

作りましょう 0.7

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 0.7

Parametric Font Family  
Proofs and pretty-printed  
source code

Matthew Skala

[mskala@ansuz.sooke.bc.ca](mailto:mskala@ansuz.sooke.bc.ca)

2013年3月7日

March 7, 2013

Proofs and pretty-printed source code for Tsukurimashou  
Copyright © 2011, 2012 Matthew Skala

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, version 3.

As a special exception, if you create a document which uses this font, and embed this font or unaltered portions of this font into the document, this font does not by itself cause the resulting document to be covered by the GNU General Public License. This exception does not however invalidate any other reasons why the document might be covered by the GNU General Public License. If you modify this font, you may extend this exception to your version of the font, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

# Contents

I Infrastructure	13
preintro.mp	15
Infrastructure . . . . .	15
Font Parameter Defaults . . . . .	16
tsuku-bk.mp	18
tsuku-kg.mp	19
tsuku-mg.mp	20
tsuku-mi.mp	21
tsuku-ps.mp	22
tsuku-bq.mp	23
tsuku-dq.mp	24
tsuku-el.mp	25
tsuku-eq.mp	26
tsuku-lw.mp	27
intro.mp	28
fntbase.mp	46
General Library Functions . . . . .	46
Prefix And Suffix Handling . . . . .	53
A Module That Finds An Envelope Of A Path Drawn With A Pen . . . . .	55
Postscript Font Generation . . . . .	67
obstack.mp	89
Object Stack Data . . . . .	89
Object Stack Methods . . . . .	90
frac-intro.mp	95
latin-intro.mp	97
accent.mp	101
bcircle.mp	112
buildkanji.mp	115
dakuten.mp	131
enclosed.mp	132
genjimon.mp	134
hiragana.mp	146
Hiragana Vowels . . . . .	146
Hiragana Kakikukeko/Gagigugego . . . . .	151
Hiragana Sashisuseso/Zajizuzezo . . . . .	155
Hiragana Tachitsuteto/Dajizudedo . . . . .	161

Hiragana Naninuneno . . . . .	166
Hiragana Hahifuheho/Babibubebo/Papipupepo . . . . .	171
Hiragana Mamimumemo . . . . .	176
Hiragana Yayuyo . . . . .	182
Hiragana Rarirurero . . . . .	184
Hiragana Wawiwewo/N/Iteration . . . . .	190
iching.mp	198
katakana.mp	200
Katakana Vowels . . . . .	200
Katakana Kakikukeko/Gagigugego . . . . .	204
Katakana Sashisuseso/Zajizuzezo . . . . .	209
Katakana Tachitsuteto/Dajizudedo . . . . .	214
Katakana Naninuneno . . . . .	219
Katakana Hahifuheho/Babibubebo/Papipupepo . . . . .	224
Katakana Mamimumemo . . . . .	230
Katakana Yayuyo . . . . .	234
Katakana Rarirurero . . . . .	237
Katakana Wawiwewo/N/Iteration . . . . .	243
latin.mp	249
numerals.mp	338
ogonek.mp	349
punct.mp	361
serif.mp	403
 II Shared kyouiku kanji	 409
gradeone.mp	411
gradetwo.mp	478
gradethree.mp	551
gradefour.mp	590
gradefive.mp	619
gradesix.mp	631
 III Other shared kanji	 641
bottomrad.mp	643
leftrad.mp	646
radical.mp	654
rightrad.mp	689
toprad.mp	693
gradeeight.mp	698

gradenine.mp	732
rare.mp	759

IV U+0000 to U+2FFF	795
tsuku-00.mp	797
Ascii . . . . .	797
Latin-1 Extra Characters . . . . .	891
Accented Latin . . . . .	912
tsuku-01.mp	970
Latin Extended A Uppercase . . . . .	970
Latin Extended A Lowercase . . . . .	1025
Latin Extended A Other . . . . .	1083
tsuku-02.mp	1085
Latin Extended B . . . . .	1085
Spacing Modifier Letters . . . . .	1086
tsuku-03.mp	1095
Combining Diacritical Marks . . . . .	1095
tsuku-20.mp	1110
General Punctuation . . . . .	1110
tsuku-21.mp	1127
Symbols Required By Mes-1 . . . . .	1127
tsuku-24.mp	1133
Circled Numerals . . . . .	1133
Circled Latin And Zero . . . . .	1153
Inverted Circled Numerals . . . . .	1206
Doubly Circled Numerals . . . . .	1216
One More Inverted Circled Numeral . . . . .	1226
tsuku-25.mp	1228
Geometric Shapes . . . . .	1228
tsuku-26.mp	1230
I Ching . . . . .	1230
tsuku-27.mp	1247
Inverted Circled Numerals . . . . .	1247
tsuku-2e.mp	1258
Cjk Radicals Supplement . . . . .	1258
tsuku-2f.mp	1295
Kangxi Radicals . . . . .	1295
Ideographic Description Characters . . . . .	1415

V	U+3000 to U+4DFF	1429
	tsuku-30.mp	1431
	Ideographic Symbols And Punctuation . . . . .	1431
	Hiragana . . . . .	1451
	Katakana . . . . .	1496
	tsuku-31.mp	1543
	Phonetic Extensions For Ainu . . . . .	1543
	tsuku-32.mp	1560
	Circled Numerals . . . . .	1560
	Circled Katakana . . . . .	1590
	tsuku-34.mp	1638
	tsuku-4d.mp	1639
	I Ching . . . . .	1639
VI	U+4E00 to U+61FF	1705
	tsuku-4e.mp	1707
	tsuku-4f.mp	1744
	tsuku-50.mp	1798
	tsuku-51.mp	1808
	tsuku-52.mp	1820
	tsuku-53.mp	1850
	tsuku-54.mp	1873
	tsuku-55.mp	1885
	tsuku-56.mp	1891
	tsuku-57.mp	1898
	tsuku-58.mp	1905
	tsuku-59.mp	1911
	tsuku-5a.mp	1924
	tsuku-5b.mp	1926
	tsuku-5c.mp	1957
	tsuku-5d.mp	1981
	tsuku-5e.mp	1989
	tsuku-5f.mp	2024
	tsuku-60.mp	2069
	tsuku-61.mp	2118
VII	U+6200 to U+75FF	2125
	tsuku-62.mp	2127
	tsuku-63.mp	2143

tsuku-65.mp	2149
tsuku-66.mp	2165
tsuku-67.mp	2184
tsuku-68.mp	2199
tsuku-69.mp	2206
tsuku-6a.mp	2212
tsuku-6b.mp	2216
tsuku-6c.mp	2225
tsuku-6d.mp	2275
tsuku-6e.mp	2288
tsuku-6f.mp	2299
tsuku-70.mp	2304
tsuku-71.mp	2309
tsuku-72.mp	2311
tsuku-73.mp	2320
tsuku-74.mp	2321
tsuku-75.mp	2325

VIII U+7600 to U+89FF	2337
tsuku-76.mp	2339
tsuku-77.mp	2346
tsuku-78.mp	2356
tsuku-79.mp	2358
tsuku-7a.mp	2369
tsuku-7b.mp	2376
tsuku-7c.mp	2389
tsuku-7d.mp	2392
tsuku-7f.mp	2408
tsuku-80.mp	2413
tsuku-81.mp	2421
tsuku-82.mp	2426
tsuku-83.mp	2457
tsuku-84.mp	2469
tsuku-85.mp	2474
tsuku-86.mp	2476
tsuku-88.mp	2477
tsuku-89.mp	2482

IX	U+8A00 to U+9FFF	2487
	tsuku-8a.mp	2489
	tsuku-8b.mp	2544
	tsuku-8c.mp	2547
	tsuku-8d.mp	2555
	tsuku-8e.mp	2559
	tsuku-8f.mp	2564
	tsuku-90.mp	2576
	tsuku-91.mp	2595
	tsuku-92.mp	2606
	tsuku-93.mp	2610
	tsuku-95.mp	2613
	tsuku-96.mp	2630
	tsuku-97.mp	2652
	tsuku-98.mp	2656
	tsuku-99.mp	2667
	tsuku-9a.mp	2671
	tsuku-9b.mp	2675
	tsuku-9c.mp	2676
	tsuku-9e.mp	2678
	tsuku-9f.mp	2681
X	U+A000 to U+10FFFF	2683
	tsuku-f7.mp	2685
	Latin Small Caps . . . . .	2685
	tsuku-ff.mp	2712
	Full-Width Forms . . . . .	2712
	Half-Width Punctuation . . . . .	2738
	Half-Width Katakana . . . . .	2741
	tsuku-1f1.mp	2805
	Squared Latin . . . . .	2805
	Inverse Circled Latin . . . . .	2831
	Inverse Squared Latin . . . . .	2857
	tsuku-200.mp	2884
	tsuku-21c.mp	2885
	tsuku-f17.mp	2886
	Combining Dots For I Ching . . . . .	2886
	Miscellaneous . . . . .	2895
	Tomoe Ornaments . . . . .	2901



Heavy Metal Umlaut . . . . .	2909
Genjimon . . . . .	2926
tsuku-ff0.mp . . . . .	2980
Fraction Numerators . . . . .	2980
tsuku-ff1.mp . . . . .	2987
Fraction Denominators . . . . .	2987
 XI Jieubsida core . . . . .	 2989
hangul.mp . . . . .	2991
Jamo Combining Operations . . . . .	2993
jamo-basic.mp . . . . .	3001
Filler Jamo . . . . .	3001
Sios/Cieuc/Chieuch Family . . . . .	3001
Kiyeok . . . . .	3004
Nieun . . . . .	3005
Tikeut . . . . .	3007
Rieul . . . . .	3009
Mieum . . . . .	3010
Pieup . . . . .	3011
Sios . . . . .	3012
Ieung . . . . .	3012
Cieuc . . . . .	3013
Chieuch . . . . .	3014
Khieukh . . . . .	3015
Thieuth . . . . .	3016
Phieuph . . . . .	3017
Hieuh . . . . .	3019
Mixed Tails . . . . .	3020
Vowels . . . . .	3021
jamo-extra.mp . . . . .	3030
Pansios . . . . .	3033
Yesieung . . . . .	3033
Yeorinhieuh . . . . .	3034
Chitueum And Ceongchieum Variants . . . . .	3036
Kapyeoun Variants . . . . .	3037
hglextb.mp . . . . .	3038
Hangul Extension B . . . . .	3038
Hangul Jungseong (Vowel) Jamo Extension B . . . . .	3038
Hangul Jongseong (Tail) Jamo Extension B . . . . .	3061

jieub-bt.mp	3111
jieub-do.mp	3112
jieub-sm.mp	3113
hglpage.mp	3114
jieub-11.mp	3116
Hangul Choseong (Lead) Jamo . . . . .	3116
Hangul Jungseong (Vowel) Jamo . . . . .	3211
Hangul Jongseong (Tail) Jamo . . . . .	3282
jieub-31.mp	3371
Hangul Compatibility Jamo . . . . .	3371
jieub-a9.mp	3465
Hangul Choseong (Lead) Jamo Extended A . . . . .	3465
jieub-ac.mp	3495
 XII Jieubsida alternates	 3549
jieub-ff2.mp	3551
Hangul Jungseong (Vowel) Jamo . . . . .	3551
Hangul Jungseong (Vowel) Jamo Extension B . . . . .	3622
jieub-ff3.mp	3646
Hangul Choseong (Lead) Jamo . . . . .	3646
jieub-ff4.mp	3771
Hangul Choseong (Lead) Jamo . . . . .	3771
jieub-ff5.mp	3896
Hangul Choseong (Lead) Jamo . . . . .	3896
jieub-ff6.mp	4021
Hangul Choseong (Lead) Jamo . . . . .	4021
jieub-ff7.mp	4146
Hangul Choseong (Lead) Jamo . . . . .	4146
Hangul Choseong (Lead) Jamo Extended A . . . . .	4241
 XIII TsuIta	 4271
tsuita-common.mp	4273
tsuita-at.mp	4284
tsuita-so.mp	4285
Additional Proofs	4287
 XIV Blackletter Lolita	 4341
bll.mp	4343

bll-co.mp	4348
pentacross.mp	4349
Utilities For Pentagrams And Crosses . . . . .	4349
bll-f5c.mp	4351
Pentagrams . . . . .	4351



Volume XIII

TsuIta

tsuita-common.mp

**COMM**

tsuita-at.mp

**AT**

tsuita-so.mp

**SO**

Additional Proofs

**PROO**



## tsuita-common.mp

COMM

```
1 %
2 % Common code for TsuIta
3 % Copyright (C) 2012 Matthew Skala
4 %
5-29 [Standard copyright notice]
30
31 inclusion_lock(tsuitacommon);
32
33 

---


34
35 familyname="TsuIta";
36
37 rescale_slant:=0;
38
39 if (pagename="00")
40 or (pagename="01")
41 or (pagename="02")
42 or (pagename="03")
43 or (pagename="20")
44 or (pagename="21")
45 or (pagename="24")
46 or (pagename="25")
47 or (pagename="26")
48 or (pagename="27")
49 or (pagename="32")
50 or (pagename="4d")
51 or (pagename="f7")
52 or (pagename="ff")
53 or (pagename="1f1")
54 or (pagename="f17")
55 or (pagename="ff0")
56 or (pagename="ff1"):
57   rescale_slant:=190;
58 fi;
59
60 tsu_punct_size:=tsu_punct_size*0.85;
61
62 boolean genji_rounded;
63 genji_rounded:=true;
64
65 

---


66
67 vardef latin.lowa =
68   latin.single_lowa;
69 enddef;
70
```

```

71 vardef latin.lowaogonek =
72   push_pbox_toexpand("latin.lowaogonek");
73   latin.lowa;
74
75   y9=0.5[y10,y1];
76   y10=latin_wide_desc_r;
77   y11=0.2[y10,y1];
78
79   x11-x9=(x1-x4)*((y1-y10)/(y3-y1));
80   x10=0.4[x9,x11];
81   x11=x1;
82
83   replace_strokep(0)(oldp{dir 210}{.z9..z10{right}{.z11};
84   replace_strokep(0)(insert_nodes(oldp)(length(oldp)-2.5));
85   replace_strokeq(0)(oldq-(1.4,1.4)-(1.3,1.3)-(1.4,1.4)-(1,1));
86   set_botip(0,length(get_strokep(0))-4,1);
87   expand_pbox;
88 enddef;
89
90 vardef latin.lowe =
91   push_pbox_toexpand("latin.lowe");
92   y2=0.57[y5,y3];
93   y3=latin_wide_xheight_r;
94   y4=0.49[y5,y3];
95   y5=latin_wide_low_r;
96   y6=0.35[y5,y2];
97   y0=0.7[y2,y3];
98
99   (x2+x4)/2=500;
100  (x2-x4)=0.86*(y3-y5);
101  x3=0.49[x4,x2];
102  x5=0.52[x4,x2];
103  x6=1.04[x4,x2]-(if sharp_corners: 0 else: (mbrush_width/3) fi);
104  x0=0.74[x3,x2];
105
106  push_stroke(z0..z3{left}{.z4{down}{.z5{right}{.z6,
107    (1.6,1.6)-(1.6,1.6)-(1.4,1.4)-(1.6,1.6)-
108    (1.6,1.6)-(1.6,1.6)-(1.6,1.6));
109  z1=get_strokep(0) intersectionpoint ((z2+(-500,-120))-z2+(-10,0));
110  replace_strokep(0)(insert_nodes((z1+(2,2)){curl 0}{.
111    tension 1.3..oldp)(0.5));
112  replace_strokep(0)((point 0 of oldp){curl 0}{.
113    ((0,-20)+point 1 of oldp)..
114    (subpath (2,infinity) of oldp));
115  expand_pbox;
116 enddef;
117
118 vardef punct.atsign =

```



```

119 push_pbox_toexpand("punct.atsign");
120 begingroup
121   save xsp,ysp;
122   xsp:=sp;
123   latin.lowa;
124   set_boserif(0,6,whatever);
125   set_botip(0,6,0);
126   ysp:=sp;
127
128   numeric x[],y[];
129   x1-x2=x2-x3=y2-y1;
130   x2=x4=500;
131   y1=y3=0.49[y4,y2];
132   y2=latin_wide_high_r;
133   y4=latin_wide_low_r;
134
135   transform shrinka;
136   (0.5[lcorner get_strokep(0),urcorner get_strokep(0)])
137     transformed shrinka=0.5[z3,z1];
138   (0.5[lrcorner get_strokep(0),urcorner get_strokep(0)])
139     transformed shrinka=0.71[z3,z1];
140   (0.5[ulcorner get_strokep(0),urcorner get_strokep(0)])
141     transformed shrinka=z2+(0.07-1)*0.29*(x1-x3);
142   sp:=xsp;
143   tsu_xform(shrinka shifted (-10,0))(sp:=ysp);
144
145   z5=point infinity of get_strokep(0);
146   y6=ypart lrcorner get_strokep(0);
147   x6=0.8[x2,x1];
148   replace_strokep(0)((subpath (0,length(oldp)-1) of oldp)
149     ..{curl 1}z5{curl 1}..z6..
150     (subpath (0,3.85) of (z1..z2..z3..z4..cycle)));
151   replace_strokep(0)(insert_nodes(oldp)((length oldp-4.5)));
152   replace_strokeq(0)(oldq-(1.6,1.6)-(1.6,1.6)-(1.6,1.6)-
153     (1.6,1.6)-(1.6,1.6)-(0,0));
154 endgroup;
155 expand_pbox;
156 endif;
157
158
159
160 if do_italic_hook:
161
162 vardef latin.lowf =
163   push_pbox_toexpand("latin.lowf");
164   (x2-x1)=290;
165   x5=x6=490=0.52[x1,x2];
166   x3-x5=2*(y4-y5);

```

```

167 x4=0.38[x5,x3];
168 x6-x7=1.15*(x4-x5);
169 x6-x8=1.15*(x3-x5);
170
171 y1=y2=0.75[latin_wide_low_h,latin_wide_xheight_h];
172 y5=0.52[latin_wide_xheight_h,y4];
173 y3=0.73[latin_wide_xheight_h,y4];
174 y4=latin_wide_high_r;
175 y7=latin_wide_desc_r;
176 y6-y7=1.10*(y4-y5);
177 y8-y7=1.10*(y4-y3);
178
179 push_stroke(z1-z2,(1.6,1.6)-(1.6,1.6));
180
181 push_stroke(z3{curl 0.6}..z4{left}..
182     {dir 268}z5{down}-z6{dir 268}..
183     {left}z7.{curl 0.6}z8,
184     (1.6,1.6)-(1.6,1.6)-(1.6,1.6)-(1.6,1.6)-(1.6,1.6)-(1.6,1.6));
185 replace_strokep(0)(subpath (0.23,4.77) of oldp);
186 expand_pbox;
187 enddef;
188
189 vardef latin.lowi =
190     push_pbox_toexpand("latin.lowi");
191     x2=x3=x4=450;
192
193     y2=latin_wide_xheight_r;
194     y3=latin_wide_low_r;
195     y4=0.5[y2,latin_wide_high_r]+mbrush_width;
196
197     push_stroke(z2-z3,(1.6,1.6)-(1.6,1.6));
198     set_boserif(0,1,11);
199
200     push_lcblob(fullcircle rotated 45 scaled (mbrush_width*2.7+15)
201         shifted (z4 transformed tsu_rescale_xform)
202         transformed inverse tsu_rescale_xform);
203     expand_pbox;
204 enddef;
205
206 vardef latin.lowj =
207     push_pbox_toexpand("latin.lowj");
208     x3=x5=450;
209     (x2-x5)=0.25*italic_hook_radius;
210
211     y2=latin_wide_xheight_r;
212     y3=latin_wide_low_v;
213     y5=0.5[y2,latin_wide_high_r]+mbrush_width;
214

```

```

215  z4=z3+(-210,-140);
216
217  push_stroke((z2-(z3+(0,55))).{curl 0.8}z4,
218    (1.6,1.6)-(1.6,1.6)-(1.6,1.6));
219  set_boserif(0,0,11);
220
221  push_lcblob(fullcircle scaled (mbrush_width*2.7+15)
222    shifted (z5 transformed tsu_rescale_xform)
223    transformed inverse tsu_rescale_xform);
224  expand_pbox;
225 endif;
226
227 vardef latin.lowv =
228   push_pbox_toexpand("latin.lowv");
229   0.4[x1,x4]=x3=480;
230   x1=x2;
231   x5=0.8[x1,x4];
232
233   y1=y5=latin_wide_xheight_v;
234   y3=latin_wide_low_h;
235   y2=0.333[y3,y1];
236   y4=0.45[y3,y5];
237
238   (x4-x1)=(y1-y3)*0.65;
239
240   push_stroke(z1..z2.{dir -30}z3{dir 30}..z4..z5,
241     (1.6,1.6)-(1.6,1.6)-(1.6,1.6)-(1.6,1.6)-(1.6,1.6));
242   set_botip(0,2,0);
243   set_boserif(0,0,11);
244
245   expand_pbox;
246 endif;
247
248 vardef latin.loww =
249   push_pbox_toexpand("latin.loww");
250   0.4[x1,x4]=x3;
251   x1=x2;
252   x5=x4;
253   0.4[x5,x8]=x7;
254   x5=x6;
255   x9=0.8[x5,x8];
256   0.5[x1,x8]=500;
257
258   y1=y5=y9=latin_wide_xheight_v;
259   y3=y7=latin_wide_low_h;
260   y2=y6=0.333[y3,y1];
261   y4=y8=0.45[y3,y5];
262

```

```

263 (x4-x1)=(y1-y3)*0.55;
264 (x8-x5)=(y1-y3)*0.60;
265
266 push_stroke(z1..z2.{dir -30}z3{dir 30}..z4..z5,
267   (1.6,1.6)-(1.6,1.6)-(1.6,1.6)-(1.6,1.6)-(1.6,1.6));
268 set_botip(0,2,0);
269 set_boserif(0,0,11);
270
271 push_stroke(z5..z6.{dir -30}z7{dir 30}..z8..z9,
272   (1.6,1.6)-(1.6,1.6)-(1.6,1.6)-(1.6,1.6));
273 replace_strokep(0)(subpath (xpart
274   (oldp intersectiontimes get_strokep(-1)),infinity) of oldp);
275
276 expand_pbox;
277 enddef;
278
279 vardef latin.lowy =
280   push_pbox_toexpand("latin.lowy");
281   (x1+x3)/2=(x2+x4)/2=510;
282   (x2+x3-x1-x4)=((y1-y2)*0.58)*2;
283   (x3-x1)=(x2-x4)*0.93;
284   x5=x4-0.1*(x2-x4);
285
286   y1=y3=latin_wide_xheight_v;
287   y2=y4;
288   y5=0.5[y4,latin_wide_low_h]=latin_wide_desc_h;
289
290   z6=0.66[z1,z3];
291
292   push_stroke(z1-z2,(1.6,1.6)-(1.6,1.6));
293
294   push_stroke(z6..
295     (0.3[z3,z4]..tension 5..(0.6[z3,z4])..tension 0.8 and 3.{left}z5,
296     (1.6,1.6)-(1.6,1.6)-(1.6,1.6)-(1.6,1.6));
297
298   numeric xchgtime;
299   xchgtime:=ypart (get_strokep(-1) intersectiontimes get_strokep(0));
300
301   replace_strokep(-1)(z1-subpath (xchgtime,infinity) of get_strokep(0));
302   replace_strokeq(-1)
303     ((1.6,1.6)-subpath (xchgtime,infinity) of get_strokeq(0));
304
305   replace_strokep(0)(subpath (0,xchgtime) of oldp);
306   replace_strokeq(0)(subpath (0,xchgtime) of oldq);
307
308   set_boserif(-1,0,11);
309   set_botip(-1,1,1);
310   set_boalternate(0);

```

```

311
312 if do_alteration:
313     replace_stroke(-1)(oldp shifted (alternate_adjust*left/2));
314     replace_stroke(0)(oldp shifted (alternate_adjust*right/2));
315 fi;
316 expand_pbox;
317 enddef;
318
319 vardef latin.lowz =
320     push_pbox_toexpand("latin.lowz");
321     y2=y4=latin_wide_xheight_h;
322     y5=y7=latin_wide_low_h;
323     y2-y1=y8-y7=0.15*(y4-y5);
324     y4-y3=y6-y5=0.09*(y4-y5);
325
326     x1=x5;
327     x4=x8;
328     (x1+x4)/2=500;
329     (x4-x1)=(y4-y5)*0.92;
330     x2=x6=0.25[x1,x4];
331     x3=x7=0.75[x1,x4];
332
333     push_stroke(z1..z2..z3..z4-z5..z6..z7..z8,
334         (1,6,1,6)-(1,6,1,6)-(1,6,1,6)-(1,6,1,6)-
335         (1,6,1,6)-(1,6,1,6)-(1,6,1,6)-(1,6,1,6));
336     set_botip(0,4,0);
337     set_botip(0,5,0);
338     expand_pbox;
339 enddef;
340
341
342
343 def tsu_render_segment(expr i,p,q) =
344     begingroup
345         save start_time,end_time,niba,nibb,nibc,myp,pext,x,y;
346         numeric start_time,end_time,x,y;
347         path myp,niba,nibb,nibc,pext,glyph;
348
349         myp:=p;
350
351         niba:=fix_nib(obstackna.bosize[i]*tsu_brush_max
352             *tsu_brush_shape,
353             obstackna.bosize[i]*tsu_brush_max
354             *tsu_brush_shape,
355             0);
356
357         if known obstacknaa.boserif[i][floor(ltime[0]+0.5)]:
358             if obstacknaa.boserif[i][floor(ltime[0]+0.5)]=11:

```

```

359 start_time:=xpart (myp intersectiontimes (fullcircle
360   scaled (italic_hook_radius*(obstackna.bosize[i]/100)*2)
361   shifted (point 0 of myp)));
362 nibb:=tsu_brush_tip(0,myq,obstackna.bosize[i],s<1,
363   t>(length obstackp[i])-1,obstackba.boalternate[i]);
364 if do_alteration and obstackba.boalternate[i]:
365   niba:=nibb;
366   nibc:=nibb;
367 else:
368   niba:=nibb scaled tsu_brush_shape;
369   nibc:=nibb scaled sqrt(sqrt(tsu_brush_shape));
370 fi;
371 z1=(point 0 of myp)-
372   italic_hook_radius*(obstackna.bosize[i]/100)*dir(italic_hook_dir);
373 z3=point start_time of myp;
374 x2=0.5[xpart point 0 of myp,x1];
375 y2=ypart point 0 of myp;
376 pext:=z1{curl 0}..z2..{direction start_time of myp}z3;
377 pen_stroke(tip(niba)(0) tip(nibc)(1) tip(nibb)(2))(pext)(glyph);
378 glstk[nlgs]:=regenerate(glyph);
379 nlgs:=nlgs+1;
380 myp:=subpath (start_time,infinity) of myp;
381 fi;
382 fi;
383
384 if known obstacknaa.boserif[i][floor(ltime[length(myp)]+0.5)]:
385   if obstacknaa.boserif[i][floor(ltime[length(myp)]+0.5)]=11:
386     end_time:=(length myp)-xpart ((reverse myp)
387       intersectiontimes (fullcircle
388         scaled (italic_hook_radius*(obstackna.bosize[i]/100)*2)
389         shifted (point infinity of myp)));
390     nibb:=tsu_brush_tip(length(myp),myq,obstackna.bosize[i],s<1,
391       t>(length obstackp[i])-1,obstackba.boalternate[i]);
392   if do_alteration and obstackba.boalternate[i]:
393     niba:=nibb;
394     nibc:=nibb;
395   else:
396     niba:=nibb scaled tsu_brush_shape;
397     nibc:=nibb scaled sqrt(sqrt(tsu_brush_shape));
398   fi;
399   z4=point end_time of myp;
400   z6=(point infinity of myp)+
401     italic_hook_radius*(obstackna.bosize[i]/100)*dir(italic_hook_dir);
402   x5=0.5[xpart point infinity of myp,x6];
403   y5=ypart point infinity of myp;
404   pext:=z4{direction end_time of myp}..z5..{curl 0}z6;
405   pen_stroke(tip(nibb)(0) tip(nibc)(1) tip(niba)(2))(pext)(glyph);
406   glstk[nlgs]:=regenerate(glyph);

```

```

407     ngls:=ngls+1;
408     myp:=subpath (0,end_time) of myp;
409   fi;
410 fi;
411
412 if do_alteration and obstackba.boalterate[i]:
413   default_nib:=fix_nib(obstackna.bosize[i]*tsu_brush_max
414                         *tsu_brush_shape,
415                         obstackna.bosize[i]*tsu_brush_max
416                         *tsu_brush_shape,
417                         0);
418 else:
419   default_nib:=fix_nib(obstackna.bosize[i]*tsu_brush_max,
420                         obstackna.bosize[i]*tsu_brush_max
421                         *tsu_brush_shape,
422                         tsu_brush_angle);
423 fi;
424 path mytip[],glyph;
425 for l=0 step 1 until length(myp):
426   mytip[l]:=tsu_brush_tip(l,myp,q,obstackna.bosize[i],s<1,
427                           t>(length obstackp[i])-1,obstackba.boalterate[i]);
428 endfor;
429 pen_stroke(for l=0 step 1 until length(myp):
430   if sharp_corners and known obstacknaa.botip[i][ltime[l]]:
431     tip(obstacknaa.botip[i][ltime[l]])(l)
432   else:
433     tsu_brush_opt(mytip[l])(l)
434   fi
435 endfor)(myp)(glyph);
436 glstk[ngls]:=regenerate(glyph);
437 ngls:=ngls+1;
438 for l=0 step 1 until length(myp):
439   si:=floor (ltime[l]+0.5);
440   if (abs(ltime[l]-si)<0.05) and known obstacknaa.boserif[i][si]:
441     tsu_serif.choose(obstacknaa.boserif[i][si],
442                      point l of myp,direction l of myp,l,
443                      obstackna.bosize[i],tsu_brush_tip_size(l,q));
444     write ("SERIF "&(decimal obstacknaa.boserif[i][si])&" "&
445           (decimal xpart point l of myp)&","&
446           (decimal ypart point l of myp)) to "proof.prf";
447   fi;
448 endfor;
449 endgroup;
450 enddef;
451
452 def tsu_rescale_xform =
453   begingroup
454     save t,st,cp,xadj;

```

```

455 transform t,st;
456 numeric xadj;
457 st:=tsu_slant_xform;
458 t:=st;
459 % check if rescaling is active
460 if (rescale_from.left<>rescale_to.left)
461 or (rescale_from.right<>rescale_to.right): begingroup
462     save i,xa,xb,lf,rf,wf,lt,rt,wt;
463     numeric i,xa,xb,lf,rf,wf,lt,rt,wt;
464     transform t;
465     % find the bounds of the paths
466     if find_stroke(0)<=0:
467         xa:=0.5[rescale_from.left,rescale_from.right];
468         xb:=0.5[rescale_from.left,rescale_from.right];
469     else:
470         xa:=infinity;
471         xb:=-infinity;
472         for i=1 upto sp-1:
473             if obstacktype[i]=otstroke:
474                 xadj:=0;
475                 if known obstacknaa.boserif[i][0]:
476                     if obstacknaa.boserif[i][0]=11:
477                         xadj:=-((2/3)*italic_hook_radius*xpart dir italic_hook_dir;
478                     fi;
479                 fi;
480                 if (xadj+xpart llcorner obstackp[i])<xa:
481                     xa:=xadj+xpart llcorner obstackp[i];
482                 fi;
483                 xadj:=0;
484                 if known obstacknaa.boserif[i][length(obstackp[i])]:
485                     if obstacknaa.boserif[i][length(obstackp[i])]=11:
486                         xadj:=(2/3)*italic_hook_radius*xpart dir italic_hook_dir;
487                     fi;
488                 fi;
489                 if (xadj+xpart lrcorner obstackp[i])>xb:
490                     xb:=xadj+xpart lrcorner obstackp[i];
491                 fi;
492             fi;
493         endfor;
494     fi;
495     % compute bearings and widths
496     lf:=xa-rescale_from.left;
497     rf:=rescale_from.right-xb;
498     lf+rf+wf=rescale_from.right-rescale_from.left;
499     lt+rt+wt=rescale_to.right-rescale_to.left;
500     (lt,rt)=whatever[(0,0),(lf,rf)];
501     wt=ypart (width_curve intersectionpoint ((wf,-infinity)-(wf,infinity)));
502     % find transformation

```



```

503     if wf>0:
504         (rescale_from.right-rf,rescale_from.bottom) transformed t=
505         (rescale_to.right-rt,rescale_to.bottom+rescale_skew);
506         (rescale_from.left+lf,rescale_from.bottom) transformed t=
507         (rescale_to.left+lt,rescale_to.bottom-rescale_skew);
508         (rescale_from.left+lf,rescale_from.top) transformed t=
509         (rescale_to.left+lt,rescale_to.top-rescale_skew);
510     else:
511         (rescale_from.left+lf,rescale_from.bottom) transformed t=
512         (rescale_to.left+lt,rescale_to.bottom);
513         (rescale_from.left+lf,rescale_from.top) transformed t=
514         (rescale_to.left+lt,rescale_to.top);
515         (rescale_from.left+lf+1,rescale_from.bottom) transformed t=
516         (rescale_to.left+lt+1,rescale_to.bottom);
517     fi;
518     pair cp;
519     transform st;
520     cp:=((rescale_to.left+rescale_to.right)/2,rescale_to.bottom);
521     cp transformed st=cp;
522     cp+(100,0) transformed st=cp+(100,0);
523     cp+(0,100) transformed st=cp+(rescale_slant/10,100);
524     t:=t transformed st;
525     endgroup; fi;
526     t
527 endgroup
528 endif;
529
530 fi;

```

## tsuita-at.mp

AT

```
1 %
2 % TsuIta Atama
3 % Copyright (C) 2012 Matthew Skala
4 %
5-29 [Standard copyright notice]
30
31 % TSUITA ATAMA
32
33 input preintro.mp;
34
35 stylename:="Atama";
36
37 (0,4) transformed tsu_brush_xf = (4,0.77);
38 (1,1) transformed tsu_brush_xf = (1,0.73);
39 (4,0) transformed tsu_brush_xf = (0,0.77);
40
41 tsu_brush_min:=0.73;
42 tsu_brush_max:=0.77;
43
44 def tsu_brush_opt(expr n,l) =
45   if rescale_slant>10: cut(n,rel 90)(l) fi enddef;
46 def sharp_corners = (rescale_slant>10) enddef;
47
48 input intro.mp;
49
50 let old_rescale_half = tsu_rescale_half;
51
52 def tsu_rescale_half =
53   old_rescale_half;
54   width_curve:=((-1,-1)-(100,100))..(480,330)..{right}(2000,440);
55 enddef;
56
57 let old_rescale_half_lc = tsu_rescale_half_lc;
58
59 def tsu_rescale_half_lc =
60   old_rescale_half_lc;
61   width_curve:=((-1,-1)-(100,100))..(610,330)..{right}(2000,420);
62 enddef;
63
64 include_late("tsuita-common.mp");
65
66 % overridden per-page later
67 rescale_slant:=190;
```

## tsuita-so.mp

```
1 %
2 % TsuIta Soku
3 % Copyright (C) 2012 Matthew Skala
4 %
5-29 [Standard copyright notice]
30
31 % TSUITA SOKU
32
33 input preintro.mp;
34
35 stylename:="Soku";
36
37 mincho:=0.8;
38
39 (0,4) transformed tsu_brush_xf = (0.0,0.80);
40 (1,1) transformed tsu_brush_xf = (1.2,0.45);
41 (4,0) transformed tsu_brush_xf = (4.8,0.80);
42
43 tsu_brush_min:=0.45;
44 tsu_brush_max:=0.80;
45
46 tsu_brush_shape:=0.60;
47 tsu_brush_angle:=1;
48
49 tsu_pbrush_size:=45;
50 tsu_pbrush_shape:=1;
51 tsu_pbrush_angle:=1;
52
53 input serif.mp;
54
55 tsu_do_serif[1]:=true;
56 tsu_do_serif[2]:=true;
57 tsu_do_serif[3]:=true;
58
59 do_alteration:=true;
60
61 genji_outline:=true;
62
63 input intro.mp;
64
65 let old_render_segment = tsu_render_segment;
66
67 def tsu_render_segment(expr i,p,q) =
68   begingroup
69     save start_time,end_time;
70     numeric start_time,end_time;
```

SO

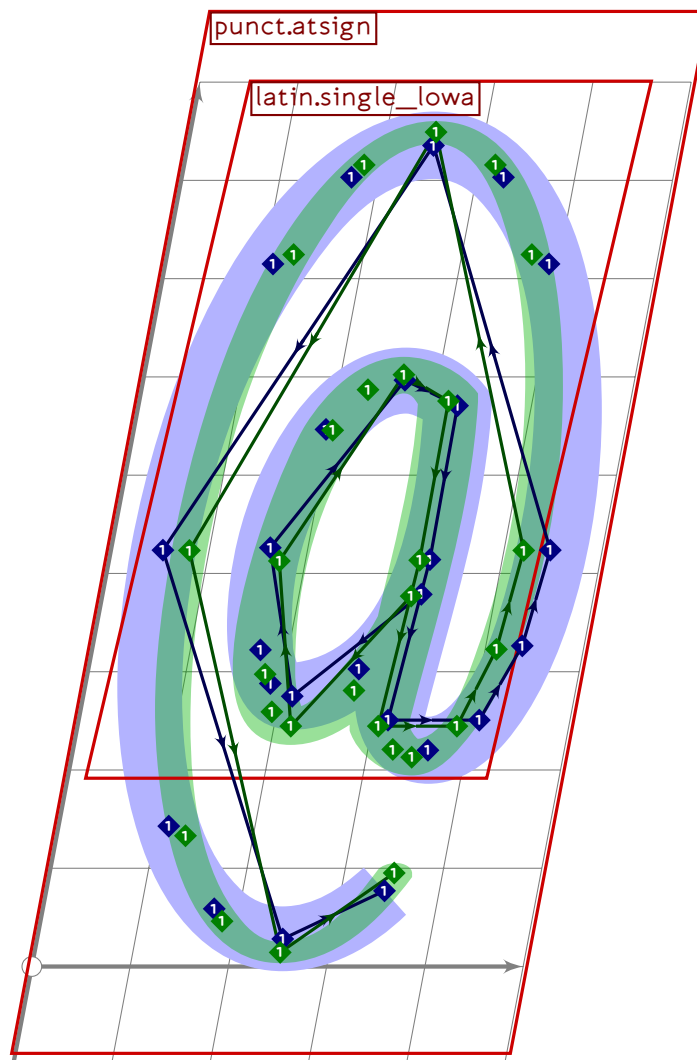
```

71  endgroup;
72  old_render_segment(i,p,q);
73  enddef;
74
75  let old_rescale_half = tsu_rescale_half;
76
77  def tsu_rescale_half =
78    old_rescale_half;
79    width_curve:=((-1,-1)-(100,100))..(560,310)..{right}(2000,400);
80  enddef;
81
82  let old_rescale_half_lc = tsu_rescale_half_lc;
83
84  def tsu_rescale_half_lc =
85    old_rescale_half_lc;
86    width_curve:=((-1,-1)-(100,100))..(610,310)..{right}(2000,400);
87  enddef;
88
89  do_italic_hook:=true;
90  italic_hook_radius:=170;
91  italic_hook_dir:=25;
92
93  include_late("tsuita-common.mp");
94
95  % overridden per-page later
96  rescale_slant:=190;

```

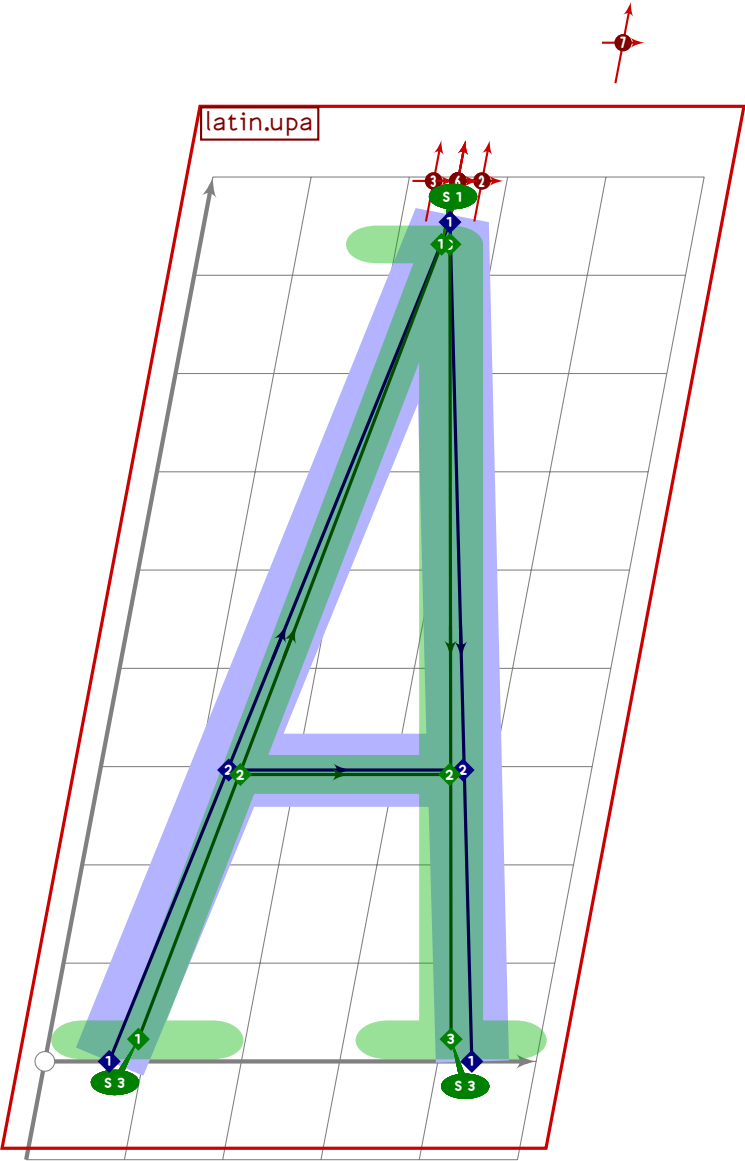
## Additional Proofs

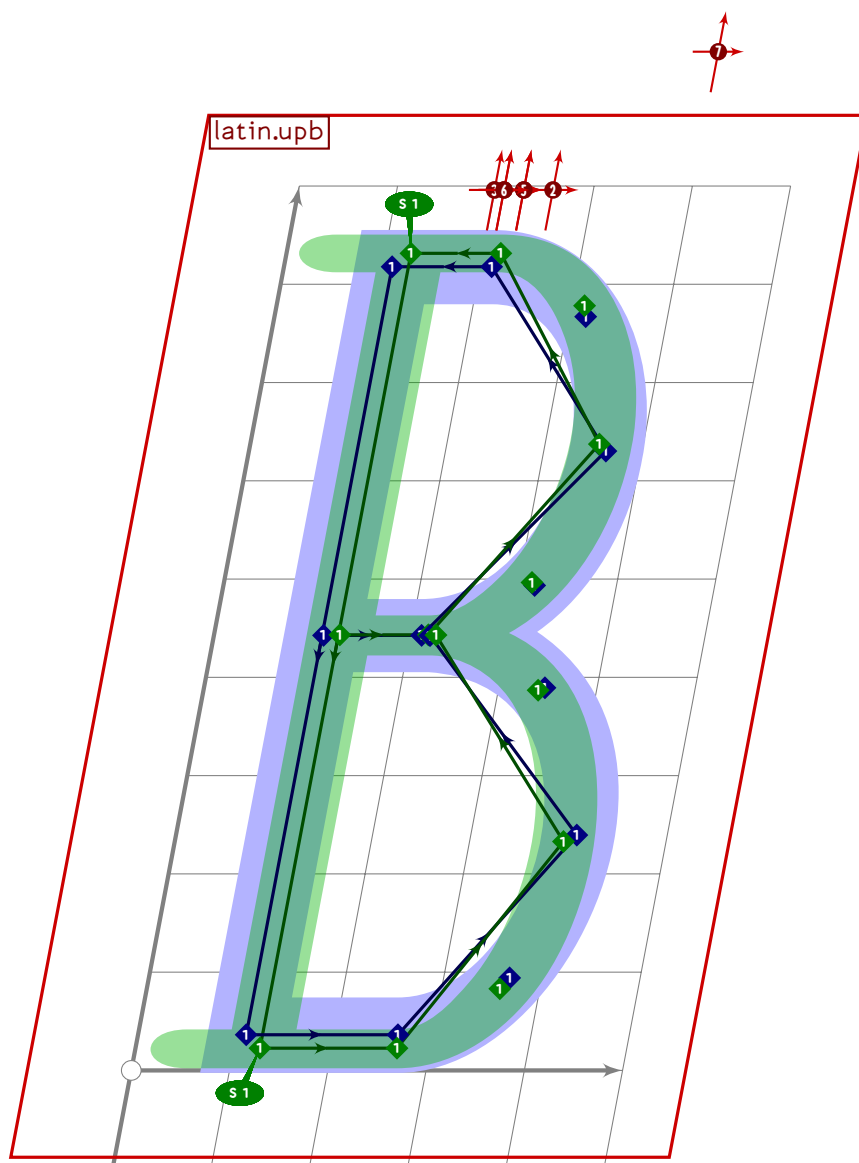
Most glyphs in TsuIta are drawn by code inherited from Tsukurimashou, and proof images for those glyphs are shown with the corresponding Tsukurimashou code. However, different parameters between Tsukurimashou and TsuIta, and overriding code in the tsuita-common.mp file above, cause many of the glyphs 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.



PROO

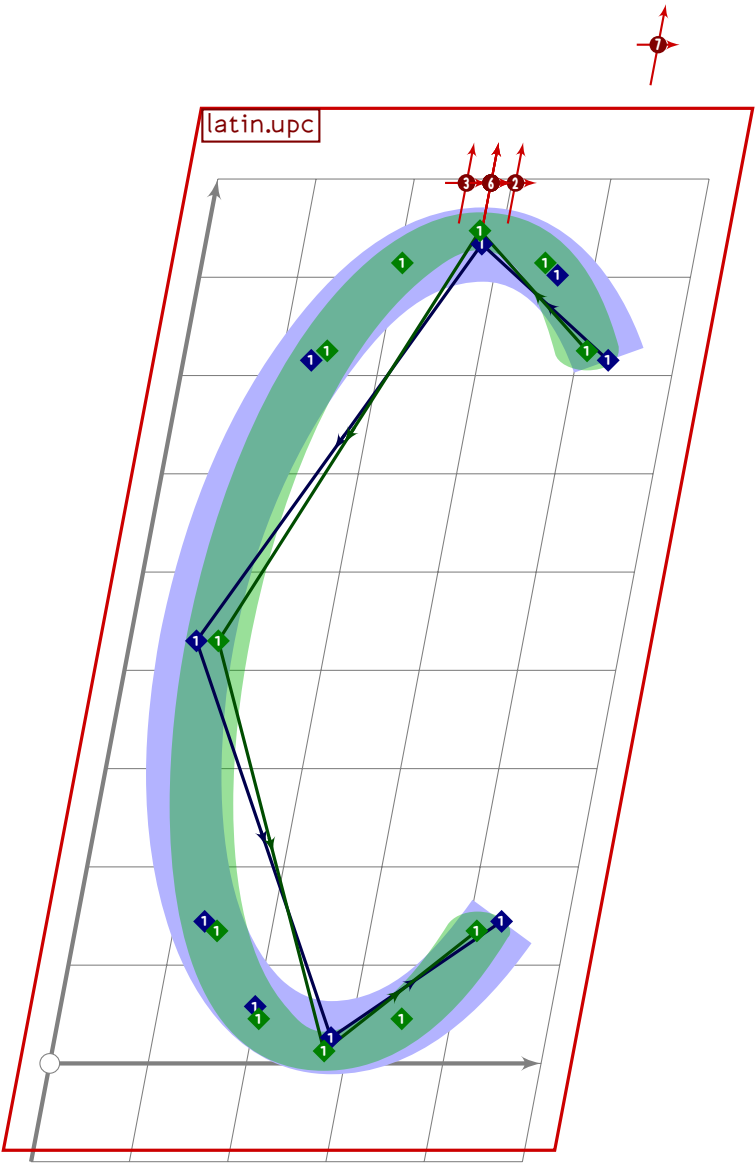
PROO



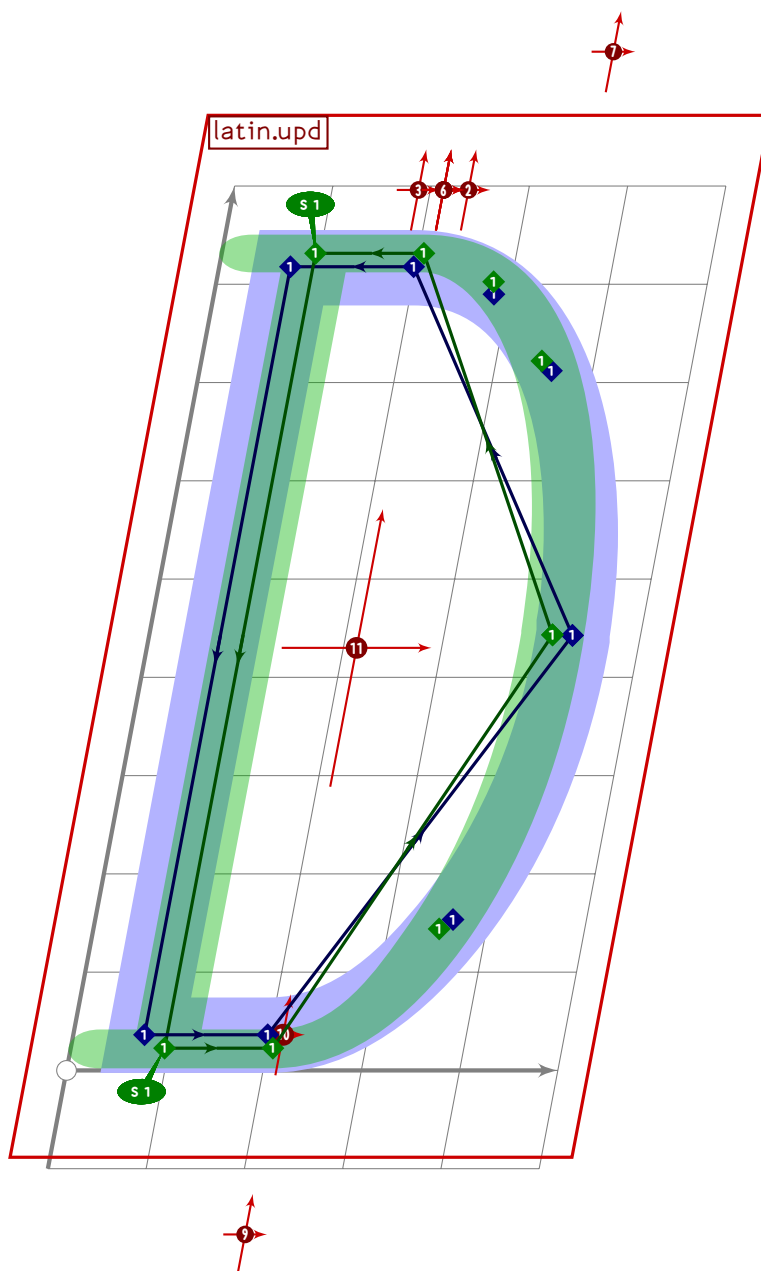


PROO

PROO

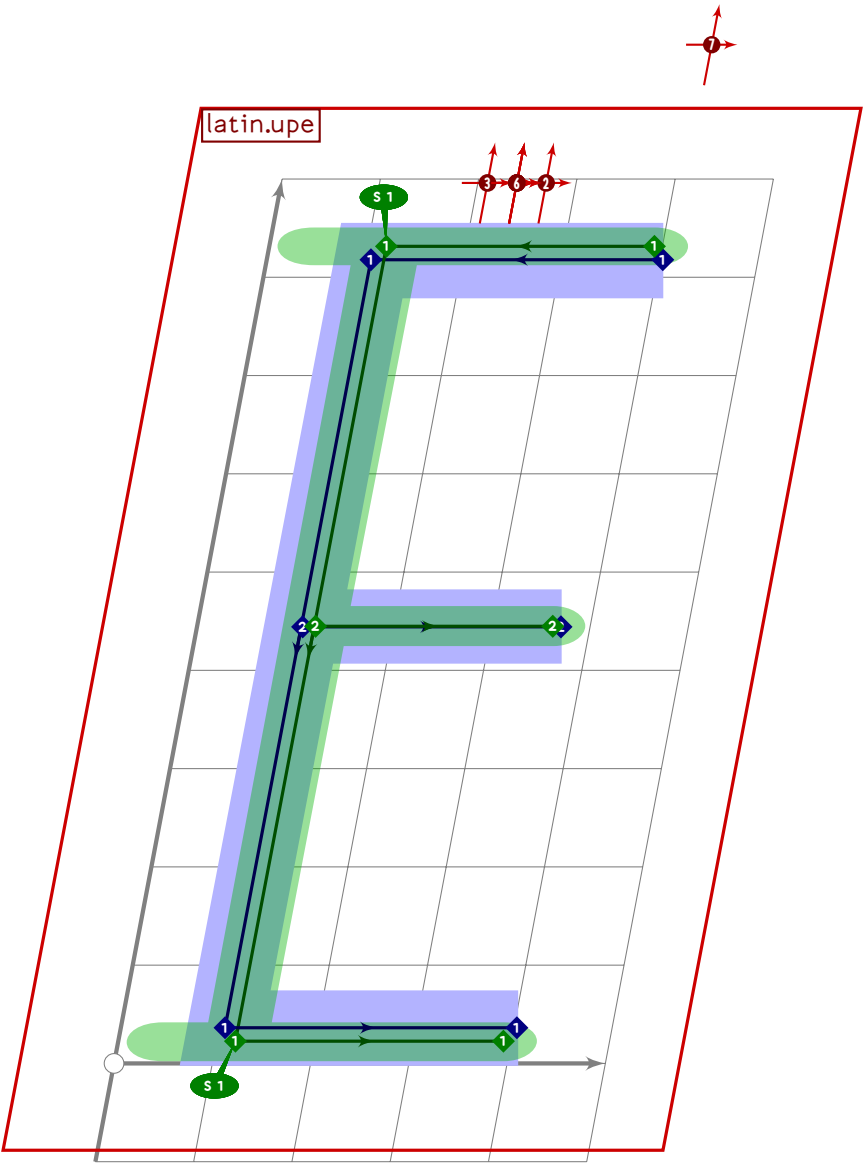


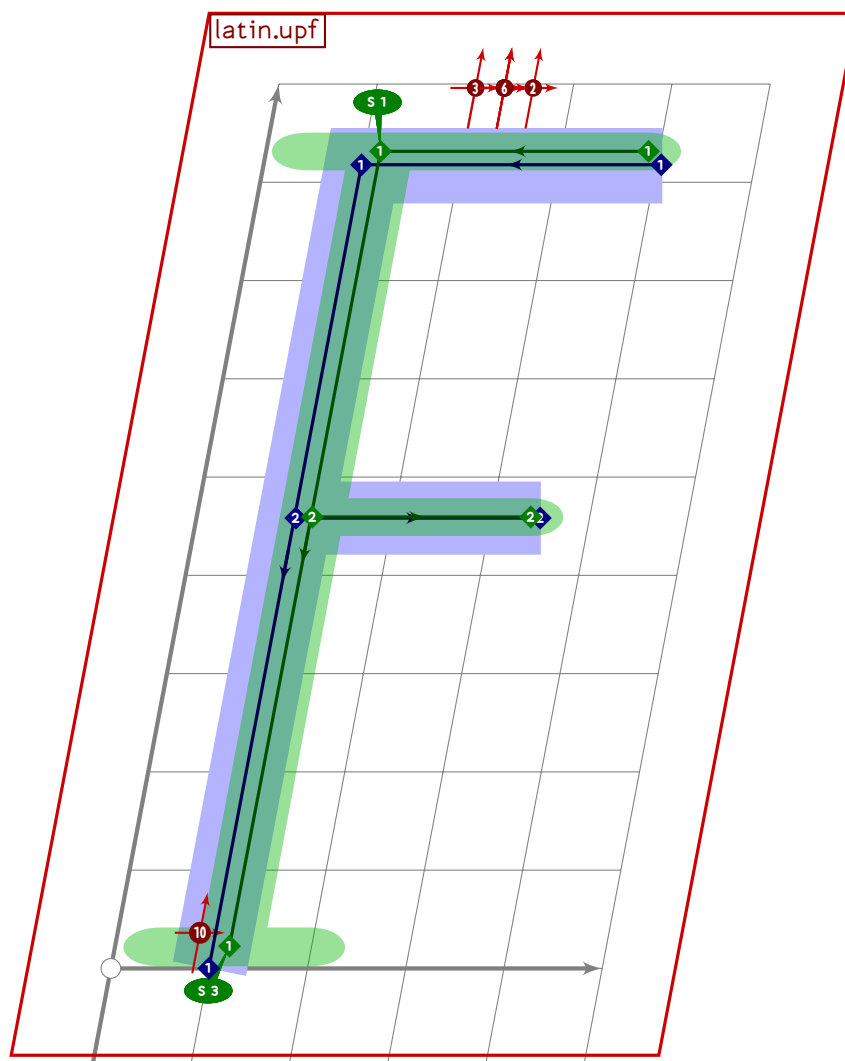




PROO

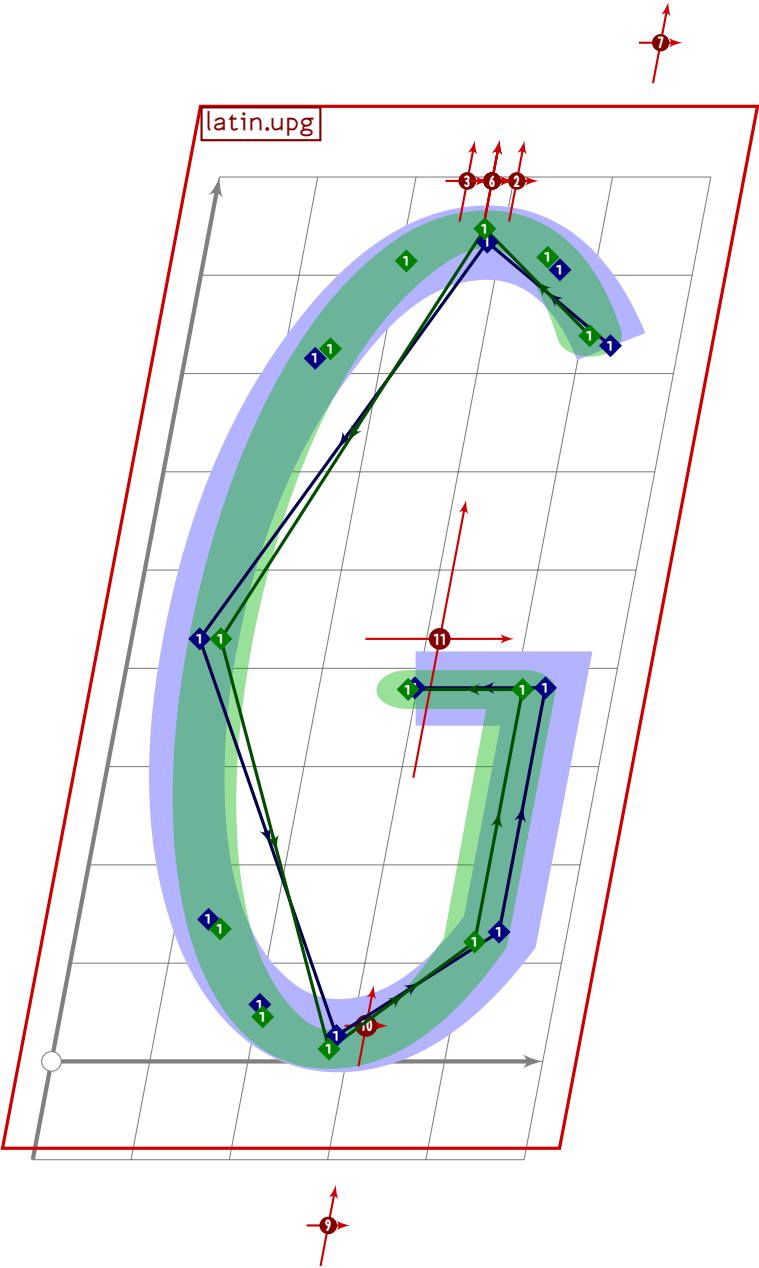
PROO

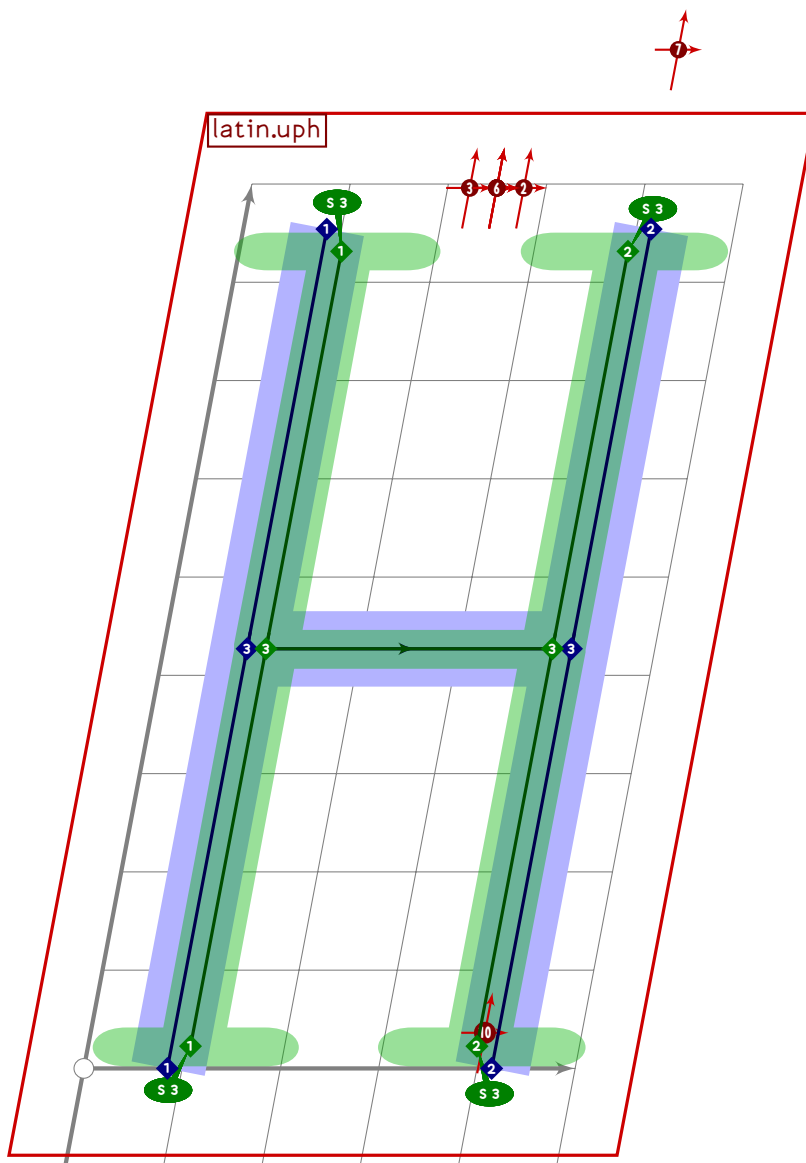




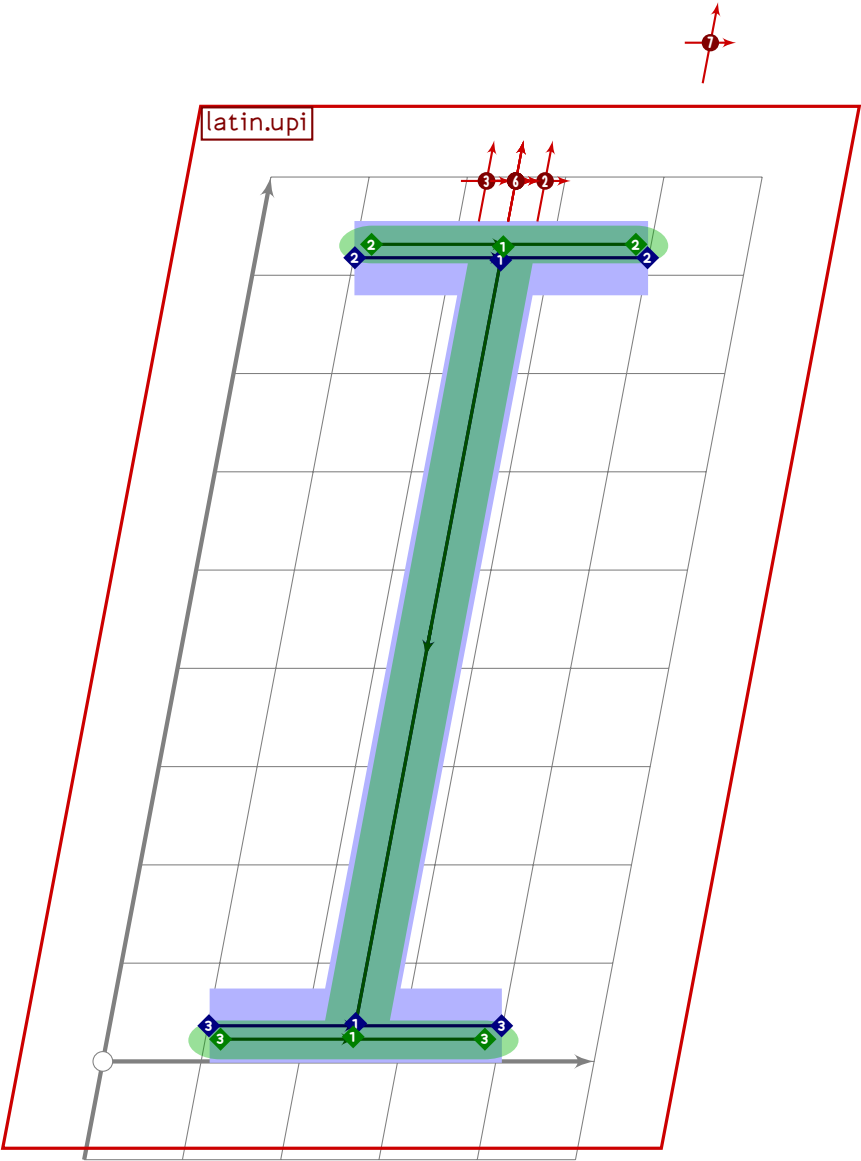
**PROO**

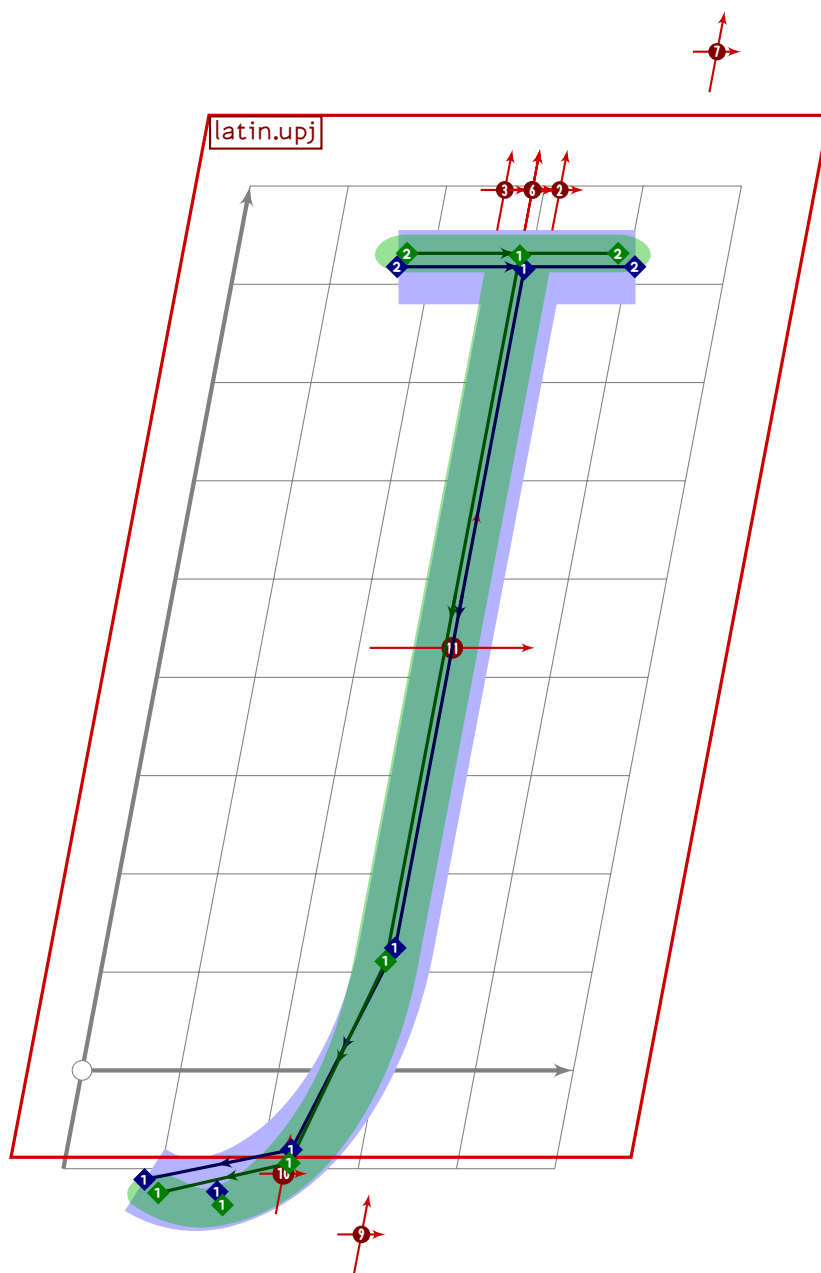
PROO





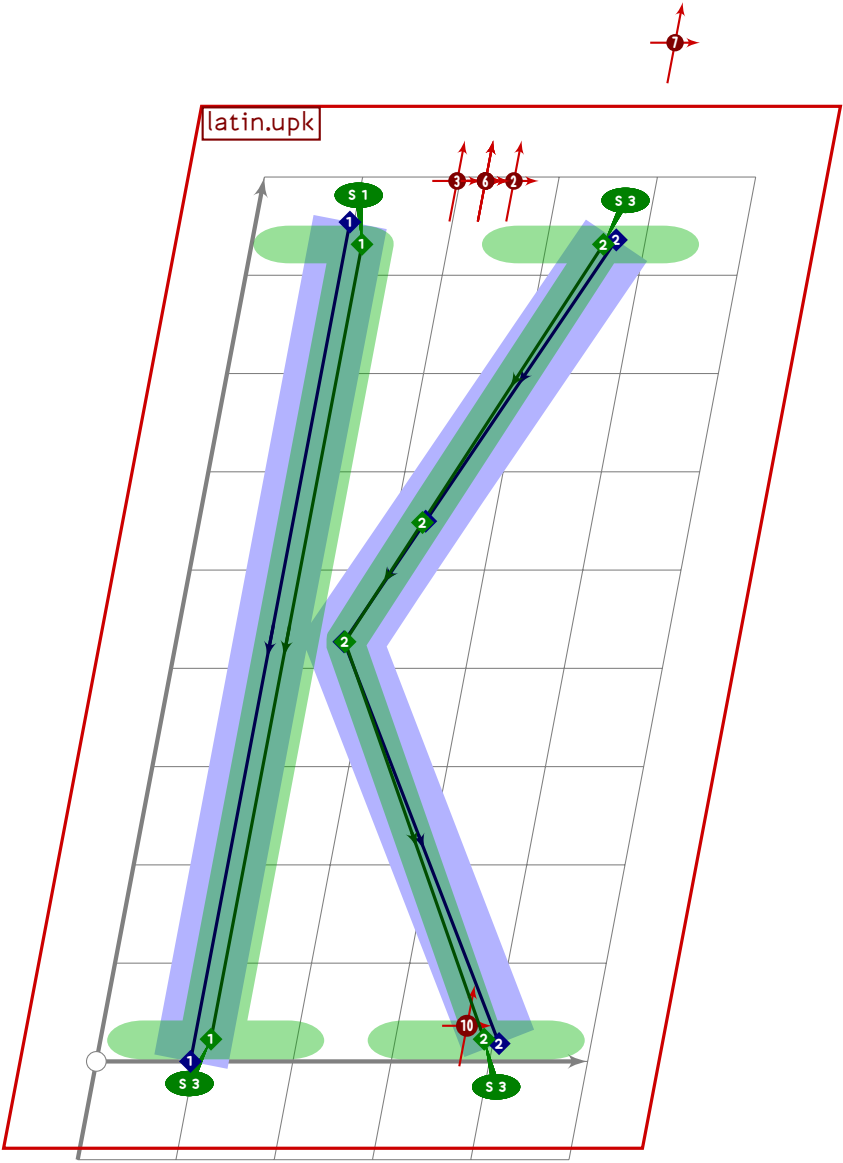
PROO



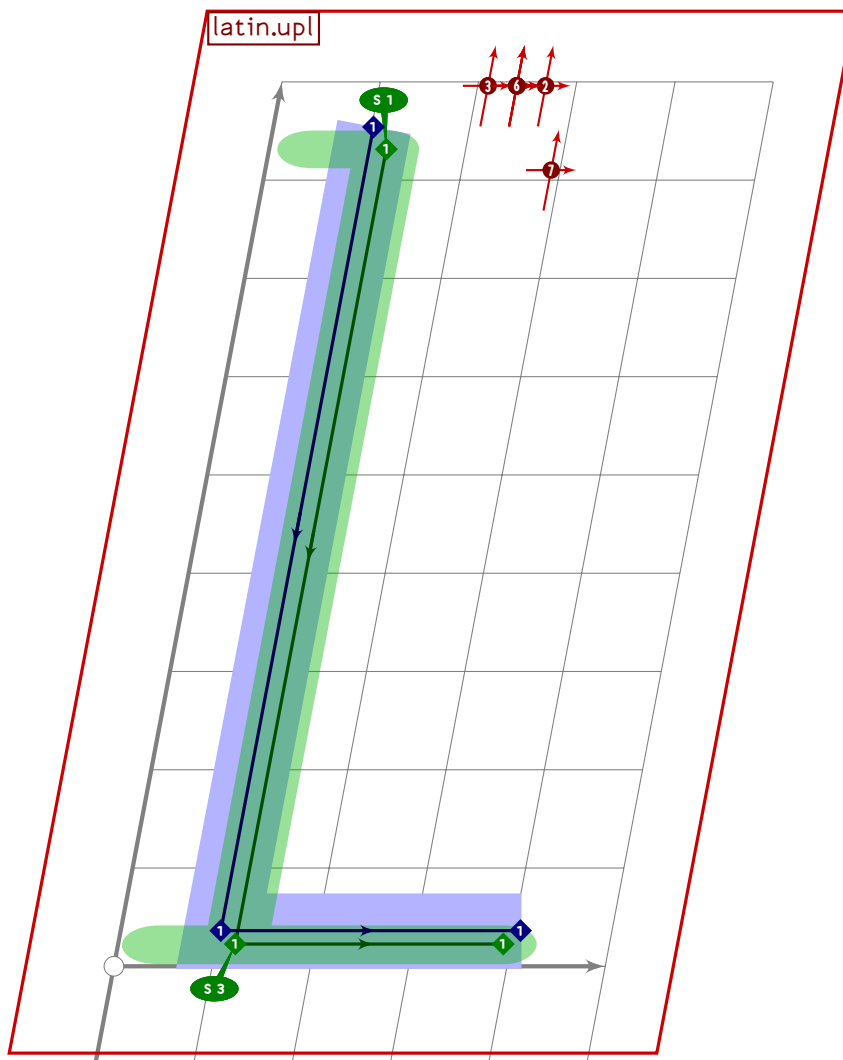


PROO

PROO

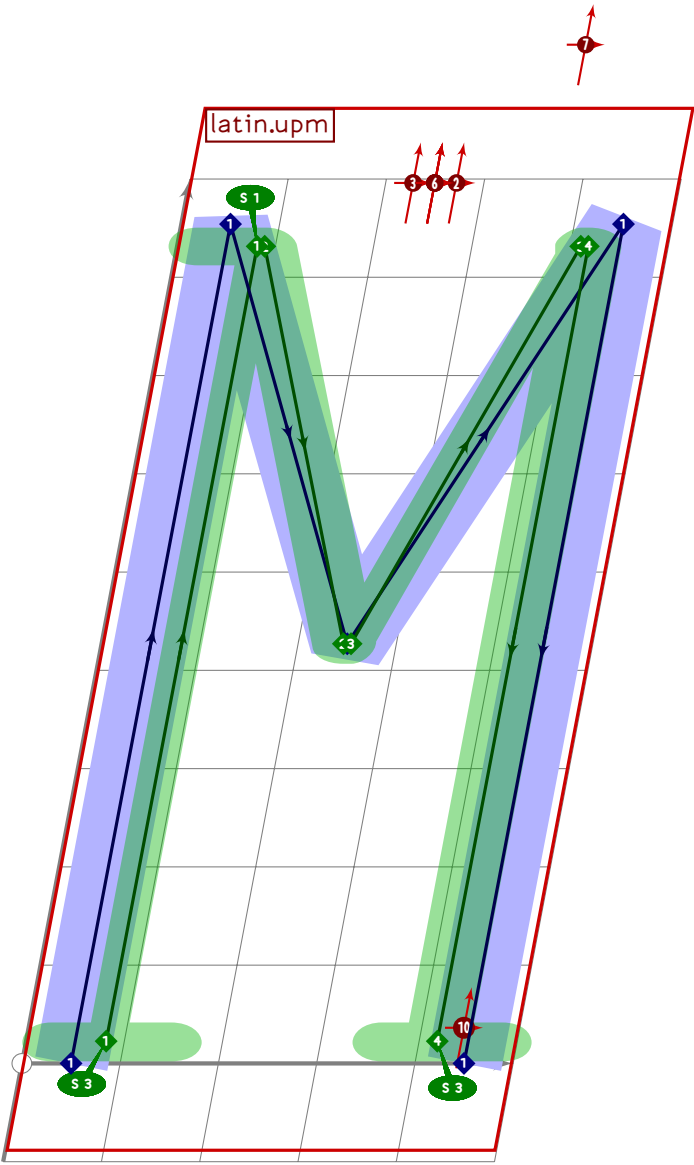


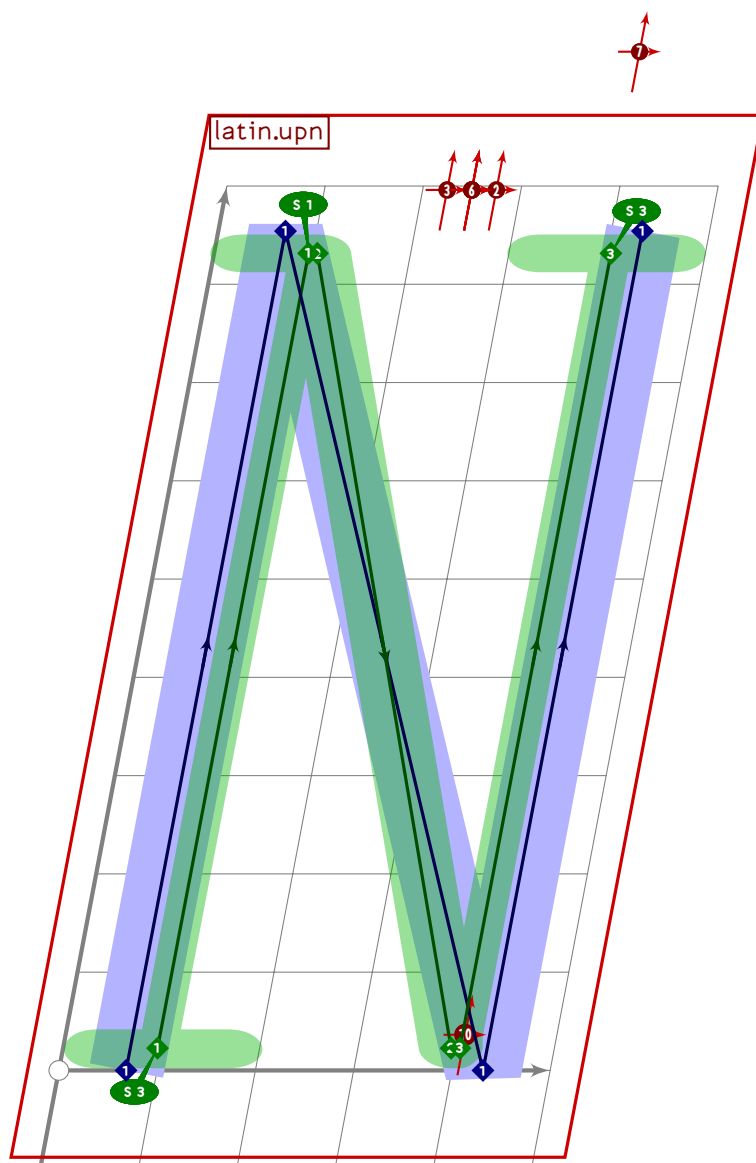




PROO

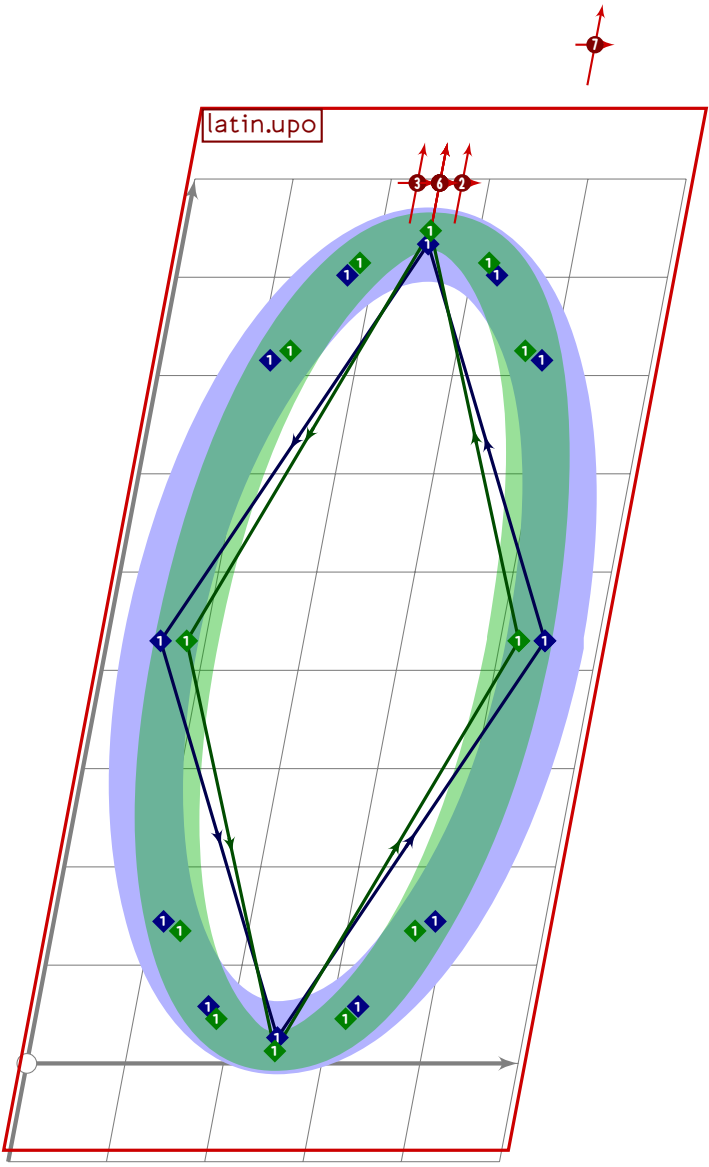
PROO

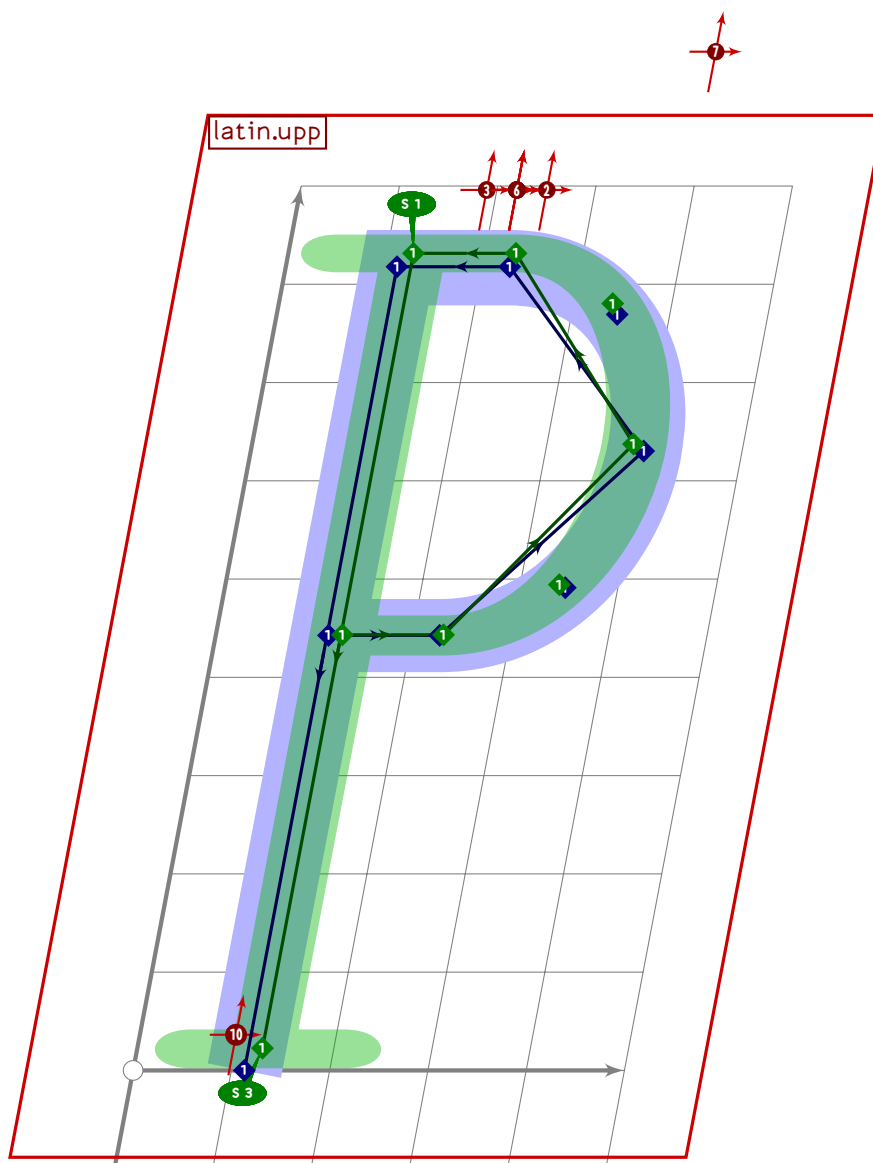




PROO

PROO

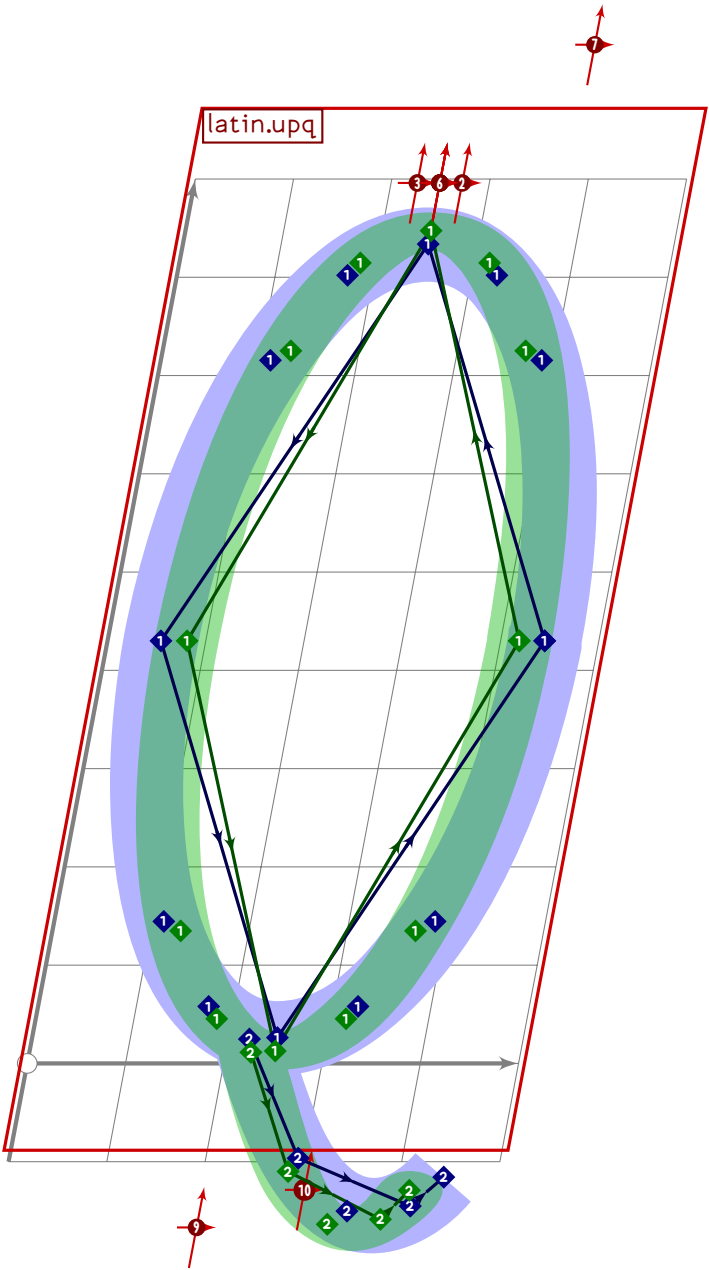


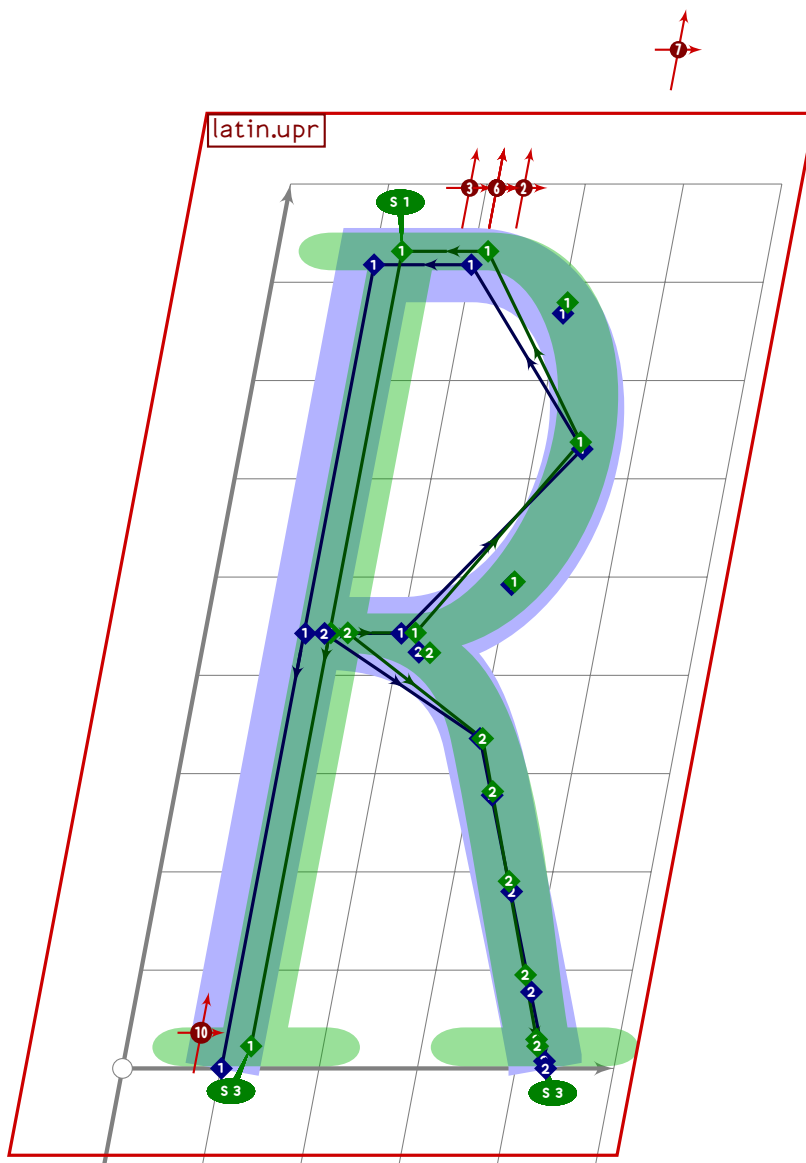


PROO

U+0051  
tsuita.Q

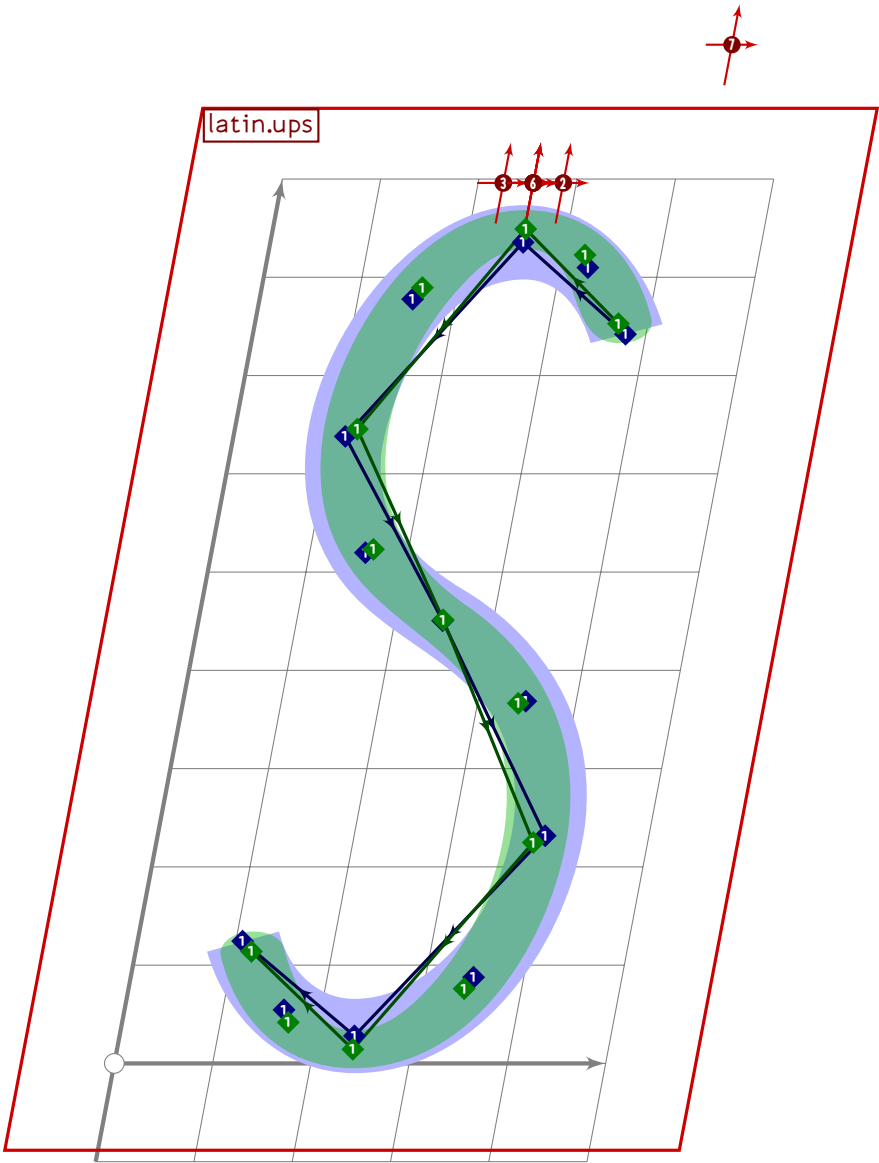
PROO



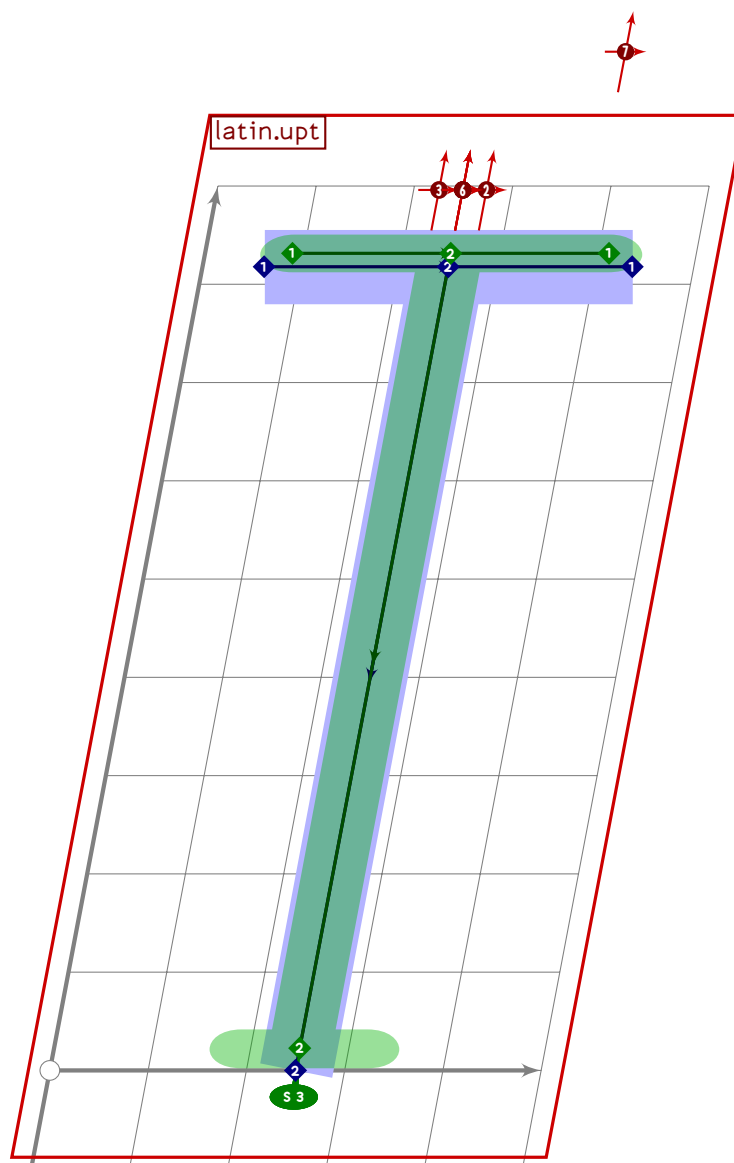


PROO

PROO

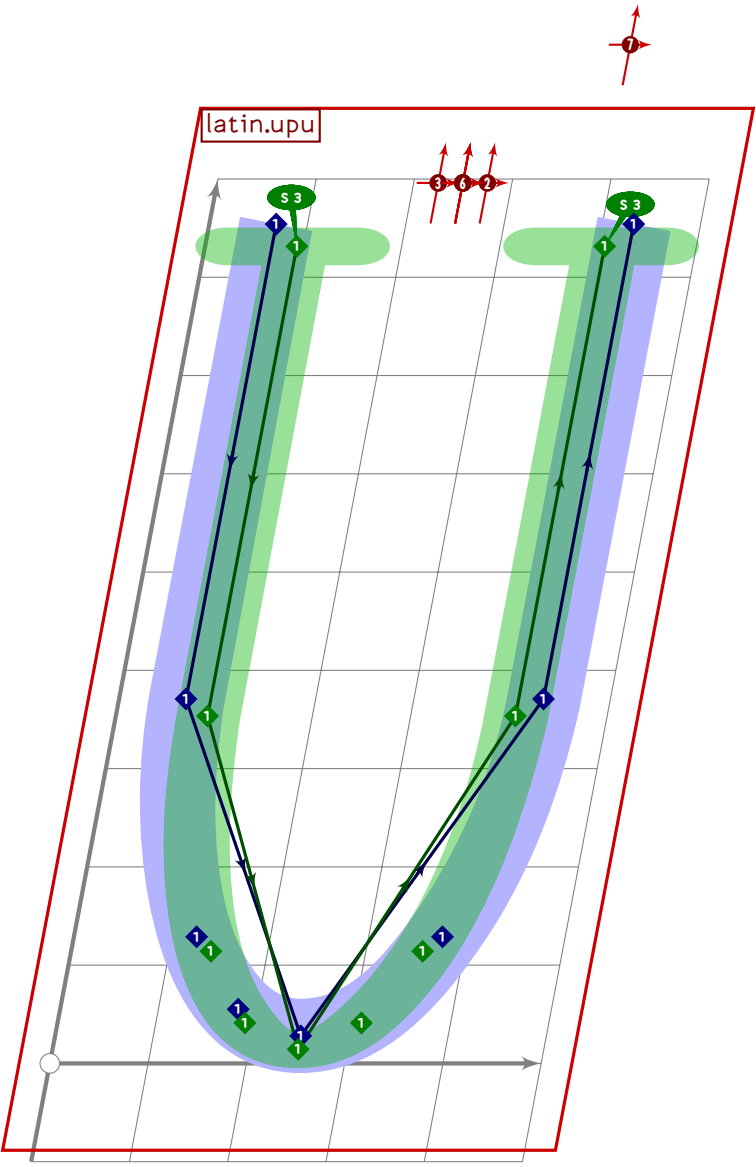


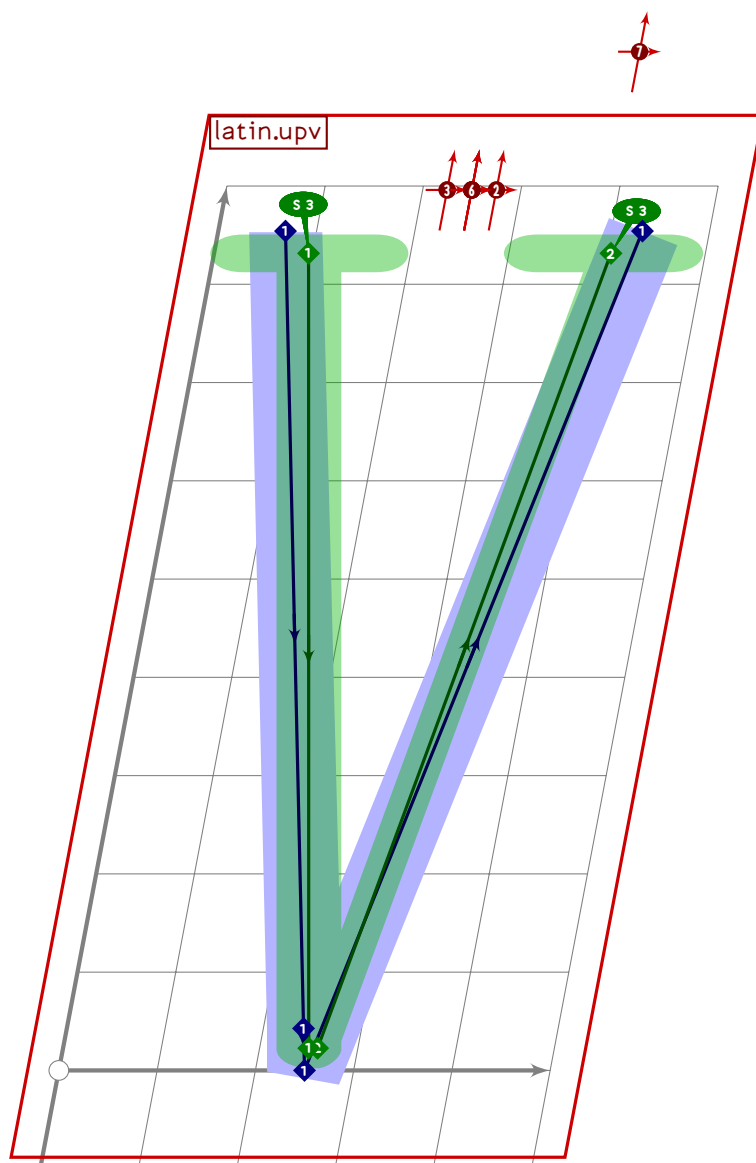


U+0054  
tsuita.T

**PROO**

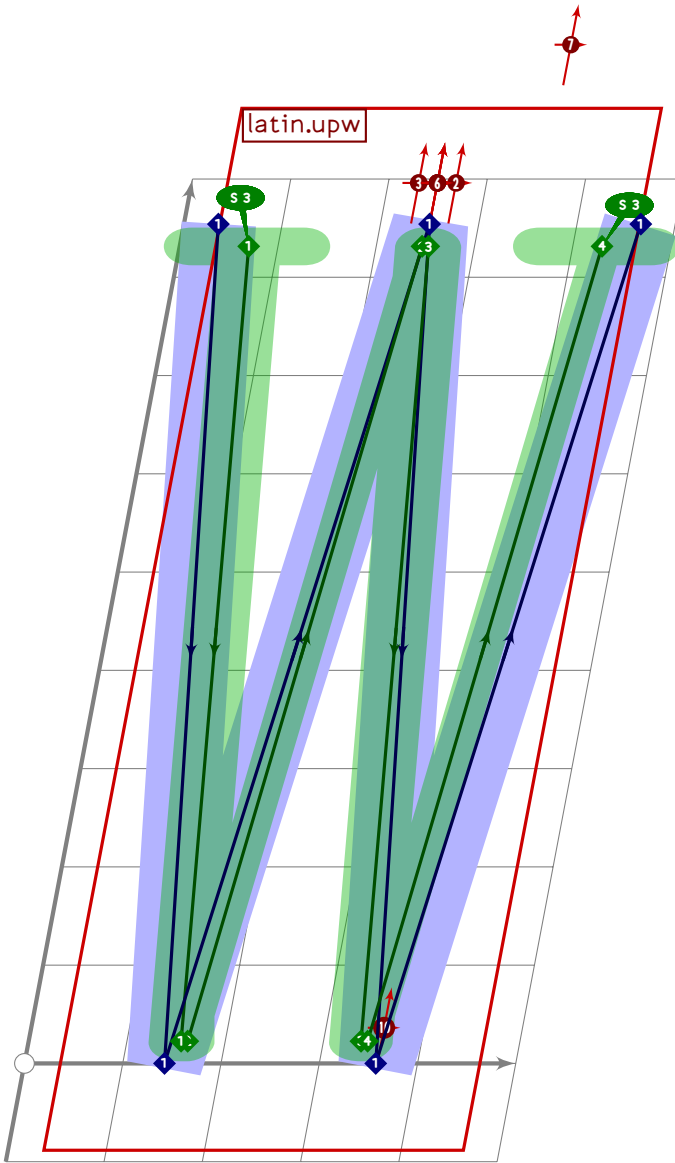
PROO





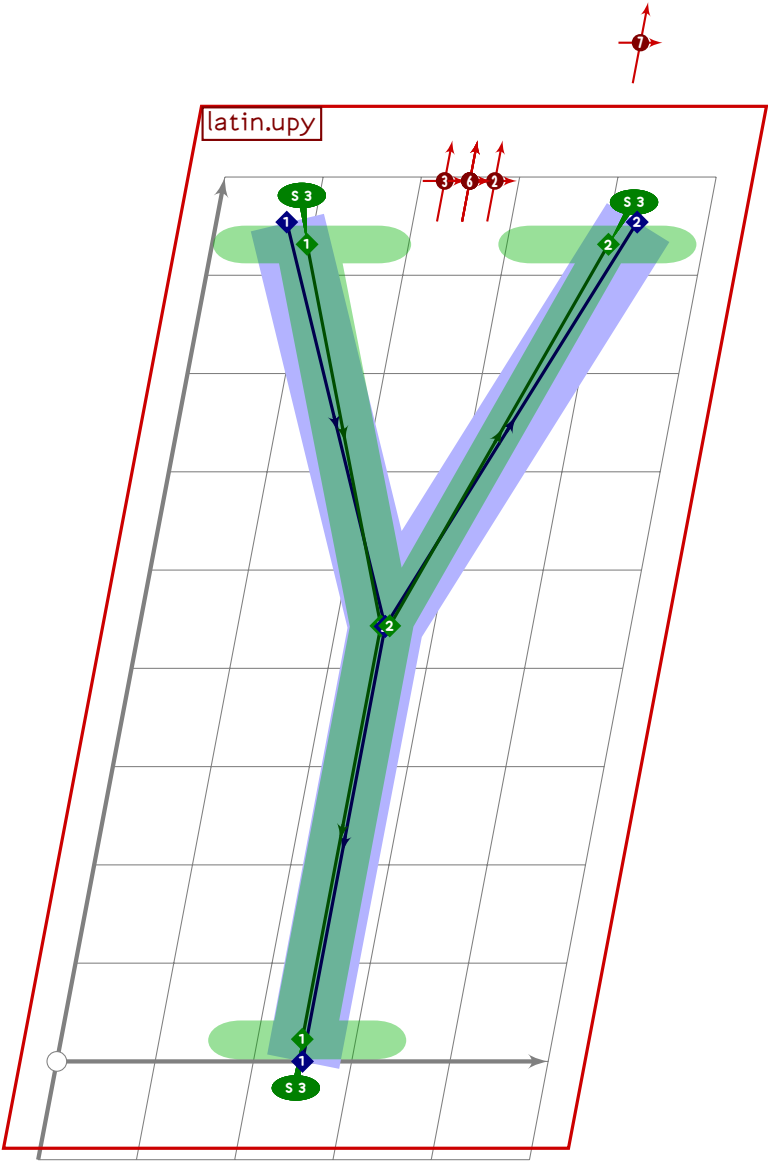
PROO

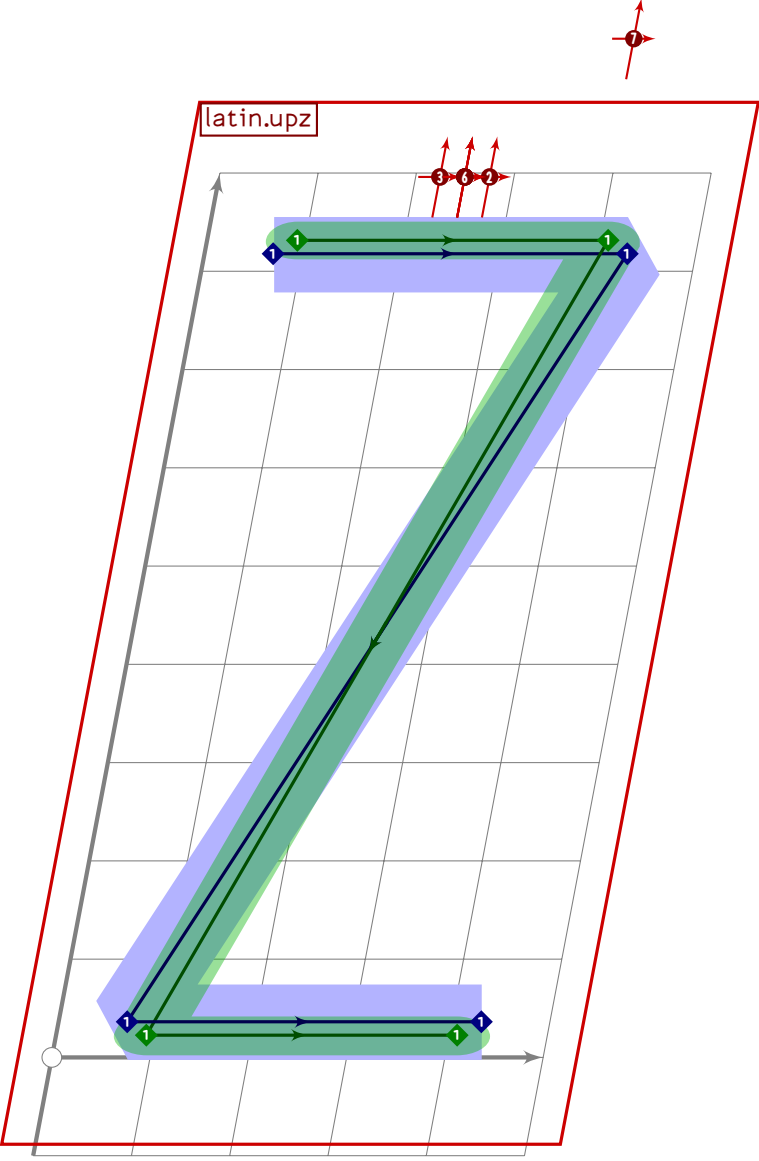
PROO





PROO



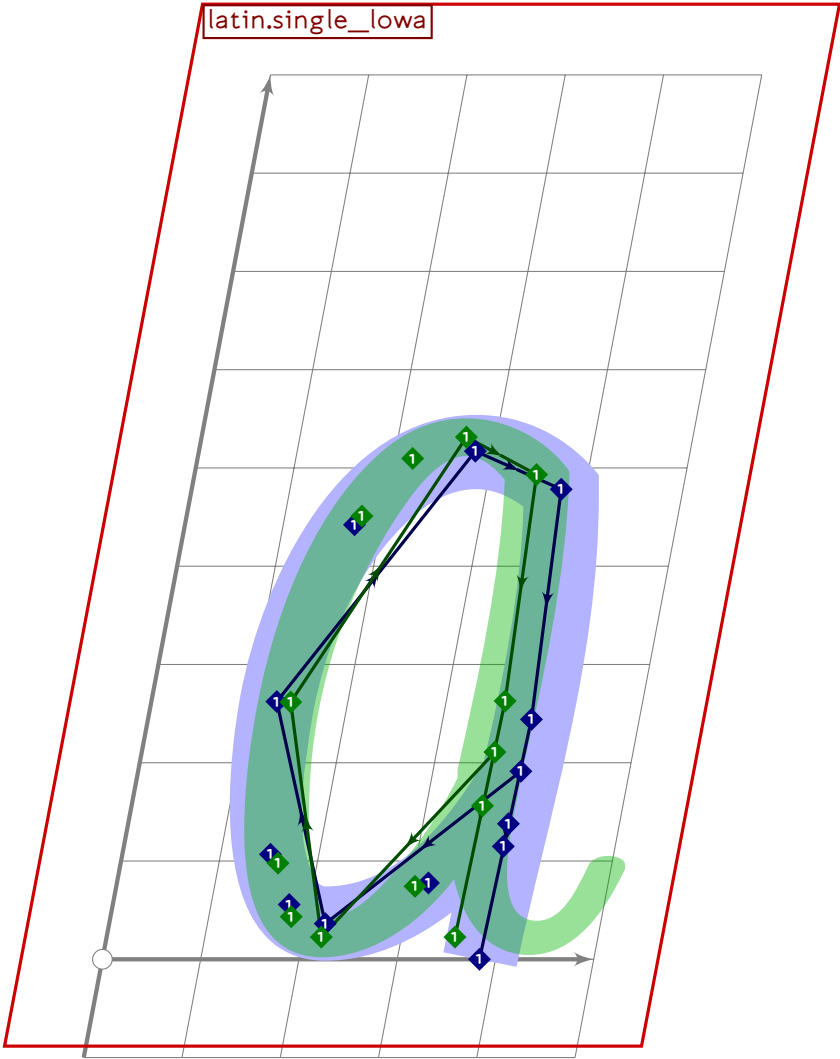


PROO

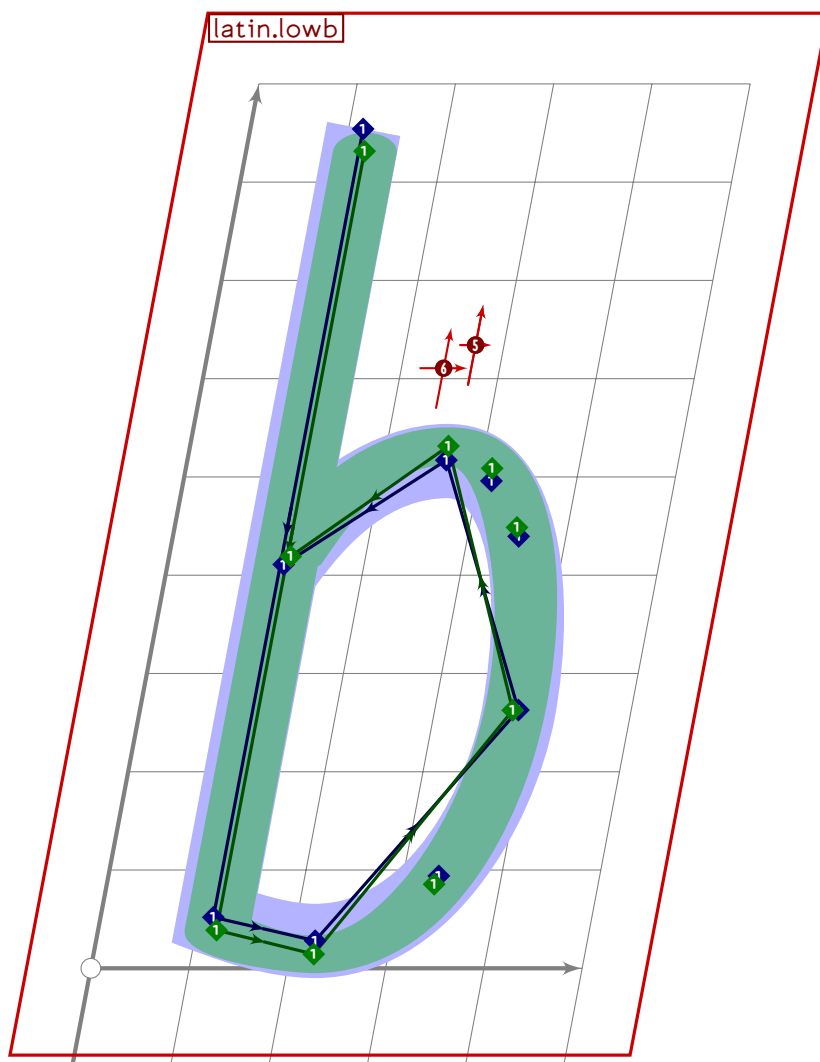
U+0061  
tsuita.a

latin.single\_lowa

PROO

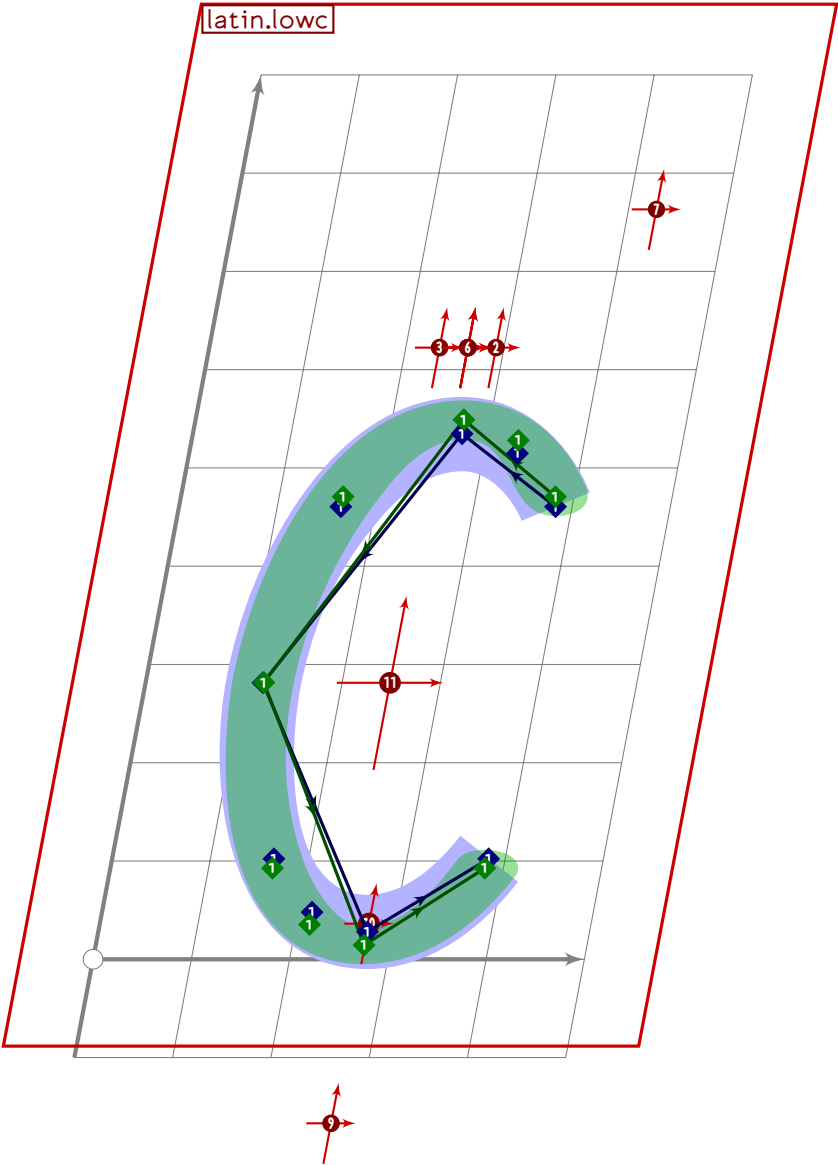


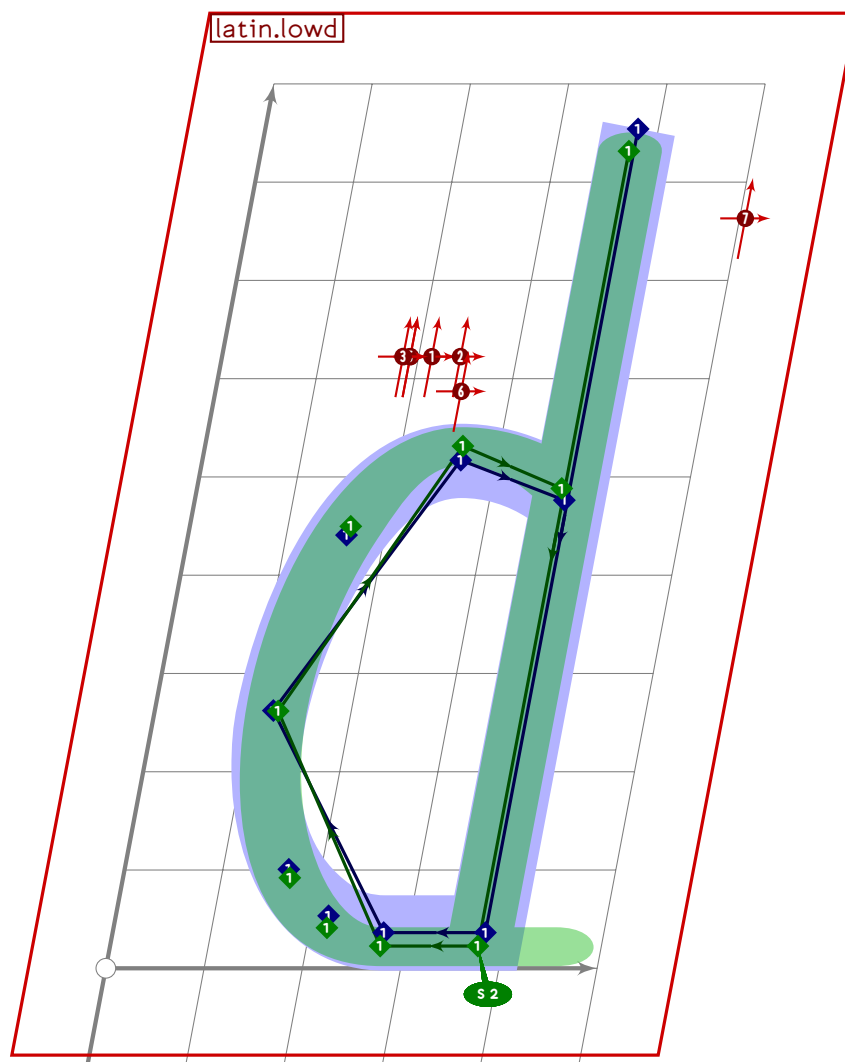




PROO

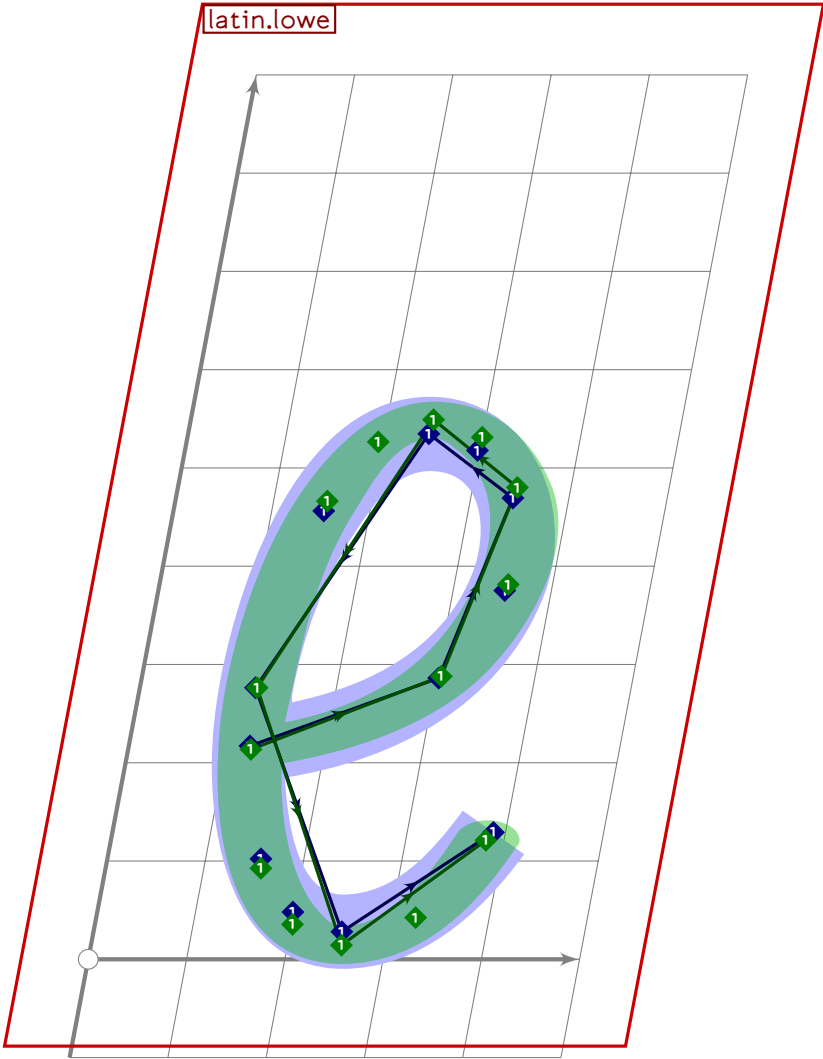
PROO

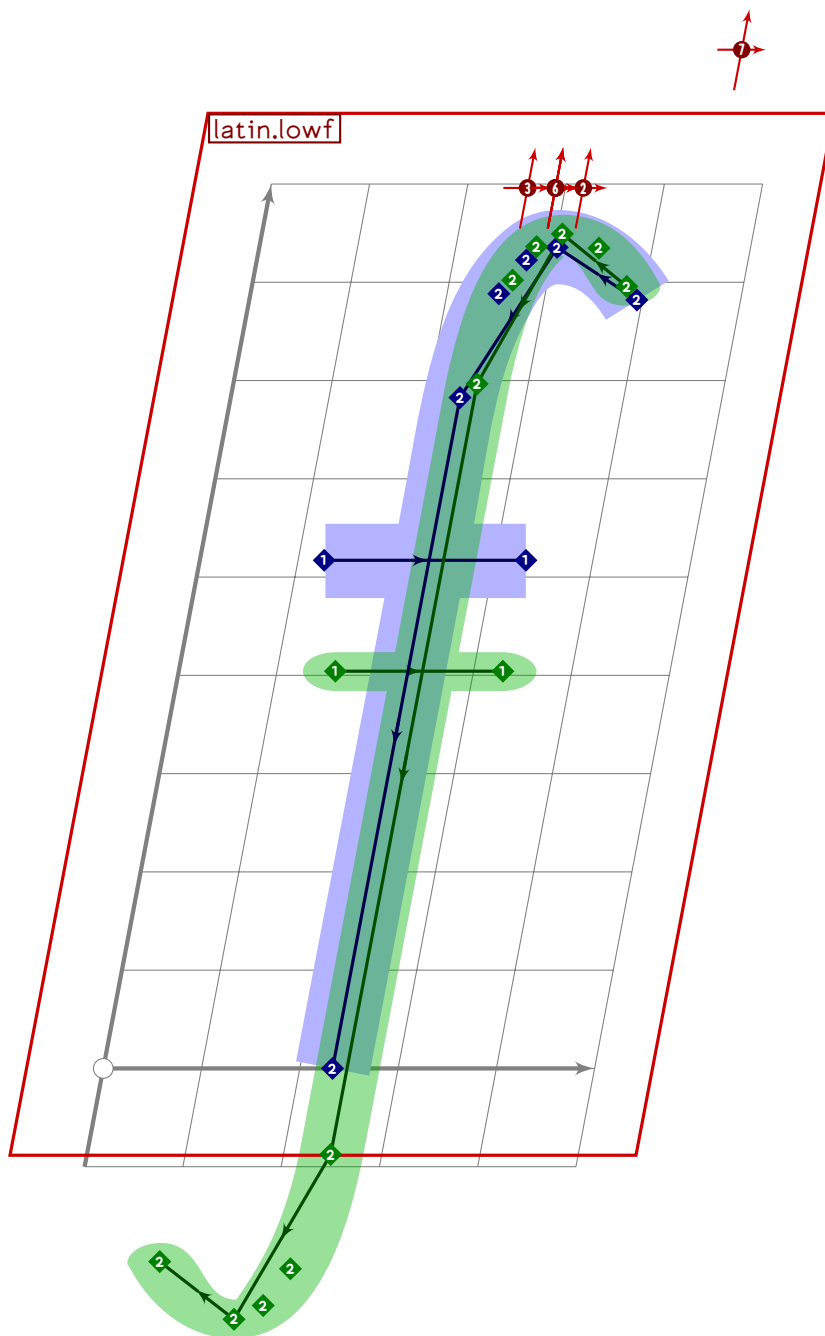




PROO

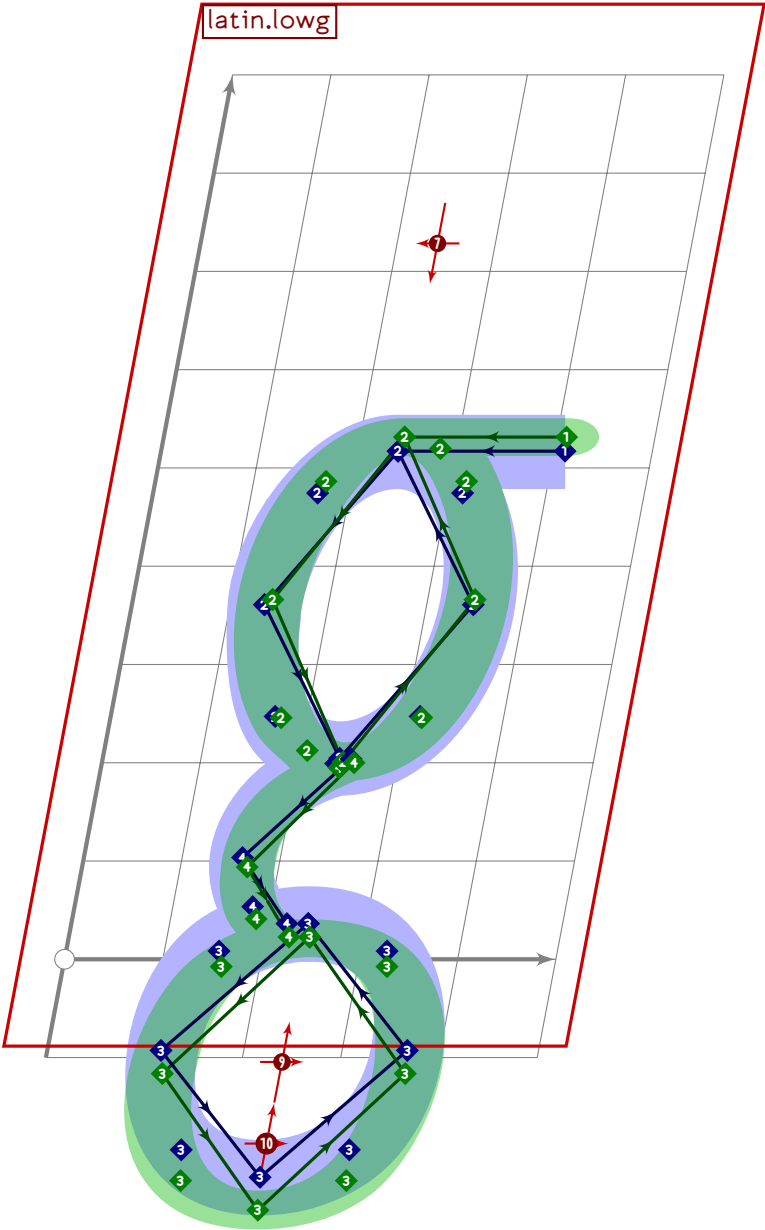
PROO

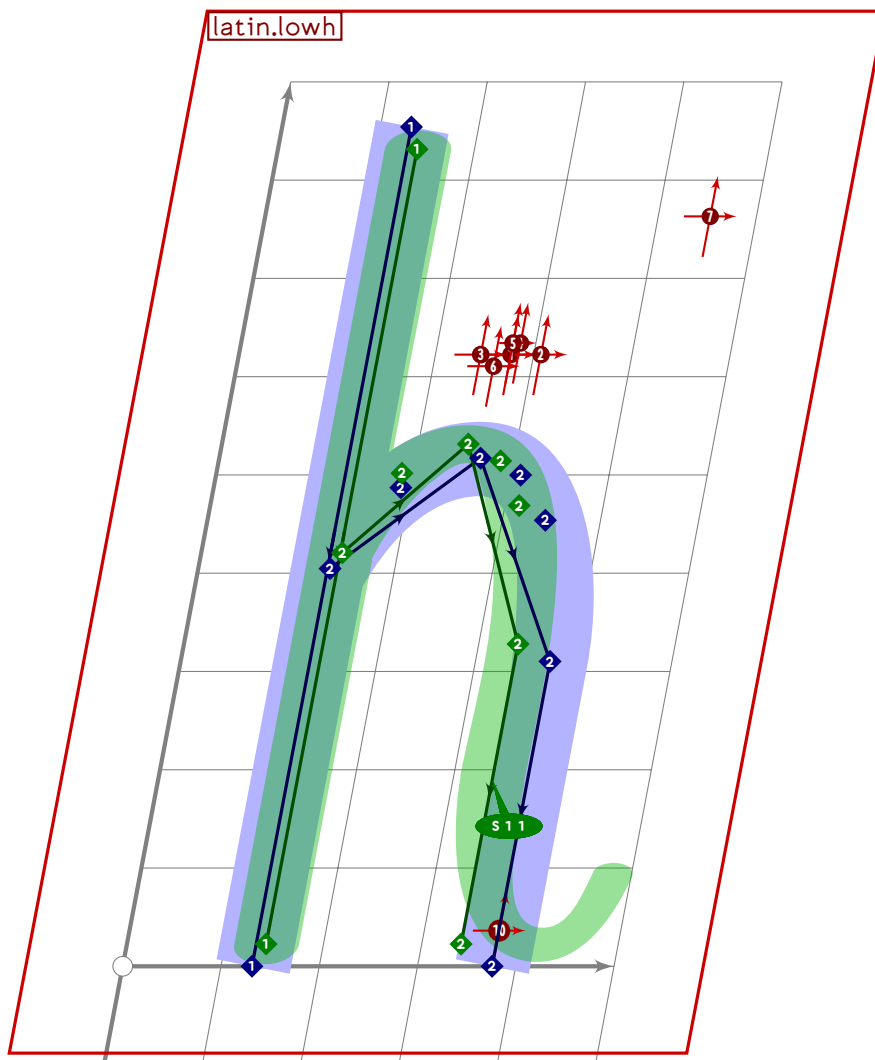




PROO

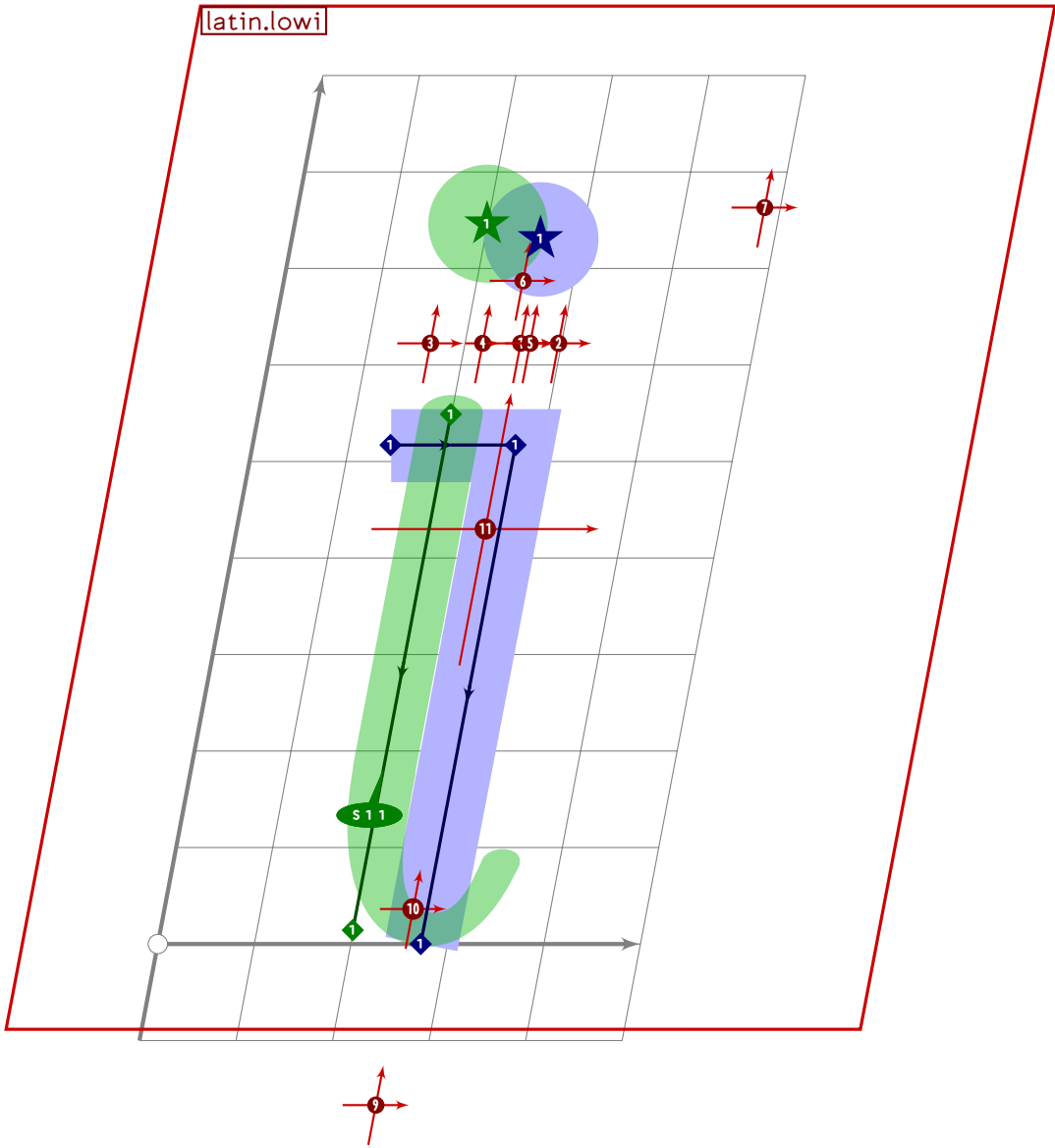
PROO



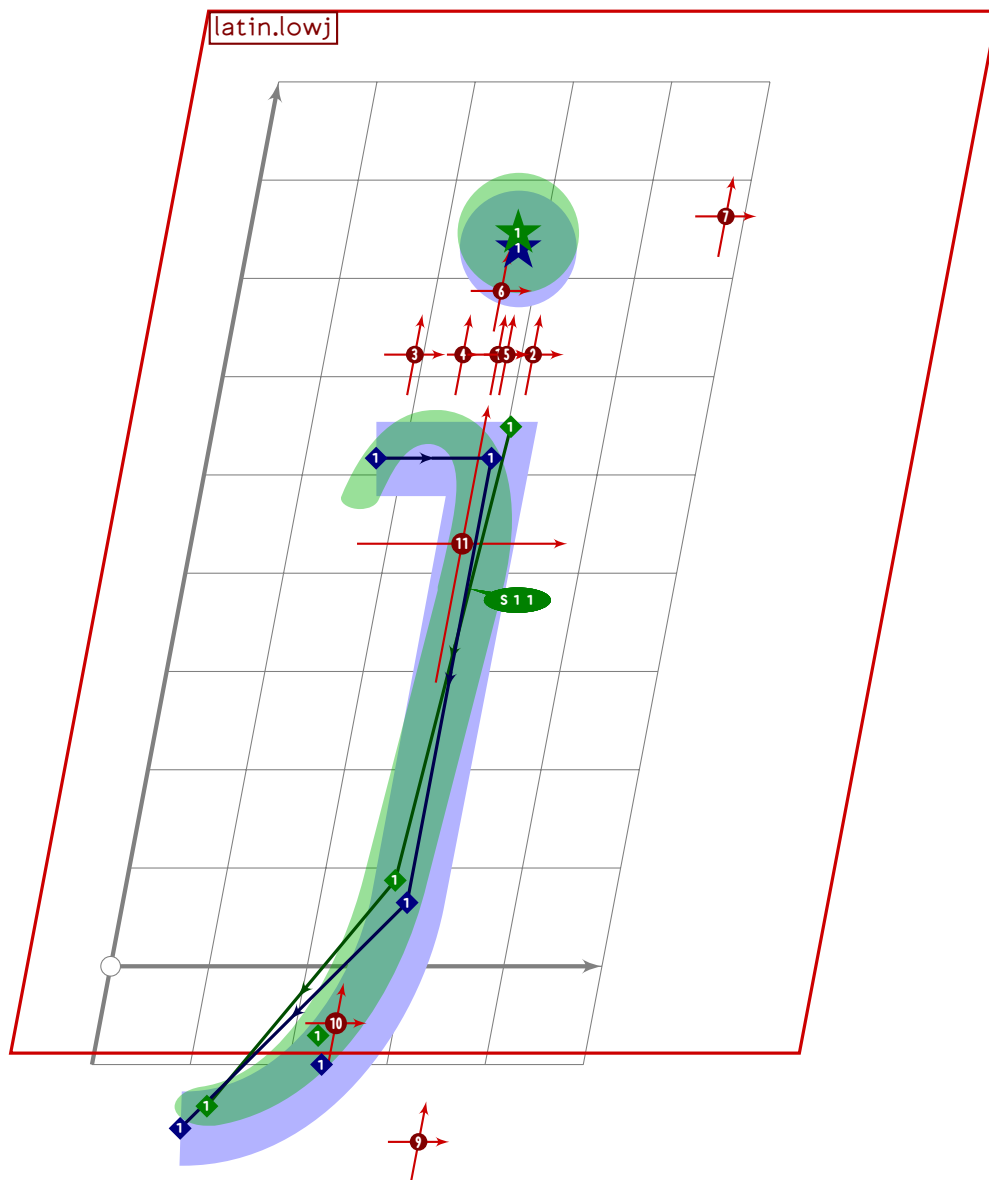


PROO

PROO

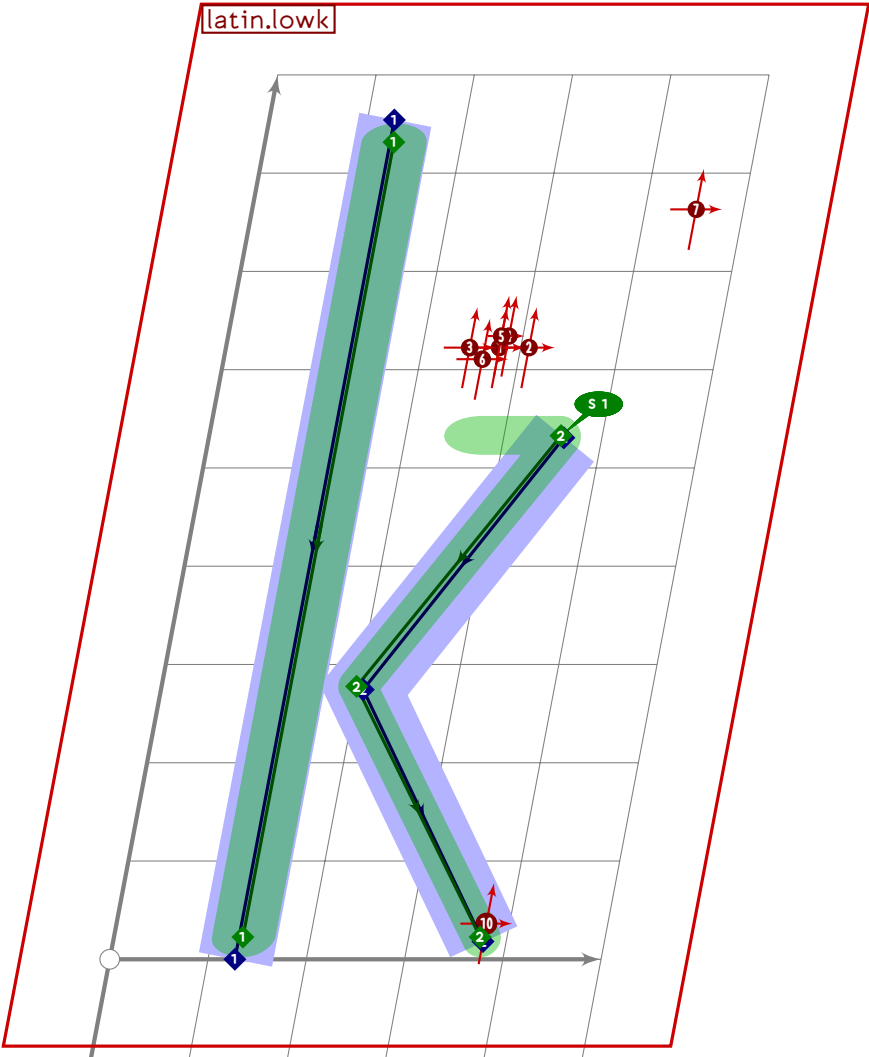


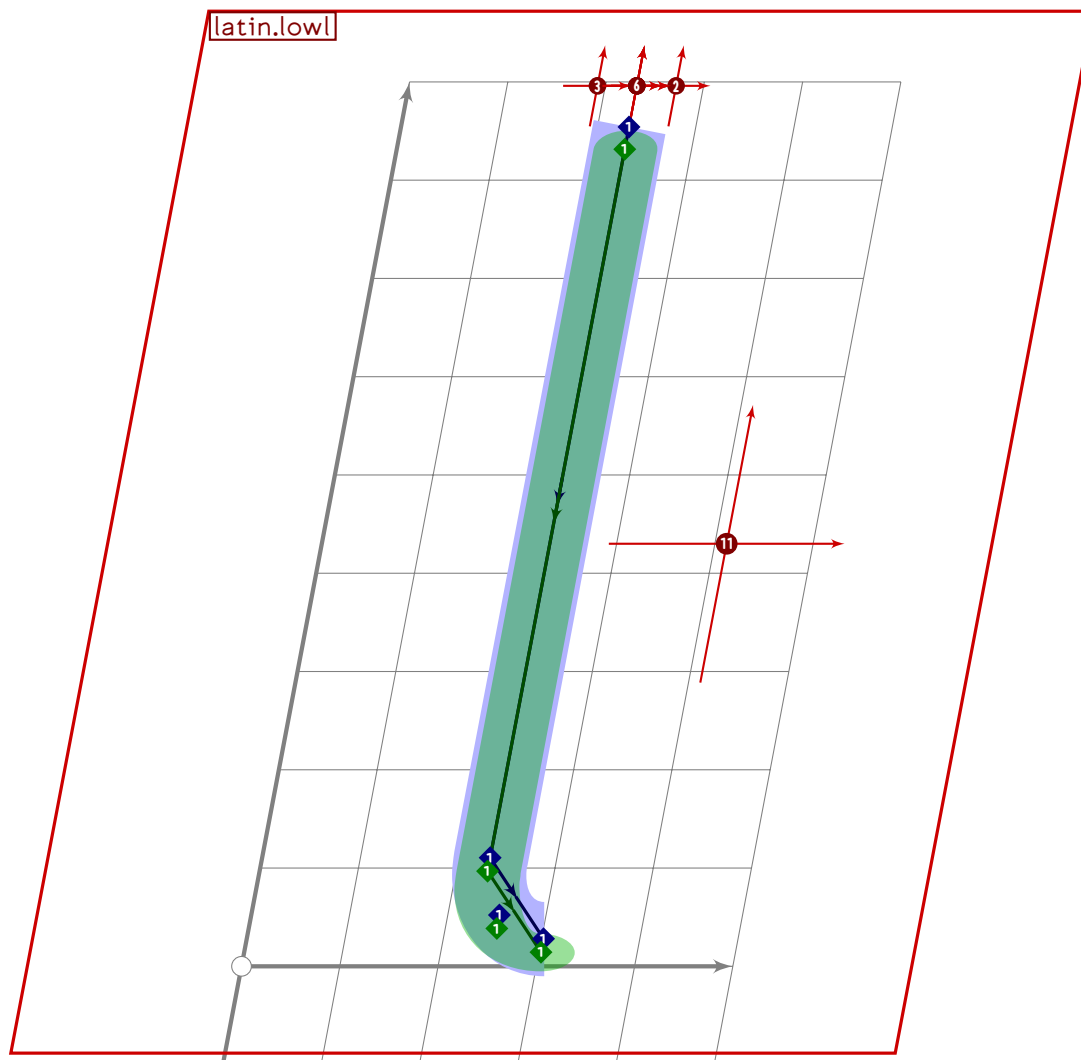




PROO

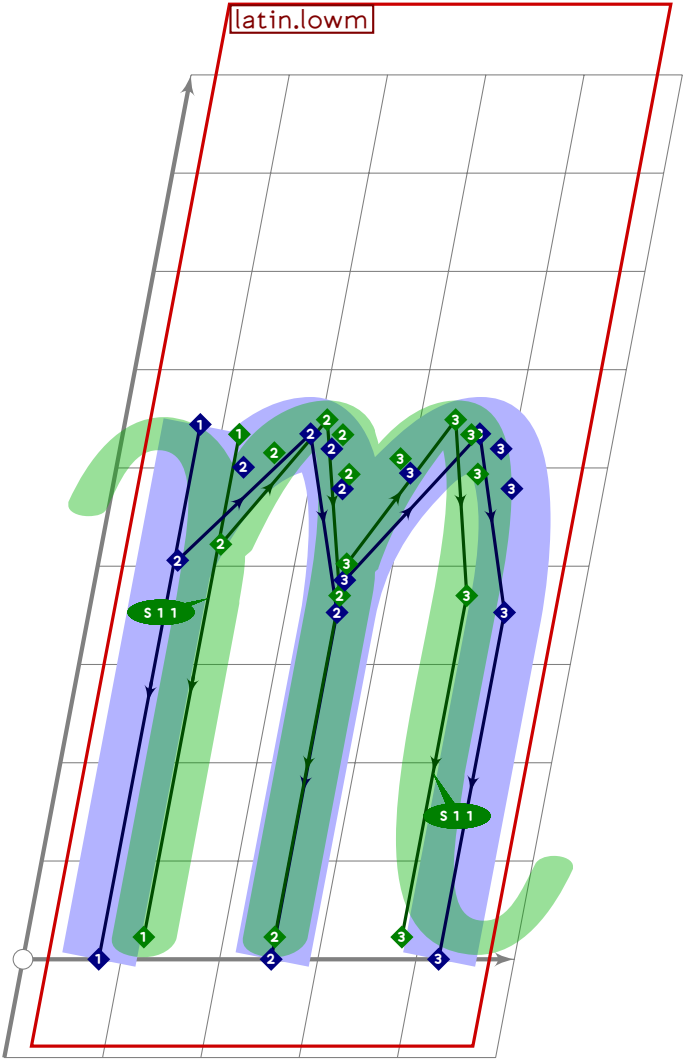
PROO

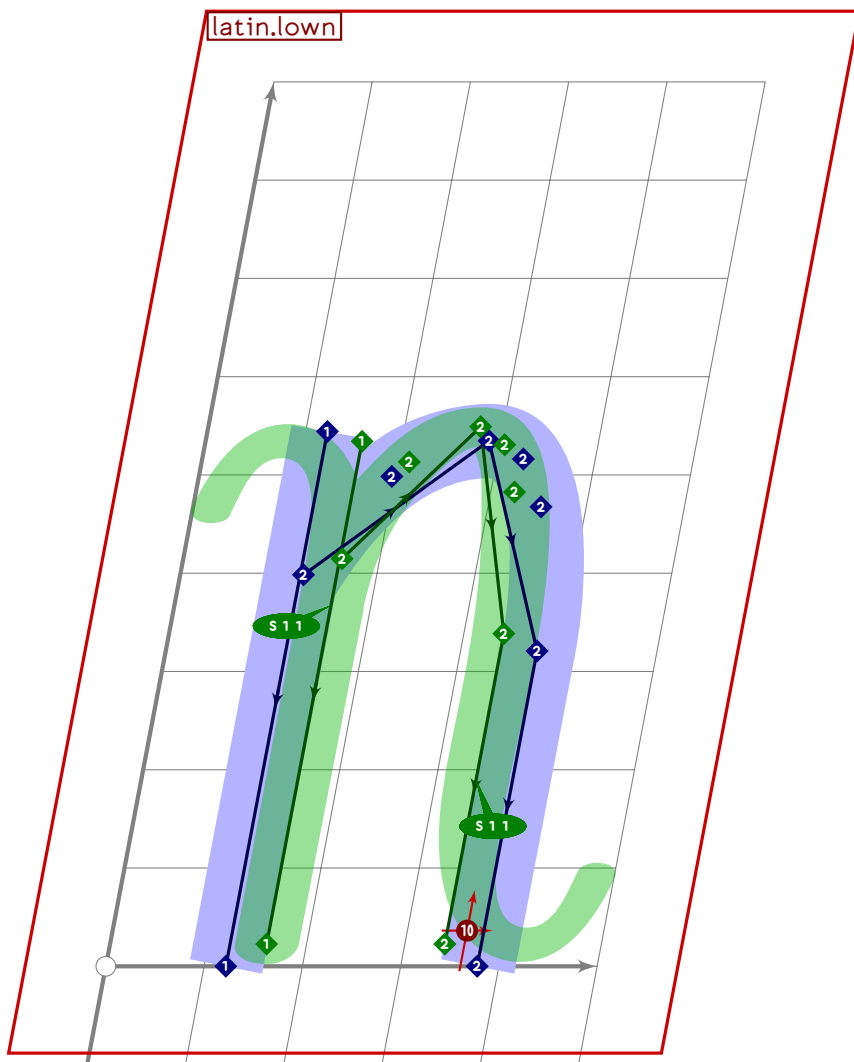




PROO

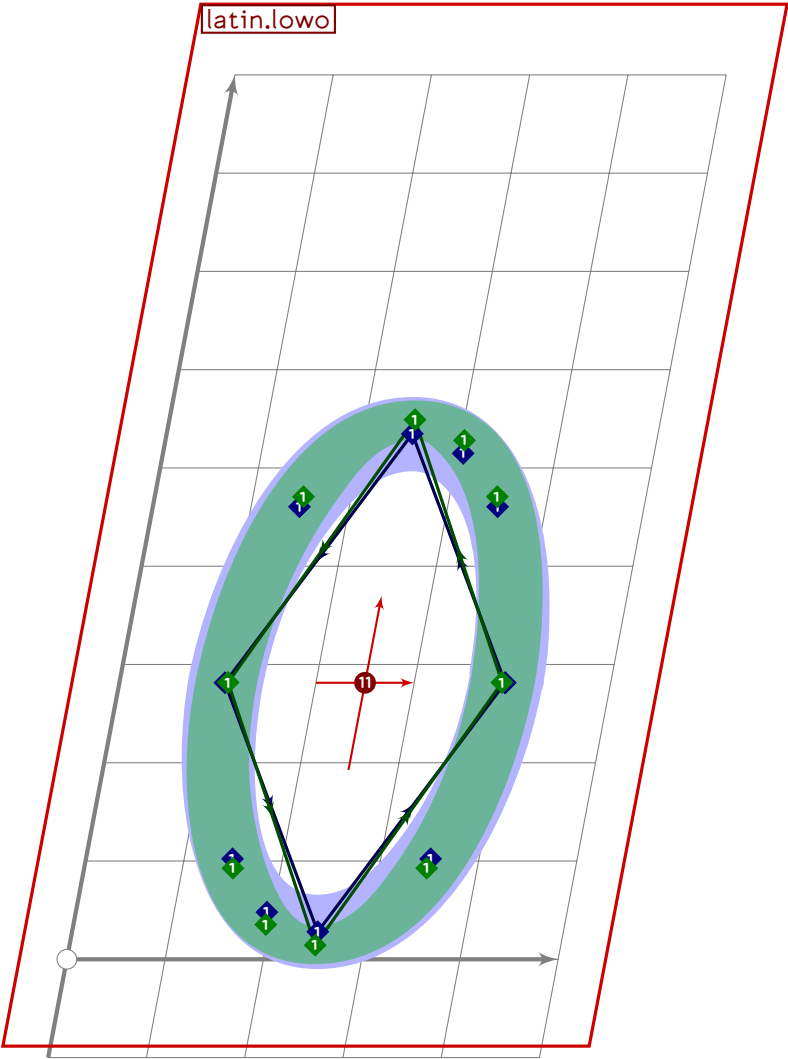
PROO





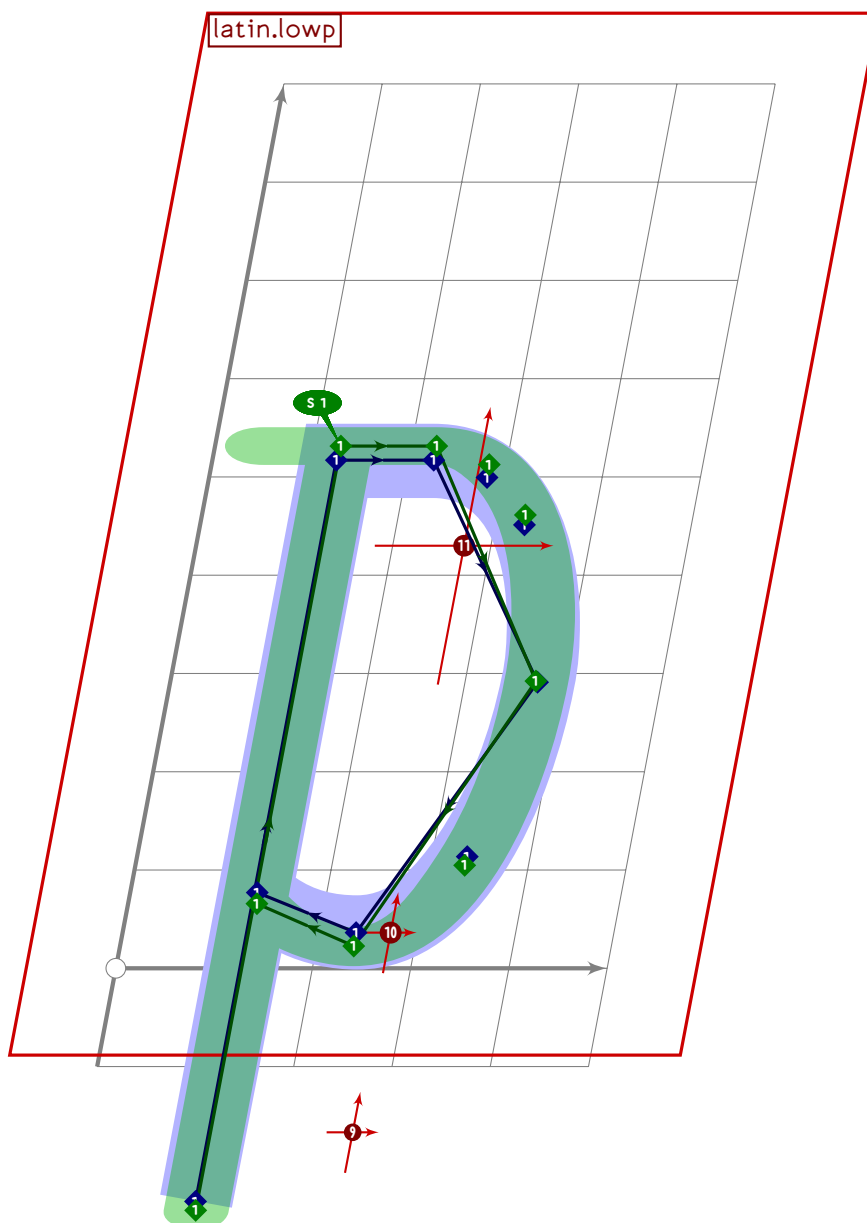
PROO

PROO

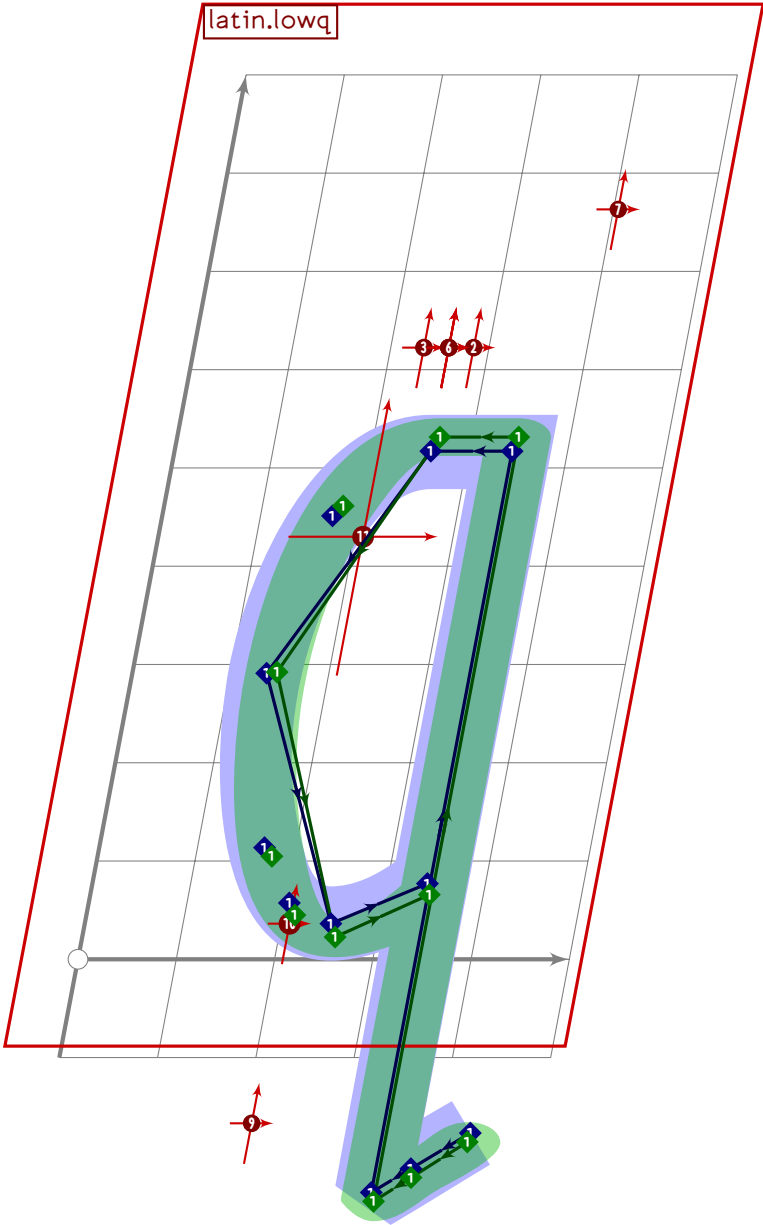


U+0070  
tsuita.p

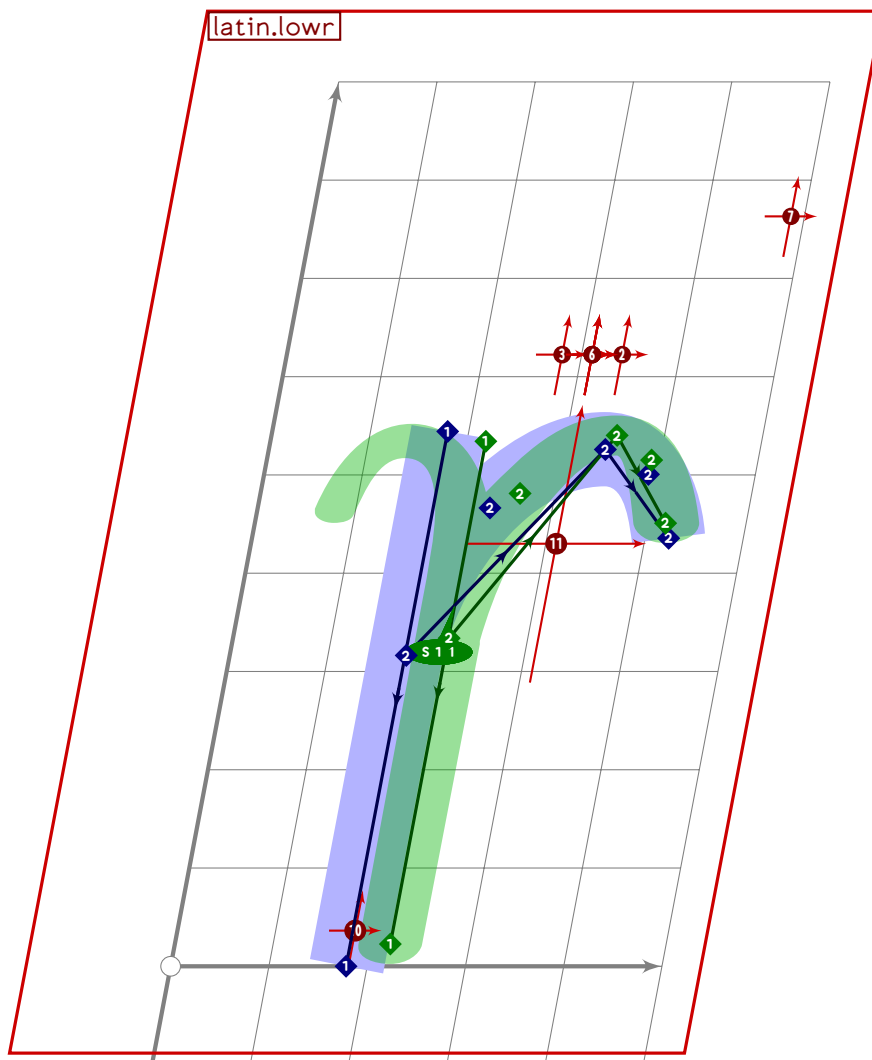
PROO



PROO

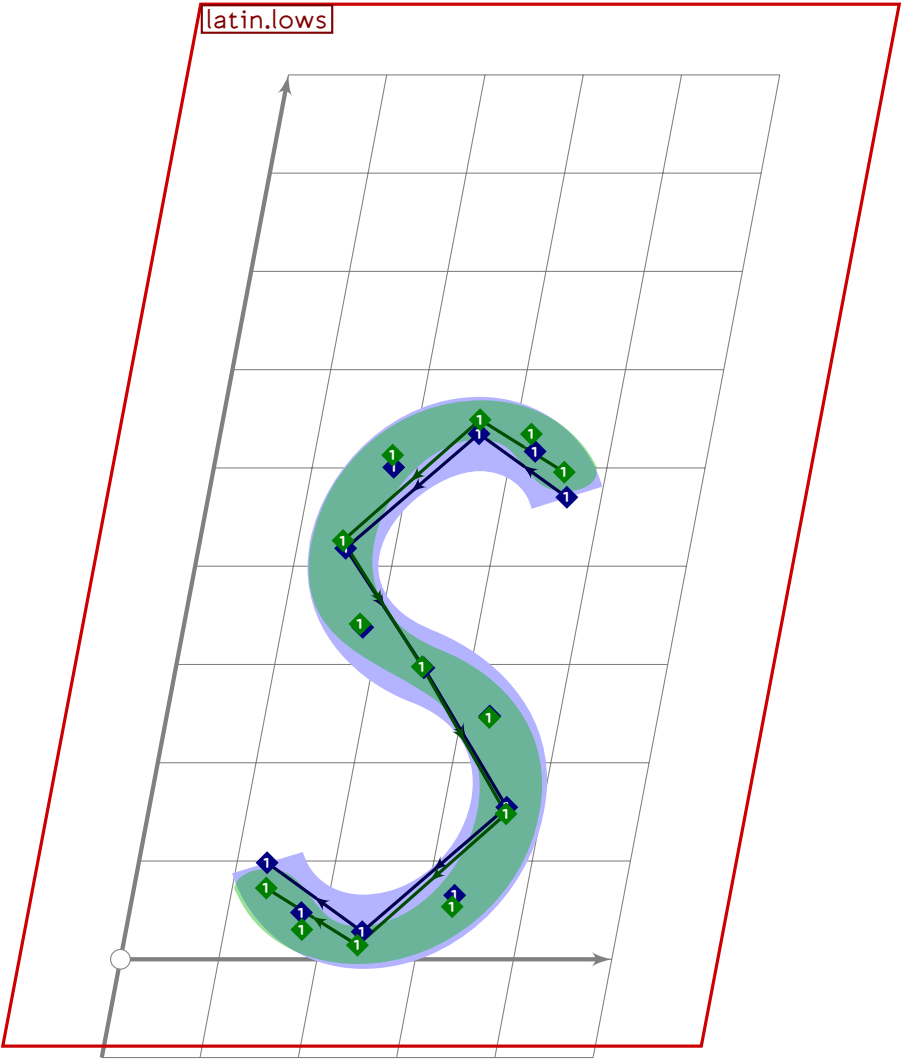


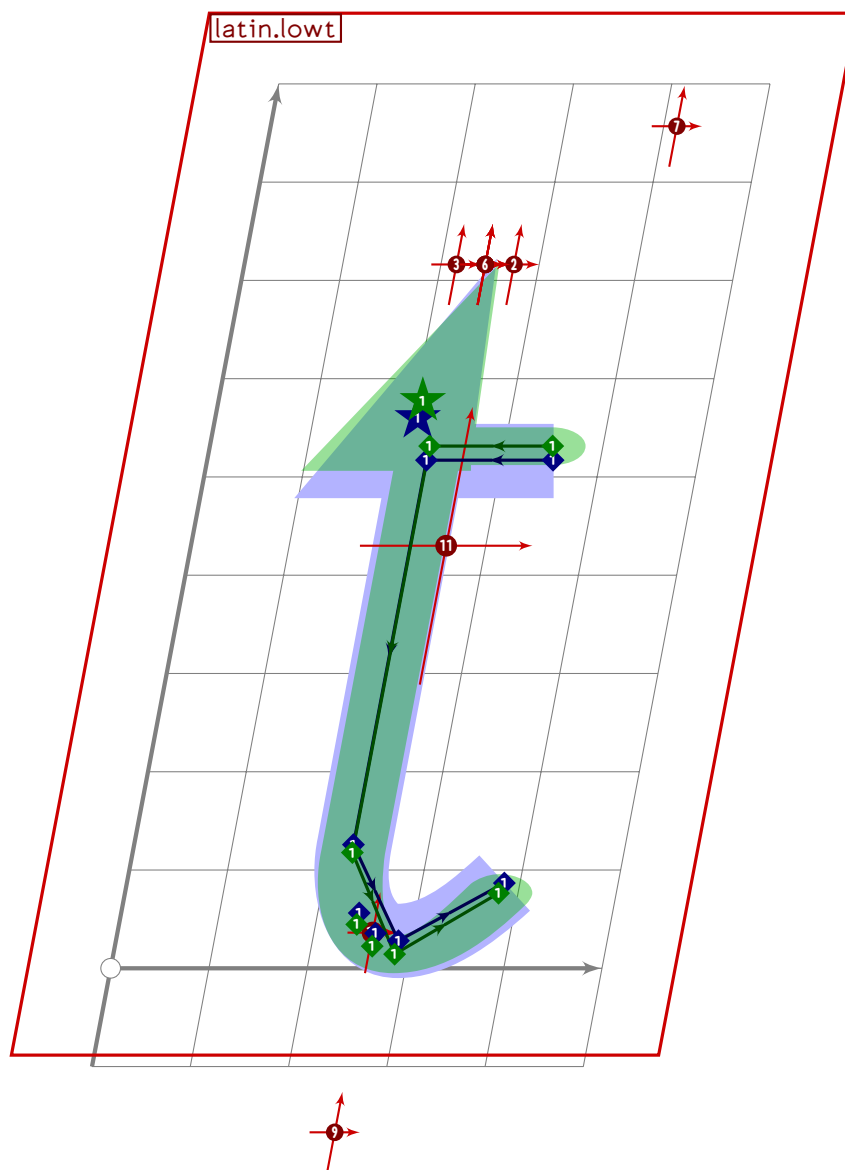




PROO

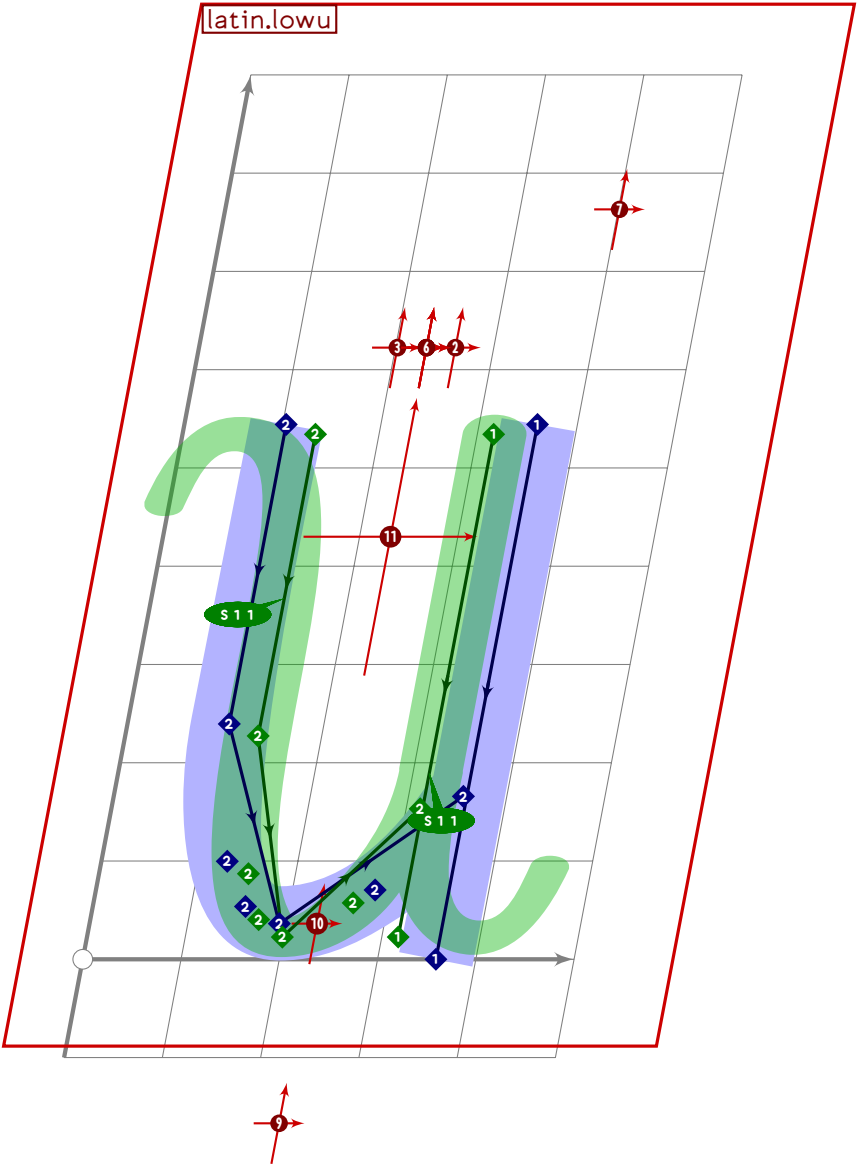
PROO

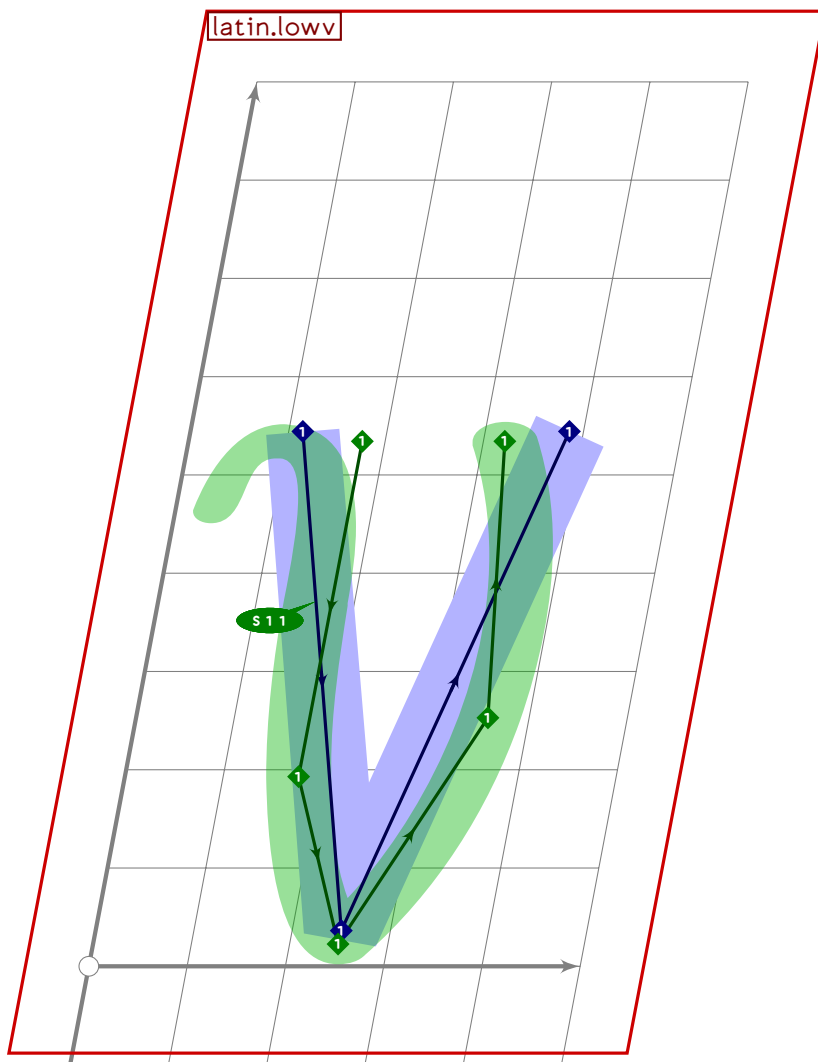




PROO

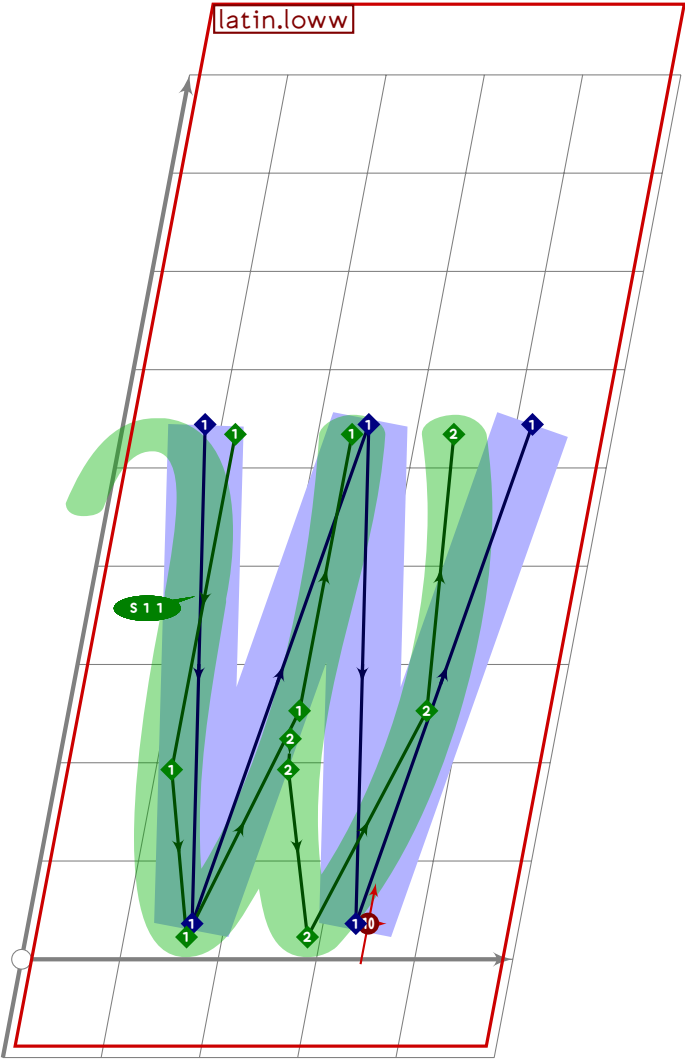
PROO

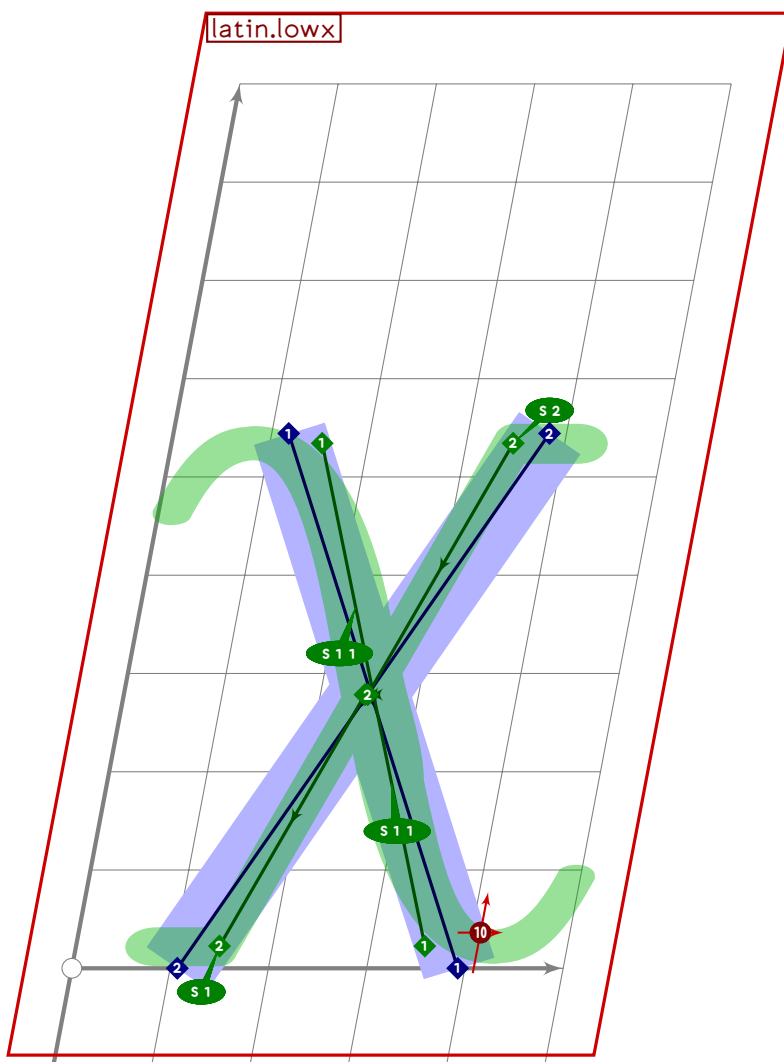




PROO

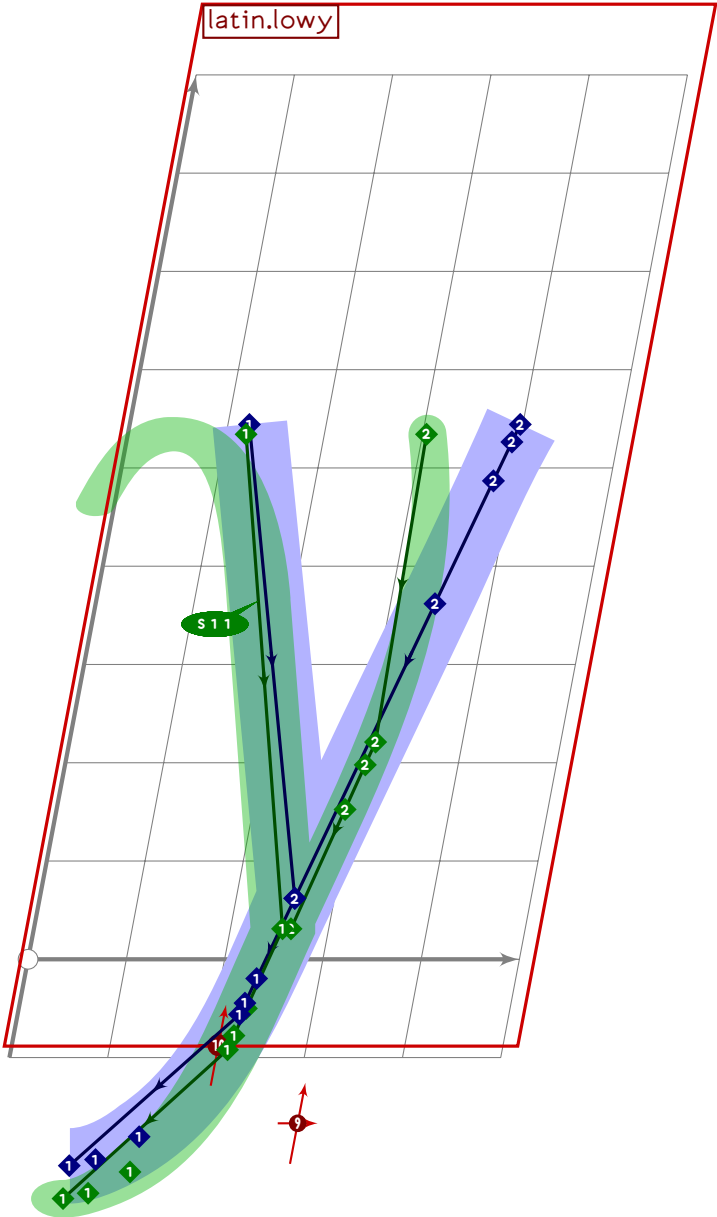
PROO



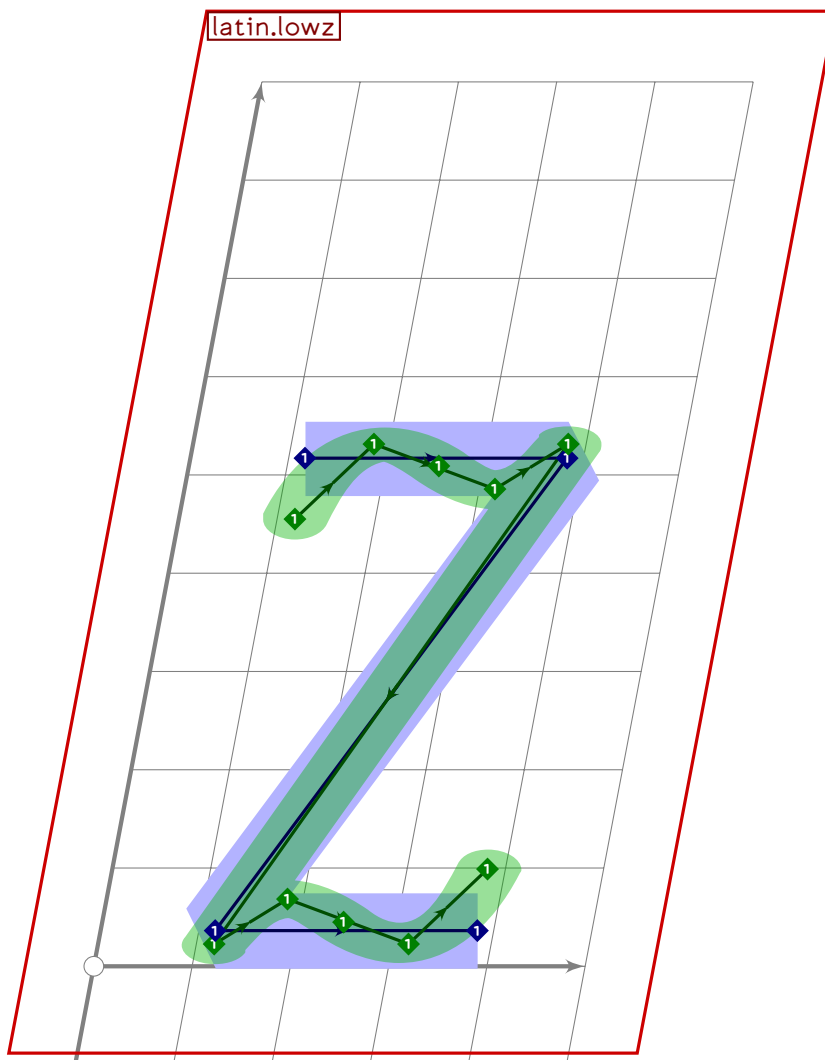


PROO

PROO







PROO

latin.lowaogonek  
latin.single\_lowa

PROO

