作りましょう 0.11

パラメタ方式フォントファミリ
校とプリティプリントのソース

Tsukurimashou 0.11

Parametric Font Family
Proofs and pretty-printed source code

Matthew Skala
mskala@ansuz.sooke.bc.ca

2021年7月1日 July 1, 2021

Contents

Volume XVII

TsuIta

```
%
% Common code for TsuIta
% Copyright (C) 2012, 2013, 2021 Matthew Skala
%
[Standard copyright notice]

inclusion_lock(tsuitacommon);

---

rescale_slant:=0;

if (pagename="00")
or (pagename="01")
or (pagename="02")
or (pagename="03")
or (pagename="20")
or (pagename="21")
or (pagename="24")
or (pagename="25")
or (pagename="26")
or (pagename="27")
or (pagename="32")
or (pagename="4d")
or (pagename="f7")
or (pagename="ff")
or (pagename="1f1")
or (pagename="f17")
or (pagename="ff0")
or (pagename="ff1"):
  rescale_slant:=190;
fi;

tsu_punct_size:=tsu_punct_size*0.85;

boolean genji_rounded;
genji_rounded:=true;

---

vardef latin.lowa =
  latin.single_lowa;
enddef;

vardef latin.lowaogonek =
  push_pbox_toexpand("latin.lowaogonek");
```

```
  latin.lowa;

  y9=0.5[y10,y1];
  y10=latin_wide_desc_r;
  y11=0.2[y10,y1];

  x11-x9=(x1-x4)*((y1-y10)/(y3-y1));
  x10=0.4[x9,x11];
  x11=x1;

  replace_strokep(0)(oldp{dir 210}..z9..z10{right}..z11);
  replace_strokep(0)(insert_nodes(oldp)(length(oldp)-2.5));
  replace_strokeq(0)(oldq--(1.4,1.4)--(1.3,1.3)--(1.4,1.4)--(1,1));
  set_botip(0,length(get_strokep(0))-4,1);
  expand_pbox;
enddef;

vardef latin.lowe =
  push_pbox_toexpand("latin.lowe");
  y2=0.57[y5,y3];
  y3=latin_wide_xheight_r;
  y4=0.49[y5,y3];
  y5=latin_wide_low_r;
  y6=0.35[y5,y2];
  y0=0.7[y2,y3];

  (x2+x4)/2=500;
  (x2-x4)=0.86*(y3-y5);
  x3=0.49[x4,x2];
  x5=0.52[x4,x2];
  x6=1.04[x4,x2]-(if sharp_corners: 0 else: (mbrush_width/3) fi);
  x0=0.74[x3,x2];

  push_stroke(z0..z3{left}..z4{down}..z5{right}..z6,
    (1.6,1.6)--(1.6,1.6)--(1.4,1.4)--(1.6,1.6)--
      (1.6,1.6)--(1.6,1.6)--(1.6,1.6));
  z1=get_strokep(0) intersectionpoint ((z2+(-500,-120))--z2+(-10,0));
  replace_strokep(0)(insert_nodes((z1+(2,2)){curl 0}..
    tension 1.3..oldp)(0.5));
  replace_strokep(0)((point 0 of oldp){curl 0}..
    ((0,-20)+point 1 of oldp)..
    (subpath (2,infinity) of oldp));
  expand_pbox;
enddef;

vardef punct.atsign =
  push_pbox_toexpand("punct.atsign");
  begingroup
```

```
    save xsp,ysp;
    xsp:=sp;
    latin.lowa;
    set_boserif(0,6,whatever);
    set_botip(0,6,0);
    ysp:=sp;

    numeric x[],y[];
    x1-x2=x2-x3=y2-y1;
    x2=x4=500;
    y1=y3=0.49[y4,y2];
    y2=latin_wide_high_r;
    y4=latin_wide_low_r;

    transform shrinka;
    (0.5[llcorner get_strokep(0),urcorner get_strokep(0)])
      transformed shrinka=0.5[z3,z1];
    (0.5[lrcorner get_strokep(0),urcorner get_strokep(0)])
      transformed shrinka=0.71[z3,z1];
    (0.5[ulcorner get_strokep(0),urcorner get_strokep(0)])
      transformed shrinka=z2+(0.07,-1)*0.29*(x1-x3);
    sp:=xsp;
    tsu_xform(shrinka shifted (-10,0))(sp:=ysp);

    z5=point infinity of get_strokep(0);
    y6=ypart lrcorner get_strokep(0);
    x6=0.8[x2,x1];
    replace_strokep(0)((subpath (0,length(oldp)-1) of oldp)
       ..{curl 1}z5{curl 1}..z6..
       (subpath (0,3.85) of (z1..z2..z3..z4..cycle)));
    replace_strokep(0)(insert_nodes(oldp)((length oldp-4.5)));
    replace_strokeq(0)(oldq--(1.6,1.6)--(1.6,1.6)--(1.6,1.6)--(1.6,1.6)--
      (1.6,1.6)--(1.6,1.6)--(0,0));
  endgroup;
  expand_pbox;
enddef;

---

if do_italic_hook:

vardef latin.lowf =
  push_pbox_toexpand("latin.lowf");
  (x2-x1)=290;
  x5=x6=490=0.52[x1,x2];
  x3-x5=2*(y4-y5);
  x4=0.38[x5,x3];
  x6-x7=1.15*(x4-x5);
```

```
  x6-x8=1.15*(x3-x5);

  y1=y2=0.75[latin_wide_low_h,latin_wide_xheight_h];
  y5=0.52[latin_wide_xheight_h,y4];
  y3=0.73[latin_wide_xheight_h,y4];
  y4=latin_wide_high_r;
  y7=latin_wide_desc_r;
  y6-y7=1.10*(y4-y5);
  y8-y7=1.10*(y4-y3);

  push_stroke(z1--z2,(1.6,1.6)--(1.6,1.6));

  push_stroke(z3{curl 0.6}..z4{left}..
      {dir 268}z5{down}--z6{dir 268}..
      {left}z7..{curl 0.6}z8,
    (1.6,1.6)--(1.6,1.6)--(1.6,1.6)--(1.6,1.6)--(1.6,1.6)--(1.6,1.6));
  replace_strokep(0)(subpath (0.23,4.77) of oldp);
  expand_pbox;
enddef;

vardef latin.lowi =
  push_pbox_toexpand("latin.lowi");
  x2=x3=x4=450;

  y2=latin_wide_xheight_r;
  y3=latin_wide_low_r;
  y4=0.5[y2,latin_wide_high_r]+mbrush_width;

  push_stroke(z2--z3,(1.6,1.6)--(1.6,1.6));
  set_boserif(0,1,11);

  push_lcblob(fullcircle rotated 45 scaled (mbrush_width*2.7+15)
    shifted (z4 transformed tsu_rescale_xform)
    transformed inverse tsu_rescale_xform);
  expand_pbox;
enddef;

vardef latin.lowj =
  push_pbox_toexpand("latin.lowj");
  x3=x5=450;
  (x2-x5)=0.25*italic_hook_radius;

  y2=latin_wide_xheight_r;
  y3=latin_wide_low_v;
  y5=0.5[y2,latin_wide_high_r]+mbrush_width;

  z4=z3+(-210,-140);

```

```
  push_stroke((z2--(z3+(0,55)))..{curl 0.8}z4,
    (1.6,1.6)--(1.6,1.6)--(1.6,1.6));
  set_boserif(0,0,11);

  push_lcblob(fullcircle scaled (mbrush_width*2.7+15)
    shifted (z5 transformed tsu_rescale_xform)
    transformed inverse tsu_rescale_xform);
  expand_pbox;
enddef;

vardef latin.lowv =
  push_pbox_toexpand("latin.lowv");
  0.4[x1,x4]=x3=480;
  x1=x2;
  x5=0.8[x1,x4];

  y1=y5=latin_wide_xheight_v;
  y3=latin_wide_low_h;
  y2=0.333[y3,y1];
  y4=0.45[y3,y5];

  (x4-x1)=(y1-y3)*0.65;

  push_stroke(z1..z2..{dir -30}z3{dir 30}..z4..z5,
    (1.6,1.6)--(1.6,1.6)--(1.6,1.6)--(1.6,1.6)--(1.6,1.6));
  set_botip(0,2,0);
  set_boserif(0,0,11);

  expand_pbox;
enddef;

vardef latin.loww =
  push_pbox_toexpand("latin.loww");
  0.4[x1,x4]=x3;
  x1=x2;
  x5=x4;
  0.4[x5,x8]=x7;
  x5=x6;
  x9=0.8[x5,x8];
  0.5[x1,x8]=500;

  y1=y5=y9=latin_wide_xheight_v;
  y3=y7=latin_wide_low_h;
  y2=y6=0.333[y3,y1];
  y4=y8=0.45[y3,y5];

  (x4-x1)=(y1-y3)*0.55;
  (x8-x5)=(y1-y3)*0.60;
```

```
  push_stroke(z1..z2..{dir -30}z3{dir 30}..z4..z5,
    (1.6,1.6)—(1.6,1.6)—(1.6,1.6)—(1.6,1.6)—(1.6,1.6));
  set_botip(0,2,0);
  set_boserif(0,0,11);

  push_stroke(z5..z6..{dir -30}z7{dir 30}..z8..z9,
    (1.6,1.6)—(1.6,1.6)—(1.6,1.6)—(1.6,1.6)—(1.6,1.6));
  replace_strokep(0)(subpath (xpart
    (oldp intersectiontimes get_strokep(-1)),infinity) of oldp);

  expand_pbox;
enddef;

vardef latin.lowy =
  push_pbox_toexpand("latin.lowy");
  (x1+x3)/2=(x2+x4)/2=510;
  (x2+x3-x1-x4)=((y1-y2)*0.58)*2;
  (x3-x1)=(x2-x4)*0.93;
  x5=x4-0.1*(x2-x4);

  y1=y3=latin_wide_xheight_v;
  y2=y4;
  y5=0.5[y4,latin_wide_low_h]=latin_wide_desc_h;

  z6=0.66[z1,z3];

  push_stroke(z1—z2,(1.6,1.6)—(1.6,1.6));

  push_stroke(z6..
      (0.3[z3,z4])..tension 5..(0.6[z3,z4])..tension 0.8 and 3..{left}z5,
    (1.6,1.6)—(1.6,1.6)—(1.6,1.6)—(1.6,1.6));

  numeric xchgtime;
  xchgtime:=ypart (get_strokep(-1) intersectiontimes get_strokep(0));

  replace_strokep(-1)(z1—subpath (xchgtime,infinity) of get_strokep(0));
  replace_strokeq(-1)
    ((1.6,1.6)—subpath (xchgtime,infinity) of get_strokeq(0));

  replace_strokep(0)(subpath (0,xchgtime) of oldp);
  replace_strokeq(0)(subpath (0,xchgtime) of oldq);

  set_boserif(-1,0,11);
  set_botip(-1,1,1);
  set_bobrush(0,bralternate);

  if do_alternation:
```

```
    replace_strokep(-1)(oldp shifted (alternate_adjust*left/2));
    replace_strokep(0)(oldp shifted (alternate_adjust*right/2));
  fi;
  expand_pbox;
enddef;

vardef latin.lowz =
  push_pbox_toexpand("latin.lowz");
  y2=y4=latin_wide_xheight_h;
  y5=y7=latin_wide_low_h;
  y2-y1=y8-y7=0.15*(y4-y5);
  y4-y3=y6-y5=0.09*(y4-y5);

  x1=x5;
  x4=x8;
  (x1+x4)/2=500;
  (x4-x1)=(y4-y5)*0.92;
  x2=x6=0.25[x1,x4];
  x3=x7=0.75[x1,x4];

  push_stroke(z1..z2..z3..z4--z5..z6..z7..z8,
    (1.6,1.6)--(1.6,1.6)--(1.6,1.6)--(1.6,1.6)--
    (1.6,1.6)--(1.6,1.6)--(1.6,1.6)--(1.6,1.6));
  set_botip(0,4,0);
  set_botip(0,5,0);
  expand_pbox;
enddef;

---

def tsu_render_segment(expr i,p,q,b) =
  begingroup
    save start_time,end_time,niba,nibb,nibc,myp,pext,x,y;
    numeric start_time,end_time,x,y;
    path myp,niba,nibb,nibc,pext,glyph;

    myp:=p;

    niba:=fix_nib(obstackna.bosize[i]*tsu_brush_max[b]
                    *tsu_brush_shape[b],
                  obstackna.bosize[i]*tsu_brush_max[b]
                    *tsu_brush_shape[b],
                  0);

    if known obstacknaa.boserif[i][floor(ltime[0]+0.5)]:
      if obstacknaa.boserif[i][floor(ltime[0]+0.5)]=11:
        start_time:=xpart (myp intersectiontimes (fullcircle
          scaled (italic_hook_radius*(obstackna.bosize[i]/100)*2)
```

```
      shifted (point 0 of myp)));
    nibb:=tsu_brush_tip(0,myp,q,obstackna.bosize[i],s<1,
      t>(length obstackp[i])-1,b);
    niba:=nibb scaled tsu_brush_shape[b];
    nibc:=nibb scaled sqrt(sqrt(tsu_brush_shape[b]));
    z1=(point 0 of myp)-
      italic_hook_radius*(obstackna.bosize[i]/100)*dir(italic_hook_dir);
    z3=point start_time of myp;
    x2=0.5[xpart point 0 of myp,x1];
    y2=ypart point 0 of myp;
    pext:=z1{curl 0}..z2..{direction start_time of myp}z3;
    pen_stroke(tip(niba)(0) tip(nibc)(1) tip(nibb)(2))(pext)(glyph);
    glstk[ngls]:=regenerate(glyph);
    ngls:=ngls+1;
    myp:=subpath (start_time,infinity) of myp;
  fi;
fi;

if known obstacknaa.boserif[i][floor(ltime[length(myp)]+0.5)]:
  if obstacknaa.boserif[i][floor(ltime[length(myp)]+0.5)]=11:
    end_time:=(length myp)-xpart ((reverse myp)
      intersectiontimes (fullcircle
      scaled (italic_hook_radius*(obstackna.bosize[i]/100)*2)
      shifted (point infinity of myp)));
    nibb:=tsu_brush_tip(length(myp),myp,q,obstackna.bosize[i],s<1,
      t>(length obstackp[i])-1,b);
    niba:=nibb scaled tsu_brush_shape[b];
    nibc:=nibb scaled sqrt(sqrt(tsu_brush_shape[b]));
    z4=point end_time of myp;
    z6=(point infinity of myp)+
      italic_hook_radius*(obstackna.bosize[i]/100)*dir(italic_hook_dir);
    x5=0.5[xpart point infinity of myp,x6];
    y5=ypart point infinity of myp;
    pext:=z4{direction end_time of myp}..z5..{curl 0}z6;
    pen_stroke(tip(nibb)(0) tip(nibc)(1) tip(niba)(2))(pext)(glyph);
    glstk[ngls]:=regenerate(glyph);
    ngls:=ngls+1;
    myp:=subpath (0,end_time) of myp;
  fi;
fi;

default_nib:=fix_nib(obstackna.bosize[i]*tsu_brush_max[b],
                     obstackna.bosize[i]*tsu_brush_max[b]
                       *tsu_brush_shape[b],
                     tsu_brush_angle[b]);
path mytip[],glyph;
for l=0 step 1 until length(myp):
  mytip[l]:=tsu_brush_tip(l,myp,q,obstackna.bosize[i],s<1,
```

```
        t>(length obstackp[i])-1,b);
    endfor;
    pen_stroke(for l=0 step 1 until length(myp):
        if sharp_corners and known obstacknaa.botip[i][ltime[l]]:
          tip(obstacknaa.botip[i][ltime[l]])(l)
        else:
          tsu_brush_opt(mytip[l])(l)
        fi
      endfor)(myp)(glyph);
    glstk[ngls]:=regenerate(glyph);
    ngls:=ngls+1;
    for l=0 step 1 until length(myp):
      si:=floor (ltime[l]+0.5);
      if (abs(ltime[l]-si)<0.05) and known obstacknaa.boserif[i][si]:
        tsu_serif.choose(obstacknaa.boserif[i][si],
          point l of myp,direction l of myp,l,
          obstackna.bosize[i],tsu_brush_tip_size(l,q,b),b);
        write ("SERIF "&(decimal obstacknaa.boserif[i][si])&" "&
              (decimal xpart point l of myp)&","&
              (decimal ypart point l of myp)) to "proof.prf";
      fi;
    endfor;
  endgroup;
enddef;

def tsu_rescale_xform =
  begingroup
    save t,st,cp,xadj;
    transform t,st;
    numeric xadj;
    st:=tsu_slant_xform;
    t:=st;
    % check if rescaling is active
    if (rescale_from.left<>rescale_to.left)
    or (rescale_from.right<>rescale_to.right): begingroup
      save i,xa,xb,lf,rf,wf,lt,rt,wt;
      numeric i,xa,xb,lf,rf,wf,lt,rt,wt;
      transform t;
      % find the bounds of the paths
      if find_stroke(0)<=0:
        xa:=0.5[rescale_from.left,rescale_from.right];
        xb:=0.5[rescale_from.left,rescale_from.right];
      else:
        xa:=infinity;
        xb:=-infinity;
        for i=1 upto sp-1:
          if obstacktype[i]=otstroke:
            xadj:=0;
```

```
        if known obstacknaa.boserif[i][0]:
          if obstacknaa.boserif[i][0]=11:
            xadj:=-(2/3)*italic_hook_radius*xpart dir italic_hook_dir;
          fi;
        fi;
        if (xadj+xpart llcorner obstackp[i])<xa:
          xa:=xadj+xpart llcorner obstackp[i];
        fi;
        xadj:=0;
        if known obstacknaa.boserif[i][length(obstackp[i])]:
          if obstacknaa.boserif[i][length(obstackp[i])]=11:
            xadj:=(2/3)*italic_hook_radius*xpart dir italic_hook_dir;
          fi;
        fi;
        if (xadj+xpart lrcorner obstackp[i])>xb:
          xb:=xadj+xpart lrcorner obstackp[i];
        fi;
      fi;
    endfor;
  fi;
  % compute bearings and widths
  lf=xa-rescale_from.left;
  rf=rescale_from.right-xb;
  lf+rf+wf=rescale_from.right-rescale_from.left;
  lt+rt+wt=rescale_to.right-rescale_to.left;
  (lt,rt)=whatever[(0,0),(lf,rf)];
  wt=ypart (width_curve intersectionpoint
    ((wf,-infinity)--(wf,infinity)));
  % find transformation
  if wf>0:
    (rescale_from.right-rf,rescale_from.bottom) transformed t=
      (rescale_to.right-rt,rescale_to.bottom+rescale_skew);
    (rescale_from.left+lf,rescale_from.bottom) transformed t=
      (rescale_to.left+lt,rescale_to.bottom-rescale_skew);
    (rescale_from.left+lf,rescale_from.top) transformed t=
      (rescale_to.left+lt,rescale_to.top-rescale_skew);
  else:
    (rescale_from.left+lf,rescale_from.bottom) transformed t=
      (rescale_to.left+lt,rescale_to.bottom);
    (rescale_from.left+lf,rescale_from.top) transformed t=
      (rescale_to.left+lt,rescale_to.top);
    (rescale_from.left+lf+1,rescale_from.bottom) transformed t=
      (rescale_to.left+lt+1,rescale_to.bottom);
  fi;
  pair cp;
  transform st;
  cp:=((rescale_to.left+rescale_to.right)/2,rescale_to.bottom);
  cp transformed st=cp;
```

```
      cp+(100,0) transformed st=cp+(100,0);
      cp+(0,100) transformed st=cp+(rescale_slant/10,100);
      t:=t transformed st;
    endgroup; fi;
    t
  endgroup
enddef;

fi;
```

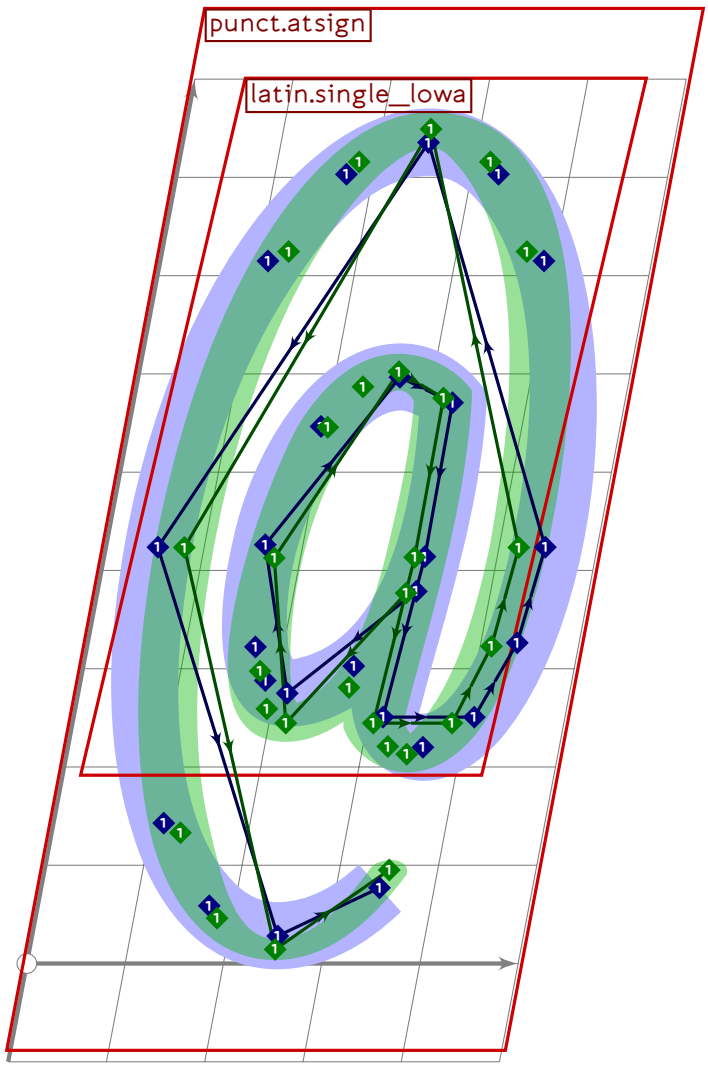
```
%
% TsuIta Atama
% Copyright (C) 2012, 2013 Matthew Skala
%
[Standard copyright notice]

% TSUITA ATAMA

input preintro.mp;

familyname:="TsuIta";
stylename:="Atama";

(0,4) transformed tsu_brush_xf.brletter = (4,0.77);
(1,1) transformed tsu_brush_xf.brletter = (1,0.73);
(4,0) transformed tsu_brush_xf.brletter = (0,0.77);

tsu_brush_min.brletter:=0.73;
tsu_brush_max.brletter:=0.77;

def tsu_brush_opt(expr n,l) =
  if rescale_slant>10: cut(n,rel 90)(l) fi enddef;
def sharp_corners = (rescale_slant>10) enddef;

input intro.mp;

let old_rescale_half = tsu_rescale_half;

def tsu_rescale_half =
  old_rescale_half;
  width_curve:=((-1,-1)--(100,100))..(480,330)..{right}(2000,440);
enddef;

let old_rescale_half_lc = tsu_rescale_half_lc;

def tsu_rescale_half_lc =
  old_rescale_half_lc;
  width_curve:=((-1,-1)--(100,100))..(610,330)..{right}(2000,420);
enddef;

include_late("tsuita-common.mp");

% overridden per-page later
rescale_slant:=190;
```

```
%
% TsuIta Soku
% Copyright (C) 2012, 2013 Matthew Skala
%
[Standard copyright notice]

% TSUITA SOKU

input preintro.mp;

familyname:="TsuIta";
stylename:="Soku";

mincho:=0.8;

(0,4) transformed tsu_brush_xf.brletter = (0.0,0.80);
(1,1) transformed tsu_brush_xf.brletter = (1.2,0.45);
(4,0) transformed tsu_brush_xf.brletter = (4.8,0.80);

tsu_brush_min.brletter:=0.45;
tsu_brush_max.brletter:=0.80;
tsu_brush_shape.brletter:=0.60;
tsu_brush_angle.brletter:=1;

tsu_brush_xf.bralternate:=identity xyscaled (1.2,0) shifted (0,0.48);
tsu_brush_min.bralternate:=0.48;
tsu_brush_max.bralternate:=0.48;
tsu_brush_shape.bralternate:=1;
tsu_brush_angle.bralternate:=0;

tsu_brush_xf.brpunct:=whatever_xf;
(0,4) transformed tsu_brush_xf.brpunct = (0,0.45);
(1,1) transformed tsu_brush_xf.brpunct = (1,0.10);
(4,0) transformed tsu_brush_xf.brpunct = (4,0.45);

tsu_brush_min.brpunct:=0.10;
tsu_brush_max.brpunct:=0.45;
tsu_brush_shape.brpunct:=1;
tsu_brush_angle.brpunct:=0;

input serif.mp;

tsu_do_serif[1]:=true;
tsu_do_serif[2]:=true;
tsu_do_serif[3]:=true;

```

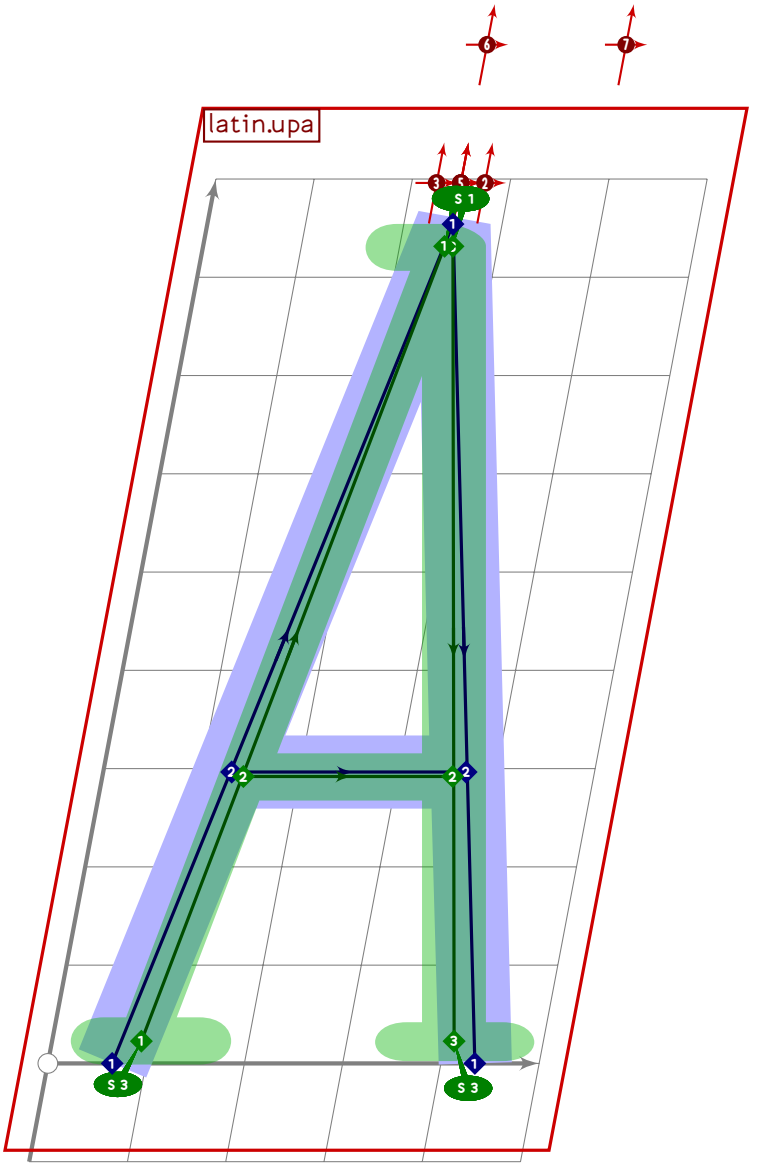
```
do_alternation:=true;

genji_outline:=true;

input intro.mp;

let old_render_segment = tsu_render_segment;

def tsu_render_segment(expr i,p,q,b) =
  begingroup
    save start_time,end_time;
    numeric start_time,end_time;
  endgroup;
  old_render_segment(i,p,q,b);
enddef;

let old_rescale_half = tsu_rescale_half;

def tsu_rescale_half =
  old_rescale_half;
  width_curve:=((-1,-1)--(100,100))..(560,310)..{right}(2000,400);
enddef;

let old_rescale_half_lc = tsu_rescale_half_lc;

def tsu_rescale_half_lc =
  old_rescale_half_lc;
  width_curve:=((-1,-1)--(100,100))..(610,310)..{right}(2000,400);
enddef;

do_italic_hook:=true;
italic_hook_radius:=170;
italic_hook_dir:=25;

include_late("tsuita-common.mp");

% overridden per-page later
rescale_slant:=190;
```

Additional Proofs

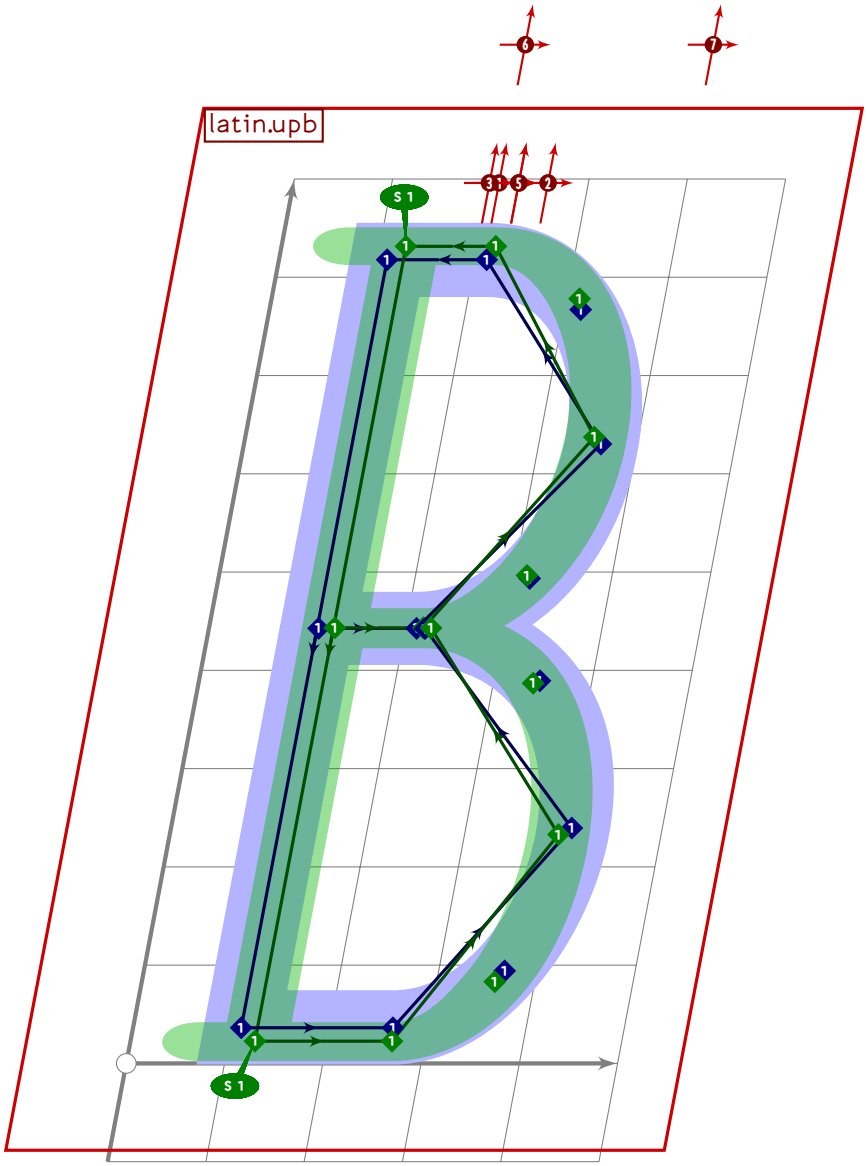
Most glyphs in TsuIta are drawn by code inherited from Tsukurimashou, and proof images for those glyphs are shown with the corresponding Tsukurimashou code. However, different parameters between Tsukurimashou and TsuIta, and overriding code in the tsuita-common.mp file above, cause many of the glyphs to change in appearance. The following pages give additional proof images for the uppercase and lowercase Latin alphabets, commercial at sign, and a-ogonek, which are the glyphs that differ most from the Tsukurimashou versions.



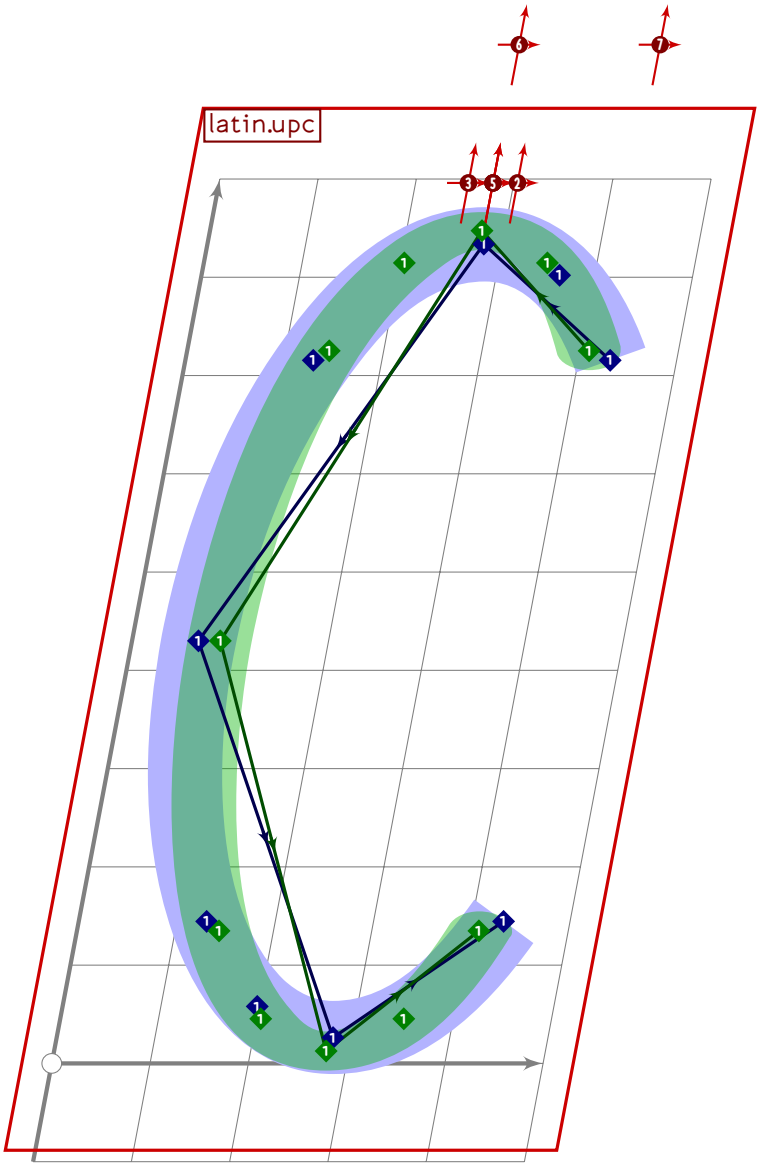
U+0041
tsuita.A



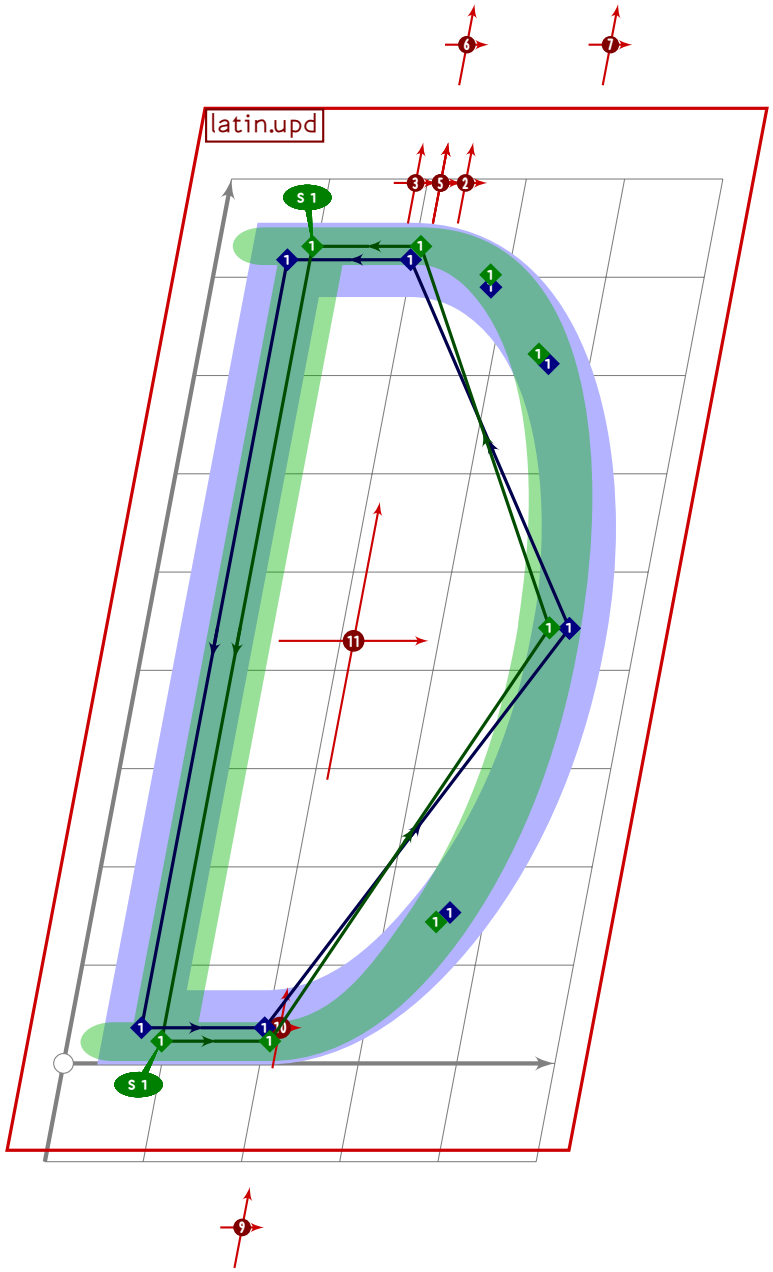
PROO



U+0043
tsuita.C



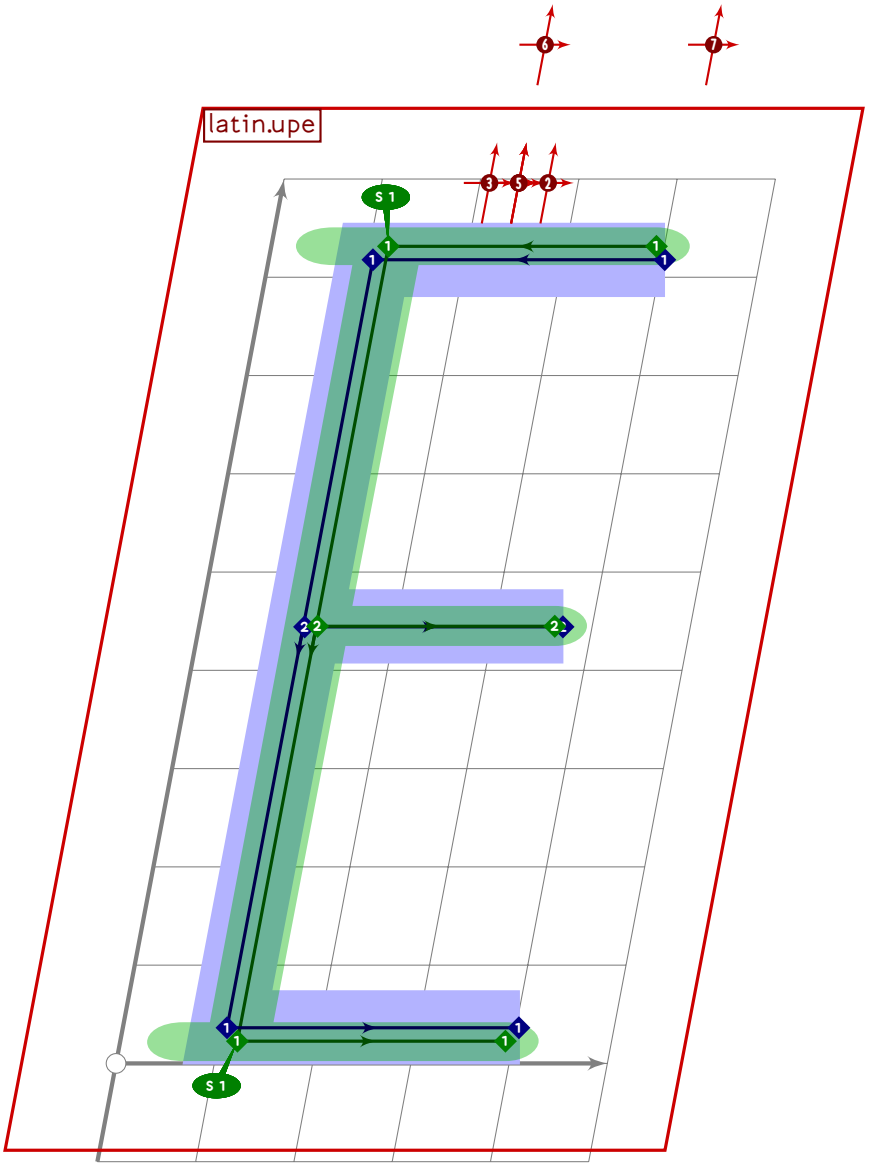
PROO

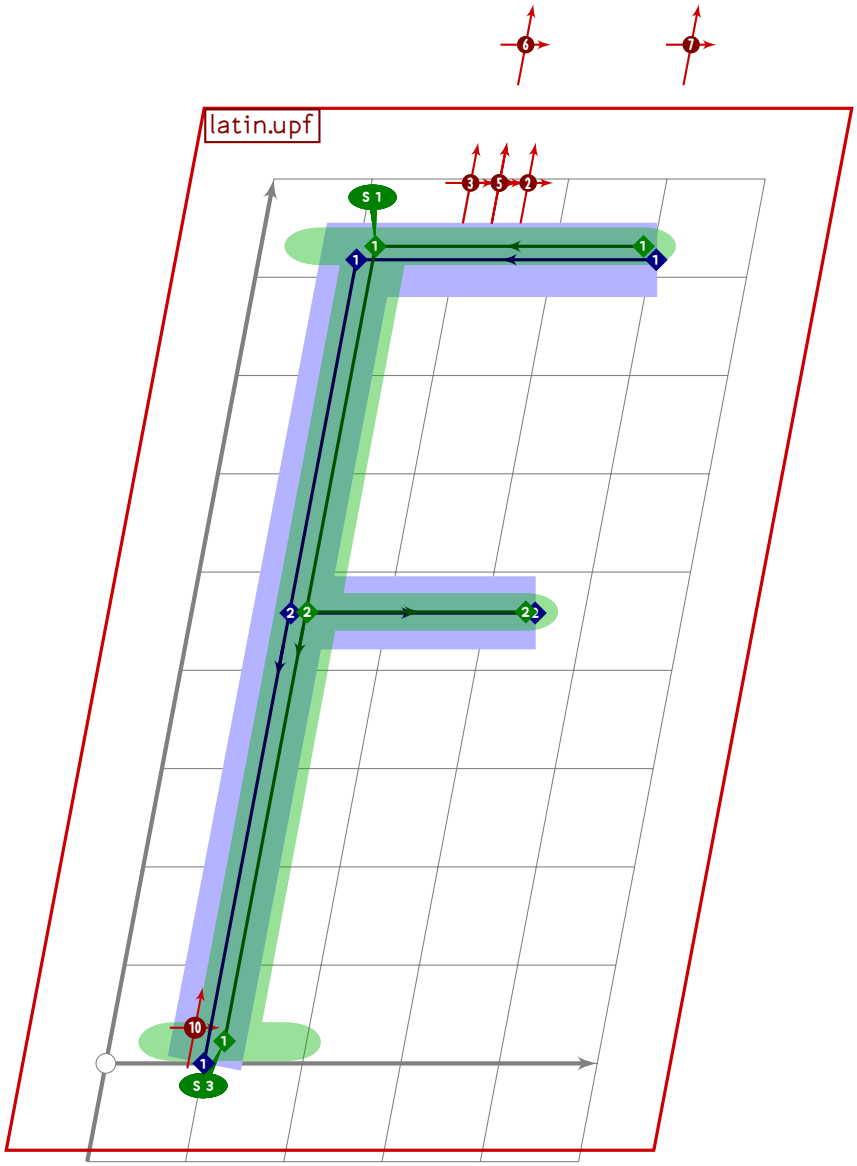


PROO

U+0045
tsuita.E

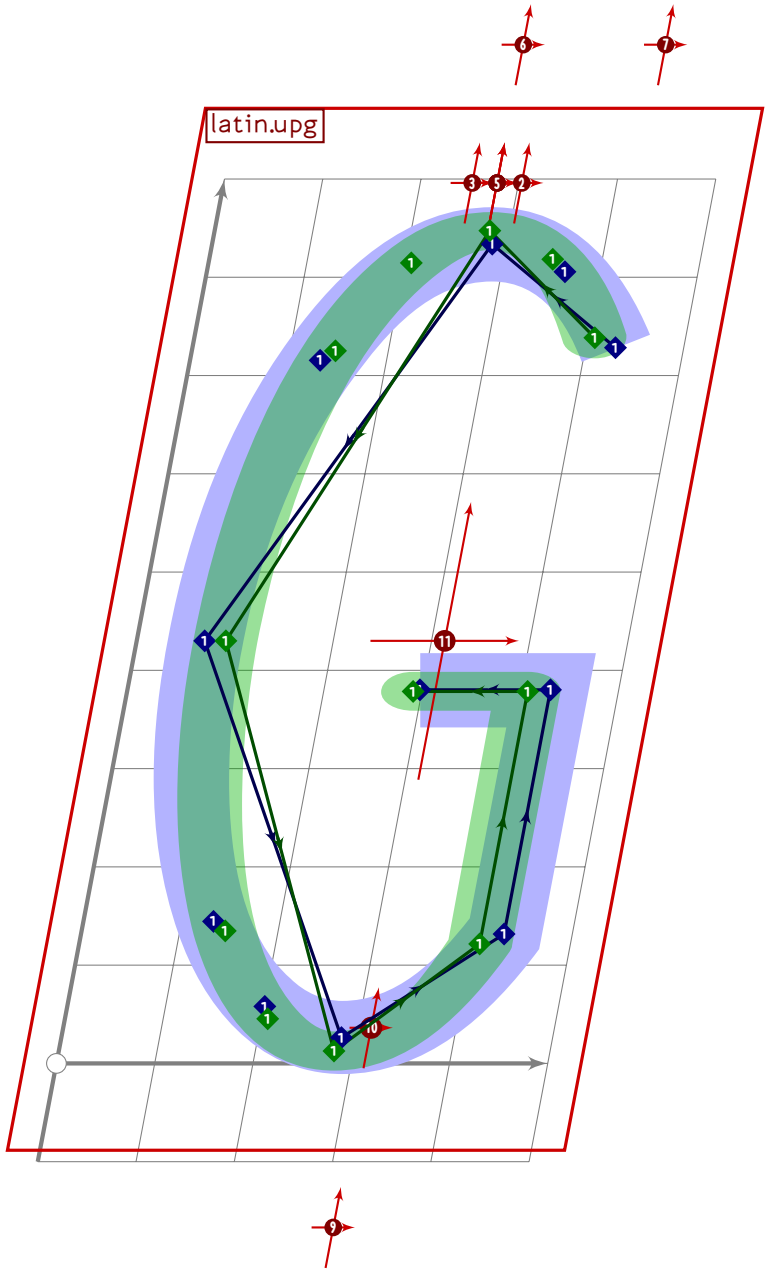
PROO





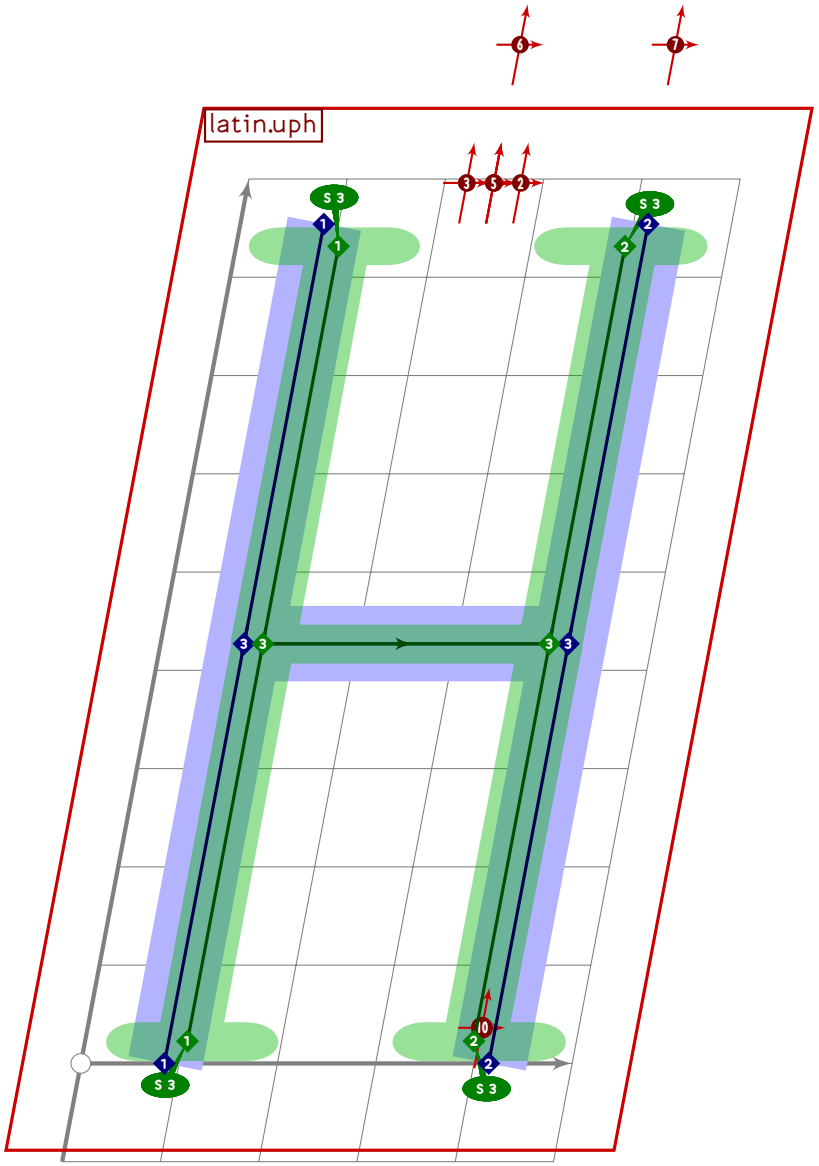
PROO

U+0047
tsuita.G



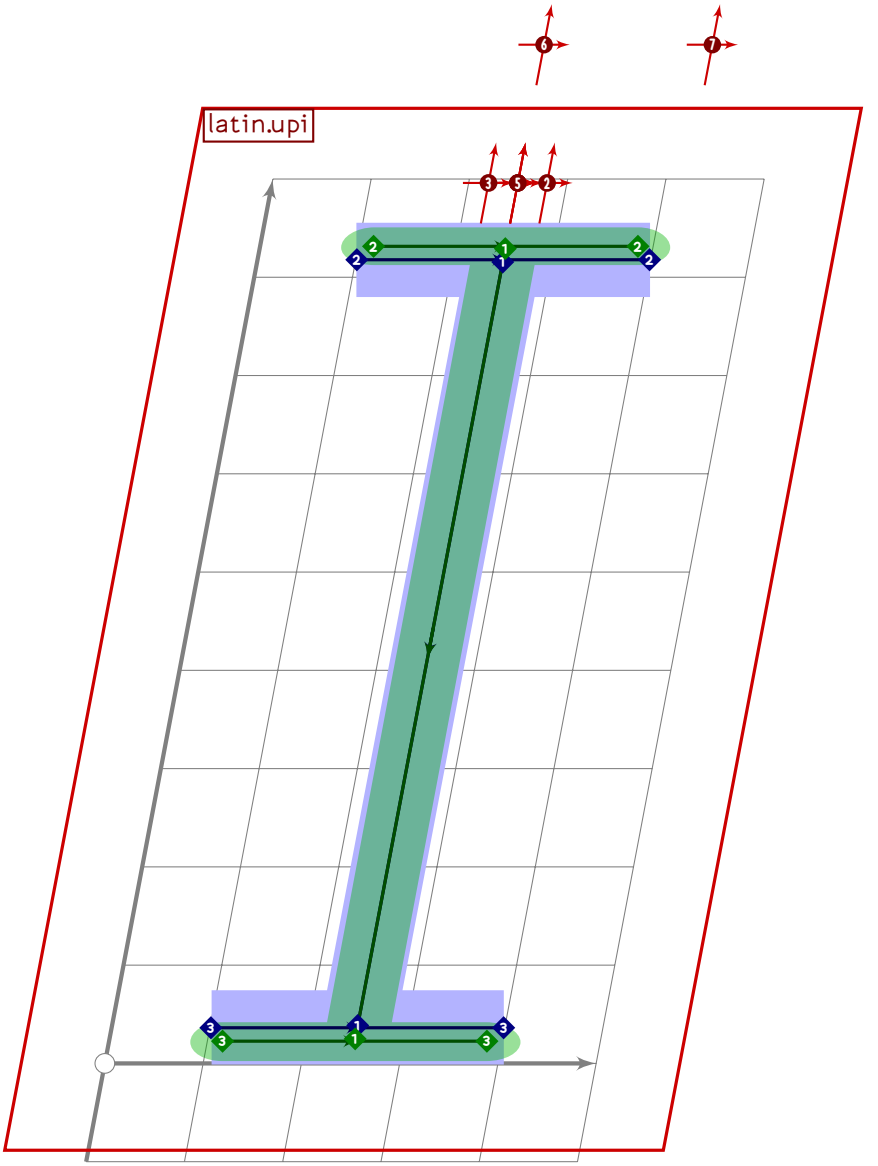
PROO

U+0048
tsuita.H

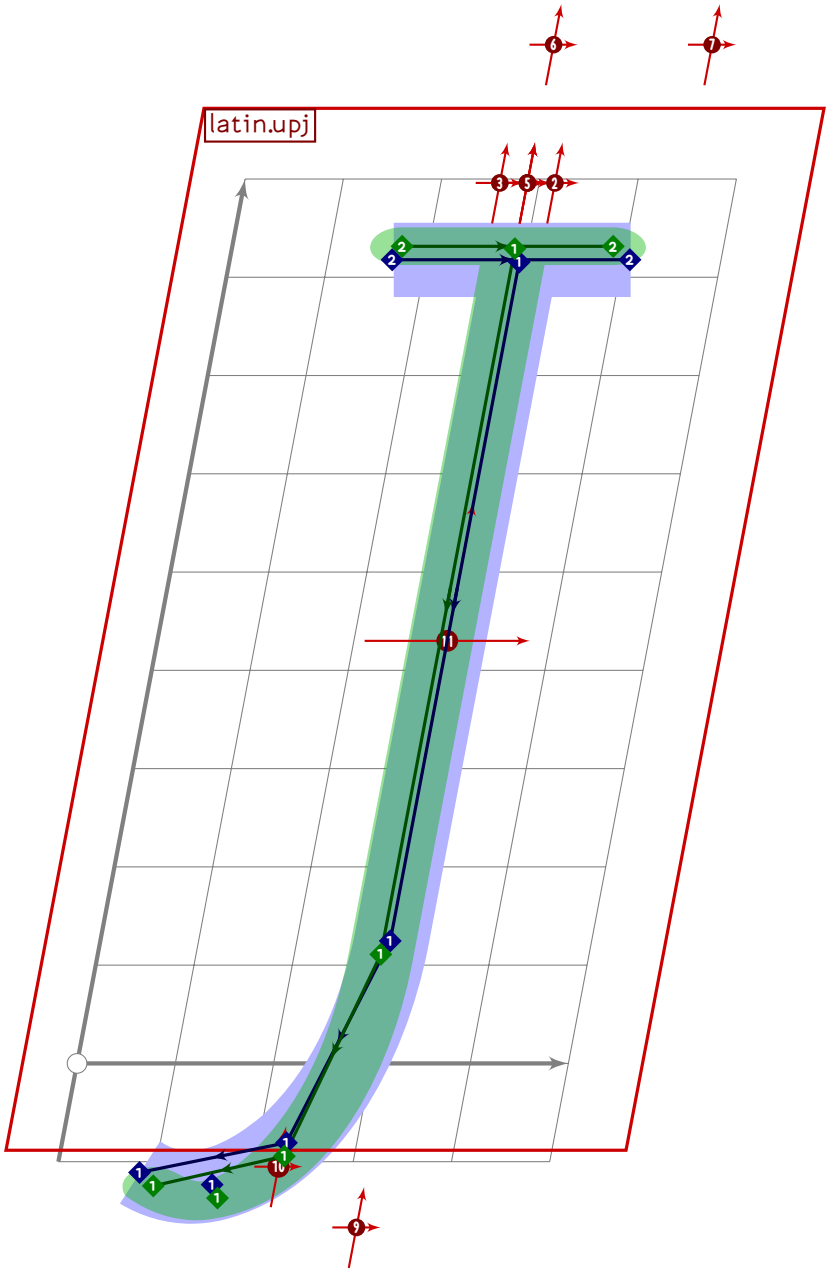


PROO

U+0049
tsuita.I

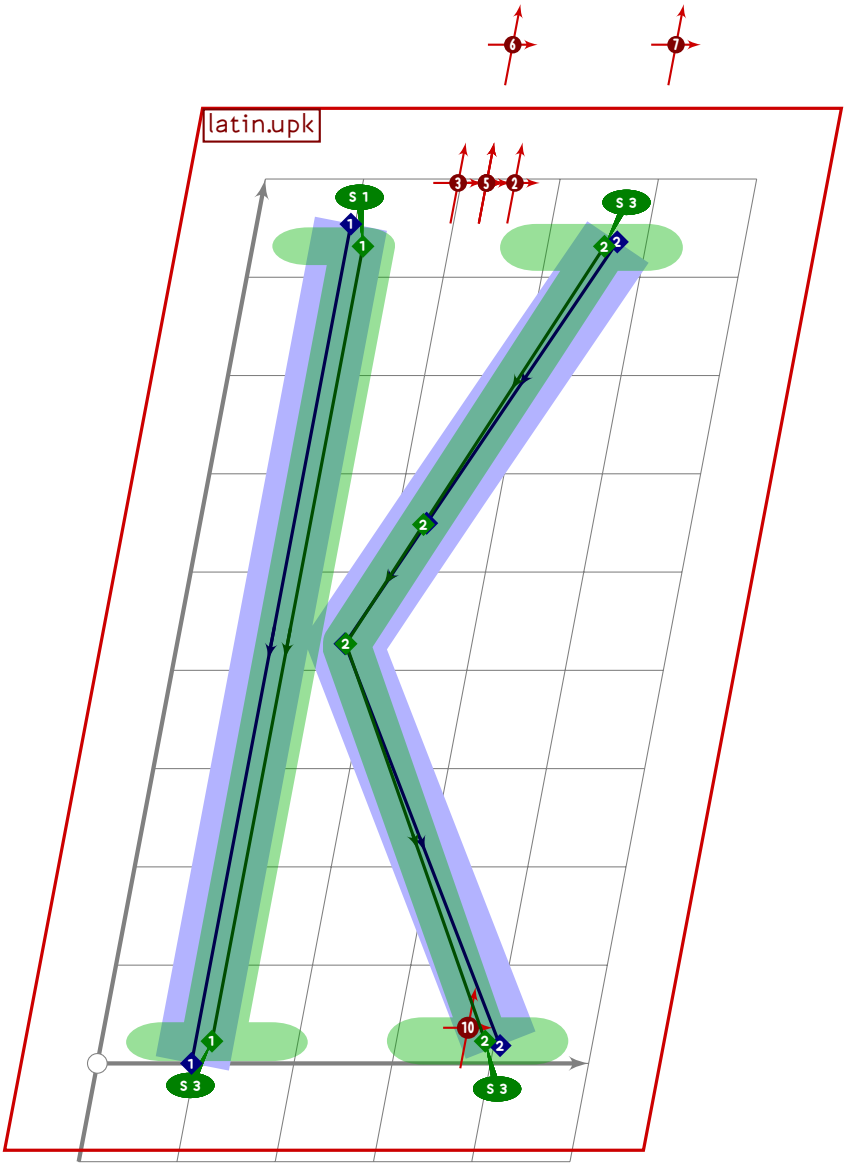


PROO

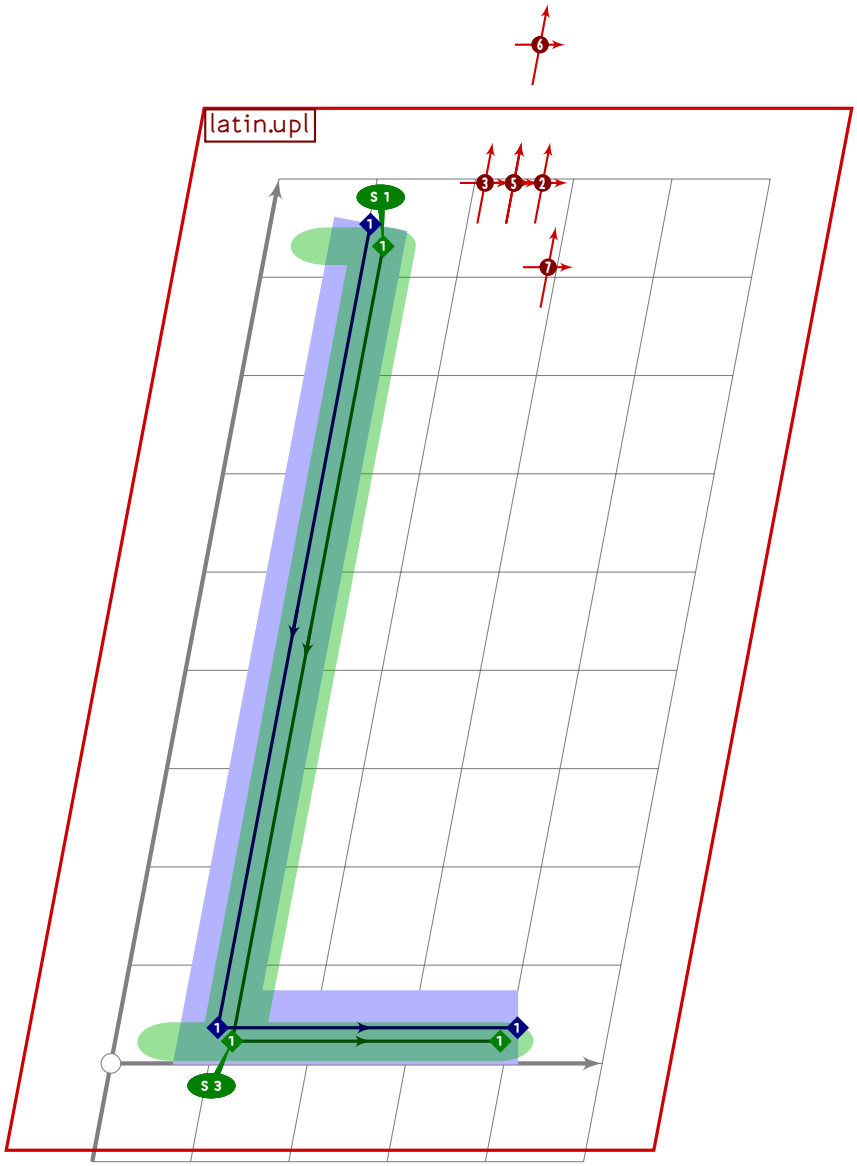


PROO

U+004B
tsuita.K

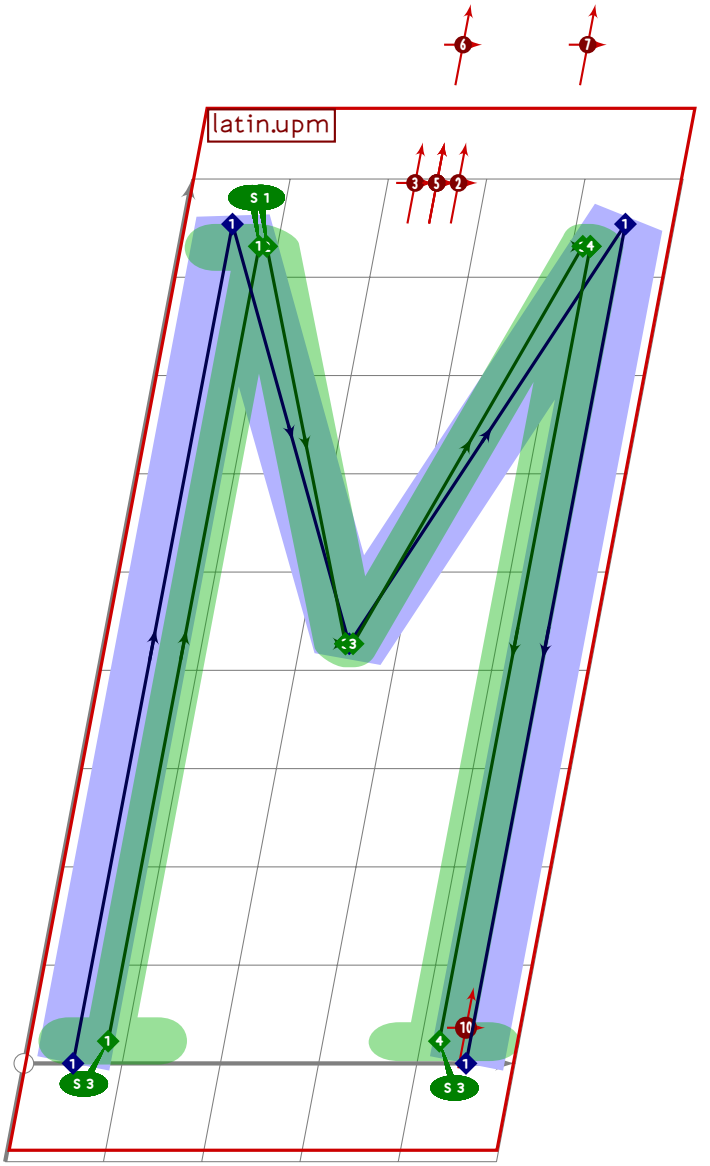


PROO



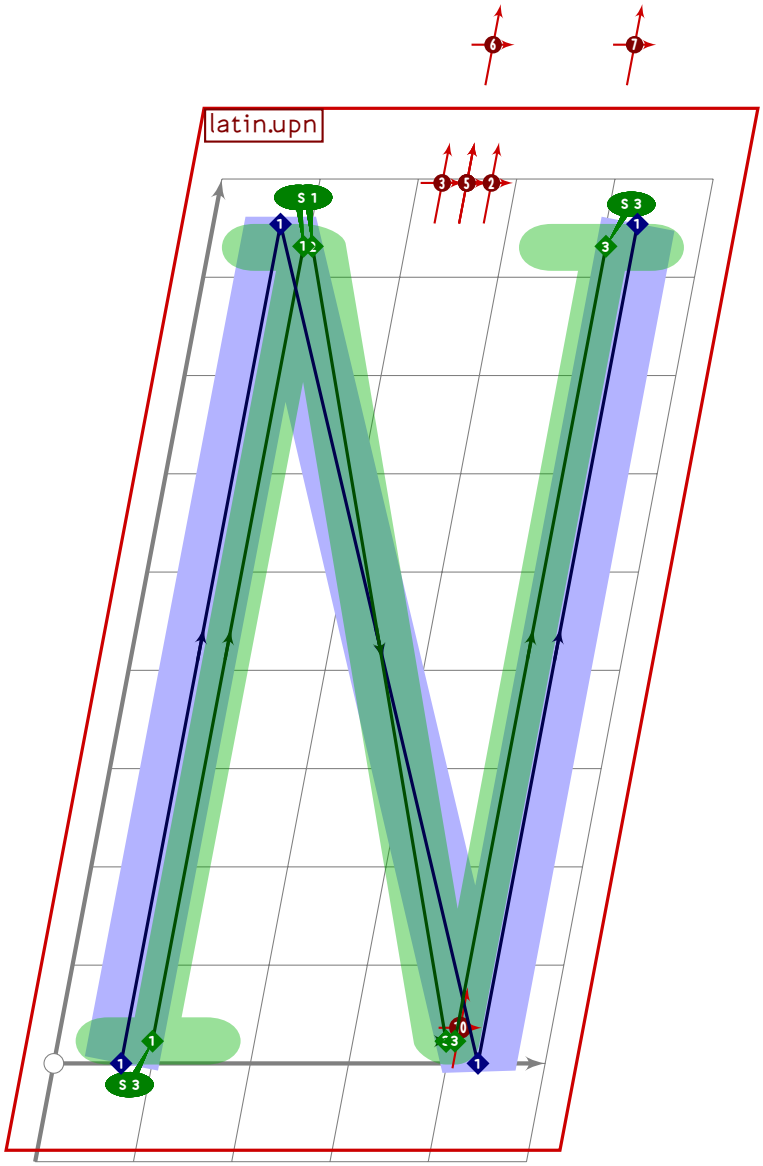
PROO

U+004D
tsuita.M



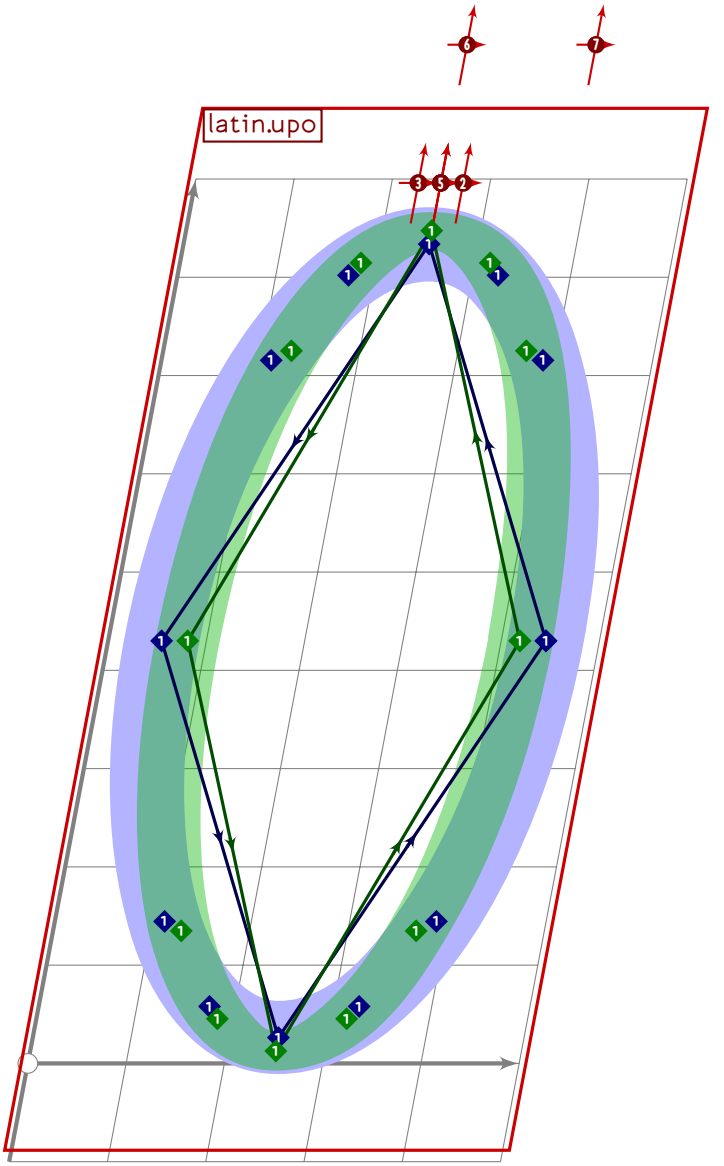
PROO

U+004E
tsuita.N

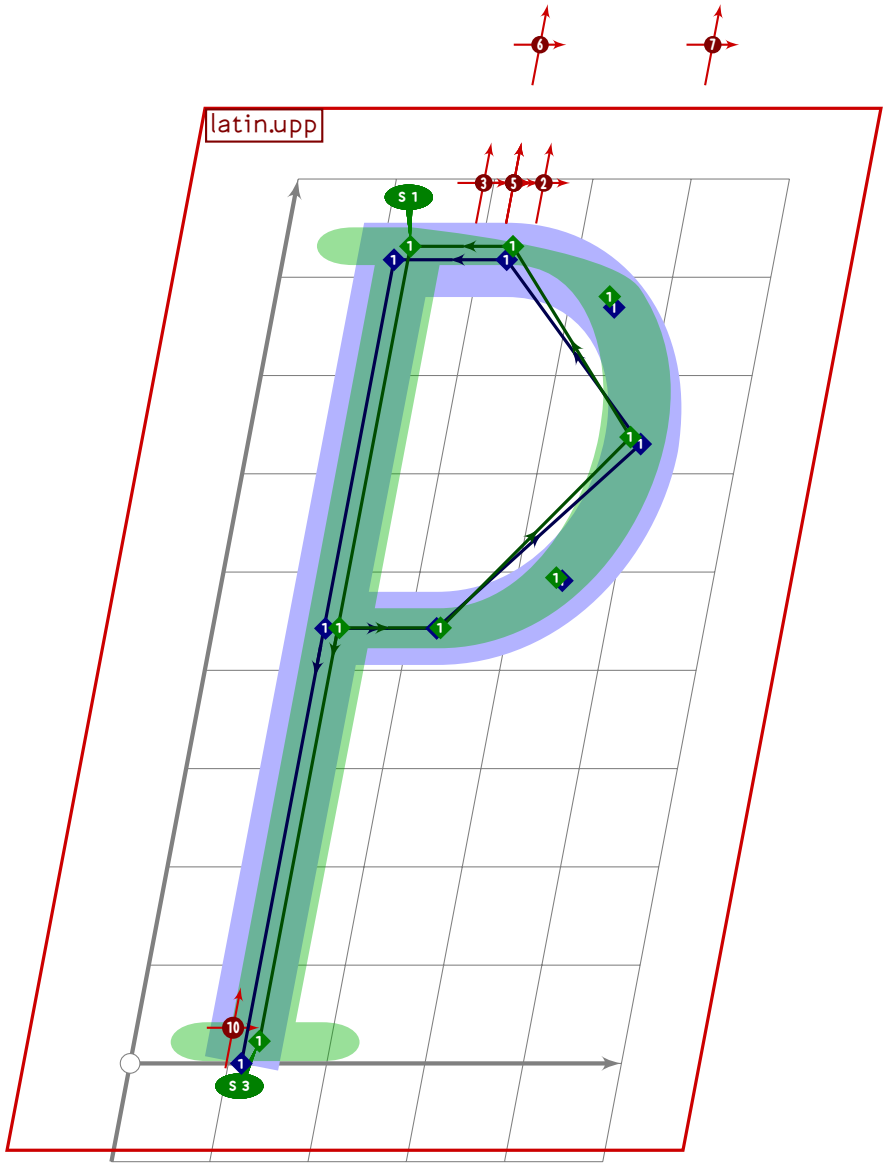


PROO

U+004F
tsuita.O

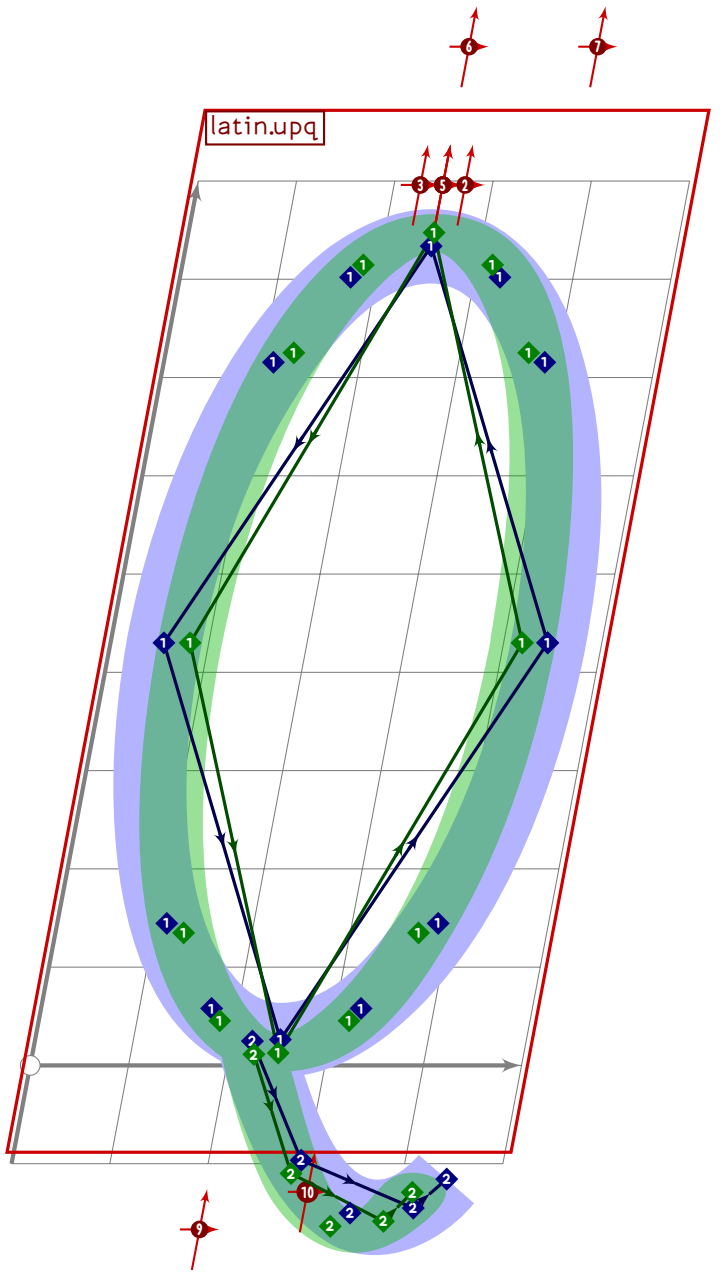


PROO



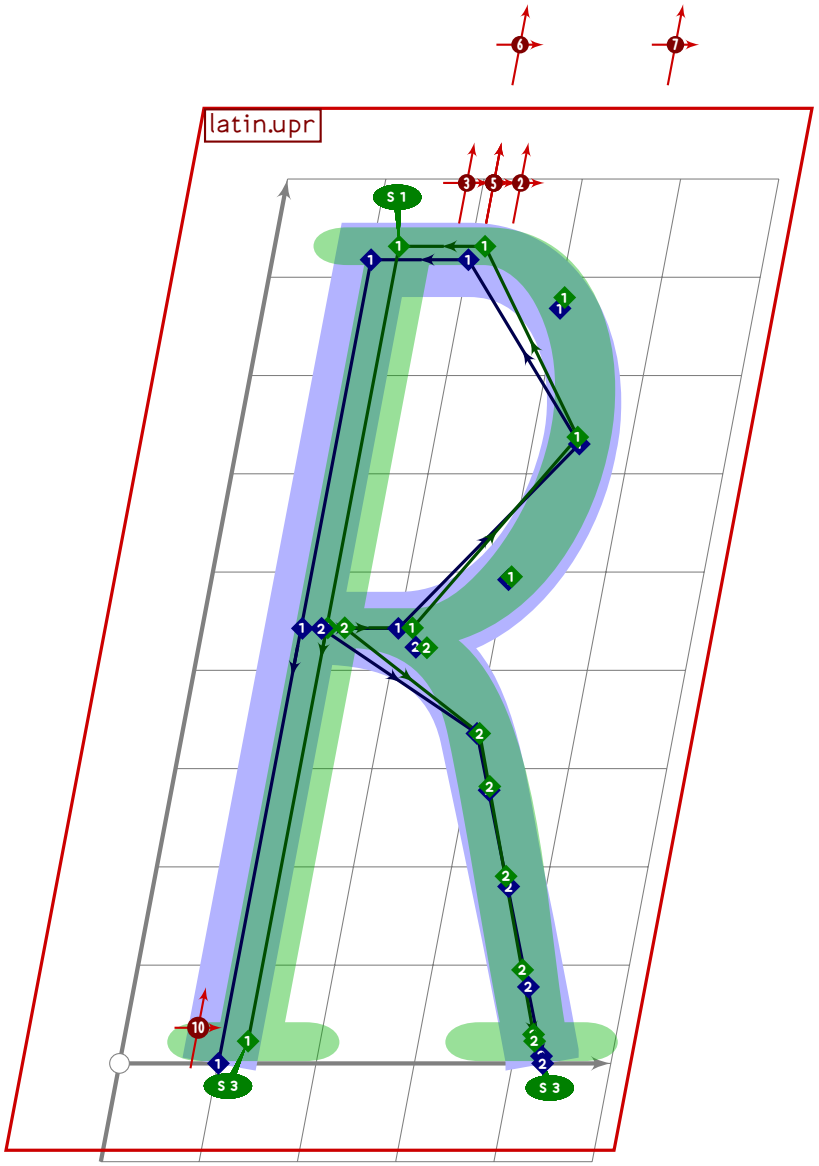
PROO

U+0051
tsuita.Q



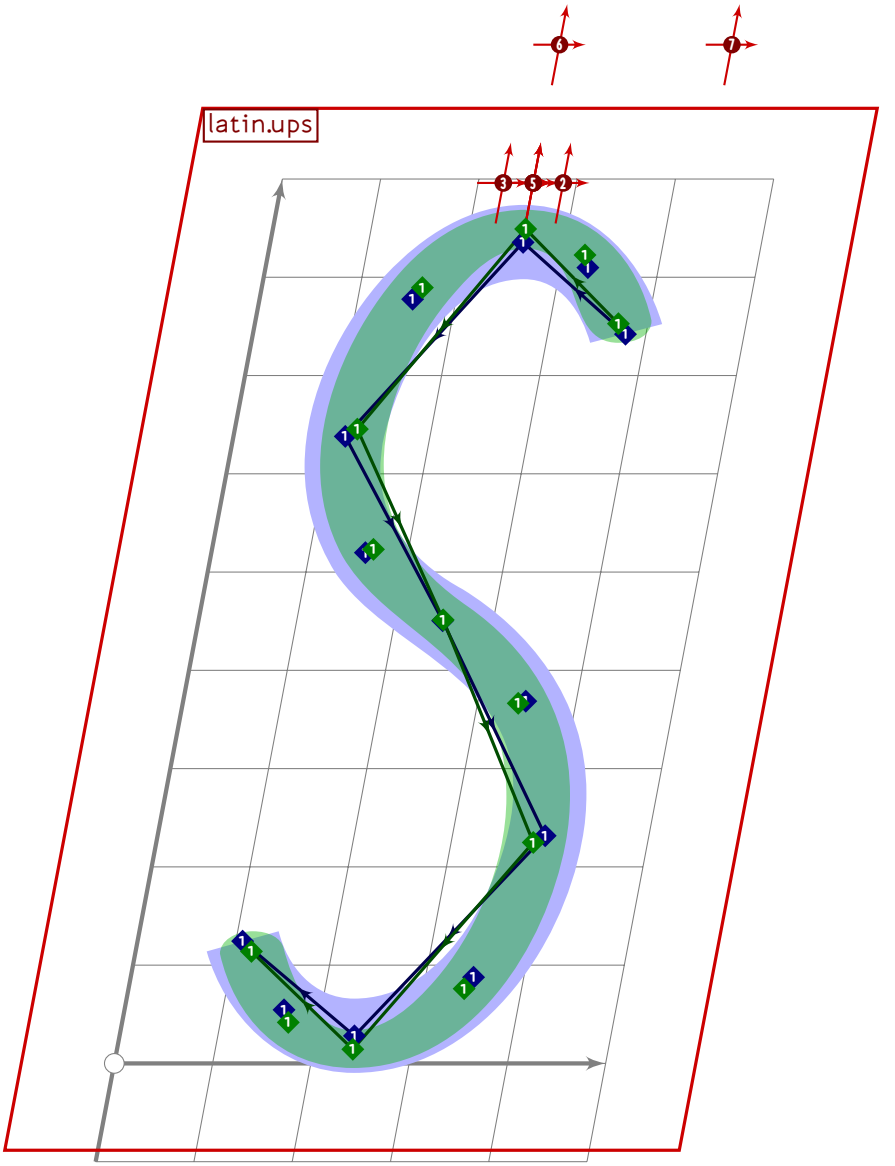
PROO

U+0052
tsuita.R

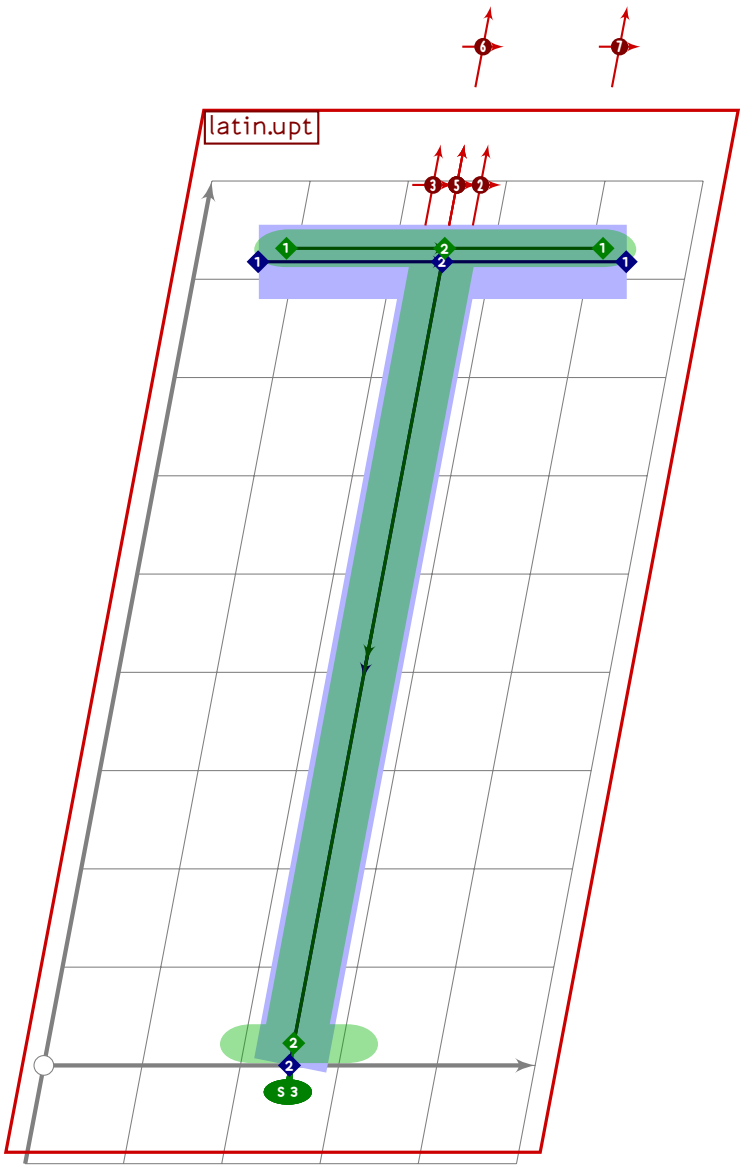


PROO

U+0053
tsuita.S



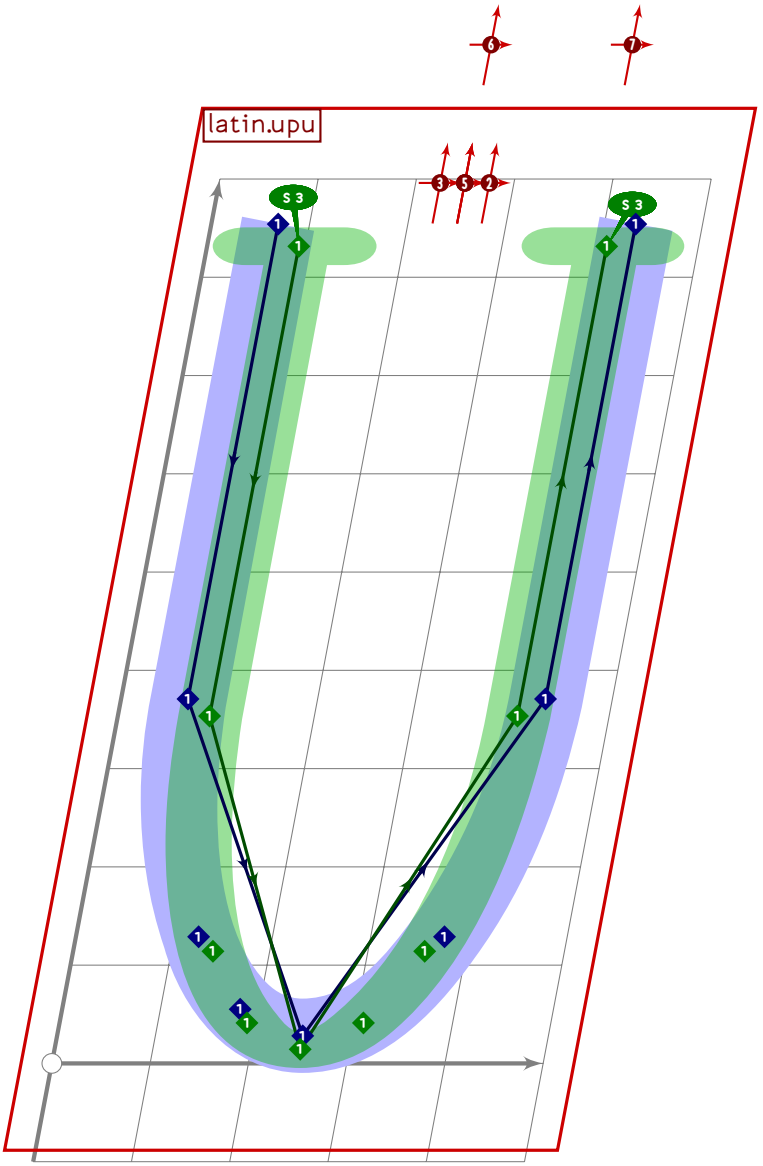
PROO

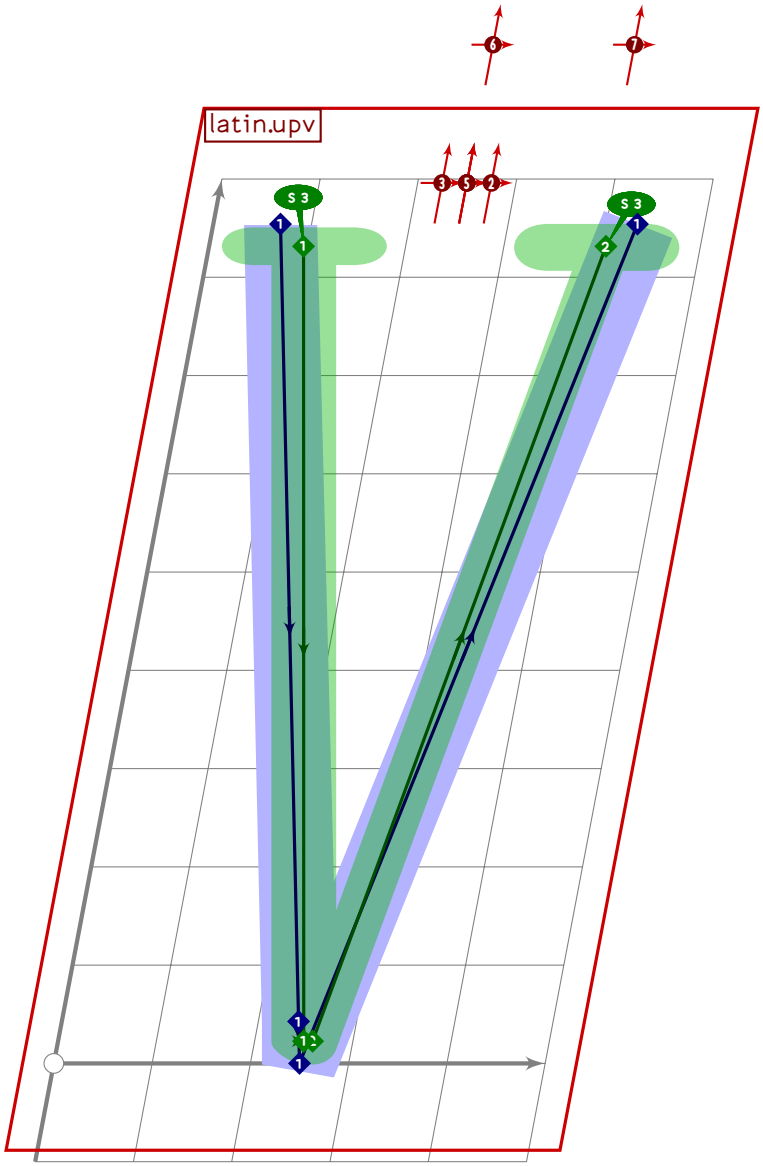


PROO

U+0055
tsuita.U

PROO

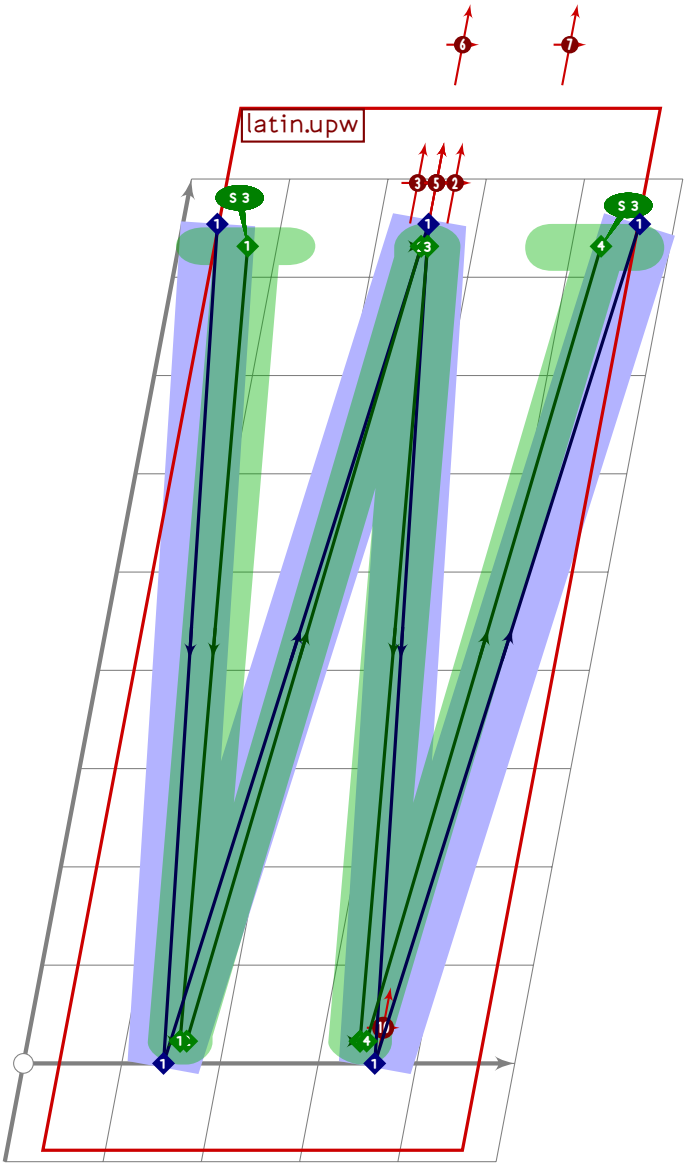




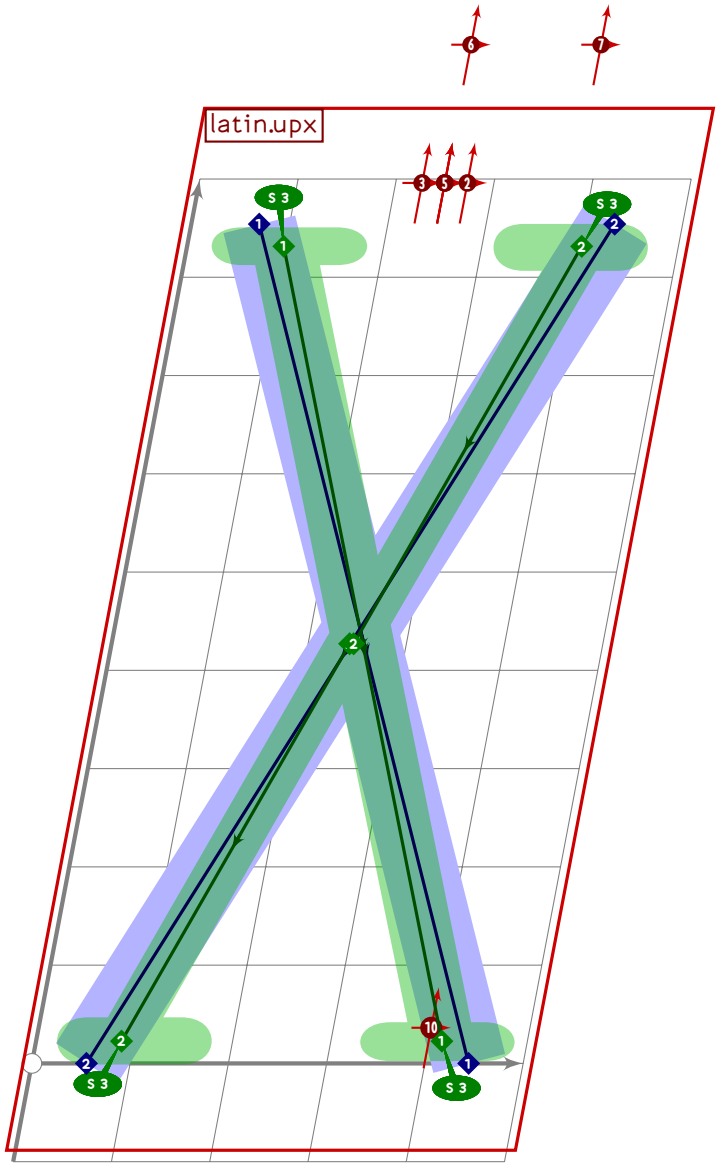
PROO

U+0057
tsuita.W

PROO

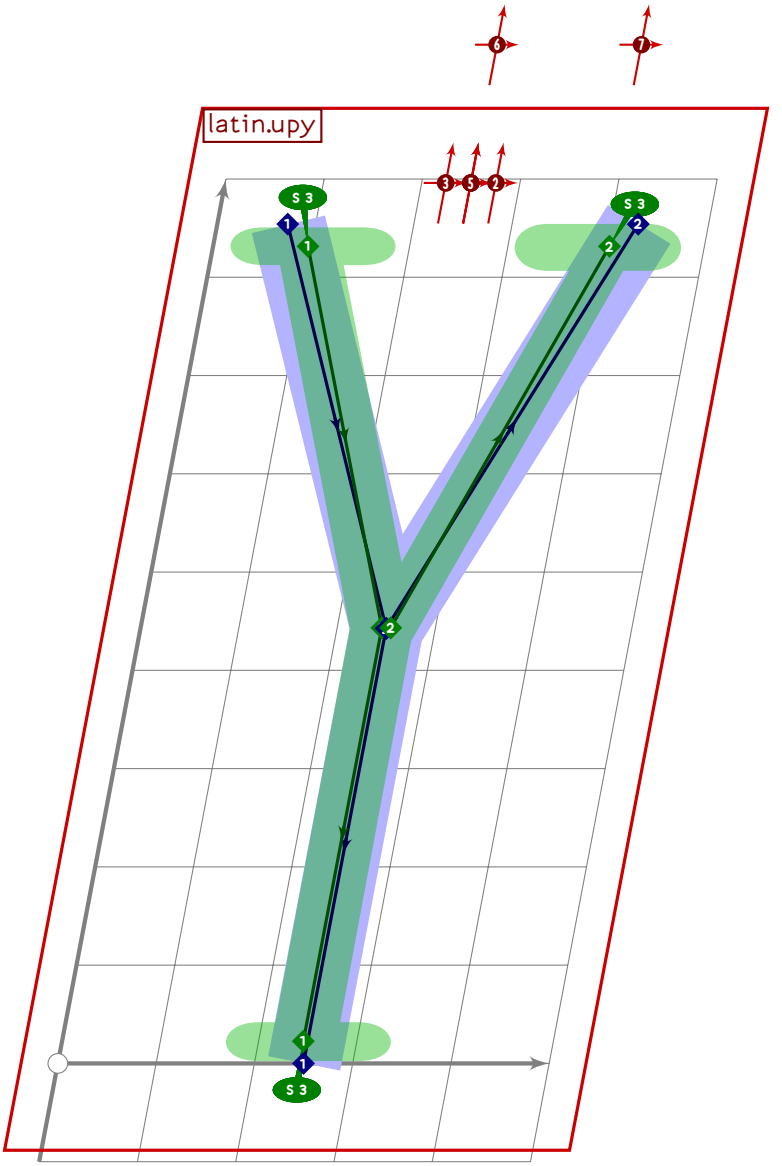


U+0058
tsuita.X

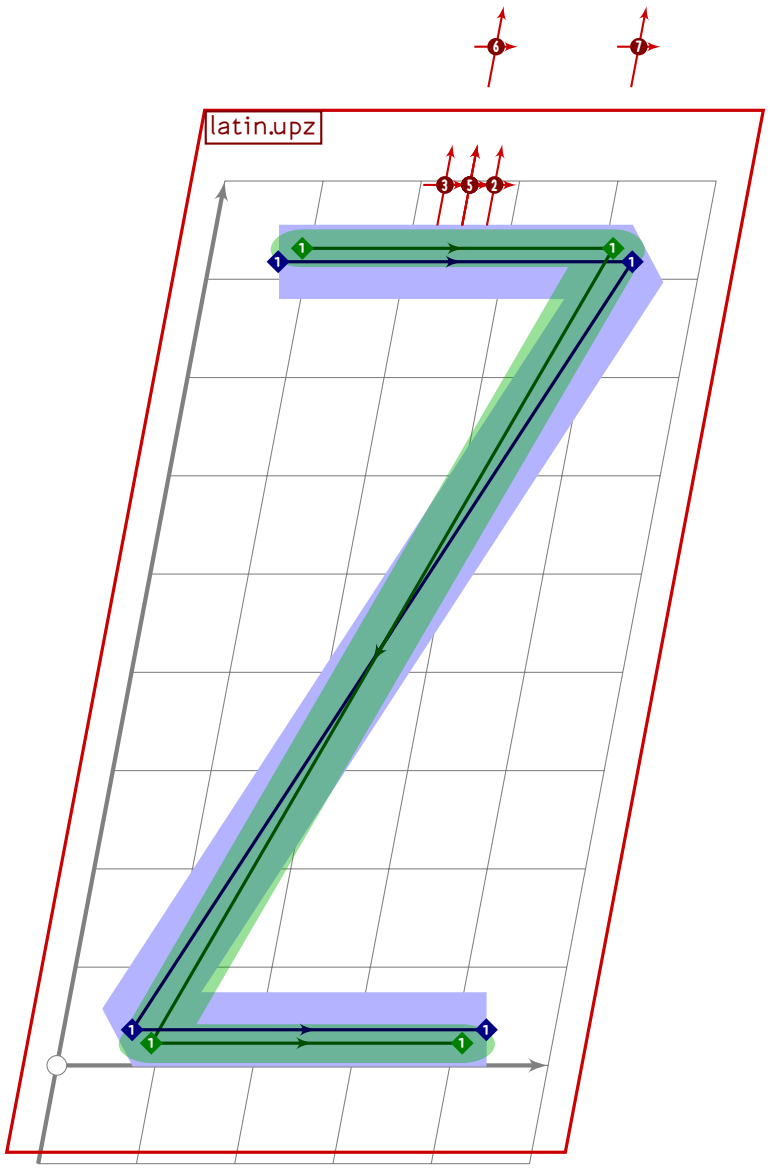


PROO

U+0059
tsuita.Y

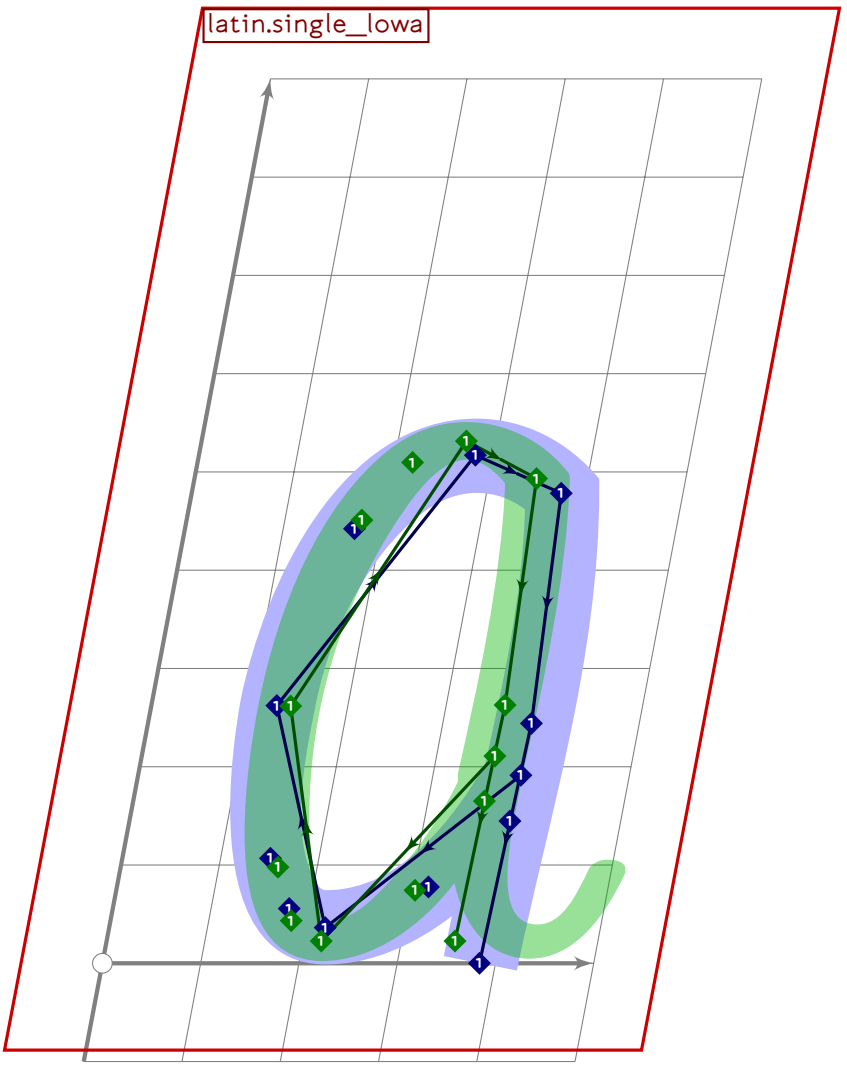


PROO



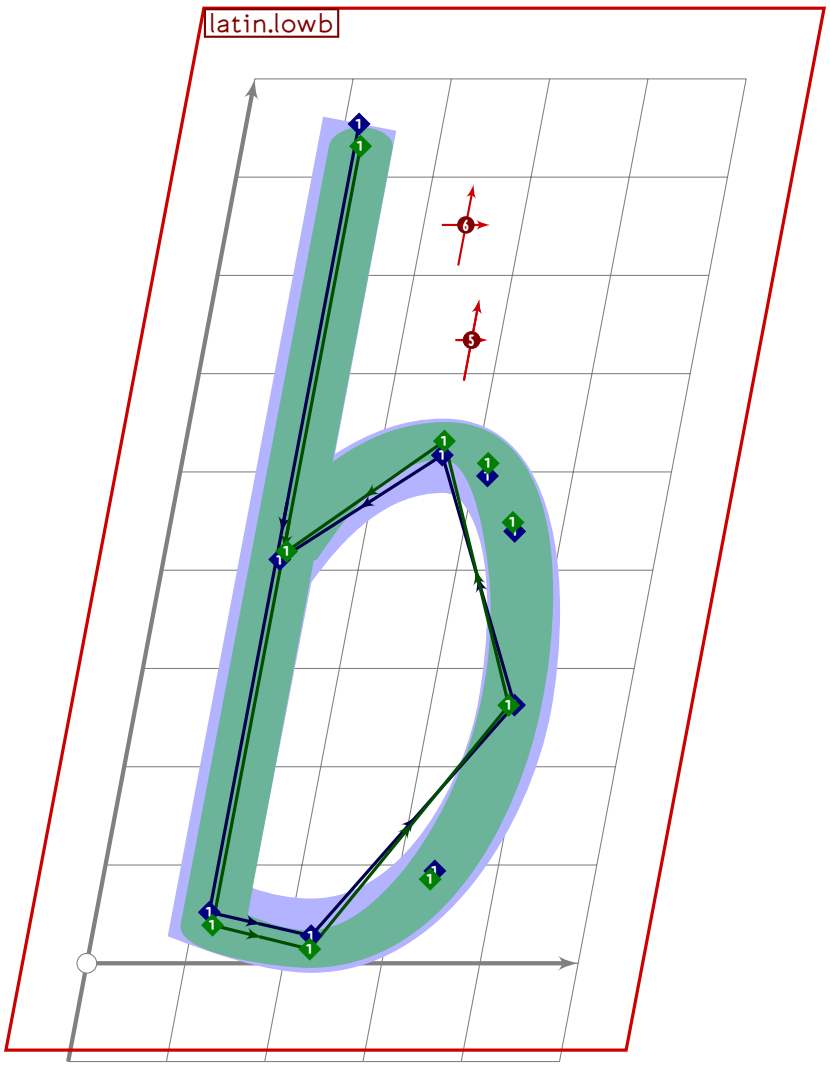
PROO

U+0061
tsuita.a



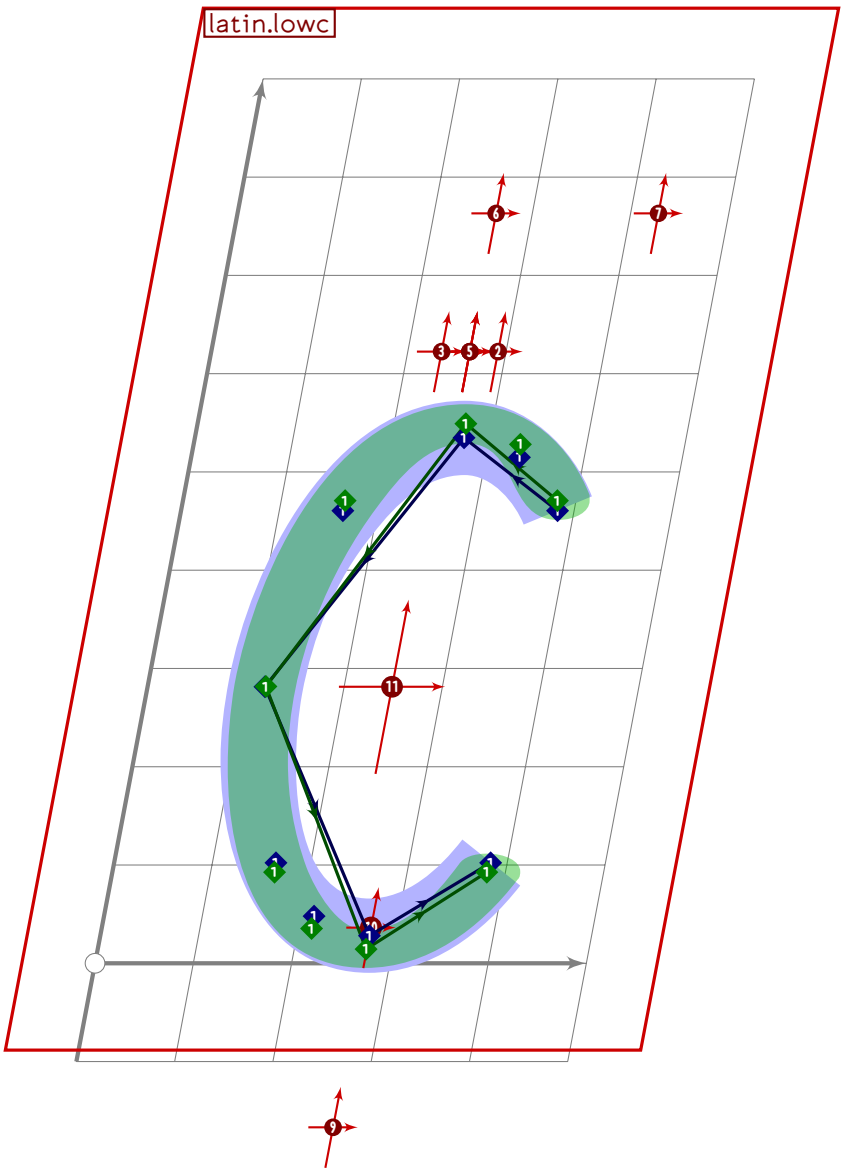
PROO

U+0062
tsuita.b

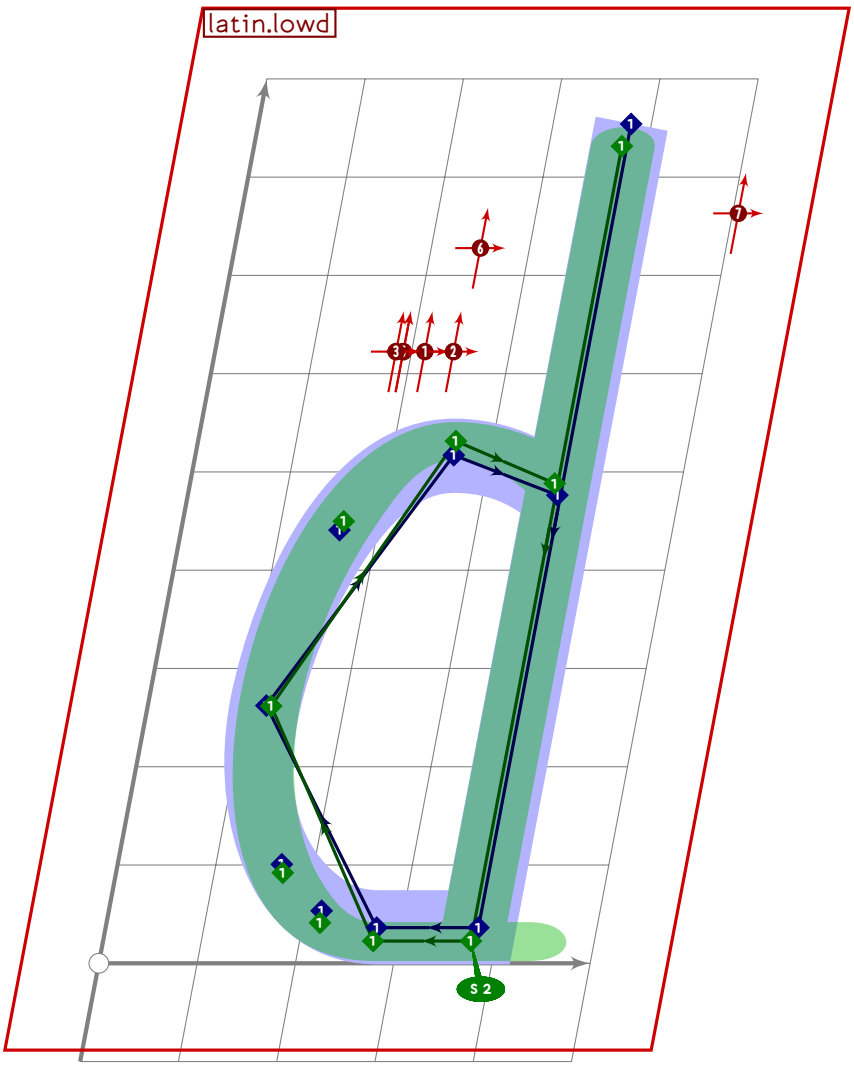


PROO

U+0063
tsuita.c

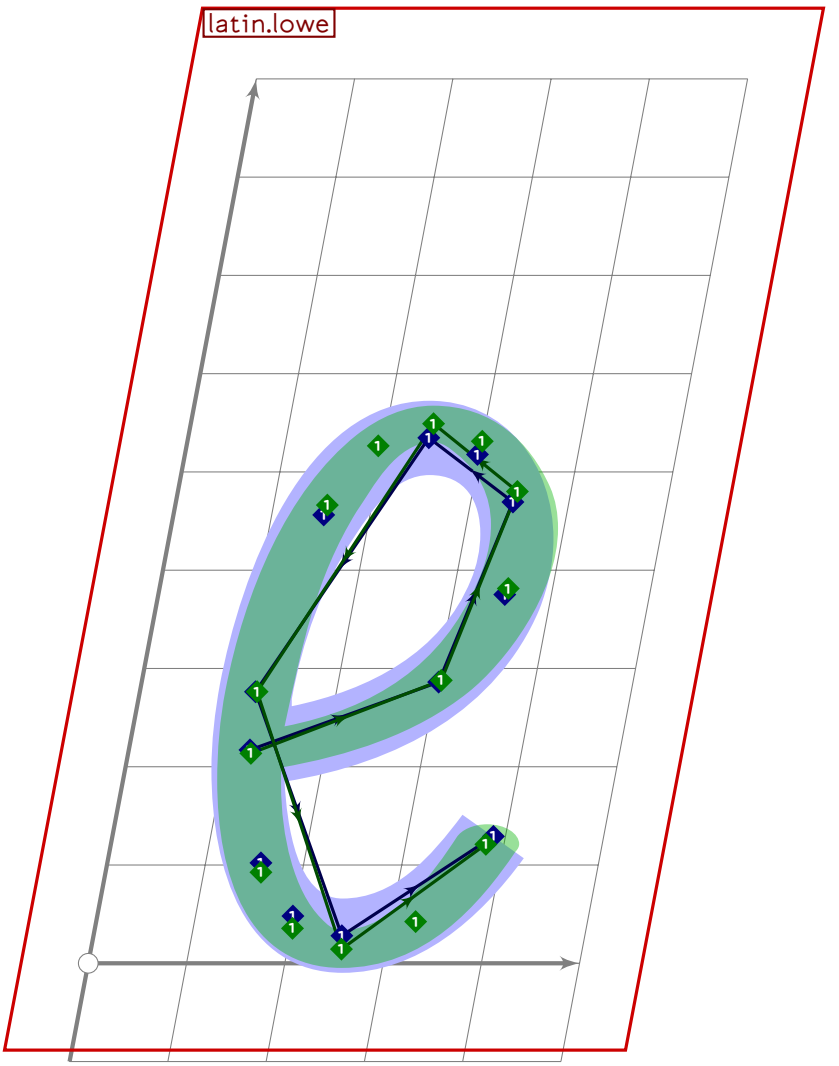


PROO

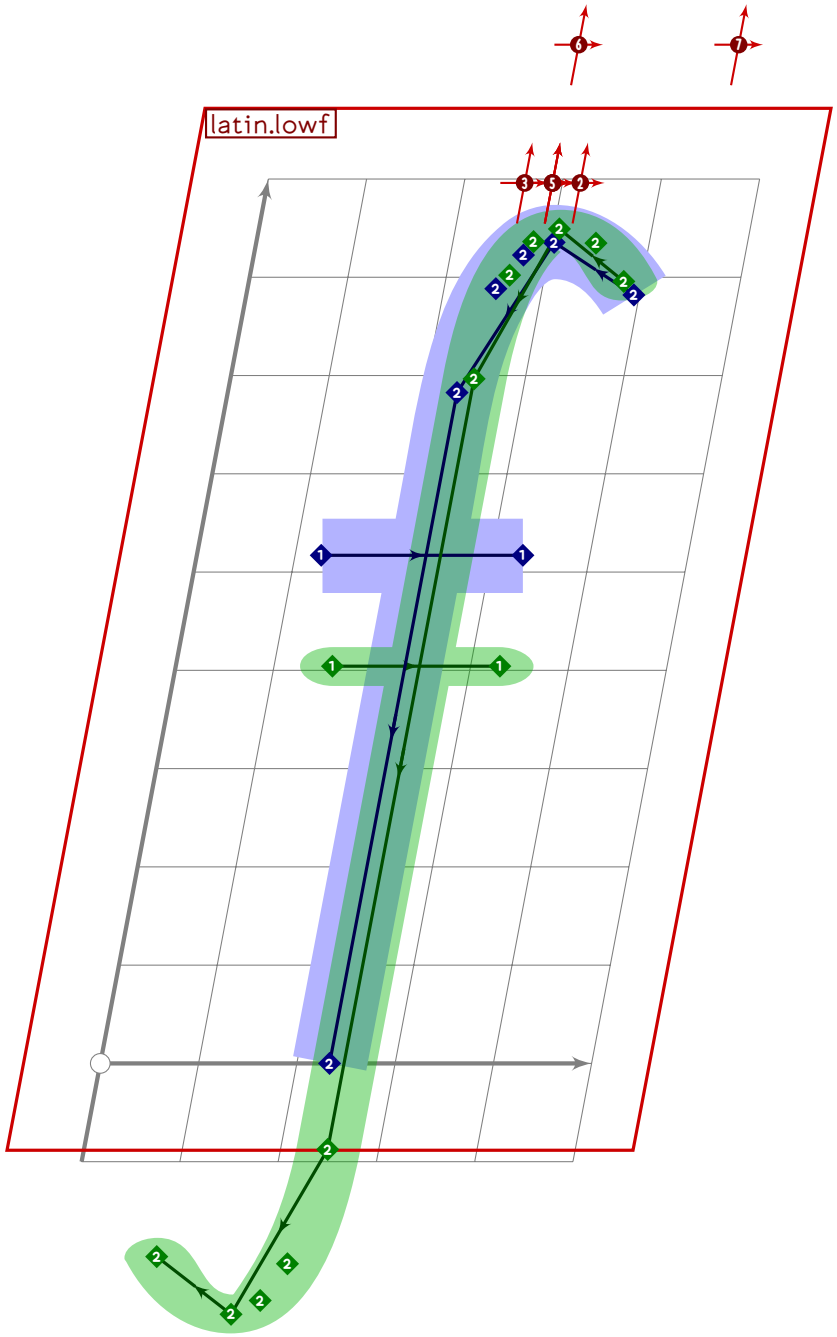


PROO

U+0065
tsuita.e

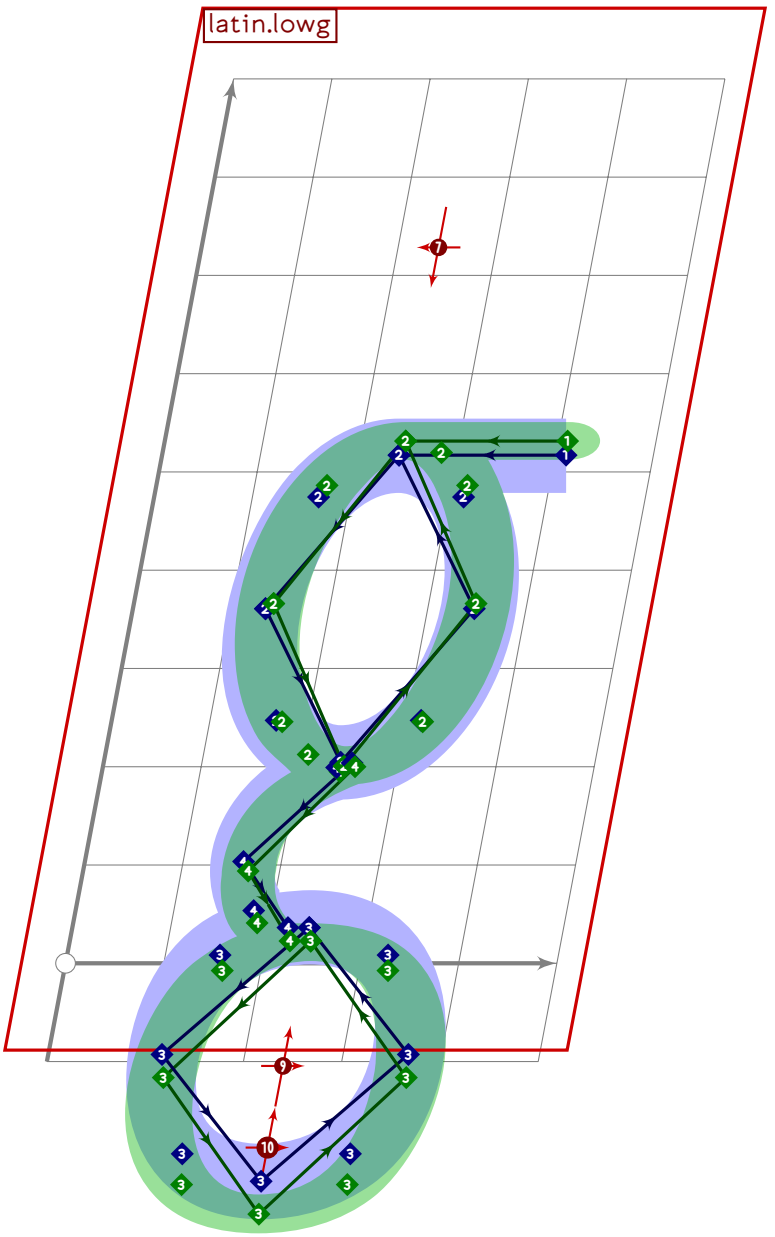


PROO

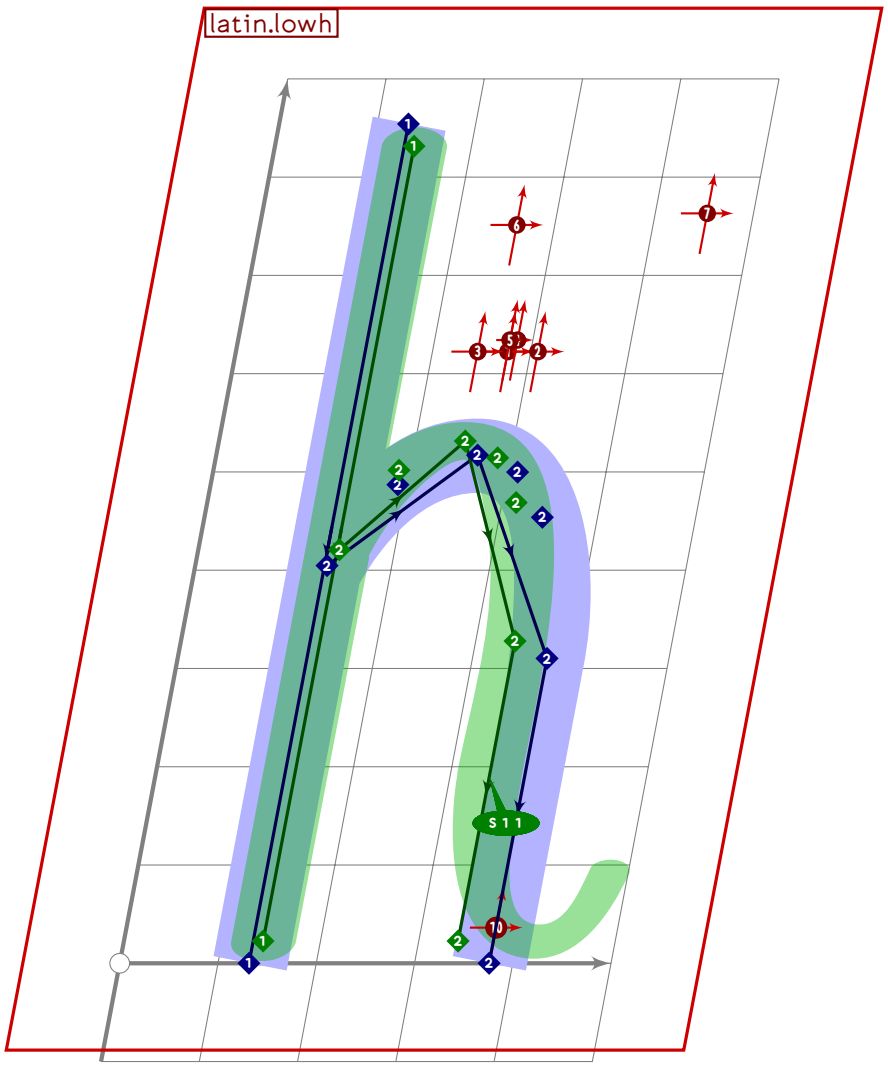


PROO

U+0067
tsuita.g



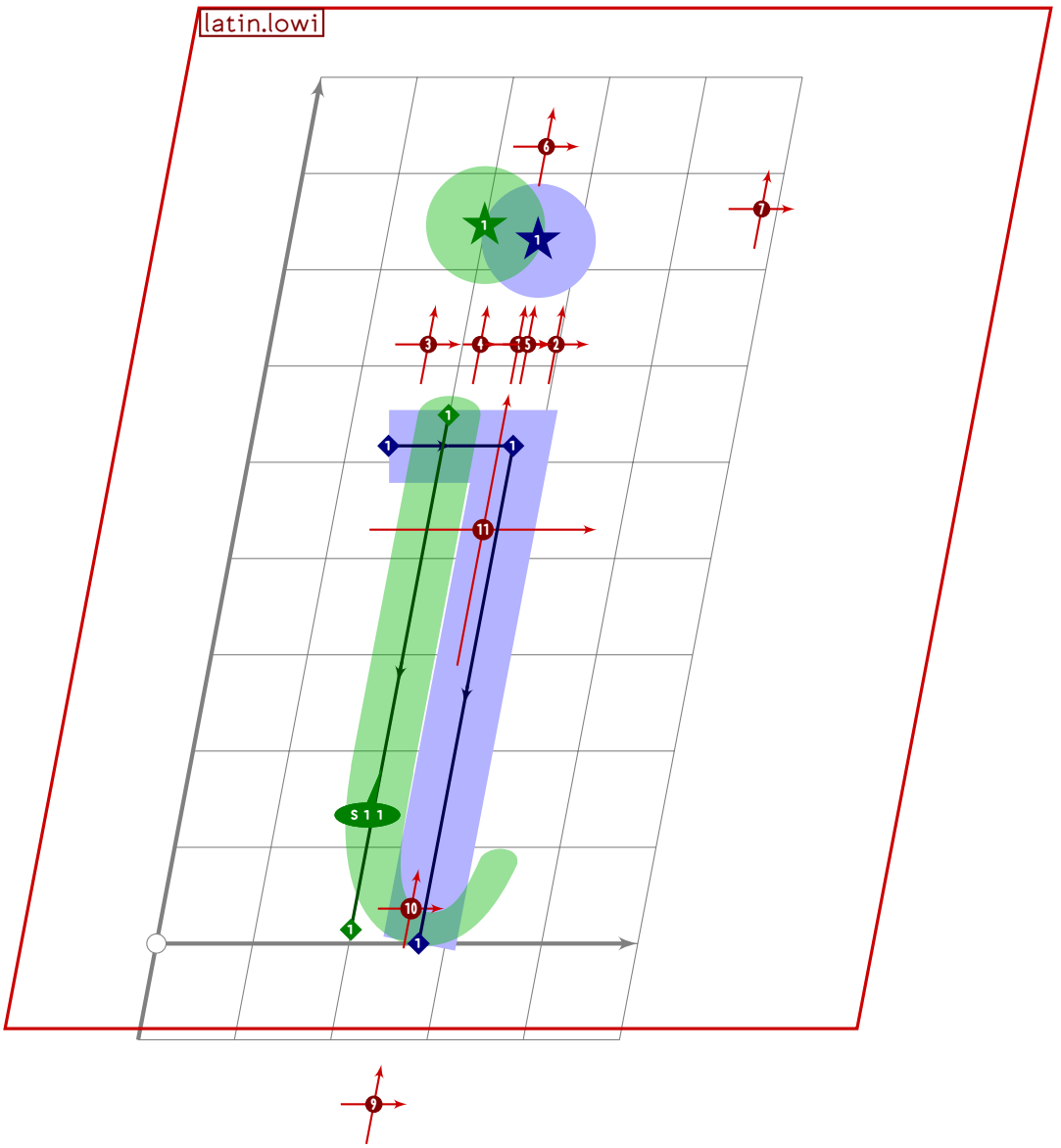
PROO

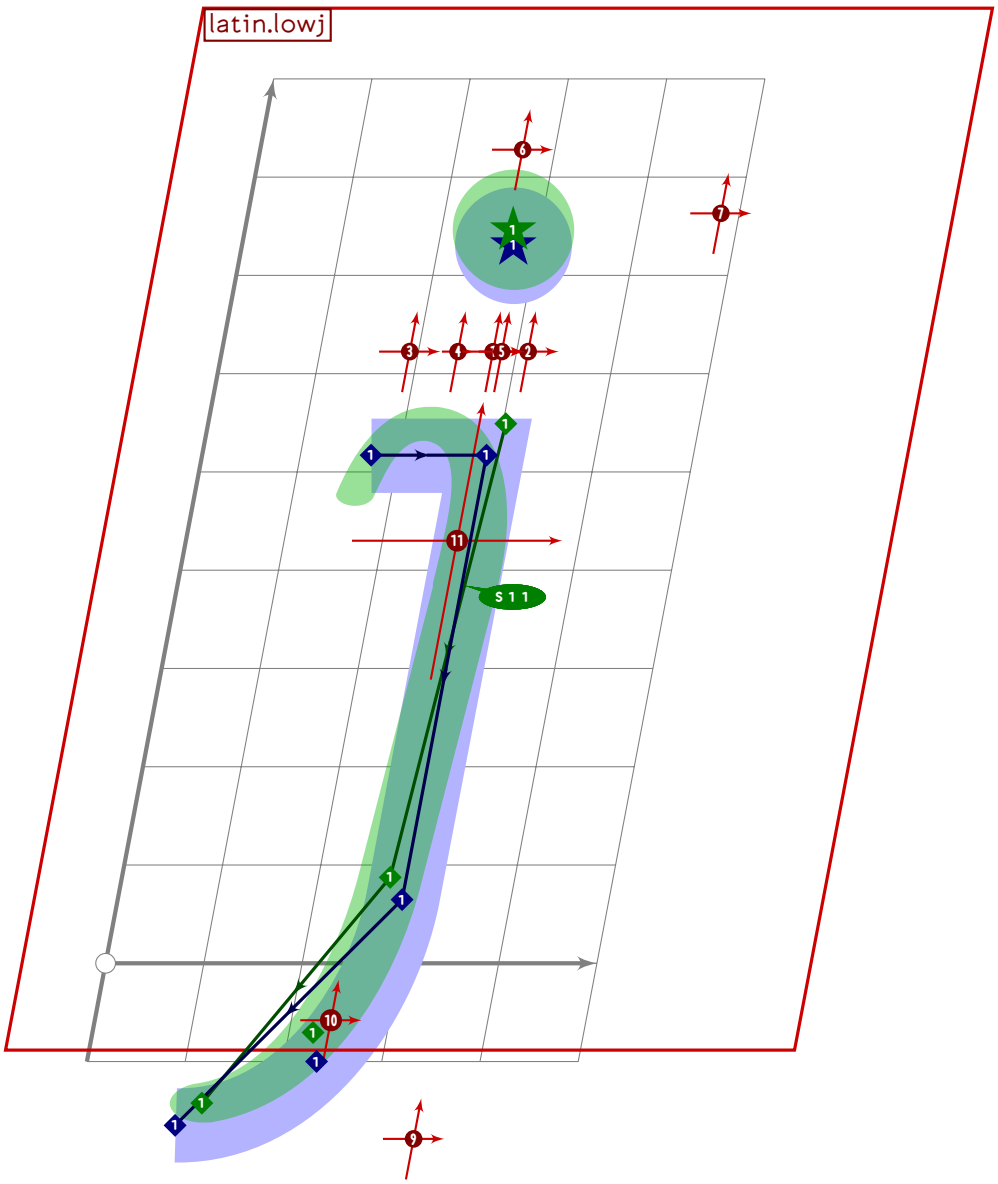


PROO

U+0069
tsuita.i

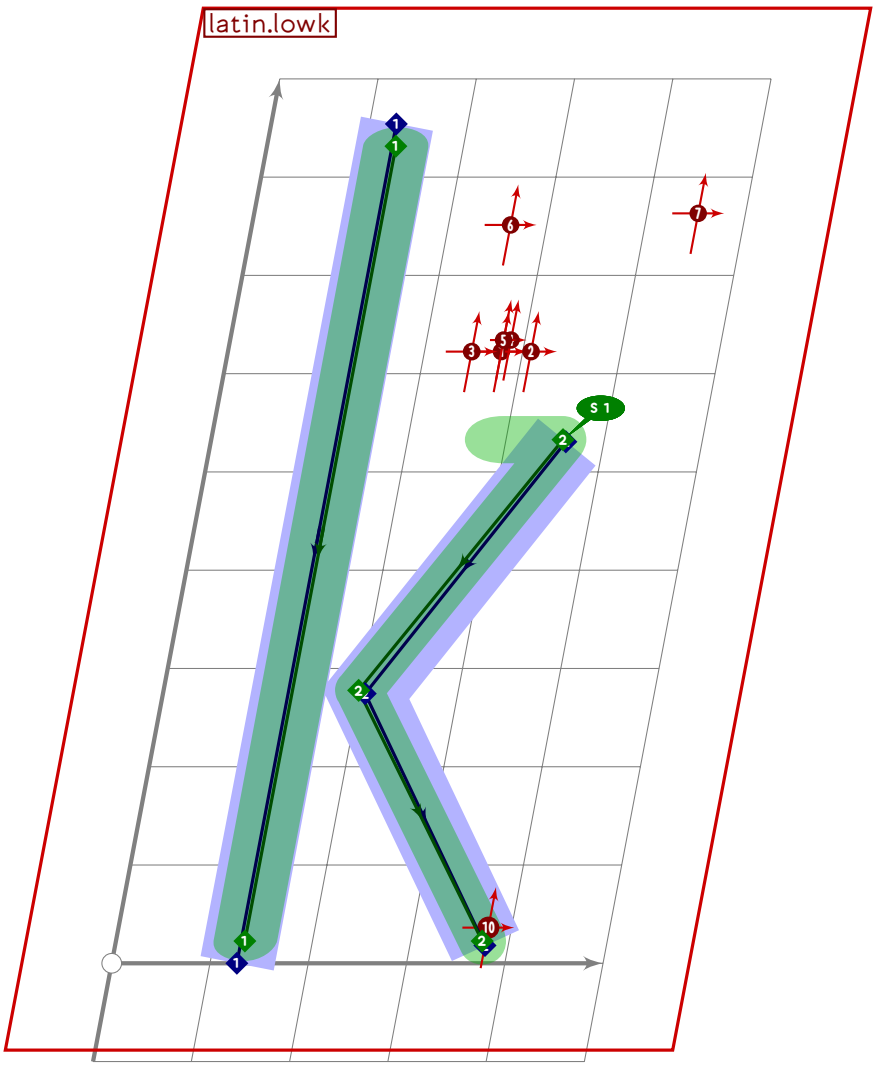
PROO



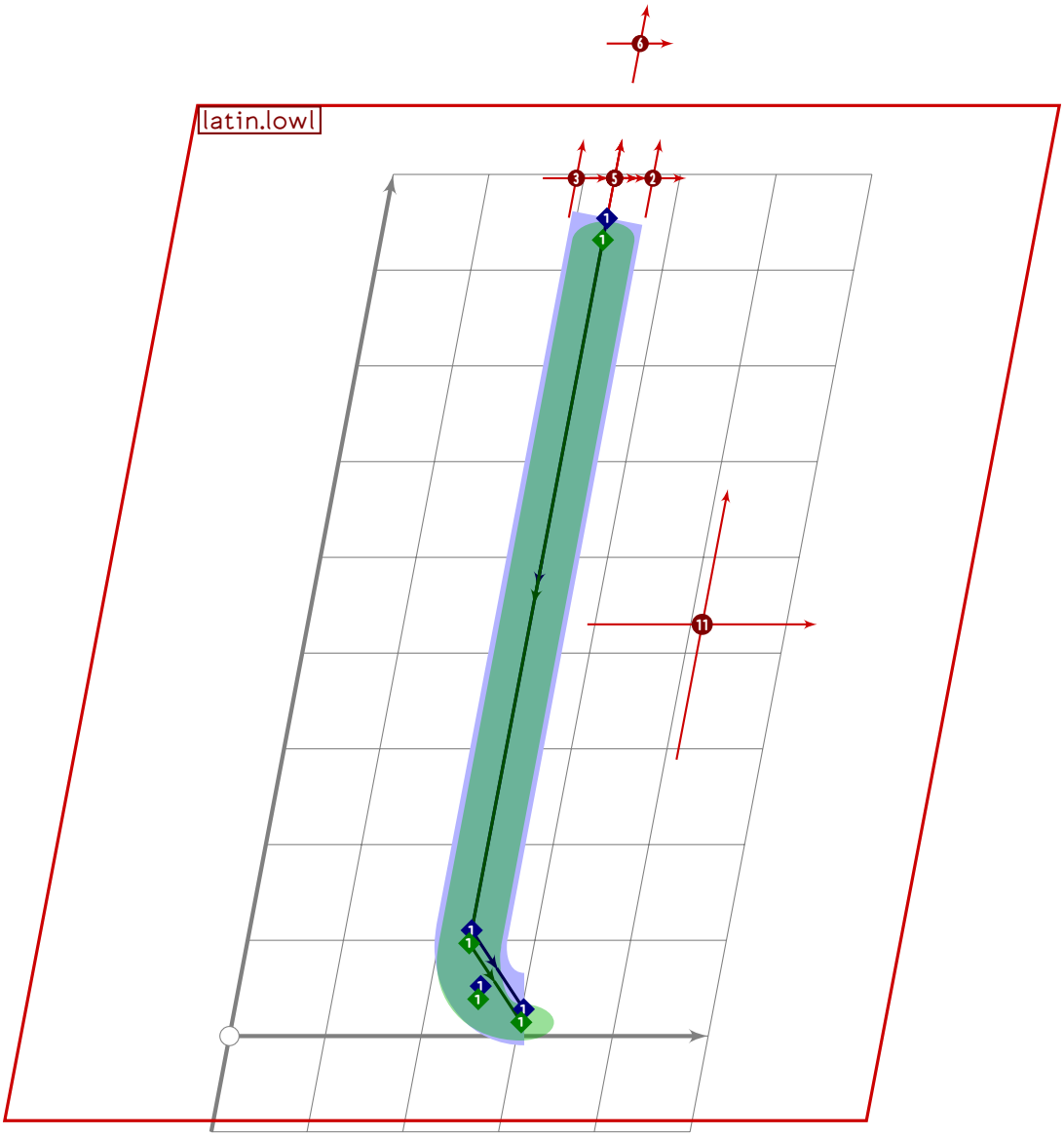


PROO

U+006B
tsuita.k

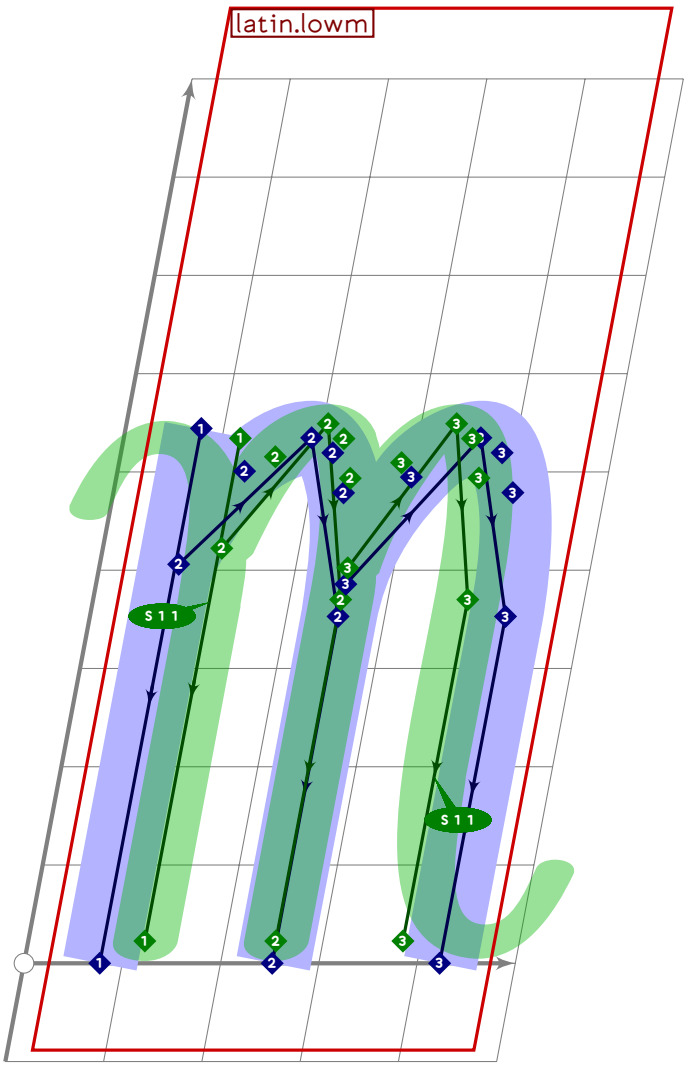


PROO

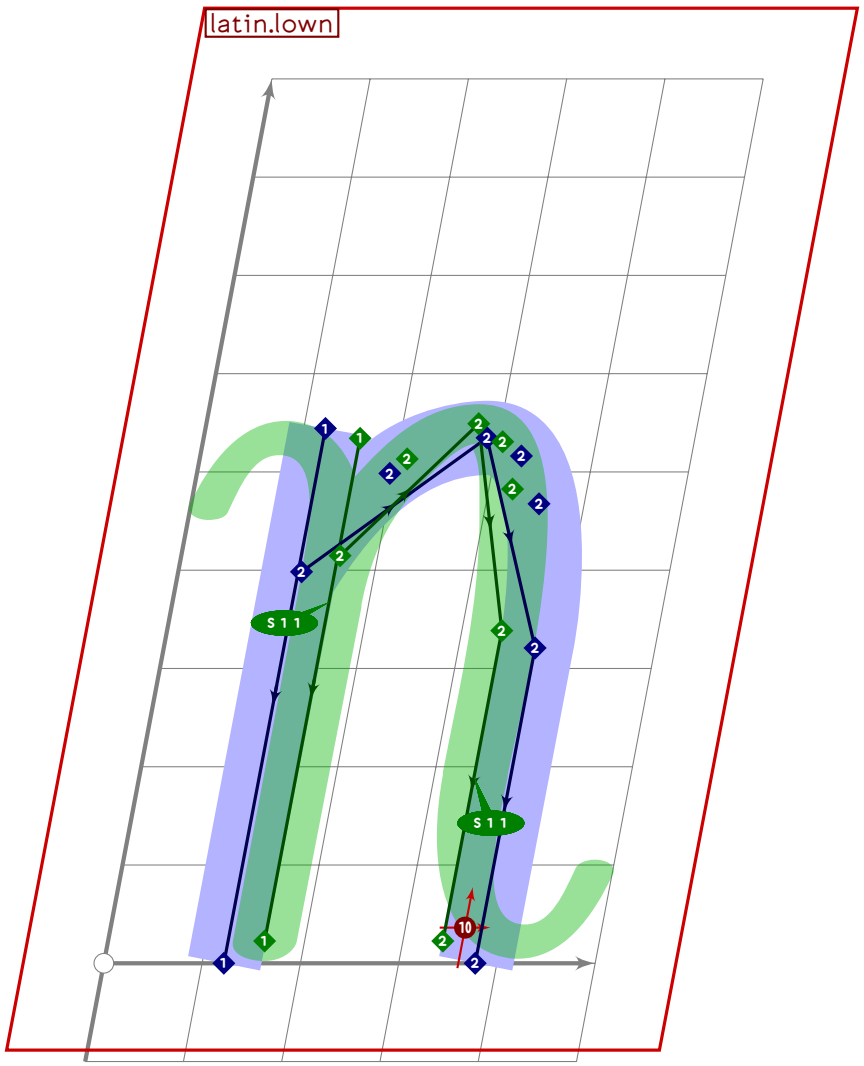


PROO

U+006D
tsuita.m

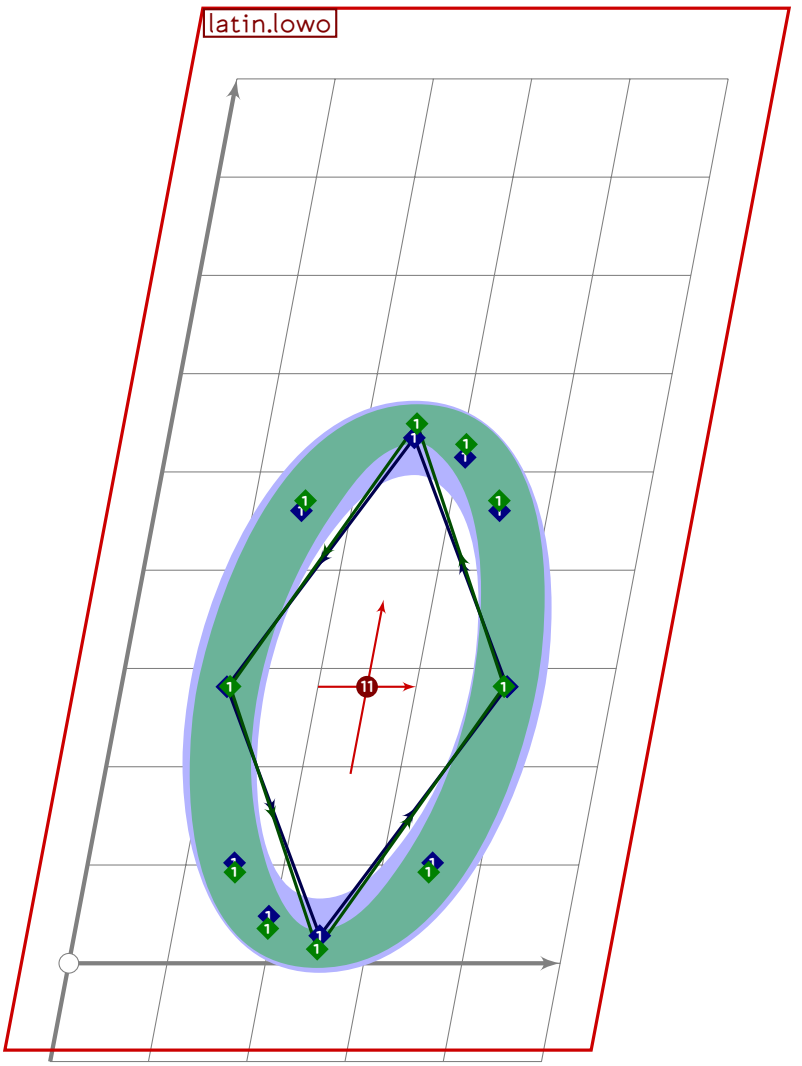


PROO

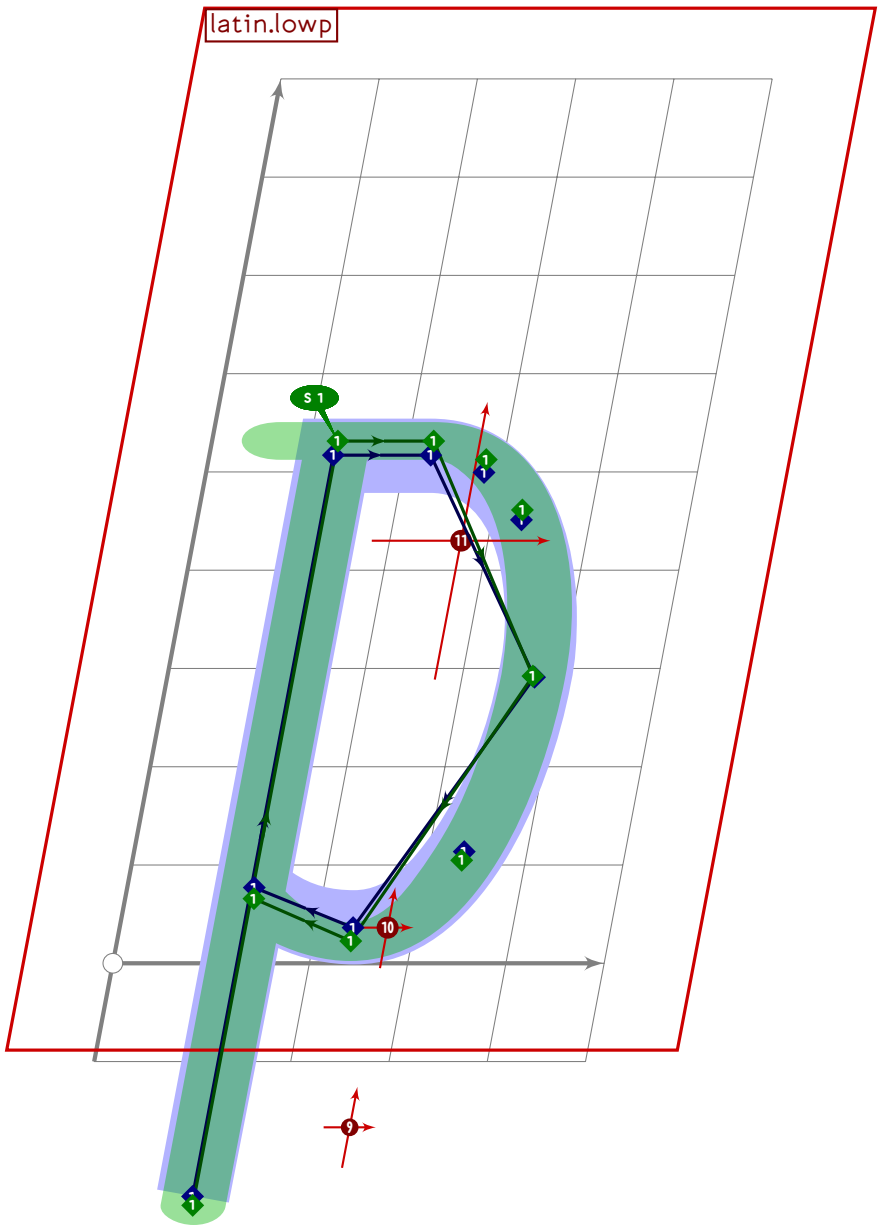


PROO

U+006F
tsuita.o

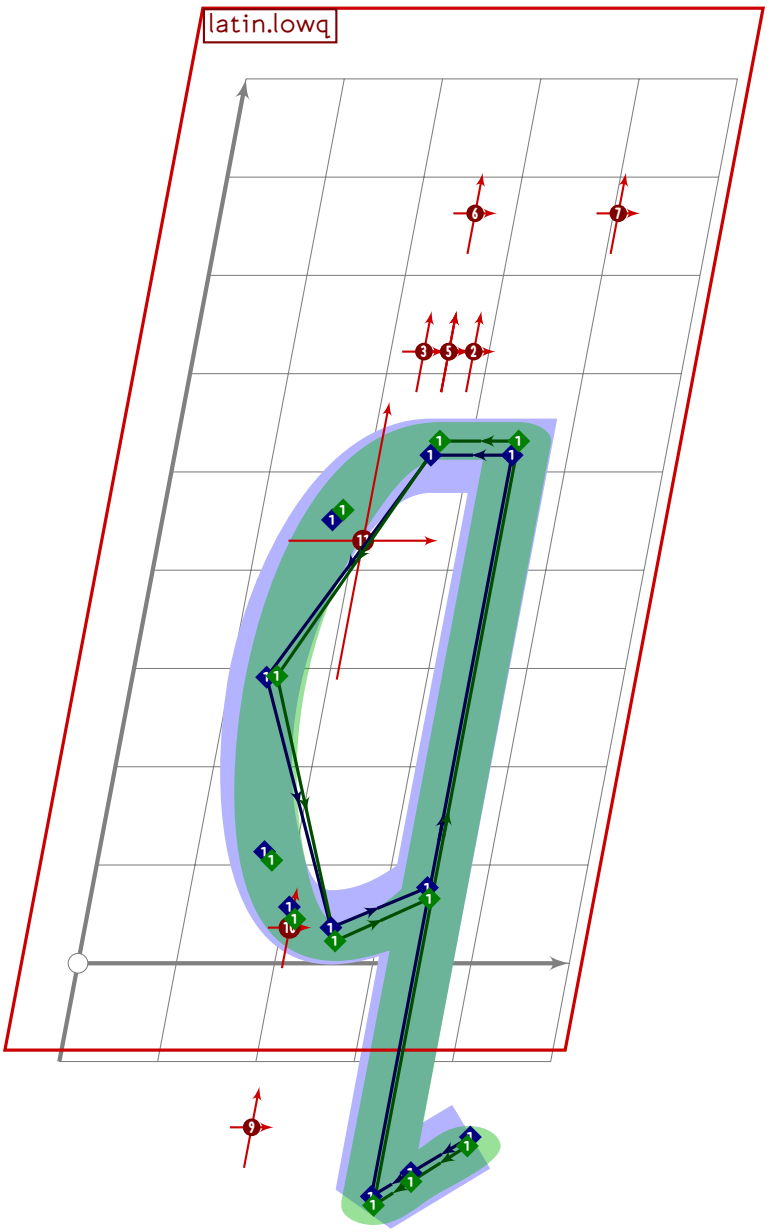


PROO

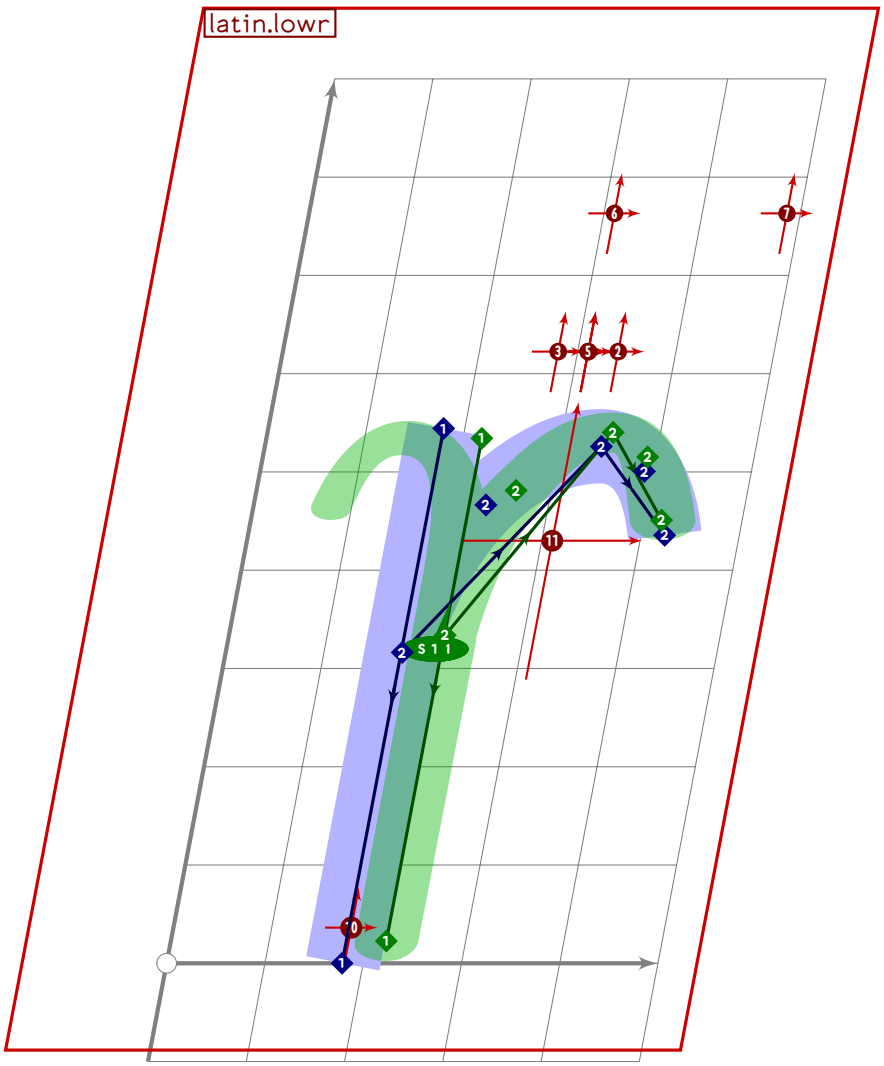


PROO

U+0071
tsuita.q

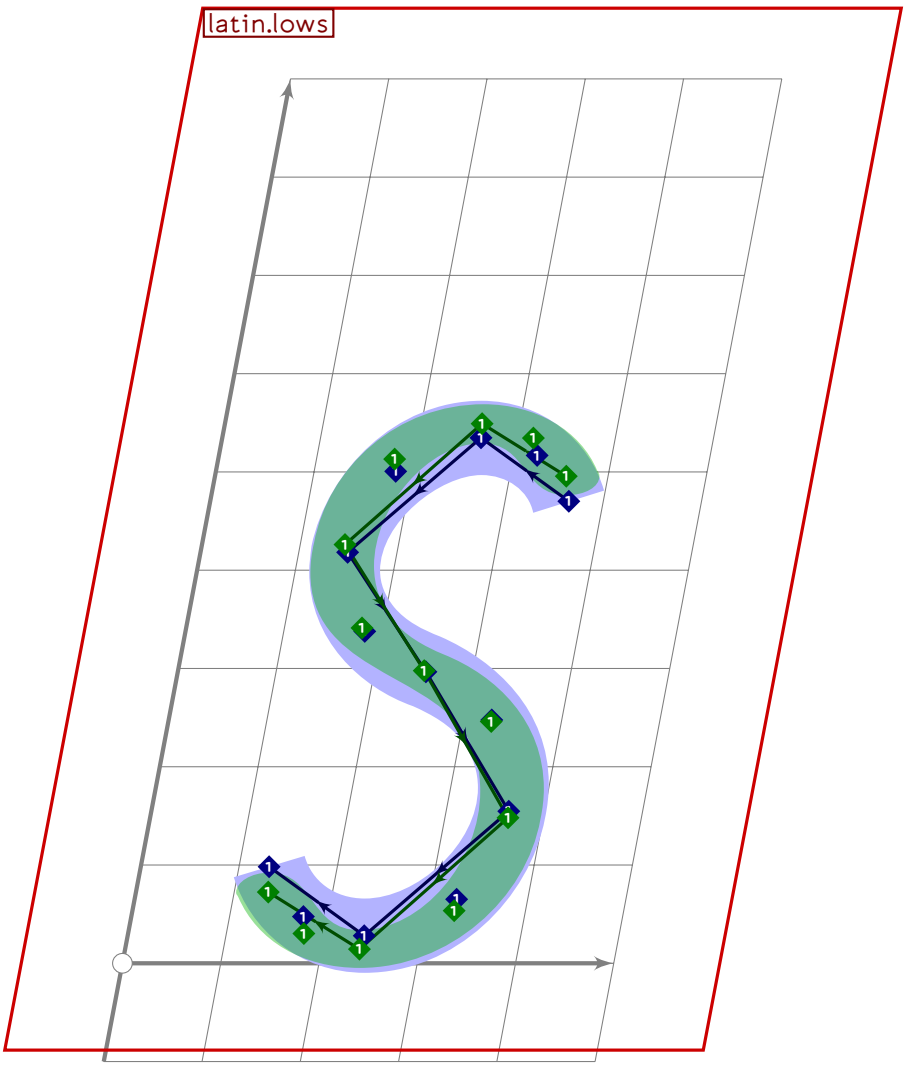


PROO

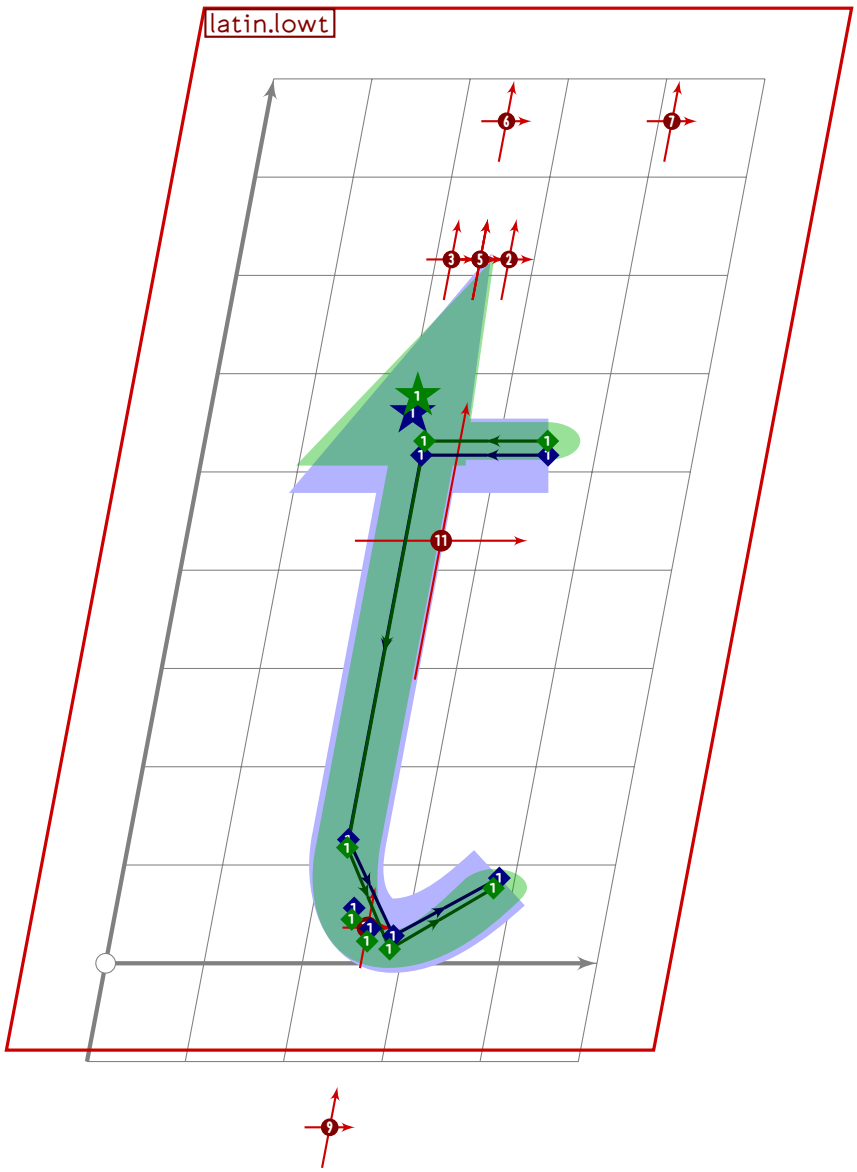


PROO

U+0073
tsuita.s

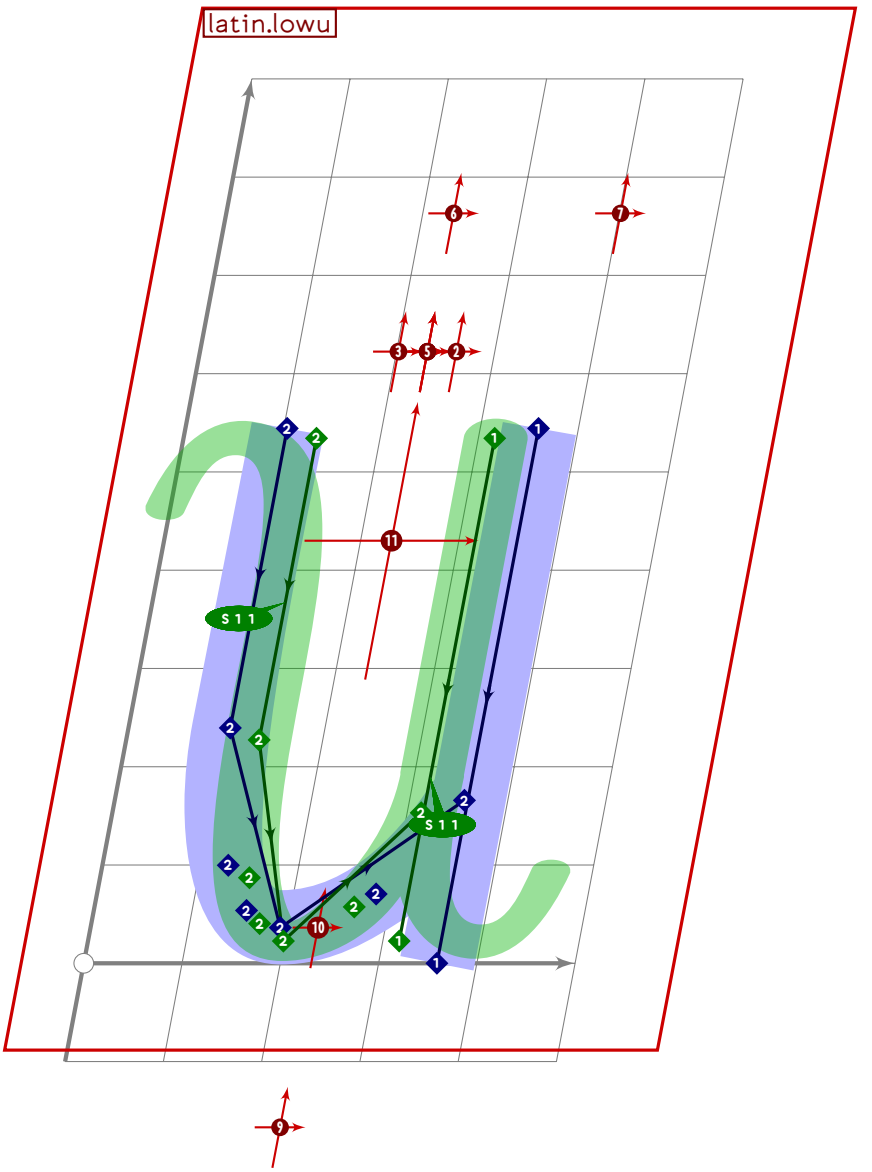


PROO

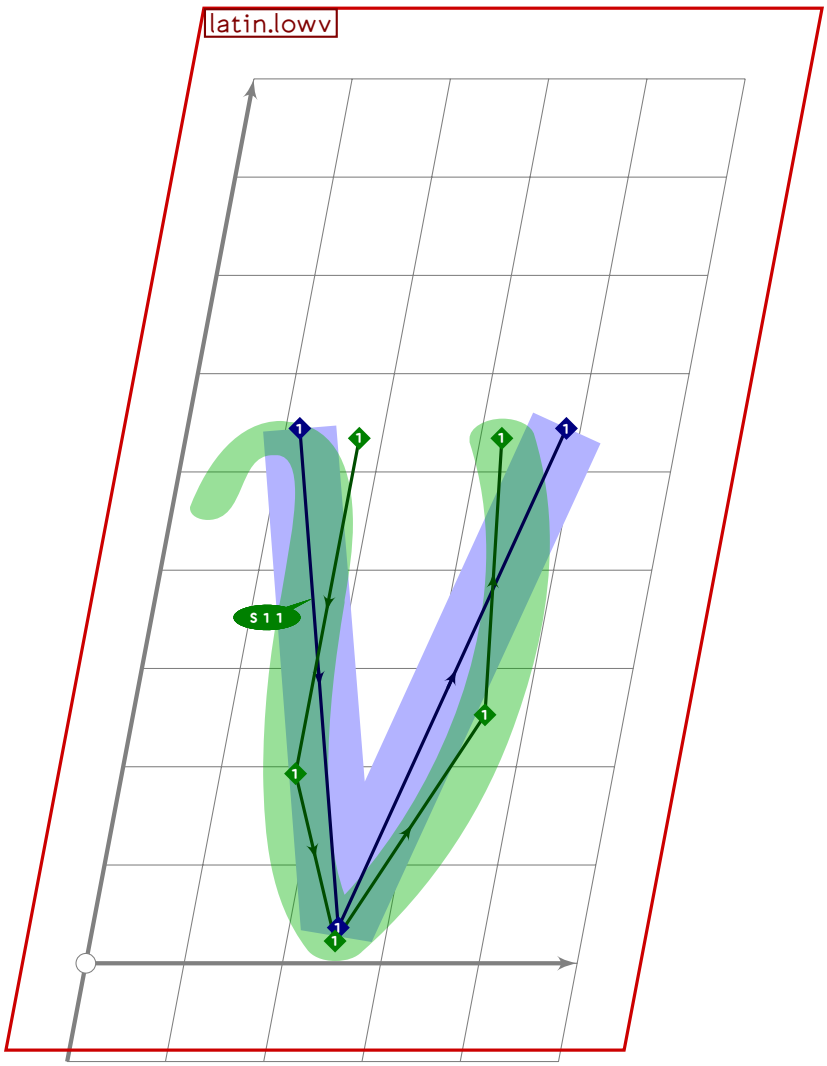


PROO

U+0075
tsuita.u

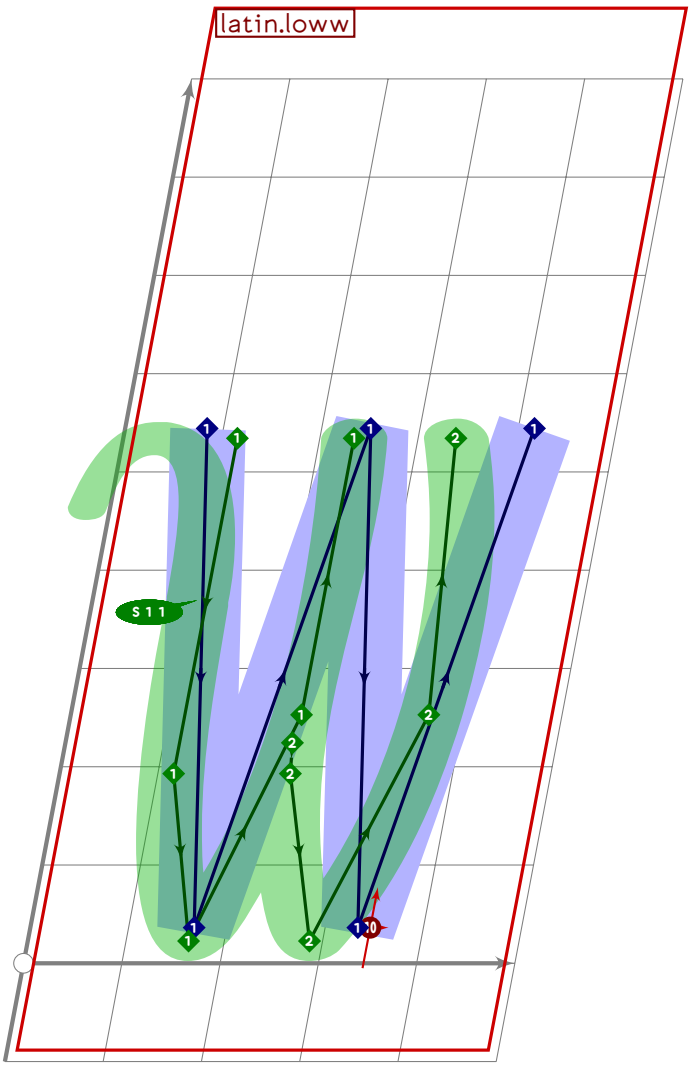


PROO

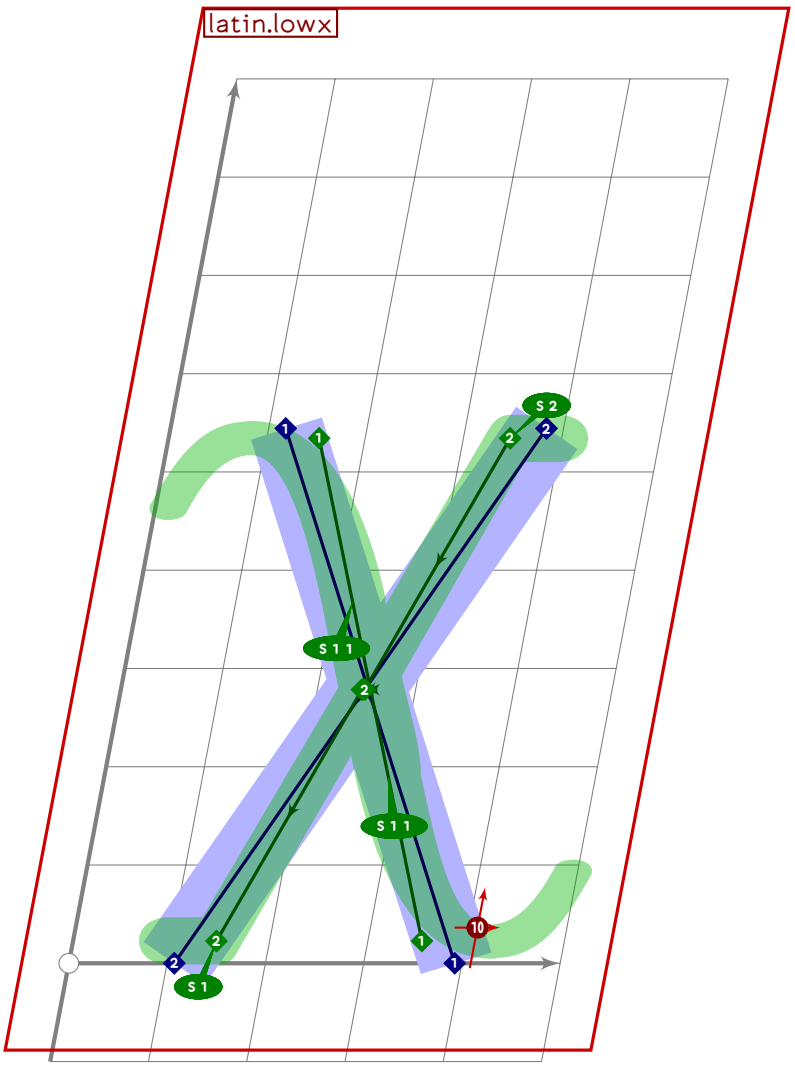


PROO

U+0077
tsuita.w

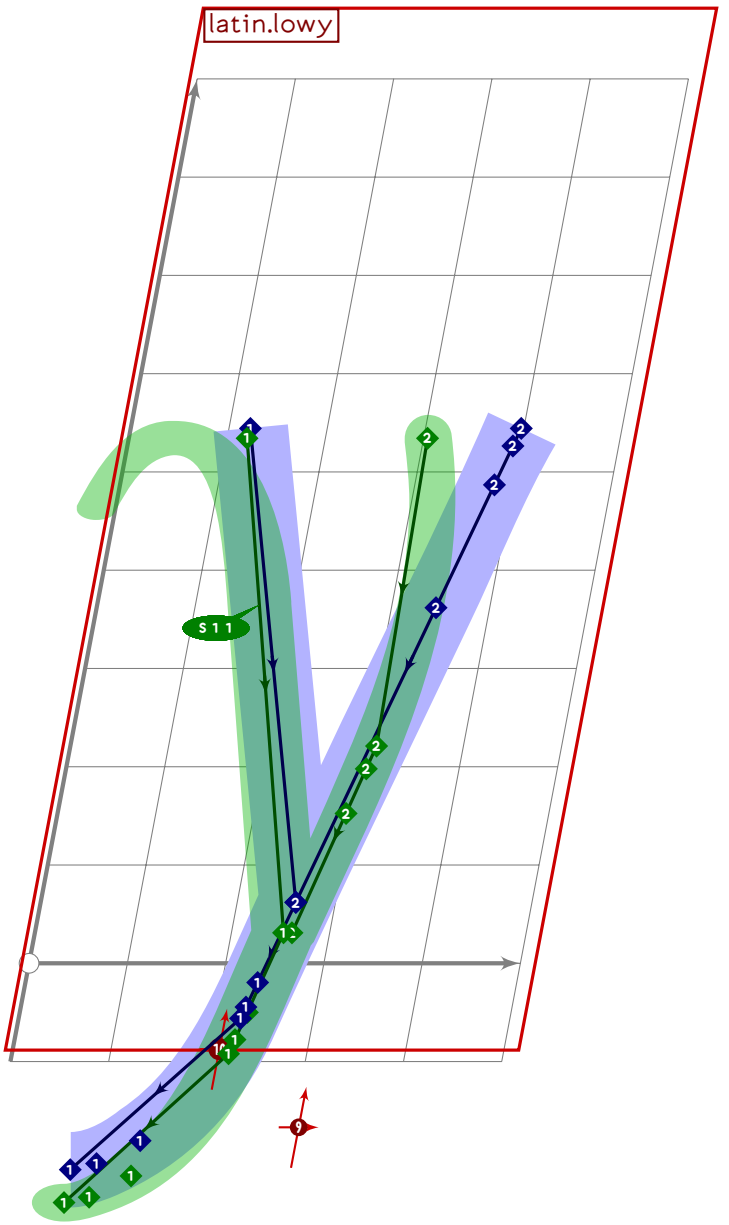


PROO

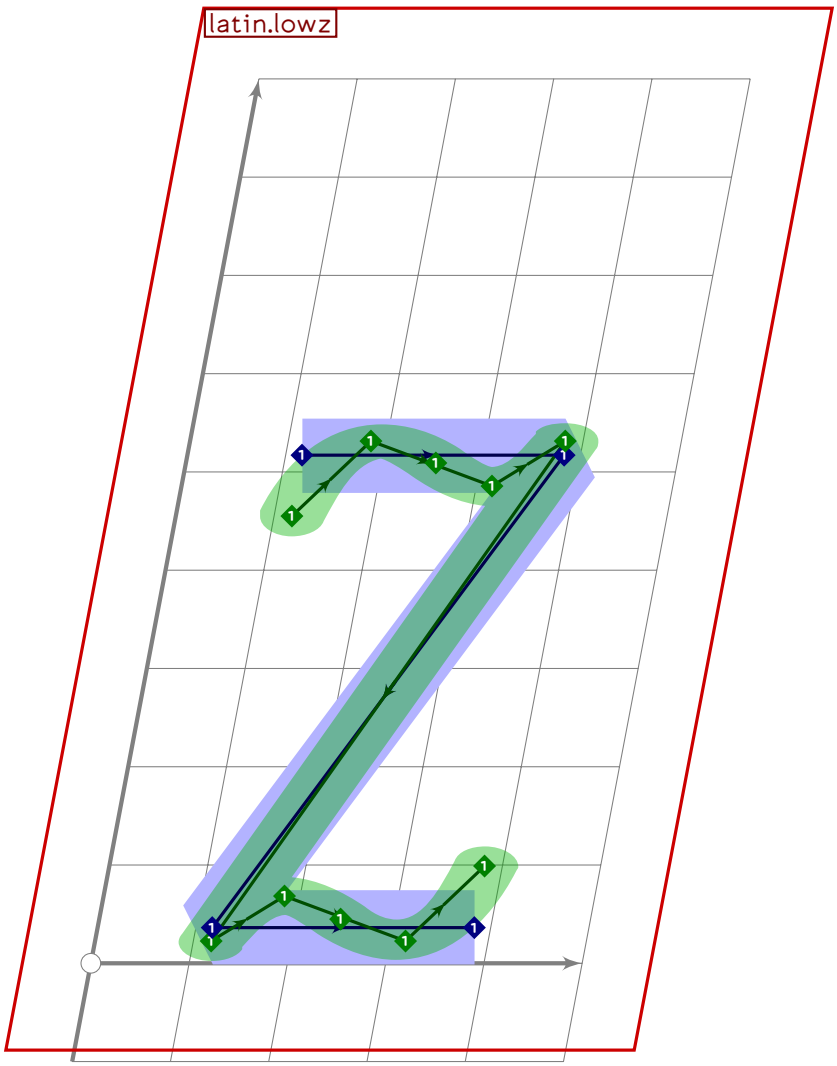


PROO

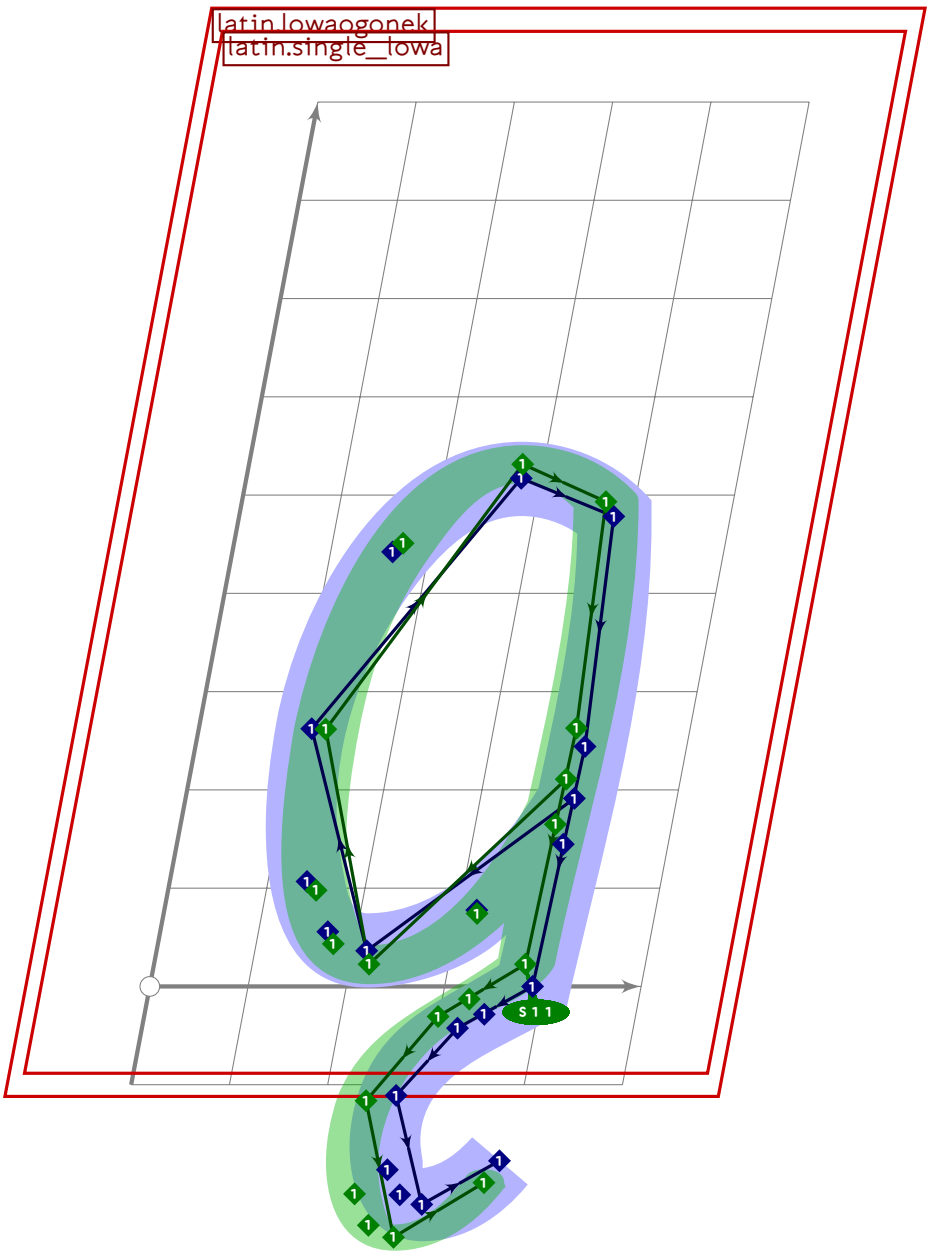
U+0079
tsuita.y



PROO



U+0105
tsuita.aogonek



PROO