(alpha*thick,0); pos6(alpha*thick,0);
z5l = z4l;
bot y6=0; z5r-z6r=whatever*(durer1-durer4);
z10 = whatever[z3r,z4r]=whatever[z6r,z5r];
filldraw z6l--z5l--z10--z6r--cycle;  % bottom leg
if serifs:
 serif(1,2,a,serif_darkness,-cap_jut); serif(1,2,b,serif_darkness,cap_jut);
 serif(2,1,c,serif_darkness,-cap_jut); serif(2,1,d,serif_darkness,cap_jut);
 numeric old_jut, old_brack; old_jut=cap_jut; old_brack=bracket;
% serifs on top leg
 bracket:=.5old_brack;cap_jut:=.5old_jut;
 if not monospace: serif(3,4,e,serif_darkness,-cap_jut); fi
 bracket:=2old_brack; cap_jut:=2old_jut;
  serif(3,4,f,.6serif_darkness,if monospace:.7 fi cap_jut);
% serifs on bottom leg
 bracket:=.5old_brack;cap_jut:=.5old_jut;
 if not monospace: serif(6,5,g,serif_darkness,-cap_jut); fi
 bracket:=2old_brack; cap_jut:=2old_jut;
  serif(6,5,h,.6serif_darkness,if monospace:.7 fi cap_jut);
 cap_jut:=old_jut; bracket:=old_brack;
fi
proofpcbb(2, pcshiftx+cap_jut+0.5thick, 0);
penlabels(0,1,2,3,4,5,6,7); endchar;
```

L The letter L. This is similar, but not identical, to the left leg and bottom arm of Pacioli's E.

```

cmchar "Pacioli's letter L";
numeric topbar_; if monospace: topbar_=11/15;
  else: topbar_=17/30; fi % Duerer's wierd value
beginchar("L", u#+cap_jut#+5.75thick#, cap_height#,0);
adjust_fit(cap_serif_fit#,0);
pcshiftx := 0.5u;
pickup tiny.nib; pos1(cap_stem,0); pos2(cap_stem,0);
lft x1l=lft x2l=.5u+cap_jut; top y1=h; bot y2=0;
filldraw stroke z1e--z2e; % stem
pickup crisp.nib; % prepare for arms and serifs
pos3(thin,90);
pos8(medium,-90); pos9(dslab,0); x8=x2; bot y8r=0; rt x9r=hround(x2l+5.75thick);
y9=y8r+1.5thick; foot(8,9,g,serif_darkness,0.75thick);
  % lower arm
if serifs: serif(1,2,a,serif_darkness,-cap_jut);
 serif(2,1,b,serif_darkness,-cap_jut); serif(1,2,d,serif_darkness,cap_jut);
 if not monospace: temp_:=slab; slab:=medium; tmp_:=bracket; bracket:=thick+medium;
  serif(2,1,c,serif_darkness,thick);
  slab:=temp_; bracket:=tmp_; fi fi
proofpcbb(2, cap_jut+0.5thick, 0);
penlabels(0,1,2,3,4,5,6,7,8,9); endchar; % end of "L"

```

M The letter M. This has slightly sloping right and left legs. The serifs extend a little way outside the square.

```

cmchar"Pacioli's letter M";
beginchar("M", u#+cap_jut#+side#, cap_height#, 0);
adjust_fit(cap_serif_fit#, cap_serif_fit#); pickup tiny.nib;
pcshiftx := 0.5u;
pos1(medium,0); pos2(medium,0);
  lft x2l=pcshiftx+cap_jut; top y1=h; bot y2=0;
  lft x1l = lft x2l + side*sind(1);  % left leg has 1 degree angle
pos3(thick,0); pos4(thick,0);
  x4r=pcshiftx+side; top y3=h; bot y4=0;
  x3r=x4r-side*sind(3);              % right leg has 3 degree angle
numeric long, short, hypot, alpha; long=side; short=1/2side;
hypot=long++short; alpha=hypot/long;
pos5(alpha*thick,0); pos6(alpha*thick,0);  % left inner
z5l=(2pcshiftx,side); bot lft z6l=(pcshiftx+1/2side,0);
pos7(alpha*medium,0); pos8(alpha*medium,0); % right inner
z7r=z6l; z8r=z3r; z0=whatever[z5r,z6r]=whatever[z7l,z8l];
z10=whatever[z5l,z6l]=whatever[z1r,z2r];
z11=whatever[z7r,z8r]=whatever[z3l,z4l];
% draw the M all in one go
filldraw z5r--z0--z8l--z3r--z4r--z4l--z11--z6l--z10--z2r--z2l--z1l--cycle;
if serifs:
 serif(2,1,a,serif_darkness,-cap_jut); serif(1,2,b,serif_darkness,-lrad);
 serif(3,4,c,serif_darkness, lrad); serif(4,3,d,serif_darkness, lrad);
 serif(2,1,e,serif_darkness,cap_jut); serif(4,3,f,serif_darkness,-cap_jut);
  % lower, inner serifs
 fi
proofpcbb(2, cap_jut+0.5medium, 0);
penlabels(0,1,2,3,4,5,6,7,8,10,11); endchar;

```

N The letter N. With serifs it fills the square, and perhaps the top left serif protrudes a tiny bit.

```

cmchar "Pacioli's letter N";
beginchar("N", u#+2cap_jut#+.8side#, cap_height#,0);
adjust_fit(cap_serif_fit#, cap_serif_fit#); pickup tiny.nib;
numeric long, short, hypot, alpha; long=h; short=side-2cap_jut;
hypot=long++short; alpha=hypot/long;
pos1(medium,0); pos2(medium,0); lft x1l=lft x2l=.5u+cap_jut; top y1=h; bot y2=0;
pos3(medium,0); pos4(medium,0); x3=x4; x3=x2+side-2cap_jut-medium; y3=y1; y4=y2;
pos 5(alpha*thick,0); pos6(alpha*thick,0);
z5r=z1r; z6l=z4r; z0=whatever[z1r,z2r]=whatever[z5l,z6l]; % under notch point
z0'=whatever[z5r,z6r]=whatever[z3l,z4l];
filldraw z1r--z1l--z2l--z2r--z0--z6l--z3r--z3l--z0'--cycle;
if serifs:
 serif(2,1,b,serif_darkness,-cap_jut); serif(2,1,c,serif_darkness,cap_jut);
  % bottom, left serifs
 serif(3,4,d,serif_darkness,-cap_jut);serif(3,4,e,serif_darkness,cap_jut);
  % top right serifs
 z.a=(x1l-1.5cap_jut,y1); pickup tiny.nib; filldraw
  z0{z0-z4r}..{left}(z.a-(0,eps))..z.a--z1r--cycle;
  % top left serif
  serif(1,2,a,serif_darkness,-lrad);
  fi
proofpcbb(2, cap_jut+0.5medium, 0);
penlabels(0,0',1,2,3,4,5,6); endchar;

```

o The letter O. He actually specifies two versions of the letter, but this one is *perfectissimo*. It is oversquare horizontally.

```

cmchar "Pacioli's original letter O";
beginchar("o", 2u#+side#+2o#, cap_height#, 0);
adjust_fit(0,0); pair durer[];
pcshiftx := u;
x2r=pcshiftx-o; x4r=w-pcshiftx+o; y1r=h; y3r=0; z0=(w/2,h/2);
z0=.5[z5,z6]; numeric axis_angle;
durer1=(x2r,y1r-o); durer2=(x4r, ypart durer1);
durer3=(xpart durer1, y3r+o); durer4=(xpart durer2, ypart durer3);
axis_angle=angle(durer2-durer3);
z5-z6=(thick,0)rotated axis_angle;
y2r=y6; y4r=y5; x1r=x5; x3r=x6;
```

```
forsuffixes $=1,2,3,4: z$=.5[z$l,z$r]; endfor
x1l=x3r; y1l-y6=y6-y3r; x5-x2l=x4r-x5; y2l=y4r;
x3l=x1r; y1r-y5=y5-y3l; x4l-x6=x6-x2r; y4l=y2r;
path p,p';
p=pulled_arc.r(1,2)&reverse pulled_arc.r(3,2)&
  pulled_arc.r(3,4)&reverse pulled_arc.r(1,4)&cycle;
path anti_diag; anti_diag=z0--durer1;
z21r=p intersectionpoint anti_diag; penpos21(thin, axis_angle+90);
x43r=w-x21r; y43r=h-y21r; penpos43(thin, axis_angle+270);
path diag; diag=z0--durer2;
z41r=diag intersectionpoint p; penpos41(thick, axis_angle);
x32r=w-x41r; y32r=h-y41r; penpos32(thick, axis_angle+180);
if not monospace: p'=
  z1l{left}..z21l..z2l{down}..z32l..
  z3l{right}..z43l..
  z4l{up}..z41l..{left}cycle;
 else:p'=z1l..z21l..z32l..z3l..z43l..z41l..cycle; fi
fill p; unfill p';
proofpcbb(0, 0.5side, 0.5side);
penlabels(0,1,2,3,4,5,6,21,32,43,41); endchar;

```

O The letter O. This includes optical adjustments at the top and bottom.

```

cmchar "Pacioli's letter O";
beginchar("O", 2u#+side#+2o#, cap_height#, 0);
adjust_fit(0,0); pair durer[];
pcshiftx := u;
x2r=pcshiftx-o; x4r=w-pcshiftx+o; y1r=h+o; y3r=-o; z0=(w/2,h/2);
z0=.5[z5,z6]; numeric axis_angle;
durer1=(x2r,y1r-o); durer2=(x4r, ypart durer1);
durer3=(xpart durer1, y3r+o); durer4=(xpart durer2, ypart durer3);
axis_angle=angle(durer2-durer3);
z5-z6=(thick,0)rotated axis_angle;
y2r=y6; y4r=y5; x1r=x5; x3r=x6;
forsuffixes $=1,2,3,4: z$=.5[z$l,z$r]; endfor
x1l=x3r; y1l-y6=y6-y3r; x5-x2l=x4r-x5; y2l=y4r;
x3l=x1r; y1r-y5=y5-y3l; x4l-x6=x6-x2r; y4l=y2r;
path p,p';
p=pulled_arc.r(1,2)&reverse pulled_arc.r(3,2)&
  pulled_arc.r(3,4)&reverse pulled_arc.r(1,4)&cycle;
path anti_diag; anti_diag=z0--durer1;
z21r=p intersectionpoint anti_diag; penpos21(thin, axis_angle+90);
x43r=w-x21r; y43r=h-y21r; penpos43(thin, axis_angle+270);
path diag; diag=z0--durer2;
z41r=diag intersectionpoint p; penpos41(thick, axis_angle);
x32r=w-x41r; y32r=h-y41r; penpos32(thick, axis_angle+180);
if not monospace: p'=
  z1l{left}..z21l..z2l{down}..z32l..
  z3l{right}..z43l..
  z4l{up}..z41l..{left}cycle;
 else:p'=z1l..z21l..z32l..z3l..z43l..z41l..cycle; fi
fill p; unfill p';
proofpcbb(0, 0.5side, 0.5side);
penlabels(0,1,2,3,4,5,6,21,32,43,41); endchar;

```

P The letter P.

```

cmchar "Pacioli's letter P";
beginchar("P", 1.5u#+P_width#, cap_height#, 0);
adjust_fit(cap_serif_fit#, 0);
pickup tiny.nib; pos1(thick,0); pos2(thick,0);
top lft z1l=(.5u+cap_jut,h); bot lft z2l=(.5u+cap_jut,0);
filldraw stroke z1e--z2e; % vertical stem
pos3(medium,90); pos4(medium,90); pos5(thick,0);
pos6(if not monospace and serifs: .5thin else:thin fi,-90);
top y3r=top y4r=h; x3=x1; x4=x1r+thick; filldraw stroke z3e--z4e;
x6=x4; y6=hround(h/2 -medium); rt x5r=hround(x2+0.65side-cap_jut-0.5medium); y5=.75side;
y7=y6; if serifs: x7=.5[rt x1r, x6]; else: x7=x3; fi
pos7(if not monospace and serifs:.5thin else:thin fi,-90);
filldraw stroke pulled_super_arc.e(4,5,eta)&pulled_super_arc.e(5,6,eta)
 ..z7e; % lobe
if serifs: serif(1,2,a,serif_darkness,-cap_jut);
 serif(2,1,b,serif_darkness,-cap_jut);
 serif(2,1,c,serif_darkness,cap_jut); fi
proofpcbb(2, cap_jut+0.5thick, 0);
penlabels(1,2,3,4,5,6,7); endchar;

```

q The letter Q. It is essentially an O with a long (not very attractive) tail.

```

cmchar "Pacioli's original letter Q";
beginchar("q", 2u#+side#+2o#, cap_height#, 0);
adjust_fit(0,0); pair durer[];
pcshiftx := u;
x2r=pcshiftx-o; x4r=w-pcshiftx+o; y1r=h; y3r=0; z0=(w/2,h/2);
z0=.5[z5,z6]; numeric axis_angle;
durer1=(x2r,y1r-o); durer2=(x4r, ypart durer1);
durer3=(xpart durer1, y3r+o); durer4=(xpart durer2, ypart durer3);
axis_angle=angle(durer2-durer3);
z5-z6=(thick,0)rotated axis_angle;
y2r=y6; y4r=y5; x1r=x5; x3r=x6;
forsuffixes $=1,2,3,4: z$=.5[z$l,z$r]; endfor
x1l=x3r; y1l-y6=y6-y3r; x5-x2l=x4r-x5; y2l=y4r;
x3l=x1r; y1r-y5=y5-y3l; x4l-x6=x6-x2r; y4l=y2r;
path p,p';
p=pulled_arc.r(1,2)&reverse pulled_arc.r(3,2)&
  pulled_arc.r(3,4)&reverse pulled_arc.r(1,4)&cycle;
path anti_diag; anti_diag=z0--durer1;
```

```
z21r=p intersectionpoint anti_diag; penpos21(thin, axis_angle+90);
x43r=w-x21r; y43r=h-y21r; penpos43(thin, axis_angle+270);
path diag; diag=z0--durer2;
z41r=diag intersectionpoint p; penpos41(thick, axis_angle);
x32r=w-x41r; y32r=h-y41r; penpos32(thick, axis_angle+180);
if not monospace: p'=
  z1l{left}..z21l..z2l{down}..z32l..
  z3l{right}..z43l..
  z4l{up}..z41l..{left}cycle;
 else:p'=z1l..z21l..z32l..z3l..z43l..z41l..cycle; fi
fill p; unfill p';
% now for the tail!
z10r=z3l; z10l=z3r;
z11r=(pcshiftx+side,-0.25side); z11l=z11r-(0,thick);
z12l=z11l+(0.5side+medium,-thick); z12r=z12l+(0,0.5thin);
filldraw z10r..z11r..z12r--z12l..z11l..z10l--cycle;
proofpcbb(0, 0.5side, 0.5side);
penlabels(0,1,2,3,4,5,6,10,11,12,13,14,15); endchar;

```

Q The letter Q. This includes the optical adjustments.

```

cmchar "Pacioli's letter Q";
beginchar("Q", 2u#+side#+2o#, cap_height#, 0);
adjust_fit(0,0); pair durer[];
pcshiftx := u;
x2r=pcshiftx-o; x4r=w-pcshiftx+o; y1r=h+o; y3r=-o; z0=(w/2,h/2);
z0=.5[z5,z6]; numeric axis_angle;
durer1=(x2r,y1r-o); durer2=(x4r, ypart durer1);
durer3=(xpart durer1, y3r+o); durer4=(xpart durer2, ypart durer3);
axis_angle=angle(durer2-durer3);
z5-z6=(thick,0)rotated axis_angle;
y2r=y6; y4r=y5; x1r=x5; x3r=x6;
forsuffixes $=1,2,3,4: z$=.5[z$l,z$r]; endfor
x1l=x3r; y1l-y6=y6-y3r; x5-x2l=x4r-x5; y2l=y4r;
x3l=x1r; y1r-y5=y5-y3l; x4l-x6=x6-x2r; y4l=y2r;
path p,p';
p=pulled_arc.r(1,2)&reverse pulled_arc.r(3,2)&
  pulled_arc.r(3,4)&reverse pulled_arc.r(1,4)&cycle;
path anti_diag; anti_diag=z0--durer1;
z21r=p intersectionpoint anti_diag; penpos21(thin, axis_angle+90);
x43r=w-x21r; y43r=h-y21r; penpos43(thin, axis_angle+270);
path diag; diag=z0--durer2;
z41r=diag intersectionpoint p; penpos41(thick, axis_angle);
x32r=w-x41r; y32r=h-y41r; penpos32(thick, axis_angle+180);
if not monospace: p'=
  z1l{left}..z21l..z2l{down}..z32l..
  z3l{right}..z43l..
  z4l{up}..z41l..{left}cycle;
 else:p'=z1l..z21l..z32l..z3l..z43l..z41l..cycle; fi
```

```
fill p; unfill p’;
% now for the tail!
z10r=z3l; z10l=z3r;
z11r=(pcshiftx+side,-0.25side); z11l=z11r-(0,thick);
z12l=z11l+(0.5side+medium,-thick); z12r=z12l+(0,0.5thin);
filldraw z10r..z11r..z12r--z12l..z11l..z10l--cycle;
proofpcbb(0, 0.5side, 0.5side);
penlabels(0,1,2,3,4,5,6,10,11,12,13,14,15); endchar;
```

R The letter R. Another character that, with serifs, fills the square.

```
cmchar "Pacioli’s letter R";
beginchar("R", u#+side#, cap_height#,0);
adjust_fit(cap_serif_fit#,0);
pickup tiny.nib; pos1(thick,0); pos2(thick,0);
top lft z1l=(.5u+cap_jut,h); bot lft z2l=(.5u+cap_jut,0);
filldraw stroke z1e--z2e; % vertical stem
pos3(thin,90); pos4(thin,90); pos5(thick,0);
pos6(thin, -90);
top y3r=top y4r=h; x3=x1; x4=x1r+thick;
x6=x4; y6=hround h/2; rt x5r=hround(.5u+P_width if not monospace:
 +1/2thin fi); y5=.5[y4,y6];
y7=y6; x7=x3; pos7(thin, -90);
filldraw stroke z3e--pulled_super_arc.e(4,5,eta)&
 pulled_super_arc.e(5,6,eta)..z7e; % lobe
numeric half_letter_width; 2half_letter_width+1/2u=w;
if serifs: half_letter_width:=half_letter_width+1/2u;
 else: half_letter_width:=half_letter_width-1.6thin; fi
path p; p=pulled_super_arc.r(5,6,eta)..z7r;
z0r=p intersectionpoint ((half_letter_width,0)--(half_letter_width,h));
path circ_; circ_= (((halfcircle scaled 2thick) rotated 180) shifted z0r);
z0l= p intersectionpoint circ_;
if not monospace: z0’=(.45[half_letter_width,w],0);
 else: z0’=(.42[half_letter_width,w],0); fi
numeric tail_angle; tail_angle=angle(z0l-z0’);
penpos8(thick, tail_angle-90); z8l=whatever[z0’, z0l];
 x8=1/5[half_letter_width,w];
penpos9(2/3thick,tail_angle-90); z9r=whatever[z0r, z8r];
 x9r=.5[half_letter_width, w];
%penpos10(if monospace:1/2 fi dslab,90); z10l=(w,0);
penpos10(if monospace:1/2 dslab else: .5thin fi,90); z10l=(w,0);
fill z0l--z8l...{right}z10l--z10r{-w,1/4h}..z9r--z0r--cycle; % tail
if serifs: serif(1,2,a,serif_darkness,-cap_jut);
 serif(2,1,b,serif_darkness,-cap_jut);
 serif(2,1,c,serif_darkness,cap_jut); fi
proofpcbb(2, 1.5cap_jut+0.5thick, 0);
penlabels(0,1,2,3,4,5,6,7,8,9,10); labels(0’); endchar;
```

S The letter S. Pacioli's instructions are unclear. I have used a modified version of Hoenig's modification of the Knuthian S, which appears to match Pacioli's drawing.

```

cmchar "The Knuthian S";
beginchar("S", 2u#+1/2 side# if not monospace:+thick# fi, cap_height#, 0);
adjust_fit(0,0); pickup tiny.nib;
pair durer[]; % points to contain corner coords of Durer square
ypart durer3=ypart durer4=0; xpart durer4-xpart durer1=side;
xpart durer1=u-1.5thick;
durer1=(xpart durer3, h); durer2=(xpart durer 4, ypart durer1);
numeric slope, diag_angle; diag_angle=angle(durer2-durer3);
slope=-if not monospace: 3/4 else: 1/2 fi h/(xpart durer4-xpart durer3);
pos0(thick,diag_angle); x0=.52[xpart durer1, xpart durer2];
y0=h/2;
lft x3l=u if not monospace:+2/3thick fi; rt x3r=lft x3l+5/6thick;
pos2(thin,-diag_angle); top y2l=h;
if x0-x3r>thick: x2l=x0 if monospace:+.4thin fi; else:
 x2l=x0r; slope:=-1/2; fi
ellipse_set(2l,3l,4l,0l); ellipse_set(2r,3r,4r,0r);
pos7(thin,-diag_angle); x7r=x0; bot y7r=0; rt x6r=w-if monospace:.75 fi u;
 lft x6l=rt x6r-thick;
ellipse_set(7l,6l,5l,0l); ellipse_set(7r,6r,5r,0r);
pos1(max(currentbreadth+eps,5/6 thick),-90);
top y1l=h-thin; rt x1=x0+2thick+o;
pos8(1.5thick,-80);lft x8r=u; bot y8r=medium;
if not monospace:
 filldraw z1l..z2l{left}..{down}z3l..z4l--z5l..{down}z6l..{left}z7l..z8l
 --z8r..{right}z7r..{up}z6r..z5r--z4r..{up}z3r..{right}z2r..z1r--cycle;
 else: draw z1l..z2l{left}..{down}z3l..z4l--z5l..{down}z6l..{left}z7l..z8l;
 draw z8r..{right}z7r..{up}z6r..z5r--z4r..{up}z3r..{right}z2r..z1r;
 fi
proofpcbb(0, 0.5side, 0.5side);
penlabels(0,1,2,3,4,5,6,7,8); endchar;

```

T The letter T.

```

cmchar "Pacioli's letter T";
beginchar("T",2u#+side#-2medium#, cap_height#,0);
adjust_fit(0,0); pickup tiny.nib;
pos1(thick,0); pos2(thick,0); .5[x1r,x1l]=w/2; .5[x2r,x2l]=w/2;
top y1=h; bot y2=0; filldraw stroke z1e--z2e; % central stem
% prepare for the top horizontal bar
numeric top_bar; top_bar=0.5side-medium;  % half length of top bar
pos3(medium,90); pos4(dslab,0);
x1=x3; top y3r=h; rt x4l=x1+top_bar;
  bot y4=vround(h-if monospace: 2 fi vstretch*(medium+cap_jut));
arm(3,4,c,serif_darkness,0);
```

```
pos5(dslab,180);
lft x5r=x1-top_bar;
  bot y5= bot y4;
arm(3,5,d,serif_darkness,0);
if serifs:
 serif(2,1,a,serif_darkness,-cap_jut);serif(2,1,b,serif_darkness,cap_jut);
 fi
proofpcbb(2, 0.5side, 0);
penlabels(1,2,3,4); endchar;

```

U The letter U. Pacioli doesn't give a U, so I have used Hoenig's U, suitably modified for leg and serif dimensions.

```

cmchar "Pacioli's letter U";
beginchar("U", u#+2cap_jut#+.8side#, cap_height#,0);
adjust_fit(cap_serif_fit#, cap_serif_fit#);
pickup tiny.nib; pos1(thick,0); pos2(thick,0); pos2'(thick,180);
pos3(medium,-90); pos4(medium,0); pos5(medium,0);
x1=x2; x3=.5[x1,x5]; x4=x5; x1l=w-x5r;
top y1=top y5=h; y2=y4=1/3h; bot y3r=0; z2'=z2;
lft x1l=hround .5u+cap_jut;
filldraw stroke z1e--z2e; % left stem
filldraw stroke pulled_super_arc.e(2',3,eta)
  & pulled_super_arc.e(3,4,eta)&z4e--z5e;
if serifs: serif(1,2,a,serif_darkness,-cap_jut);
  serif(1,2,b,serif_darkness,cap_jut);
  serif(5,4,c,serif_darkness,-cap_jut); serif(5,4,d,serif_darkness,cap_jut); fi
penlabels(1,2,3,4,5); endchar;

```

V The letter V.

```

cmchar "Pacioli's letter V";
beginchar("V",u#+side#, cap_height#, 0);
adjust_fit(cap_serif_fit#, cap_serif_fit#); pickup tiny.nib;
pcshiftx := 0.5u;
z10 = (pcshiftx+0.5side,0);
numeric alpha; alpha=((0.5h-cap_jut)++(h))/(h);
pos1(alpha*thick,0); pos2(alpha*thick,0);
pos3(alpha*medium,0); pos4(alpha*medium,0);
top y1=top y4=h; lft x1l=pcshiftx+cap_jut; x4r=x1l+side-2cap_jut;
lft x2l=hround(pcshiftx+0.5side); x3r=x2l; bot y2=bot y3=0;
z0=whatever[z1r,z2r]=whatever[z3l,z4l]; % notch point
filldraw z0--diag_end(0,4l,1,1,4r,3r)--z2l
 --diag_end(2l,1l,1,1,1r,0)--cycle;
if serifs: bracket:=cap_jut_factor*bracket; cap_jut:=cap_jut_factor*cap_jut;
 pickup tiny.nib; numeric jut_factor;
 if monospace: jut_factor=1; else: jut_factor=3/4; fi
 serif(1,2,a,.8serif_darkness,-Lrad);
 serif(4,3,d,.8serif_darkness,lrad);
 bracket:=jut_factor*bracket;
 serif(1,2,b,7/8 serif_darkness, jut_factor*cap_jut);
 serif(4,3,c,7/8 serif_darkness,-jut_factor*cap_jut); fi
proofpcbb(10, 0.5side, 0);
penlabels(0,1,2,3,4); endchar;

```

W The letter W. Again, Pacioli misses this out. I use a modified version of Hoenig's W.

```

cmchar "Pacioli's letter W";
numeric alpha;
 alpha=((cap_height#+cap_jut#)++(.4side#+cap_jut_factor*cap_jut#))/
            (cap_height#+cap_jut#);
beginchar("W", 2cap_jut_factor*cap_jut#+1.6side#-alpha*thick#, cap_height#, 0);
adjust_fit(.5cap_serif_fit#, cap_serif_fit#); pickup tiny.nib;
mid_corr=if monospace: -apex_corr else: 1/3jut fi;
pos1(alpha*thick,0); pos2(alpha*thick,0);
pos3(alpha*medium,0); pos4(alpha*medium,0);
pos5(alpha*thick,0); pos6(alpha*thick,0);
pos7(alpha*medium,0); pos8(alpha*medium,0);
numeric V_width; V_width=2cap_jut_factor*cap_jut+4/5side;
if not monospace:
 top y1=top y4=h; lft x1l=cap_jut_factor*cap_jut; x4r=V_width-x1l;
 lft x2l=hround V_width/2; x3r=x2l; bot y2=bot y3=0;
 pair shift_amt; ypart shift_amt=0;
 xpart shift_amt=V_width-2cap_jut_factor*cap_jut-alpha*(thick);
 z0=whatever[z1r,z2r]=whatever[z3l,z4l]; % notch point
 path p; p=z0--diag_end(0,4l,1,1,4r,3r)--z2l
  --diag_end(2l,1l,1,1,1r,0)--cycle;
 filldraw p; filldraw (p shifted shift_amt); % the 2 V's
 forsuffixes t=1,2,3,4: z[t+4]=z.t shifted shift_amt; endfor
else: top y1=h; lft x1l=cap_jut_factor*cap_jut; bot y2=bot y3=0;
 x2l+apex_corr=x3l;
 x8=w-x1; y8=y1; x2-x1=x4-x3=x6-x5=x8-x7;
 x6l+apex_corr=x7l; y6=y7=y2;
 y4:=vround .6h; y5=y4;
 x5=x4;
 y4l:=y4r:=y4; % ??
 z0 =whatever[z1r,z2r];
 z0'=whatever[z7l,z8l]; y0=y0'=h/9;
 z45=whatever[z3r,z4r]=whatever[z5l,z6l];
 filldraw z0--diag_end(0,4l,1,1,4r,3r)--z2l--diag_end(2l,1l,1,1,1r,0)--cycle;
 filldraw z0'--diag_end(0',8l,1,1,8r,7r)--z7l--diag_end(7l,5l,1,1,5r,0')
  --cycle; fi
if serifs: bracket:=cap_jut_factor*bracket; cap_jut:=cap_jut_factor*cap_jut;
 pickup tiny.nib; numeric jut_factor;
 if monospace: jut_factor=1.8; else: jut_factor=3/4; fi
 serif(1,2,a,.8serif_darkness,-cap_jut);
 serif(8,7,f,.8serif_darkness,cap_jut);
 if not monospace: serif(5,6,d,.8serif_darkness,cap_jut);
  serif(5,6,c,.8serif_darkness,-cap_jut); fi
 bracket:=jut_factor*bracket;
 serif(1,2,b,7/8 serif_darkness, jut_factor*cap_jut);
 serif(8,7,e,7/8 serif_darkness,-jut_factor*cap_jut);
 fi
penlabels(0,0',1,2,3,4,5,6,7,8,45); endchar;

```

`X` The letter X.

```

cmchar "Pacioli's letter X";
beginchar("X", u#+side#, cap_height#, 0);
adjust_fit(cap_serif_fit#, cap_serif_fit#);
pcshiftx := 0.5u;
z10=(pcshiftx+0.5side, 0.5side); % center of the square
numeric alpha; alpha=(h++(h-2thick))/h;
penpos1(alpha*thick,0); penpos4(alpha*thick,0);
penpos2(alpha*medium,0); penpos3(alpha*medium,0);
lft x3l = pcshiftx+thick; y3=0; y2=side;
0.5[z3l,z2l]=z10;
y4 = 0; y1 = side;
lft x1l = pcshiftx+1.5thick;
0.5[z4l,z1l]=z10;
filldraw z1l--z1r--z4r--z4l--cycle; % draw the thick limb
filldraw z2l--z2r--z3r--z3l--cycle; % draw the other limb
if serifs: bracket:=cap_jut_factor*bracket; cap_jut:=cap_jut_factor*cap_jut;
 pickup tiny.nib; numeric jut_factor;
 if monospace: cap_jut:=.5cap_jut; jut_factor=1; else: jut_factor=3/4; fi
 serif(1,4,a,.8serif_darkness,-Lrad);
 serif(4,1,d,.8serif_darkness,cap_jut);
 serif(2,3,f,.8serif_darkness,cap_jut);
 serif(3,2,g,.8serif_darkness,-lrad);
 bracket:=jut_factor*bracket;
 serif(1,4,b,7/8 serif_darkness,.75cap_jut);
 serif(4,1,c,7/8 serif_darkness,-.75cap_jut);
 serif(2,3,e,7/8 serif_darkness,-.75cap_jut);
 serif(3,2,h,7/8 serif_darkness,.75cap_jut); fi
proofpcbb(10, 0.5side, 0.5side);
penlabels(1,2,3,4); endchar;

```

`Y` The letter Y.

```

cmchar "Pacioli's letter Y";
beginchar("Y", u#+side#, cap_height#, 0);
adjust_fit(cap_serif_fit#, cap_serif_fit#); pickup pencircle scaled 0pt;
pcshiftx := 0.5u;
z100 = (pcshiftx+0.5side,0.5side);  % center of the square
```

```
numeric alpha, long, short, hypot; long=h/2;
short=0.5side-thick;
 hypot=long++short; alpha=hypot/long;
penpos1(alpha*thick,0);
lft x1l=pcshiftx+thick;
top y1=h;
penpos4(alpha*medium,0); y4=y1; x4r=x1l+side-2thick;
penpos2(alpha*thick,0); penpos3(alpha*medium,0);
y2 = y3 = 0.5side;
penpos5(thick,0); penpos6(thick,0);
z2l = z3r = z100;
z6 = (pcshiftx+0.5side,0);
z5 = (x6,side);
z0=whatever[z1r, z2r]=whatever[z3l,z4l]; % notch point
z7=whatever[z6l,z5l] = whatever[z2l,z1l]; % left stem and V
z8=whatever[z6r,z5r] = whatever[z3r,z4r]; % right stem and V
filldraw z1r--z0--z4l--z4r--z8--z6r--z6l--z7--z1l--cycle;
if serifs: numeric jut_factor;
 serif(6,5,e,serif_darkness,-cap_jut); serif(6,5,f,serif_darkness,cap_jut);
 bracket:=cap_jut_factor*bracket; cap_jut:=cap_jut_factor*cap_jut;
 if monospace: cap_jut:=.5cap_jut; jut_factor=1; else: jut_factor=3/4; fi
 % top outer serifs
 serif(1,2,a,serif_darkness,-lrad);serif(4,3,d,serif_darkness,lrad);
 bracket:=jut_factor*bracket;
 if monospace: cap_jut:=1.5cap_jut; fi
 % top inner serifs
 serif(1,2,b,7/8 serif_darkness,.6cap_jut);
 serif(4,3,c,7/8 serif_darkness,-.6cap_jut);
 fi
proofpcbb(100, 0.5side, 0.5side);
penlabels(0,1,2,3,4,5,6); endchar; % End of "Y"

```

Z The letter Z. The last of the letters that Pacioli doesn't describe. Again, use a modifed Dürer Z.

```

cmchar "Pacioli's letter Z";
beginchar("Z", 2u#+.8side#, cap_height#,0);
adjust_fit(0, 0); pickup tiny.nib;
pcshiftx := u;
numeric alpha, long, short, hypot;
long=h; short=w-2u-thick; hypot=long++short;
alpha=hypot/long;
pos1(medium,90); pos2(medium,90); pos3(alpha*thick, 0);
pos4(alpha*thick, 0); pos5(medium,-90); pos6(medium,-90);
rt x3r=w-u; top y3=h; x4l=w-x3r; bot y4=0;
top y1r=h; lft x1=u; y2=y1; rt x2=x3;
bot y5r=bot y6r=0; lft x5=x4; rt x6=w-u;
filldraw stroke z1e--z2e; % top bar
filldraw stroke z3e--z4e; % diagonal
filldraw stroke z5e--z6e; % bottom bar
z100 = (0.5(x1+x3r),0.5side);  % center of square
if serifs: pickup crisp.nib; % prepare for arms and serifs
 pos7(medium, 90); pos8(dslab,180); z7=z2; lft x8r=u;
 bot y8=h-if monospace:2 fi vstretch*thick;
 arm(7,8,a,serif_darkness,0); % upper dropped serif
 pos9(medium,-90); z9=z5; pos10(dslab,0); rt x10r=w-u;
 top y10=if monospace: 2.5 fi vstretch*thick;
 arm(9,10,b,serif_darkness,0); fi
proofpcbb(100, 0.5side, 0.5side);
penlabels(1,2,3,4,5,6,7,8,9,10); endchar;


⟨/maj⟩
```

3.4 The punctuation file

This is essentially a copy of Knuth's Computer Modern Roman punctuation file punct.mf.

```
⟨*punct⟩
% CPCPUNCT.MF This file contains punctuation marks common to
%             Pacioli roman and italic styles.

```

! The ! character.

```

cmchar "Exclamation point";
beginchar("!",5u#+width_adj#,asc_height#,0);
italcorr asc_height#*slant-2u#-.5width_adj#+.5dot_size#;
adjust_fit(0,0);
pickup tiny.nib; pos3(dot_size,0); pos4(dot_size,90);
lft x3l=hround(.5w-.5dot_size); bot y4l=0; z3=z4; dot(3,4);  % dot
numeric bot_width;
bot_width=if hefty:max(hround .8dot_size,fine.breadth) else: hair fi;
pickup fine.nib; pos1(dot_size,0); pos2(bot_width,0);
x1=x2=x3; bot y2=.25[top y4r,x_height]+1;
if square_dots: top y1=h;
 filldraw stroke z1e--z2e;  % stem
else: top z0=(x1,h+o); y1+.5dot_size=h+o;
 filldraw z1r...z0...z1l---z2l--z2r---cycle; fi  % stem and bulb
penlabels(0,1,2,3,4); endchar;

```

’ The ’ character.

```

cmchar "Apostrophe";
beginchar("’",5u#,asc_height#,0);
italcorr asc_height#*slant+.5dot_size#-2u#;
adjust_fit(0,0);
```

```
x1-.5dot_size=hround(.5w-.5dot_size); y1+.5dot_size=h;
if monospace: comma(1,a,dot_size,.28u,vround 1.5comma_depth);  % large comma
else: comma(1,a,dot_size,.25u,comma_depth); fi  % comma with increased jut
penlabels(1); endchar;

```

(The (character.

```

cmchar "Left parenthesis";
beginchar("(",7u# if monospace: -u# fi,body_height#,paren_depth#);
italcorr body_height#*slant-.5u#;
adjust_fit(0,0); pickup fine.nib;
pos1(vair,0); pos2(.75[hair,stem],0); pos3(vair,0);
rt x1r=rt x3r=hround(w-u); lft x2l=hround(x1-4u if monospace: +4/3u fi);
top y1=h; y2=.5[y1,y3]=math_axis;
filldraw stroke z1e{3(x2e-x1e),y2-y1}...z2e
 ...{3(x3e-x2e),y3-y2}z3e;  % arc
penlabels(1,2,3); endchar;

```

) The) character.

```

cmchar "Right parenthesis";
beginchar(")",7u# if monospace: -u# fi,body_height#,paren_depth#);
italcorr math_axis#*slant-.5u#;
adjust_fit(0,0); pickup fine.nib;
pos1(vair,0); pos2(.75[hair,stem],0); pos3(vair,0);
lft x1l=lft x3l=hround u; rt x2r=hround(x1+4u if monospace: -4/3u fi);
top y1=h; y2=.5[y1,y3]=math_axis;
filldraw stroke z1e{3(x2e-x1e),y2-y1}...z2e
 ...{3(x3e-x2e),y3-y2}z3e;  % arc
penlabels(1,2,3); endchar;

```

, The , character.

```

cmchar "Comma";
numeric dot_diam#; dot_diam#=if monospace: 5/4 fi\\ dot_size#;
define_whole_blacker_pixels(dot_diam);
beginchar(",",5u#,dot_diam#,comma_depth#);
adjust_fit(0,0);
x1-.5dot_diam=hround(.5w-.5dot_diam); y1-.5dot_diam=0;
comma(1,a,dot_diam,.2u,comma_depth);  % dot and tail
penlabels(1); endchar;

```

. The . character.

```

cmchar "Period";
```

```
numeric dot_diam#; dot_diam#=if monospace: 5/4 fi\\ dot_size#;
define_whole_blacker_pixels(dot_diam);
beginchar(".",5u#,dot_diam#,0);
adjust_fit(0,0); pickup fine.nib;
pos1(dot_diam,0); pos2(dot_diam,90);
lft x1l=hround(.5w-.5dot_diam); bot y2l=0; z1=z2; dot(1,2);  % dot
penlabels(1,2); endchar;

```

/ The / character.

```

cmchar "Virgule (slash)";
beginchar("/",9u#,body_height#,paren_depth#);
italcorr body_height#*slant-.5u#;
adjust_fit(0,0); pickup rule.nib;
rt x1=hround(w-u)+eps; top y1=h+eps;
lft x2=hround u-eps; bot y2=-d-eps;
draw z1--z2;  % diagonal
penlabels(1,2); endchar;

```

: The : character.

```

cmchar "Colon";
numeric dot_diam#; dot_diam#=if monospace: 5/4 fi\\ dot_size#;
define_whole_blacker_pixels(dot_diam);
beginchar(":",5u#,x_height#,0);
italcorr x_height#*slant+.5dot_diam#-2u#;
adjust_fit(0,0); pickup fine.nib;
pos1(dot_diam,0); pos2(dot_diam,90);
lft x1l=hround(.5w-.5dot_diam); top y2r=h; z1=z2; dot(1,2);  % upper dot
pos3(dot_diam,0); pos4(dot_diam,90);
x3=x1; bot y4l=0; z3=z4; dot(3,4);  % lower dot
penlabels(1,2,3,4); endchar;

```

; The ; character.

```

cmchar "Semicolon";
numeric dot_diam#; dot_diam#=if monospace: 5/4 fi\\ dot_size#;
define_whole_blacker_pixels(dot_diam);
beginchar(";",5u#,x_height#,comma_depth#);
italcorr x_height#*slant+.5dot_diam#-2u#;
adjust_fit(0,0); pickup fine.nib;
pos1(dot_diam,0); pos2(dot_diam,90);
lft x1l=hround(.5w-.5dot_diam); top y2r=h; z1=z2; dot(1,2);  % upper dot
x3-.5dot_diam=hround(.5w-.5dot_diam); y3-.5dot_diam=0;
comma(3,a,dot_diam,.05u,comma_depth);  % lower dot and tail
penlabels(1,2,3); endchar;

```

[The [character.

```

cmchar "Left bracket";
numeric wd#; wd#=max(5u#,4.5u#+.5if hefty:stem# else:rule_thickness# fi);
beginchar("[",wd#,body_height#,paren_depth#);
italcorr body_height#*slant;
adjust_fit(0,0);
numeric top_thickness,side_thickness;
if hefty: top_thickness=vair;
 side_thickness=max(crisp.breadth,stem-2stem_corr);
else: top_thickness=side_thickness=rule_thickness; fi;
pickup crisp.nib; pos1(side_thickness,0); pos2(side_thickness,0);
top y1=h; bot y2=-d; lft x1l=lft x2l=hround(2.5u-.5side_thickness)-1-eps;
filldraw stroke z1e--z2e;  % stem
pos3(top_thickness,90); pos4(top_thickness,90);
pos5(top_thickness,90); pos6(top_thickness,90);
x3=x5=x1l; rt x4=rt x6=ceiling(w-.4u)+eps; y3r=y4r=y1; y5l=y6l=y2;
filldraw stroke z3e--z4e;  % upper bar
filldraw stroke z5e--z6e;  % lower bar
penlabels(1,2,3,4,5,6); endchar;

```

] The] character.

```

cmchar "Right bracket";
numeric wd#; wd#=max(5u#,4.5u#+.5if hefty:stem# else:rule_thickness# fi);
beginchar("]",wd#,body_height#,paren_depth#);
italcorr body_height#*slant-2u#+.5if hefty:stem# else:rule_thickness# fi;
adjust_fit(0,0);
numeric top_thickness,side_thickness;
if hefty: top_thickness=vair;
 side_thickness=max(crisp.breadth,stem-2stem_corr);
else: top_thickness=side_thickness=rule_thickness; fi;
pickup crisp.nib; pos1(side_thickness,0); pos2(side_thickness,0);
top y1=h; bot y2=-d; rt x1r=rt x2r=hround(w-2.5u+.5side_thickness)+1+eps;
filldraw stroke z1e--z2e;  % stem
pos3(top_thickness,90); pos4(top_thickness,90);
pos5(top_thickness,90); pos6(top_thickness,90);
x3=x5=x1r; lft x4=lft x6=floor .4u-eps; y3r=y4r=y1; y5l=y6l=y2;
filldraw stroke z3e--z4e;  % upper bar
filldraw stroke z5e--z6e;  % lower bar
penlabels(1,2,3,4,5,6); endchar;

```

‘ The ‘ character.

```

cmchar "Reverse apostrophe";
beginchar("‘",5u#,asc_height#,0);
italcorr asc_height#*slant+.5dot_size#-2u#;
```

```
adjust_fit(0,0);
x1-.5dot_size=hround(.5w-.5dot_size); y1+.5dot_size=h-comma_depth;
if monospace: ammoc(1,a,dot_size,.28u,vround 1.5comma_depth);  % large ammoc
else: ammoc(1,a,dot_size,.25u,comma_depth); fi  % normal ammoc
penlabels(1); endchar;


⟨/punct⟩
```

3.5 The ligatures and dashes file

This is essentially a copy of the Computer Modern comlig.mf file.

```
⟨*lig⟩
% CPCLIG.MF This file defines characters common to
%           Pacioli roman and italic text fonts
%           that appear only when |ligs>0|.

```

First the ligtables.

```

ligtable "‘": "‘"=:oct"134";
ligtable "’": "’"=:oct"042", "?" kern 2u#, "!" kern 2u#;

ligtable "-": "-"=:oct"173";
ligtable oct"173": "-"=:oct"174";

```

’’ The ” ligature.

```

cmchar "Closing quotes";
beginchar(oct"042",7u#+max(2u#,dot_size#),asc_height#,0);
italcorr asc_height#*slant+dot_size#-4.1u#;
adjust_fit(0,0);
x1-.5dot_size=hround .6u; y2+.5dot_size=h;
x2-x1=hround(1.5u+max(2u,dot_size)); y2=y1;
comma(1,a,dot_size,.25u,comma_depth); % left dot and tail
comma(2,b,dot_size,.25u,comma_depth); % right dot and tail
penlabels(1,2); endchar;

```

- The - character.

```

cmchar "Hyphen";
beginchar("-",6u#,x_height#,0);
italcorr .5x_height#*slant-.5u#;
adjust_fit(0,0);
numeric thickness; thickness=if hefty:bar else:.75[hair,stem] fi;
pickup crisp.nib; pos1(thickness,90); pos2(thickness,90);
top y1r=top y2r=vround(.5h+.5thickness); rt x2=hround(w-u)+eps;
```

```
if monospace: x2=w-x1 else: lft x1=hround .2u-eps fi;
filldraw stroke z1e--z2e;  % bar
penlabels(1,2); endchar;

```

`` The “ ligature.

```

cmchar "Opening quotes";
beginchar(oct"134",7u#+max(2u#,dot_size#),asc_height#,0);
italcorr asc_height#*slant-.1u#;
adjust_fit(0,0);
x2+.5dot_size=hround(w-.6u); y1+.5dot_size=h-comma_depth;
x2-x1=hround(1.5u+max(2u,dot_size)); y2=y1;
ammoc(1,a,dot_size,.25u,comma_depth); % left dot and tail
ammoc(2,b,dot_size,.25u,comma_depth); % right dot and tail
penlabels(1,2); endchar;

```

-- The – ligature. I think that the CM en-dash is too thin for the Pacioli font, so I have thickened it a little.

```

cmchar "En dash";
beginchar(oct"173",9u#,x_height#,0);
italcorr .61803x_height#*slant+.5u#;
adjust_fit(0,0);
%%pickup crisp.nib; pos1(vair,90); pos2(vair,90);
pickup crisp.nib; pos1(medium,90); pos2(medium,90);
top y1r=top y2r=vround(.61803h+.5vair); lft x1=-eps; rt x2=w+eps;
filldraw stroke z1e--z2e;  % bar
penlabels(1,2); endchar;

```

--- The — ligature. I think that the CM em-dash is too thin for the Pacioli font, so I have thickened it a little.

```

cmchar "Em dash";
beginchar(oct"174",18u#,x_height#,0);
italcorr .61803x_height#*slant+.5u#;
adjust_fit(letter_fit#,letter_fit#);
%%pickup crisp.nib; pos1(vair,90); pos2(vair,90);
pickup crisp.nib; pos1(medium,90); pos2(medium,90);
top y1r=top y2r=vround(.61803h+.5vair); lft x1=-eps; rt x2=w+eps;
filldraw stroke z1e--z2e;  % bar
penlabels(1,2); endchar;


⟨/lig⟩
```

3.6 Extra punctuation

This is an abbreviated copy of romanp.mf.

```
⟨*xp⟩
% CPCROMANP.MF This file contains ‘\&’ and ‘?’ in the so-called roman style.

```

& The & character.

```

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;

```

? The ? character. I have changed this slightly because of the redefinition of the pulled_super_arc macro.

```

cmchar "Question mark";
beginchar("?",8.5u#,asc_height#,0);
italcorr .8asc_height#*slant;
adjust_fit(0,0);
pickup tiny.nib; pos7(dot_size,0); pos8(dot_size,90);
lft x7l=hround(.5w-.25u-.5dot_size); bot y8l=0; z7=z8; dot(7,8);  % dot
numeric bot_width;
bot_width=if hefty:max(hround .8dot_size,fine.breadth) else: hair fi;
pickup fine.nib; pos2(vair,90); pos3(curve,0);
pos4(vair,-90); pos5(bot_width,0); pos6(bot_width,0);
x2=x4=x5=x6=x7; rt x3r=hround(w-u); bot y6=1+.25[top y8r,x_height];
top y2r=h+oo; y3=.75[y6,y2]; y4=.5[y6,y2]; y5=.1[y6,y2];
{{interim superness:=more_super;
%%filldraw stroke pulled_super_arc.e(2,3)(superpull) % macro changed in cpctitle.mf
filldraw stroke pulled_super_arc.e(2,3,eta)
 & subpath (0,1) of super_arc.e(3,4) .. z5e---z6e\\}};  % arc and stem
if serifs: pos1(hair,180); pos0(flare,180);
 lft x1r=hround u; y1=y3; bulb(2,1,0);  % bulb
else:
 pos1(Vround 5/7[vair,flare],110);
 lft x1r=hround u; top y1r=vround .9[y6,top y2r];
 filldraw stroke term.e(2,1,left,1,4);
fi  % terminal
penlabels(0,1,2,3,4,5,6,7,8); endchar;


⟨/xp⟩
```

4 The font definition files

The font definition file for the OT1 encoding.

```
⟨*fdot1⟩
\DeclareFontFamily{OT1}{cpc}{}
  \DeclareFontShape{OT1}{cpc}{m}{n}{ <-> cpcr10 }{}
  \DeclareFontShape{OT1}{cpc}{bx}{n}{ <-> sub * cpc/m/n }{}
  \DeclareFontShape{OT1}{cpc}{b}{n}{ <-> sub * cpc/m/n }{}
  \DeclareFontShape{OT1}{cpc}{m}{sl}{ <-> cpcsl10 }{}
  \DeclareFontShape{OT1}{cpc}{m}{it}{ <-> sub * cpc/m/sl }{}
⟨/fdot1⟩
```

The font definition file for the T1 encoding.

```
⟨*fdt1⟩
\DeclareFontFamily{T1}{cpc}{}
```

```
1575   \DeclareFontShape{T1}{cpc}{m}{n}{ <-> cpcr10 }{}
1576   \DeclareFontShape{T1}{cpc}{bx}{n}{ <-> sub * cpc/m/n }{}
1577   \DeclareFontShape{T1}{cpc}{b}{n}{ <-> sub * cpc/m/n }{}
1578   \DeclareFontShape{T1}{cpc}{m}{sl}{ <-> cpcsl10 }{}
1579   \DeclareFontShape{T1}{cpc}{m}{it}{ <-> sub * cpc/m/sl }{}
1580 ⟨/fdt1⟩
```

5 The pacioli package code

Announce the name and version of the package, which requires LaTeX 2$_\varepsilon$.

```
1581 ⟨*usc⟩
1582 \NeedsTeXFormat{LaTeX2e}
1583 \ProvidesPackage{pacioli}[1999/03/14 v1.0 package for Pacioli fonts]
```

\cpcfamily Selects the Pacioli font family using the OT1 encoding.

```
1584 \newcommand{\cpcfamily}{\usefont{OT1}{cpc}{m}{n}}
```

\textcpc Text command for the Pacioli font family.

```
1585 \DeclareTextFontCommand{\textcpc}{\cpcfamily}
```

The end of this package.

```
1586 ⟨/usc⟩
```

References

[GMS94] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The LaTeX Companion.* Addison-Wesley Publishing Company, 1994.

[Mor94] Stanley Morison. *Pacioli's Classic Roman Alphabet.* Dover, 1994.

Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.