

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

Proofs and pretty-printed source code for Tsukurimashou
Copyright © 2011–2021 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	19
	tsuku-kg.mp	20
	tsuku-mg.mp	21
	tsuku-mi.mp	22
	tsuku-ps.mp	24
	tsuku-bq.mp	25
	tsuku-dq.mp	26
	tsuku-el.mp	27
	tsuku-eq.mp	28
	tsuku-lw.mp	29
	intro.mp	30
	fntbase.mp	49
	General Library Functions	49
	Prefix And Suffix Handling	56
	A Module That Finds An Envelope Of A Path Drawn With A Pen	58
	Postscript Font Generation	70
	obstack.mp	92
	Object Stack Data	92
	Object Stack Methods	93
	frac-intro.mp	98
	latin-intro.mp	100
	accent.mp	104
	bcircle.mp	115
	bkencl.mp	118
	buildkanji.mp	125
	dakuten.mp	138
	enclosed.mp	139
	genjimon.mp	141
	hiragana.mp	153
	Hiragana Vowels	153
	Hiragana Kakikukeko/Gagigugego	158

Hiragana Sashisuseso/Zajizuzezo	163
Hiragana Tachitsuteto/Dajizudedo	169
Hiragana Naninuneno	174
Hiragana Hahifuheho/Babibubebo/Papipupepo	180
Hiragana Mamimumemo	185
Hiragana Yayuyo	191
Hiragana Rarirurero	193
Hiragana Wawiwewo/N/Iteration	199
iching.mp	207
katakana.mp	209
Katakana Vowels	209
Katakana Kakikukeko/Gagigugego	213
Katakana Sashisuseso/Zajizuzezo	218
Katakana Tachitsuteto/Dajizudedo	223
Katakana Naninuneno	228
Katakana Hahifuheho/Babibubebo/Papipupepo	233
Katakana Mamimumemo	239
Katakana Yayuyo	243
Katakana Rarirurero	246
Katakana Wawiwewo/N/Iteration	252
latin.mp	258
numerals.mp	347
ogonek.mp	358
punct.mp	370
serif.mp	412
 II Shared kyouiku kanji	 419
gradeone.mp	421
gradetwo.mp	493
gradethree.mp	600
gradefour.mp	683
gradefive.mp	765
gradesix.mp	817
 III Other shared kanji	 841
bottomrad.mp	843
leftrad.mp	847

radical.mp	857
rightrad.mp	903
toprad.mp	909
gradeeight.mp	917
gradenine.mp	987
gradeten.mp	1031
rare.mp	1032

IV U+0000 to U+0FFF 1109

tsuku-00.mp	1111
Ascii	1111
Latin-1 Extra Characters	1205
Accented Latin	1226
tsuku-01.mp	1284
Latin Extended A Uppercase	1284
Latin Extended A Lowercase	1339
Latin Extended A Other	1397
tsuku-02.mp	1399
Latin Extended B	1399
Spacing Modifier Letters	1400
tsuku-03.mp	1409
Combining Diacritical Marks	1409

V U+1000 to U+2FFF 1425

tsuku-20.mp	1427
General Punctuation	1427
tsuku-21.mp	1444
Symbols Required By Mes-1	1444
tsuku-24.mp	1450
Circled Numerals	1450
Circled Latin And Zero	1470
Inverted Circled Numerals	1523
Doubly Circled Numerals	1533
One More Inverted Circled Numeral	1543
tsuku-25.mp	1545
Geometric Shapes	1545
tsuku-26.mp	1547

I Ching	1547
tsuku-27.mp	1564
Inverted Circled Numerals	1564
tsuku-2e.mp	1575
Cjk Radicals Supplement	1575
tsuku-2f.mp	1616
Kangxi Radicals	1616
Ideographic Description Characters	1736

VI U+3000 to U+4DFF	1751
tsuku-30.mp	1753
Ideographic Symbols And Punctuation	1753
Hiragana	1773
Katakana	1818
tsuku-31.mp	1865
Phonetic Extensions For Ainu	1865
tsuku-32.mp	1882
Circled Numerals	1882
Circled Katakana	1912
tsuku-34.mp	1960
tsuku-35.mp	1961
tsuku-4d.mp	1962
I Ching	1962

VII U+4E00 to U+56FF	2027
tsuku-4e.mp	2029
tsuku-4f.mp	2062
tsuku-50.mp	2118
tsuku-51.mp	2135
tsuku-52.mp	2150
tsuku-53.mp	2197
tsuku-54.mp	2217
tsuku-55.mp	2226
tsuku-56.mp	2231

VIII	U+5700 to U+5EFF	2239
	tsuku-57.mp	2241
	tsuku-58.mp	2361
	tsuku-59.mp	2427
	tsuku-5a.mp	2442
	tsuku-5b.mp	2448
	tsuku-5c.mp	2502
	tsuku-5d.mp	2525
	tsuku-5e.mp	2534
IX	U+5F00 to U+65FF	2567
	tsuku-5f.mp	2569
	tsuku-60.mp	2634
	tsuku-61.mp	2763
	tsuku-62.mp	2815
	tsuku-63.mp	2834
	tsuku-64.mp	2849
	tsuku-65.mp	2852
X	U+6600 to U+69FF	2887
	tsuku-66.mp	2889
	tsuku-67.mp	2972
	tsuku-68.mp	3099
	tsuku-69.mp	3218
XI	U+6A00 to U+79FF	3317
	tsuku-6a.mp	3319
	tsuku-6b.mp	3381
	tsuku-6c.mp	3394
	tsuku-6d.mp	3444
	tsuku-6e.mp	3459
	tsuku-6f.mp	3475
	tsuku-70.mp	3483
	tsuku-71.mp	3489
	tsuku-72.mp	3498
	tsuku-73.mp	3515

tsuku-74.mp	3518
tsuku-75.mp	3522
tsuku-76.mp	3535
tsuku-77.mp	3558
tsuku-78.mp	3567
tsuku-79.mp	3572

XII U+7A00 to U+89FF	3625
tsuku-7a.mp	3627
tsuku-7b.mp	3677
tsuku-7c.mp	3692
tsuku-7d.mp	3698
tsuku-7e.mp	3809
tsuku-7f.mp	3829
tsuku-80.mp	3836
tsuku-81.mp	3850
tsuku-82.mp	3855
tsuku-83.mp	3887
tsuku-84.mp	3901
tsuku-85.mp	3906
tsuku-86.mp	3908
tsuku-87.mp	3910
tsuku-88.mp	3912
tsuku-89.mp	3943

XIII U+8A00 to U+9FFF	3957
tsuku-8a.mp	3959
tsuku-8b.mp	4051
tsuku-8c.mp	4086
tsuku-8d.mp	4106
tsuku-8e.mp	4110
tsuku-8f.mp	4114
tsuku-90.mp	4131
tsuku-91.mp	4157
tsuku-92.mp	4170
tsuku-93.mp	4177
tsuku-95.mp	4181

tsuku-96.mp	4245
tsuku-97.mp	4273
tsuku-98.mp	4277
tsuku-99.mp	4340
tsuku-9a.mp	4345
tsuku-9b.mp	4350
tsuku-9c.mp	4353
tsuku-9e.mp	4355
tsuku-9f.mp	4360

XIV U+A000 to U+10FFFF	4363
tsuku-f7.mp	4365
Latin Small Caps	4365
tsuku-f9.mp	4392
tsuku-ff.mp	4394
Full-Width Forms	4394
Half-Width Punctuation	4420
Half-Width Katakana	4423
tsuku-1f1.mp	4487
Squared Latin	4487
Inverse Circled Latin	4513
Inverse Squared Latin	4539
tsuku-200.mp	4566
tsuku-20a.mp	4567
tsuku-21c.mp	4568
tsuku-295.mp	4569
tsuku-f17.mp	4570
Combining Dots For I Ching	4570
Miscellaneous	4579
Tomoe Ornaments	4585
Heavy Metal Umlaut	4593
Genjimon	4610
tsuku-ff0.mp	4664
Fraction Numerators	4664
tsuku-ff1.mp	4671
Fraction Denominators	4671

XV Mandeubsida core	4673
hangul.mp	4675
Jamo Combining Operations	4677
jamo-basic.mp	4685
Filler Jamo	4685
Sios/Cieuc/Chieuch Family	4685
Kiyek	4688
Nieun	4689
Tikeut	4691
Rieul	4693
Mieum	4694
Pieup	4695
Sios	4696
Ieung	4696
Cieuc	4698
Chieuch	4698
Khieukh	4699
Thieuth	4700
Phieuph	4701
Hieuh	4703
Mixed Tails	4704
Vowels	4705
jamo-extra.mp	4714
Pansios	4717
Yesieung	4717
Yeorinhieuh	4718
Chitueum And Ceongchieum Variants	4720
Kapyeoun Variants	4721
hglxtb.mp	4722
Hangul Extension B	4722
Hangul Jungseong (Vowel) Jamo Extension B	4722
Hangul Jongseong (Tail) Jamo Extension B	4745
mande-bt.mp	4795
mande-do.mp	4796
mande-sm.mp	4797
hglpage.mp	4798
mande-ll.mp	4800
Hangul Choseong (Lead) Jamo	4800
Hangul Jungseong (Vowel) Jamo	4895

Hangul Jongseong (Tail) Jamo	4966
mande-3l.mp	5055
Hangul Compatibility Jamo	5055
mande-a9.mp	5149
Hangul Choseong (Lead) Jamo Extended A	5149
mande-ac.mp	5179

XVI Mandeubsida alternates	5233
mande-ff2.mp	5235
Hangul Jungseong (Vowel) Jamo	5235
Hangul Jungseong (Vowel) Jamo Extension B	5306
mande-ff3.mp	5330
Hangul Choseong (Lead) Jamo	5330
mande-ff4.mp	5455
Hangul Choseong (Lead) Jamo	5455
mande-ff5.mp	5580
Hangul Choseong (Lead) Jamo	5580
mande-ff6.mp	5705
Hangul Choseong (Lead) Jamo	5705
mande-ff7.mp	5830
Hangul Choseong (Lead) Jamo	5830
Hangul Choseong (Lead) Jamo Extended A	5925

XVII TsuIta	5955
tsuita-common.mp	5957
tsuita-at.mp	5968
tsuita-so.mp	5969
Additional Proofs	5971

XVIII Blackletter Lolita	6025
bll.mp	6027
bll-co.mp	6032
pentacross.mp	6033
Utilities For Pentagrams And Crosses	6033
bll-f5c.mp	6035
Pentagrams	6035

XIX	Kazoemashou	6055
	kazoe-se.mp	6057
	kazoe-ld4.mp	6060
	Math Bold	6060
	Math Bold Italic	6067
	Math Italic	6074

Volume XVII

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, 2013, 2021 Matthew Skala
4 %
5-29 [Standard copyright notice]
30
31 inclusion_lock(tsuitacommon);
32
33 

---


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

---


64
65 vardef latin.lowa =
66   latin.single_lowa;
67 enddef;
68
69 vardef latin.lowaogonek =
70   push_pbox_toexpand("latin.lowaogonek");
```

```

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

```



```

119   save xsp,ysp;
120   xsp:=sp;
121   latin.lowf;
122   set__boserif(0,6,whatever);
123   set__botip(0,6,0);
124   ysp:=sp;
125
126   numeric x[],y[];
127   x1-x2=x2-x3=y2-y1;
128   x2=x4=500;
129   y1=y3=0.49[y4,y2];
130   y2=latin_wide_high_r;
131   y4=latin_wide_low_r;
132
133   transform shrinka;
134   (0.5[lcorner get__strokep(0),urcorner get__strokep(0)])
135     transformed shrinka=0.5[z3,z1];
136   (0.5[lrcorner get__strokep(0),urcorner get__strokep(0)])
137     transformed shrinka=0.71[z3,z1];
138   (0.5[ulcorner get__strokep(0),urcorner get__strokep(0)])
139     transformed shrinka=z2+(0.07-1)*0.29*(x1-x3);
140   sp:=xsp;
141   tsu_xform(shrinka shifted (-10,0))(sp:=ysp);
142
143   z5=point infinity of get__strokep(0);
144   y6=y part lrcorner get__strokep(0);
145   x6=0.8[x2,x1];
146   replace__strokep(0)((subpath (0,length(oldp)-1) of oldp)
147     ..{curl 1}z5{curl 1}..z6..
148     (subpath (0,3.85) of (z1..z2..z3..z4..cycle)));
149   replace__strokep(0)(insert__nodes(oldp)((length oldp-4.5)));
150   replace__strokeq(0)(oldq-(1.6,1.6)-(1.6,1.6)-(1.6,1.6)-
151     (1.6,1.6)-(1.6,1.6)-(0,0));
152   endgroup;
153   expand__pbox;
154   enddef;
155
156
157
158 if do_italic_hook:
159
160 vardef latin.lowf =
161   push__pbox__toexpand("latin.lowf");
162   (x2-x1)=290;
163   x5=x6=490=0.52[x1,x2];
164   x3-x5=2*(y4-y5);
165   x4=0.38[x5,x3];
166   x6-x7=1.15*(x4-x5);

```

```

167 x6-x8=1.15*(x3-x5);
168
169 y1=y2=0.75[latin_wide_low_h,latin_wide_xheight_h];
170 y5=0.52[latin_wide_xheight_h,y4];
171 y3=0.73[latin_wide_xheight_h,y4];
172 y4=latin_wide_high_r;
173 y7=latin_wide_desc_r;
174 y6-y7=1.10*(y4-y5);
175 y8-y7=1.10*(y4-y3);
176
177 push_stroke(z1-z2,(1.6,1.6)-(1.6,1.6));
178
179 push_stroke(z3{curl 0.6}..z4{left}..
180     {dir 268}z5{down}-z6{dir 268}..
181     {left}z7.{curl 0.6}z8,
182     (1.6,1.6)-(1.6,1.6)-(1.6,1.6)-(1.6,1.6)-(1.6,1.6)-(1.6,1.6));
183 replace_strokep(0)(subpath (0.234,77) of oldp);
184 expand_pbox;
185 enddef;
186
187 vardef latin.lowi =
188   push_pbox_toexpand("latin.lowi");
189   x2=x3=x4=450;
190
191   y2=latin_wide_xheight_r;
192   y3=latin_wide_low_r;
193   y4=0.5[y2,latin_wide_high_r]+mbrush_width;
194
195   push_stroke(z2-z3,(1.6,1.6)-(1.6,1.6));
196   set_boserif(0,1,11);
197
198   push_lcblob(fullcircle rotated 45 scaled (mbrush_width*2.7*15)
199     shifted (z4 transformed tsu_rescale_xform)
200     transformed inverse tsu_rescale_xform);
201   expand_pbox;
202 enddef;
203
204 vardef latin.lowj =
205   push_pbox_toexpand("latin.lowj");
206   x3=x5=450;
207   (x2-x5)=0.25*italic_hook_radius;
208
209   y2=latin_wide_xheight_r;
210   y3=latin_wide_low_v;
211   y5=0.5[y2,latin_wide_high_r]+mbrush_width;
212
213   z4=z3+(-210,-140);
214

```

```

215 push_stroke((z2-(z3+(0,55)))..{curl 0.8}z4,
216   (1.6,1.6)-(1.6,1.6)-(1.6,1.6));
217 set__boserif(0,0,11);
218
219 push_lcblob(fullcircle scaled (mbrush_width*2.7+15)
220   shifted (z5 transformed tsu_rescale_xform)
221   transformed inverse tsu_rescale_xform);
222 expand_pbox;
223 enddef;
224
225 vardef latin.lowv =
226   push_pbox__toexpand("latin.lowv");
227   0.4[x1,x4]=x3=480;
228   x1=x2;
229   x5=0.8[x1,x4];
230
231   y1=y5=latin_wide_xheight_v;
232   y3=latin_wide_low_h;
233   y2=0.333[y3,y1];
234   y4=0.45[y3,y5];
235
236   (x4-x1)=(y1-y3)*0.65;
237
238   push_stroke(z1..z2..{dir -30}z3{dir 30}..z4..z5,
239     (1.6,1.6)-(1.6,1.6)-(1.6,1.6)-(1.6,1.6)-(1.6,1.6));
240   set__botip(0,2,0);
241   set__boserif(0,0,11);
242
243   expand_pbox;
244 enddef;
245
246 vardef latin.loww =
247   push_pbox__toexpand("latin.loww");
248   0.4[x1,x4]=x3;
249   x1=x2;
250   x5=x4;
251   0.4[x5,x8]=x7;
252   x5=x6;
253   x9=0.8[x5,x8];
254   0.5[x1,x8]=500;
255
256   y1=y5=y9=latin_wide_xheight_v;
257   y3=y7=latin_wide_low_h;
258   y2=y6=0.333[y3,y1];
259   y4=y8=0.45[y3,y5];
260
261   (x4-x1)=(y1-y3)*0.55;
262   (x8-x5)=(y1-y3)*0.60;

```

```

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

```

```

311     replace_strokep(-1)(oldp shifted (alternate_adjust*left/2));
312     replace_strokep(0)(oldp shifted (alternate_adjust*right/2));
313     fi;
314     expand__pbox;
315 enddef;
316
317 vardef latin.lowz =
318     push_pbox_toexpand("latin.lowz");
319     y2=y4=latin_wide_xheight_h;
320     y5=y7=latin_wide_low_h;
321     y2-y1=y8-y7=0.15*(y4-y5);
322     y4-y3=y6-y5=0.09*(y4-y5);
323
324     x1=x5;
325     x4=x8;
326     (x1+x4)/2=500;
327     (x4-x1)=(y4-y5)*0.92;
328     x2=x6=0.25[x1,x4];
329     x3=x7=0.75[x1,x4];
330
331     push_stroke(z1..z2..z3..z4-z5..z6..z7..z8,
332         (1.6,1.6)-(1.6,1.6)-(1.6,1.6)-(1.6,1.6)-
333         (1.6,1.6)-(1.6,1.6)-(1.6,1.6)-(1.6,1.6));
334     set_botip(0,4,0);
335     set_botip(0,5,0);
336     expand__pbox;
337 enddef;
338
339
340
341 def tsu_render_segment(expr i,p,q,b) =
342     begingroup
343         save start_time,end_time,niba,nibb,nibc,myp,pext,x,y;
344         numeric start_time,end_time,x,y;
345         path myp,niba,nibb,nibc,pext,glyph;
346
347         myp:=p;
348
349         niba:=fix_nib(obstackna.bosize[i]*tsu_brush_max[b]
350             *tsu_brush_shape[b],
351             obstackna.bosize[i]*tsu_brush_max[b]
352             *tsu_brush_shape[b],
353             0);
354
355         if known obstacknaa.boserif[i][floor(ltime[0]+0.5)]:
356             if obstacknaa.boserif[i][floor(ltime[0]+0.5)]=11:
357                 start_time:=xpart (myp intersectiontimes (fullcircle
358                     scaled (italic_hook_radius*(obstackna.bosize[i]/100)*2)

```

```

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

```

```

407         t>(length obstackp[i])-1,b);
408     endfor;
409     pen_stroke(for l=0 step 1 until length(myp):
410         if sharp_corners and known obstacknaa.botip[i][itime[l]]:
411             tip(obstacknaa.botip[i][itime[l]])(l)
412         else:
413             tsu_brush_opt(mytip[l])(l)
414         fi
415     endfor)(myp)(glyph);
416     glstk[ngls]:=regenerate(glyph);
417     ngls:=ngls+1;
418     for l=0 step 1 until length(myp):
419         si:=floor (itime[l]+0.5);
420         if (abs(ltime[l]-si)<0.05) and known obstacknaa.boserif[i][si]:
421             tsu_serif.choose(obstacknaa.boserif[i][si],
422                 point l of myp,direction l of myp,l,
423                 obstackna.bosize[i],tsu_brush_tip_size(l,q,b),b);
424             write ("SERIF "&(decimal obstacknaa.boserif[i][si])&" "&
425                 (decimal xpart point l of myp)&"&
426                 (decimal ypart point l of myp)) to "proof.prf";
427         fi;
428     endfor;
429 endgroup;
430 enddef;
431
432 def tsu_rescale_xform =
433     begingroup
434         save t,st,cp,xadj;
435         transform t,st;
436         numeric xadj;
437         st:=tsu_slant_xform;
438         t:=st;
439         % check if rescaling is active
440         if (rescale_from.left<>rescale_to.left)
441         or (rescale_from.right<>rescale_to.right): begingroup
442             save i,xa,xb,lf,rf,wf,lt,rt,wt;
443             numeric i,xa,xb,lf,rf,wf,lt,rt,wt;
444             transform t;
445             % find the bounds of the paths
446             if find_stroke(0)<=0:
447                 xa:=0.5[rescale_from.left,rescale_from.right];
448                 xb:=0.5[rescale_from.left,rescale_from.right];
449             else:
450                 xa:=infinity;
451                 xb:=-infinity;
452                 for i=1 upto sp-1:
453                     if obstacktype[i]=otstroke:
454                         xadj:=0;

```

```

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

```



```
503      cp+(100,0) transformed st=cp+(100,0);
504      cp+(0,100) transformed st=cp+(rescale_slant/10,100);
505      t:=t transformed st;
506  endgroup; fi;
507      t
508  endgroup
509 enddef;
510
511 fi;
```

COMM

tsuita-at.mp

```

1 %
2 % TsuIta Atama
3 % Copyright (C) 2012, 2013 Matthew Skala
4 %
5-29 [Standard copyright notice]
30
31 % TSUITA ATAMA
32
33 input preintro.mp;
34
35 familyname:="TsuIta";
36 stylename:="Atama";
37
38 (0,4) transformed tsu_brush_xf.brletter = (4,0.77);
39 (1,1) transformed tsu_brush_xf.brletter = (1,0.73);
40 (4,0) transformed tsu_brush_xf.brletter = (0,0.77);
41
42 tsu_brush_min.brletter:=0.73;
43 tsu_brush_max.brletter:=0.77;
44
45 def tsu_brush_opt(expr n,l) =
46   if rescale_slant>10: cut(n,rel 90)(l) fi enddef;
47 def sharp_corners = (rescale_slant>10) enddef;
48
49 input intro.mp;
50
51 let old_rescale_half = tsu_rescale_half;
52
53 def tsu_rescale_half =
54   old_rescale_half;
55   width_curve:=((-1,-1)-(100,100)).(480,330)..{right}(2000,440);
56 enddef;
57
58 let old_rescale_half_lc = tsu_rescale_half_lc;
59
60 def tsu_rescale_half_lc =
61   old_rescale_half_lc;
62   width_curve:=((-1,-1)-(100,100)).(610,330)..{right}(2000,420);
63 enddef;
64
65 include_late("tsuita-common.mp");
66
67 % overridden per-page later
68 rescale_slant:=190;

```

tsuita-so.mp

```
1 %
2 % TsuIta Soku
3 % Copyright (C) 2012, 2013 Matthew Skala
4 %
5-29 [Standard copyright notice]
30
31 % TSUITA SOKU
32
33 input preintro.mp;
34
35 familyname:="TsuIta";
36 stylename:="Soku";
37
38 mincho:=0.8;
39
40 (0,4) transformed tsu_brush_xf.brletter = (0.0,0.80);
41 (1,1) transformed tsu_brush_xf.brletter = (1.2,0.45);
42 (4,0) transformed tsu_brush_xf.brletter = (4.8,0.80);
43
44 tsu_brush_min.brletter:=0.45;
45 tsu_brush_max.brletter:=0.80;
46 tsu_brush_shape.brletter:=0.60;
47 tsu_brush_angle.brletter:=1;
48
49 tsu_brush_xf.bralternate:=identity xyscaled (1.2,0) shifted (0,0.48);
50 tsu_brush_min.bralternate:=0.48;
51 tsu_brush_max.bralternate:=0.48;
52 tsu_brush_shape.bralternate:=1;
53 tsu_brush_angle.bralternate:=0;
54
55 tsu_brush_xf.brpunct:=whatever_xf;
56 (0,4) transformed tsu_brush_xf.brpunct = (0,0.45);
57 (1,1) transformed tsu_brush_xf.brpunct = (1,0.10);
58 (4,0) transformed tsu_brush_xf.brpunct = (4,0.45);
59
60 tsu_brush_min.brpunct:=0.10;
61 tsu_brush_max.brpunct:=0.45;
62 tsu_brush_shape.brpunct:=1;
63 tsu_brush_angle.brpunct:=0;
64
65 input serif.mp;
66
67 tsu_do_serif[1]:=true;
68 tsu_do_serif[2]:=true;
69 tsu_do_serif[3]:=true;
70
```

SO

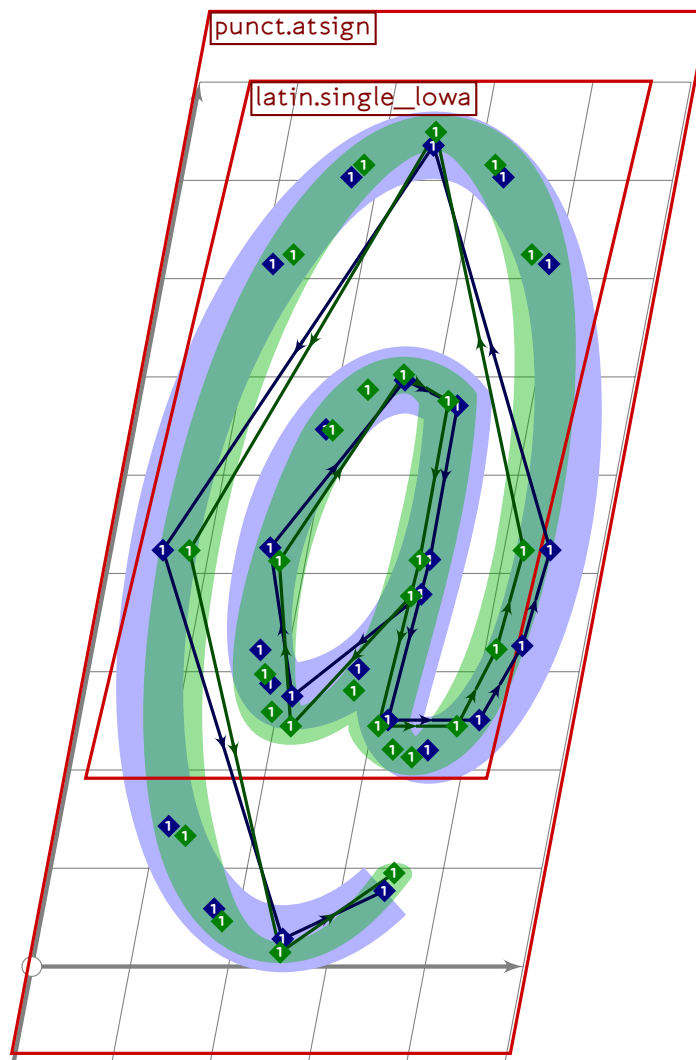
```

71 do_alteration:=true;
72
73 genji_outline:=true;
74
75 input intro.mp;
76
77 let old_render_segment = tsu_render_segment;
78
79 def tsu_render_segment(expr i,p,q,b) =
80   begingroup
81     save start_time,end_time;
82     numeric start_time,end_time;
83   endgroup;
84   old_render_segment(i,p,q,b);
85 enddef;
86
87 let old_rescale_half = tsu_rescale_half;
88
89 def tsu_rescale_half =
90   old_rescale_half;
91   width_curve:=((-1,-1)-(100,100))..(560,310)..{right}(2000,400);
92 enddef;
93
94 let old_rescale_half_lc = tsu_rescale_half_lc;
95
96 def tsu_rescale_half_lc =
97   old_rescale_half_lc;
98   width_curve:=((-1,-1)-(100,100))..(610,310)..{right}(2000,400);
99 enddef;
100
101 do_italic_hook:=true;
102 italic_hook_radius:=170;
103 italic_hook_dir:=25;
104
105 include_late("tsuita-common.mp");
106
107 % overridden per-page later
108 rescale_slant:=190;

```

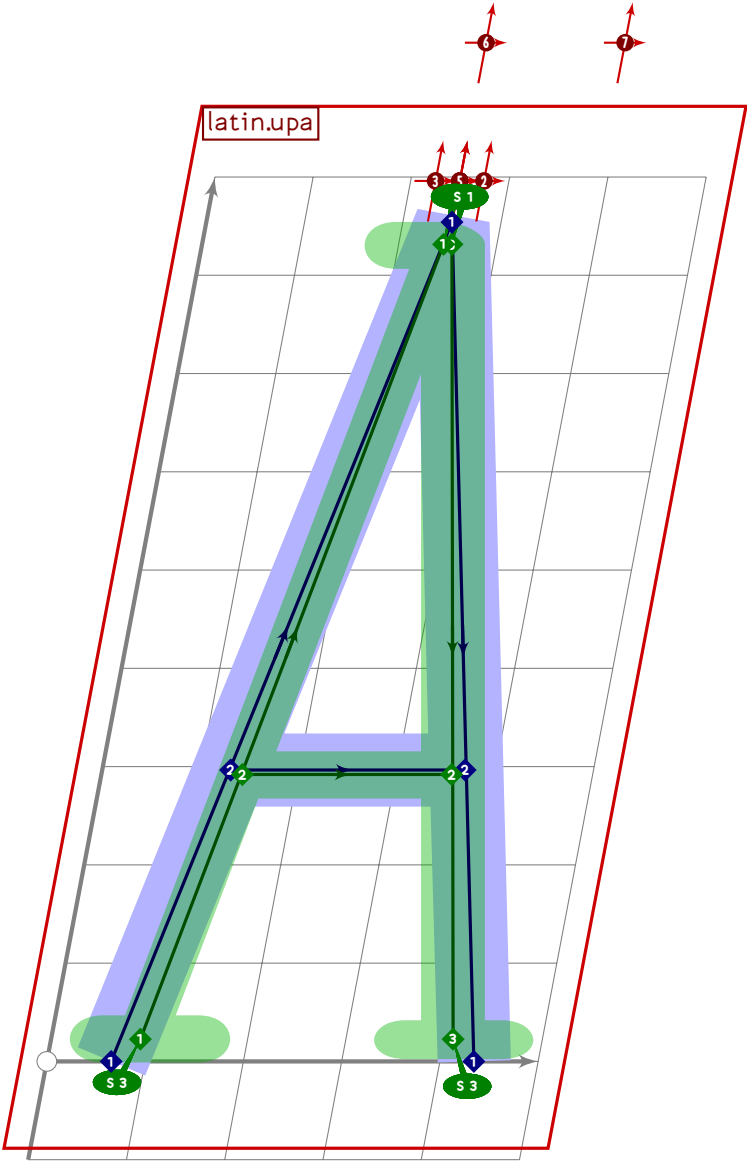
Additional Proofs

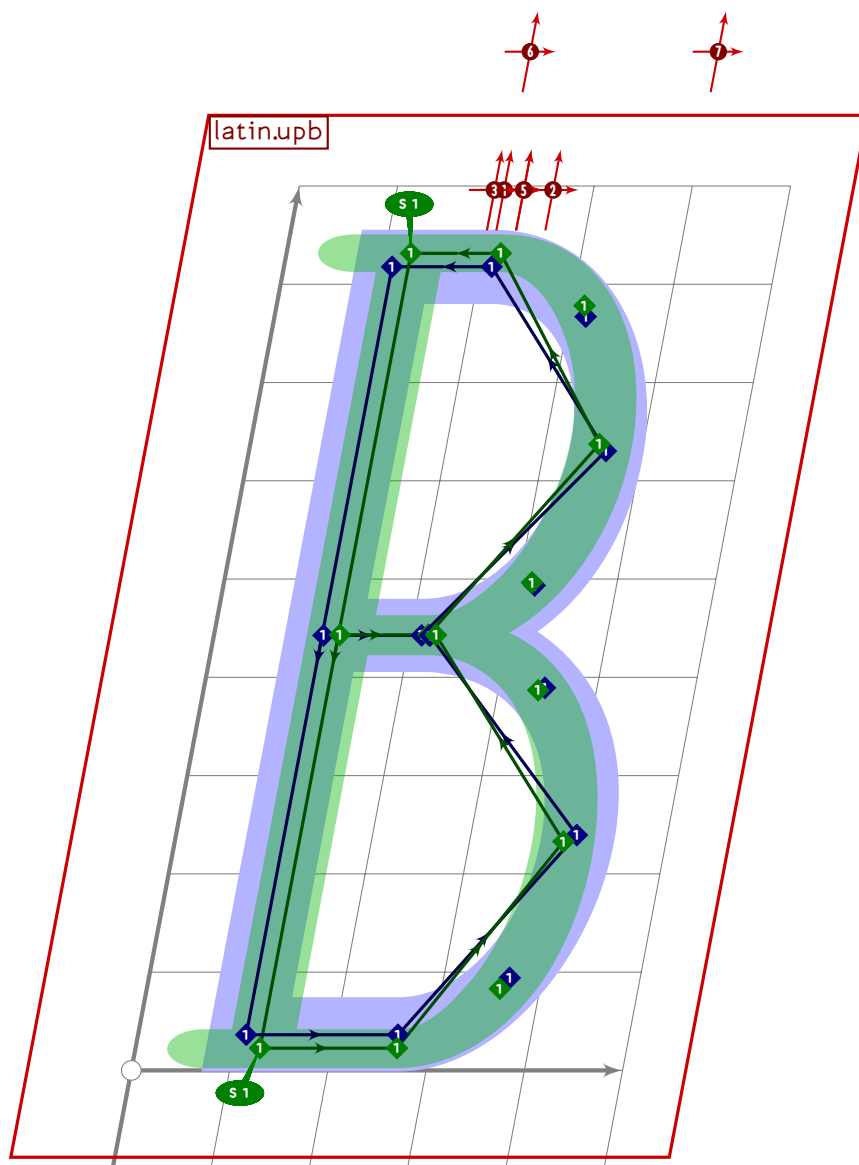
Most glyphs in TsuIta are drawn by code inherited from Tsukurimashou, and proof images for those glyphs are shown with the corresponding Tsukurimashou code. However, different parameters between Tsukurimashou and TsuIta, and overriding code in the tsuita-common.mp file above, cause many of the glyphs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

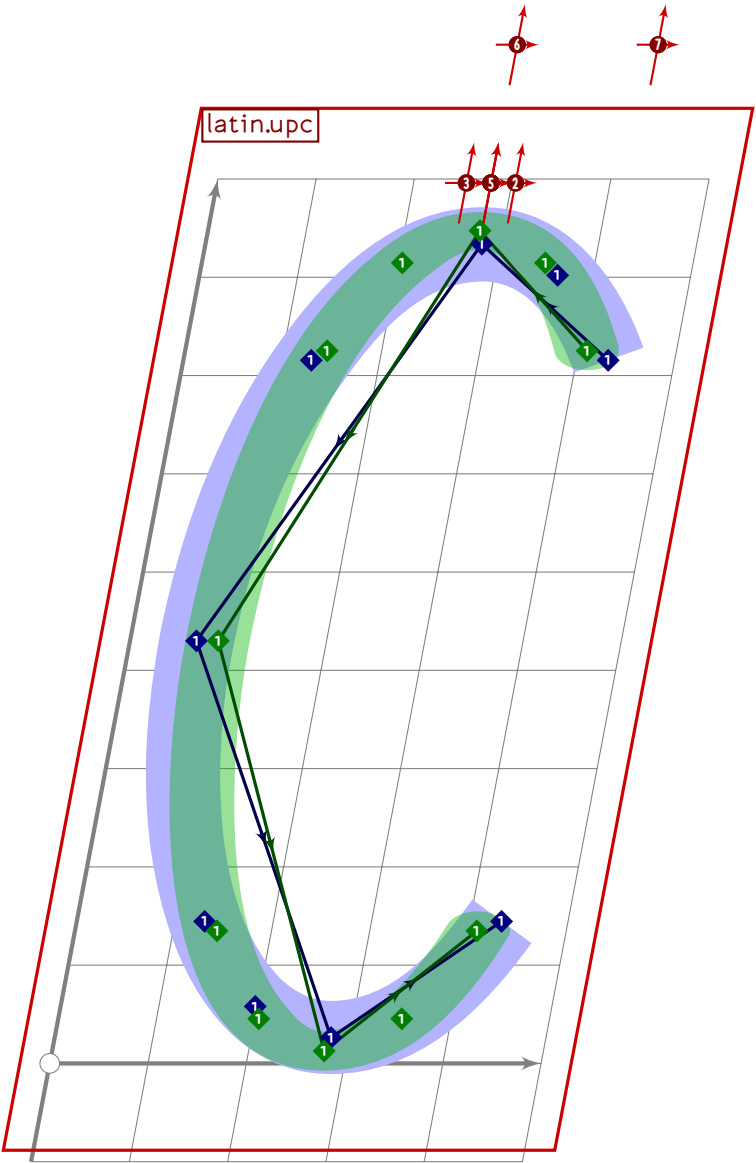
PROO

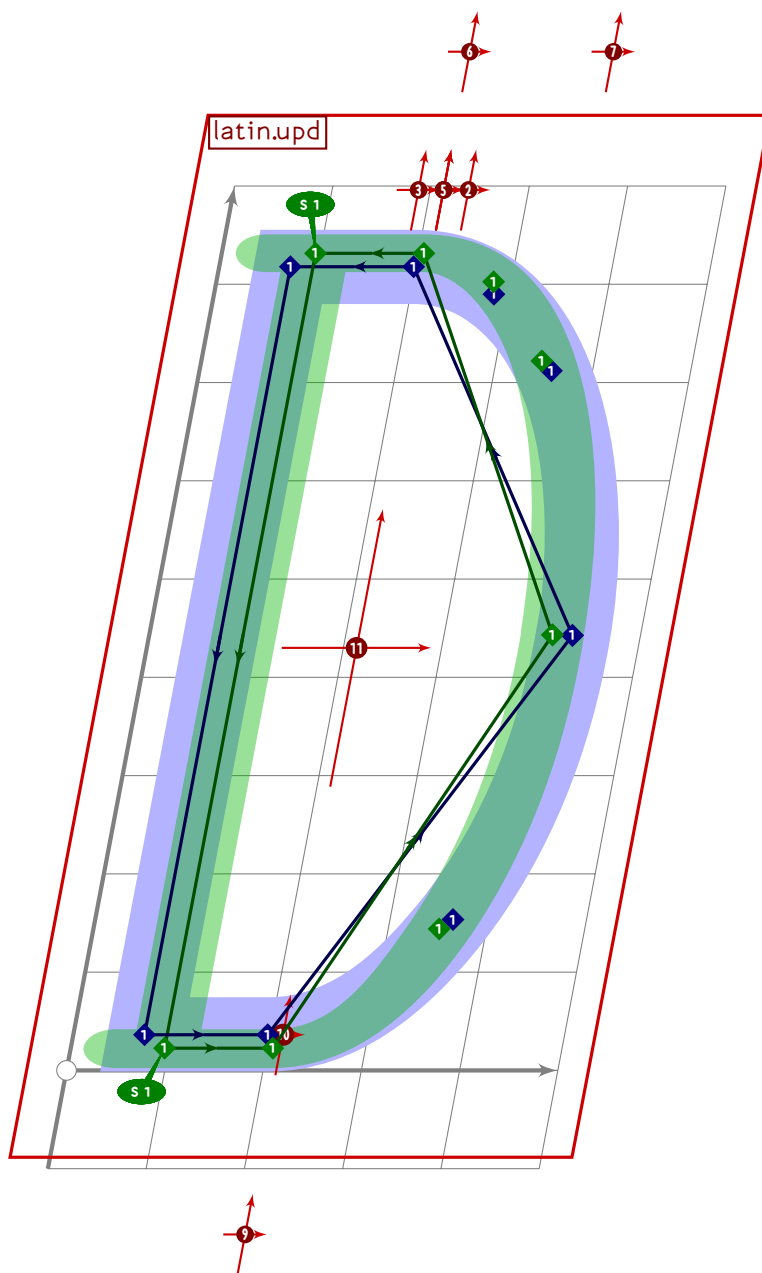




PROO

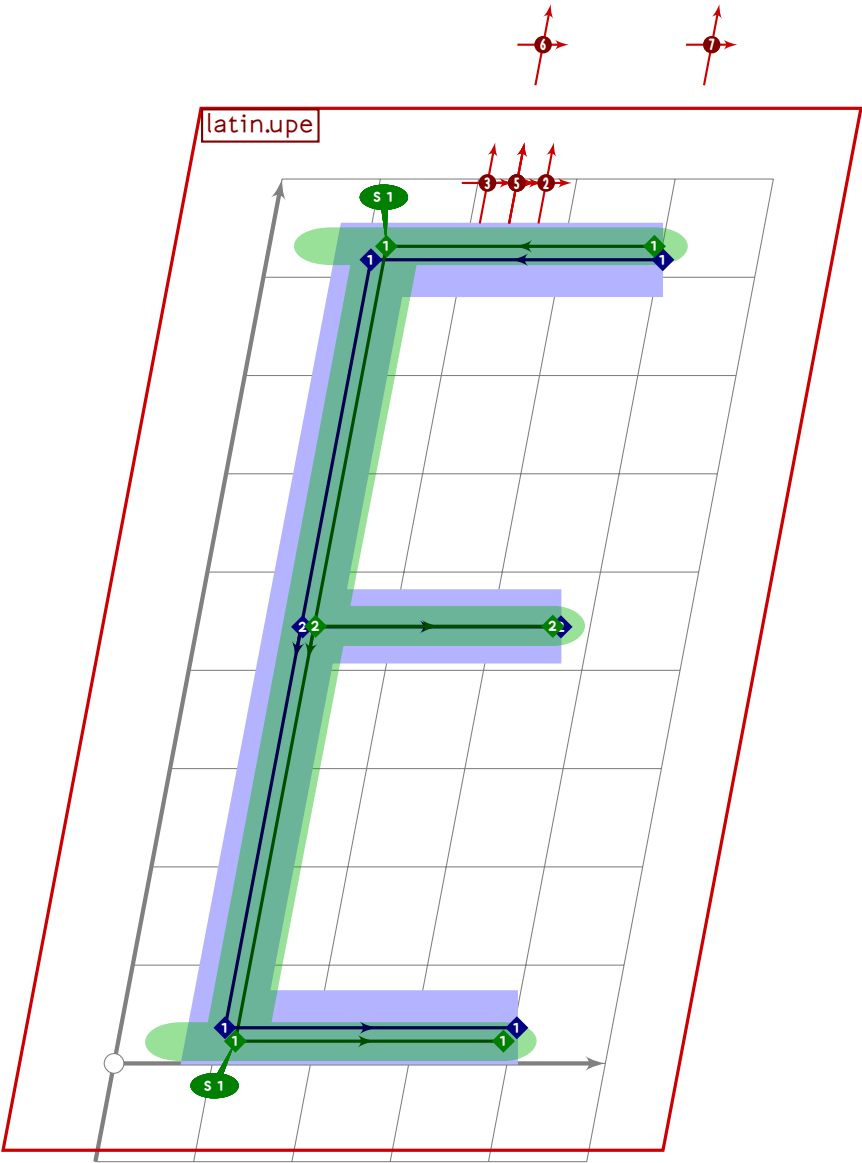
PROO



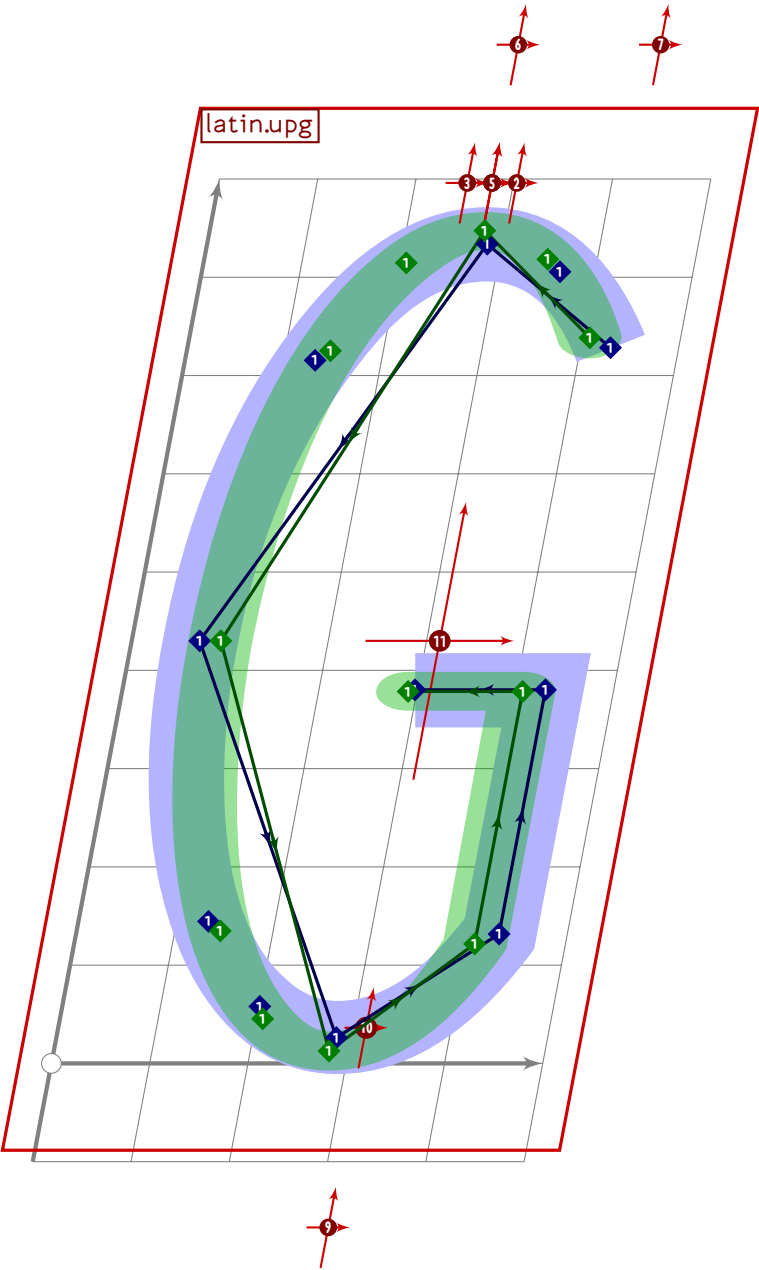


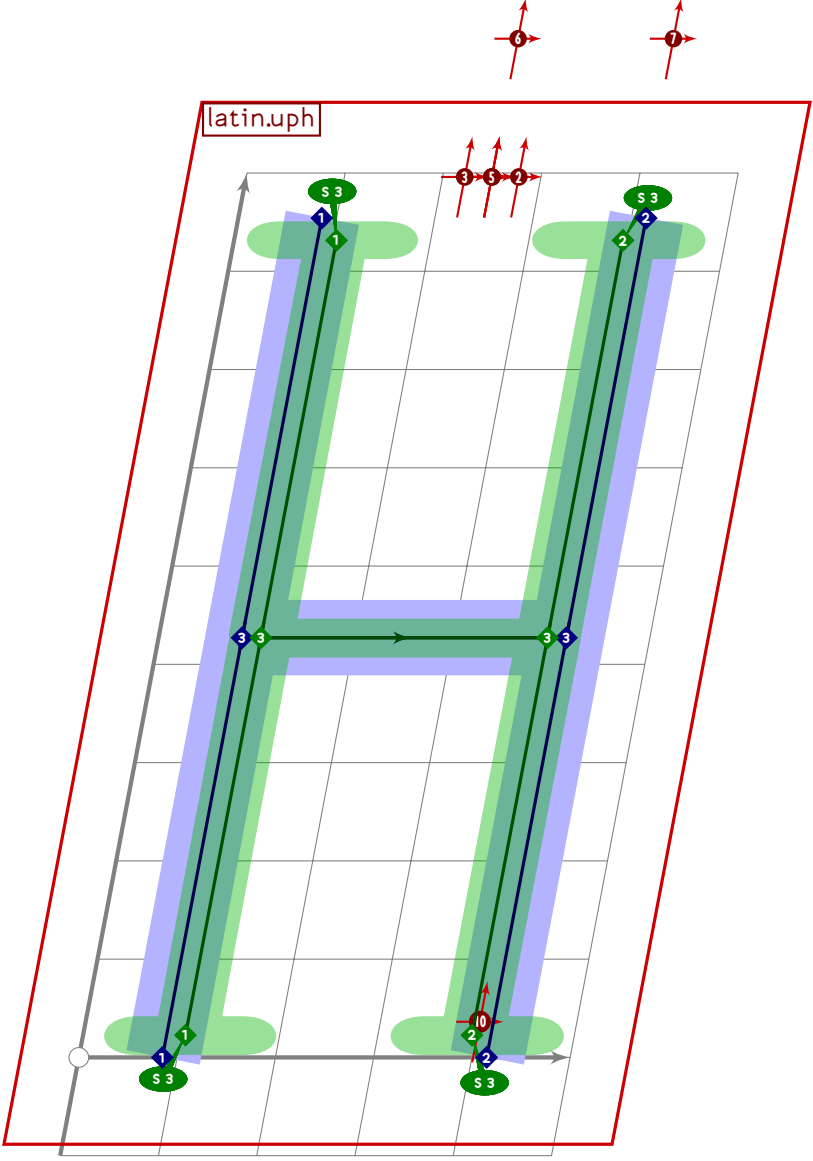
PROO

PROO



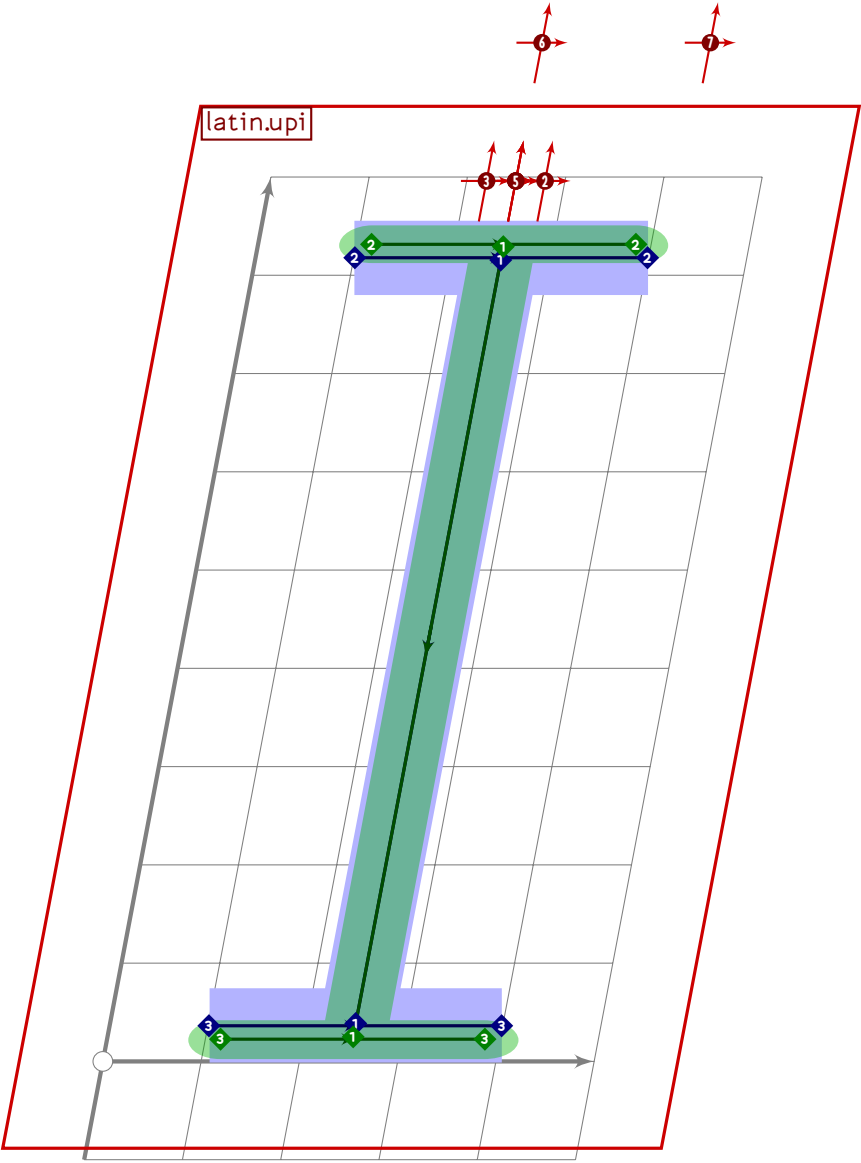
PROO

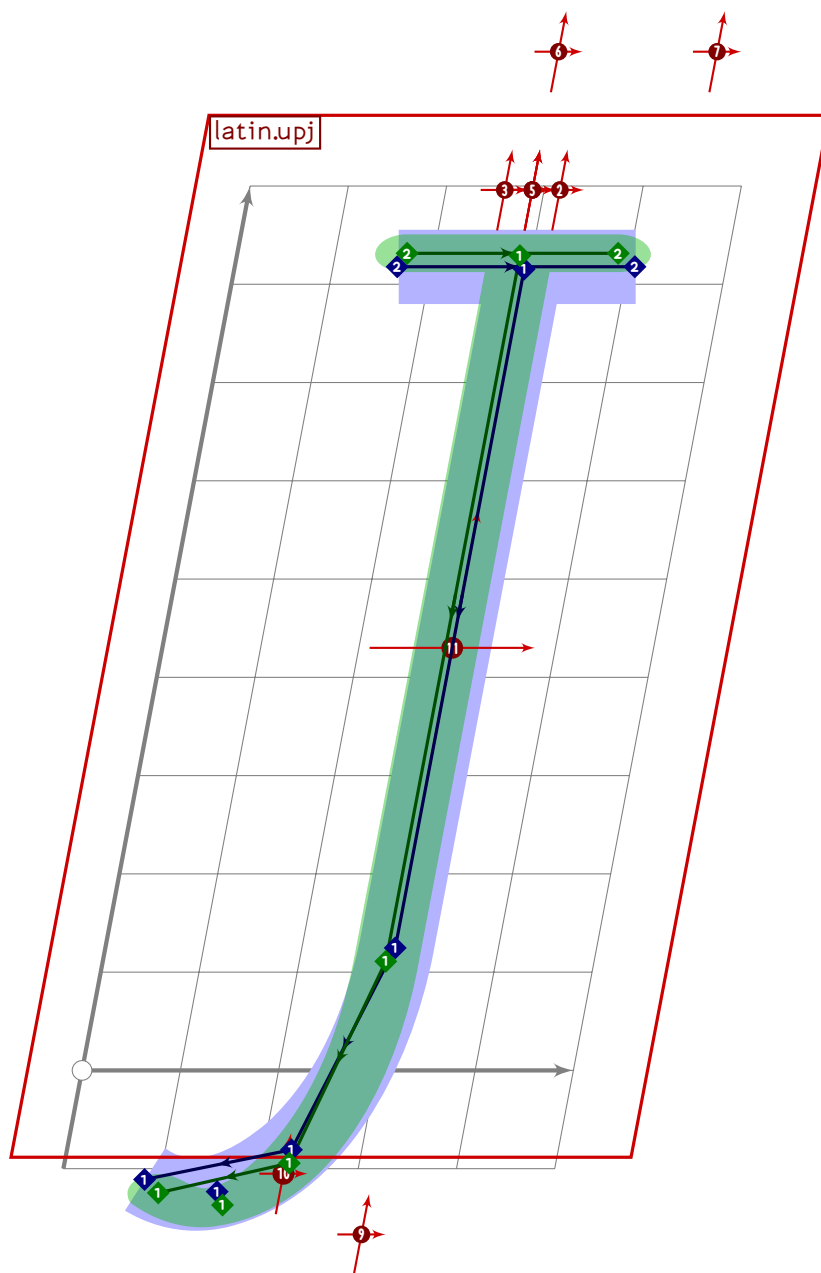




PROO

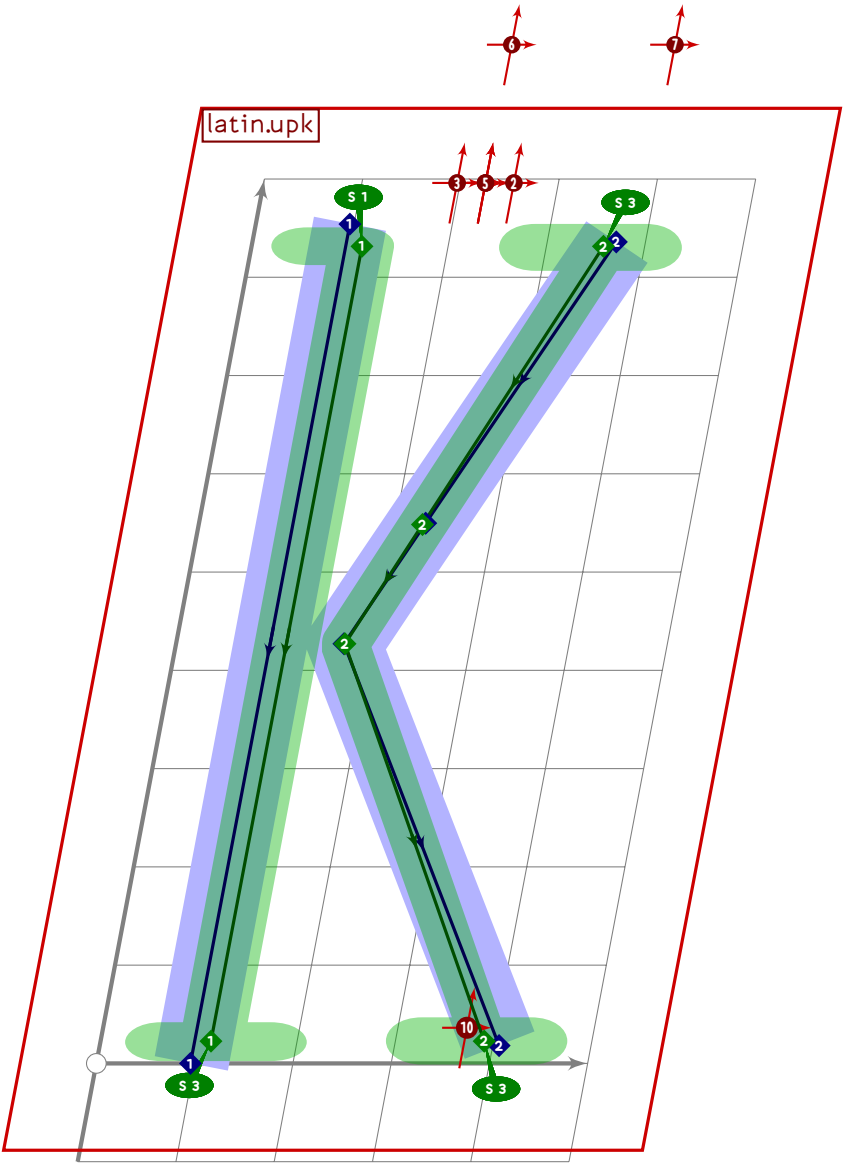
PROO

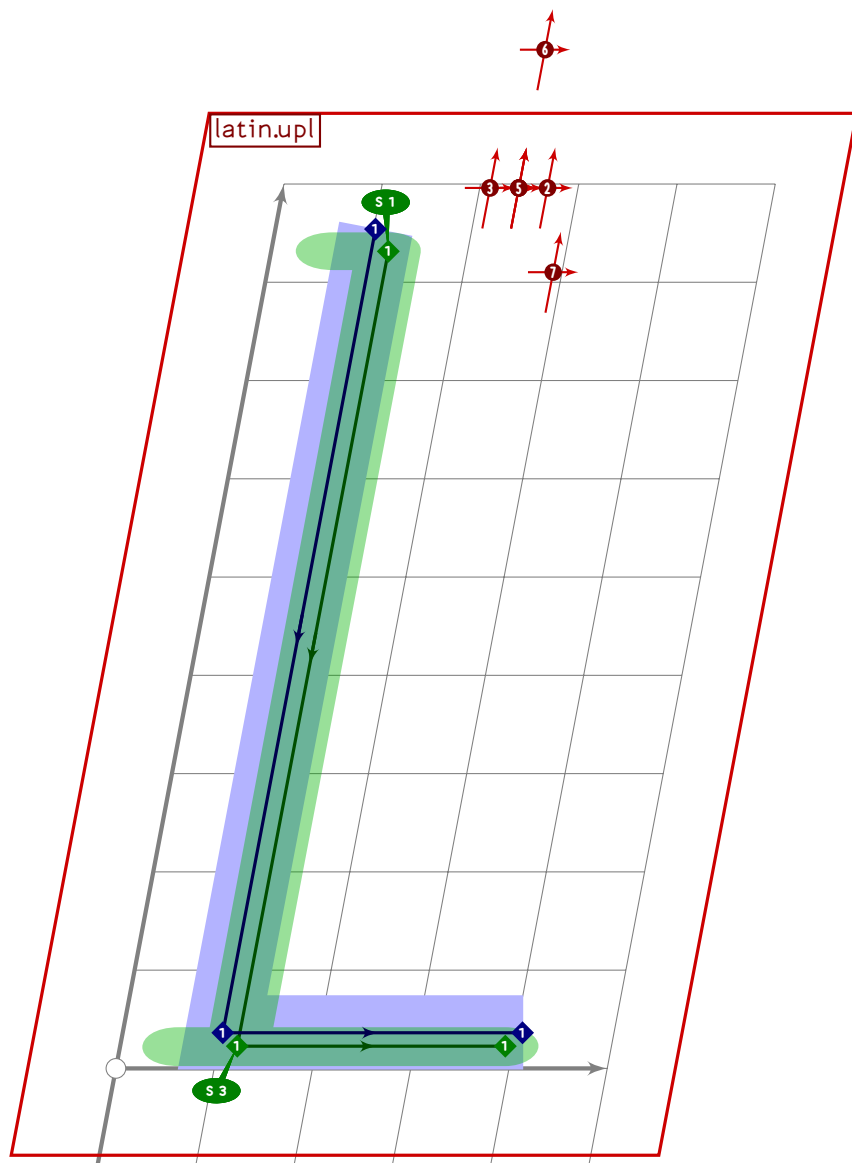




PROO

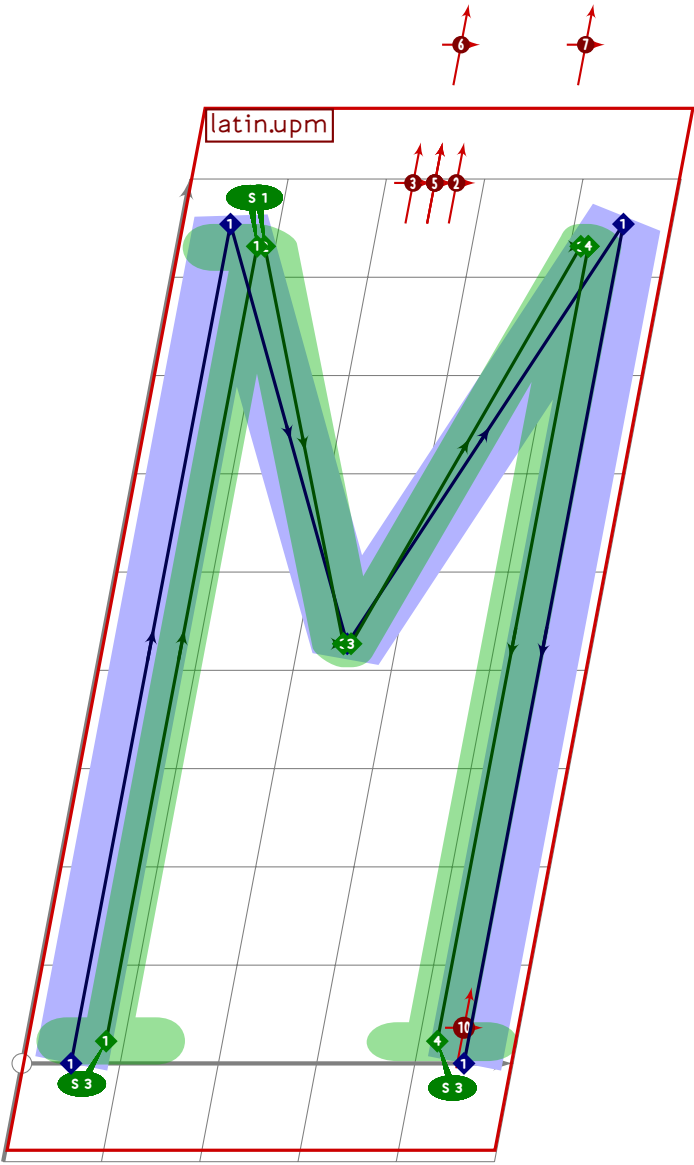
PROO

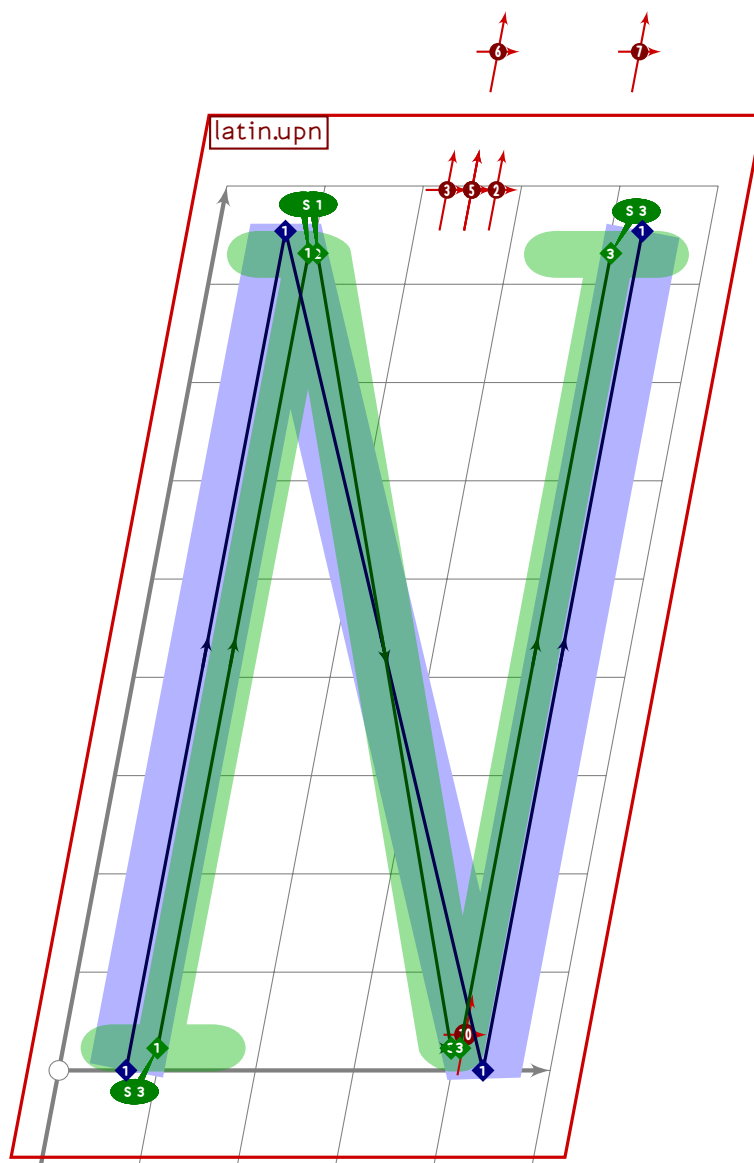




PROO

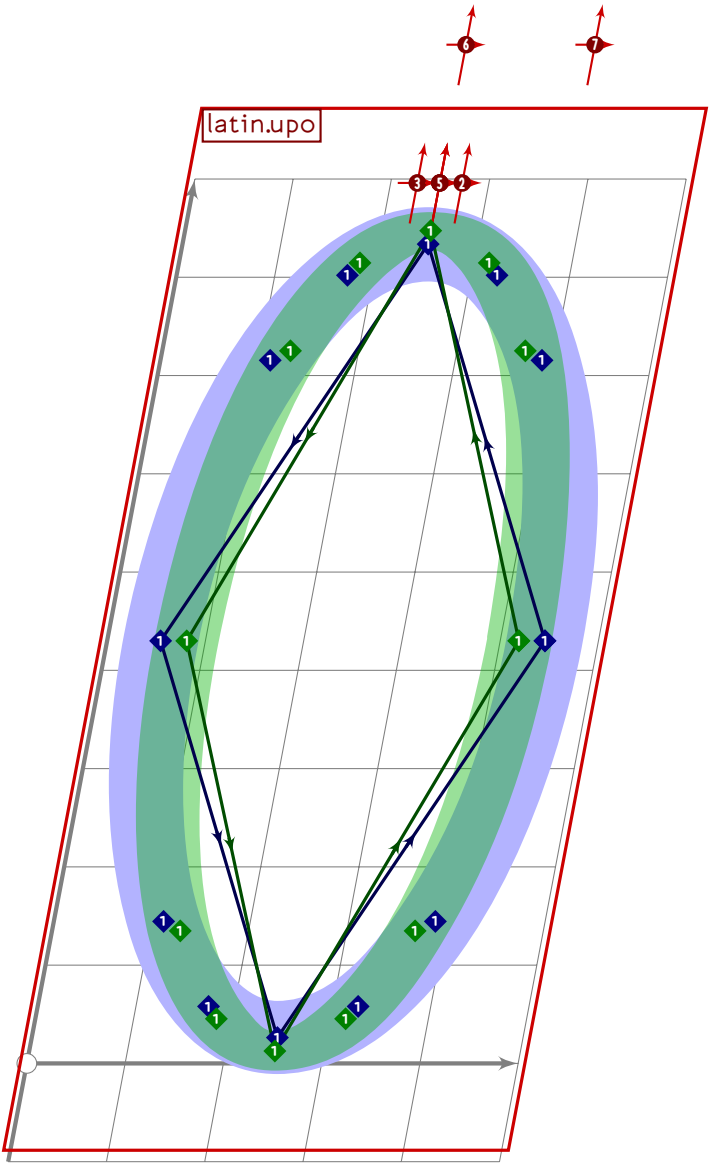
PROO



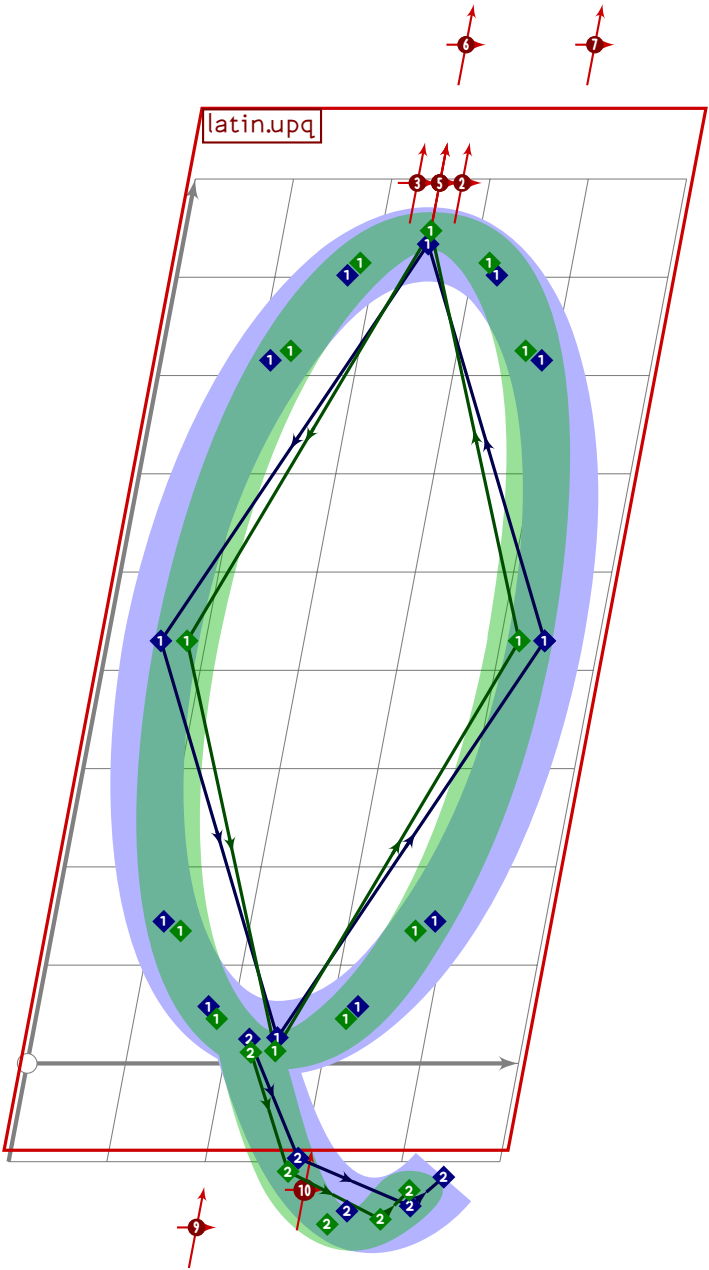


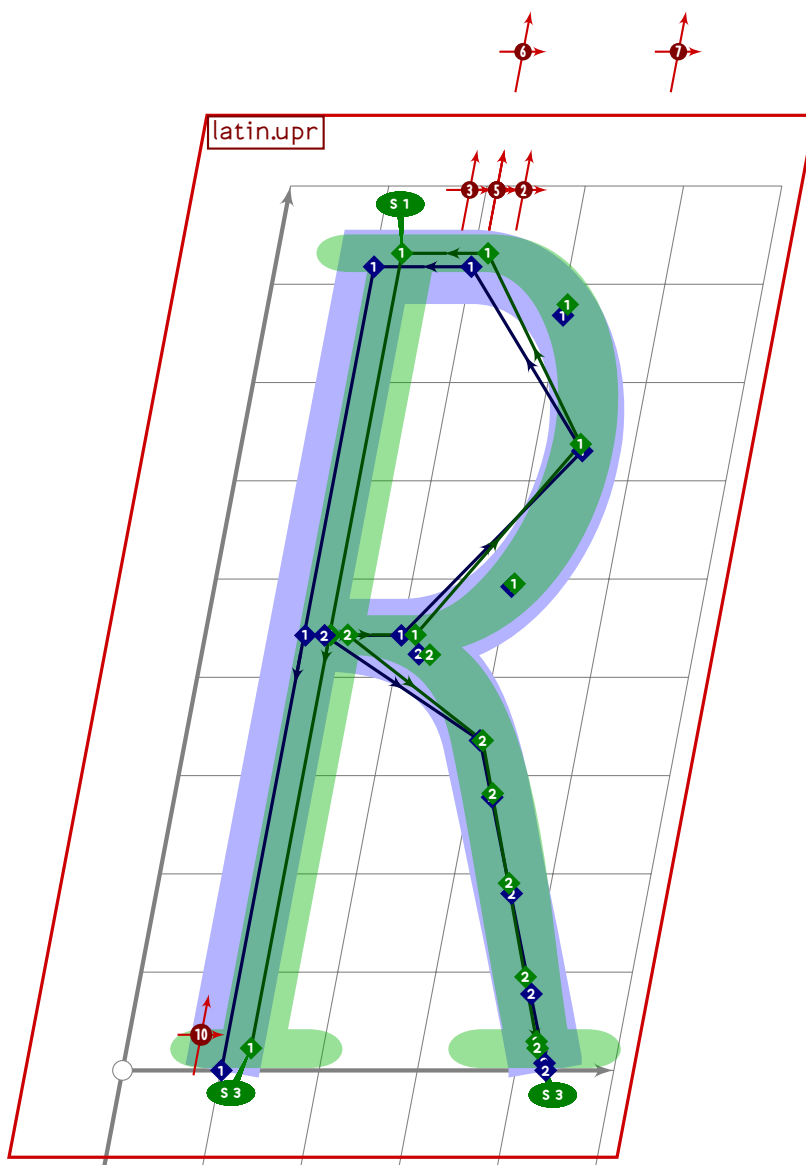
PROO

PROO



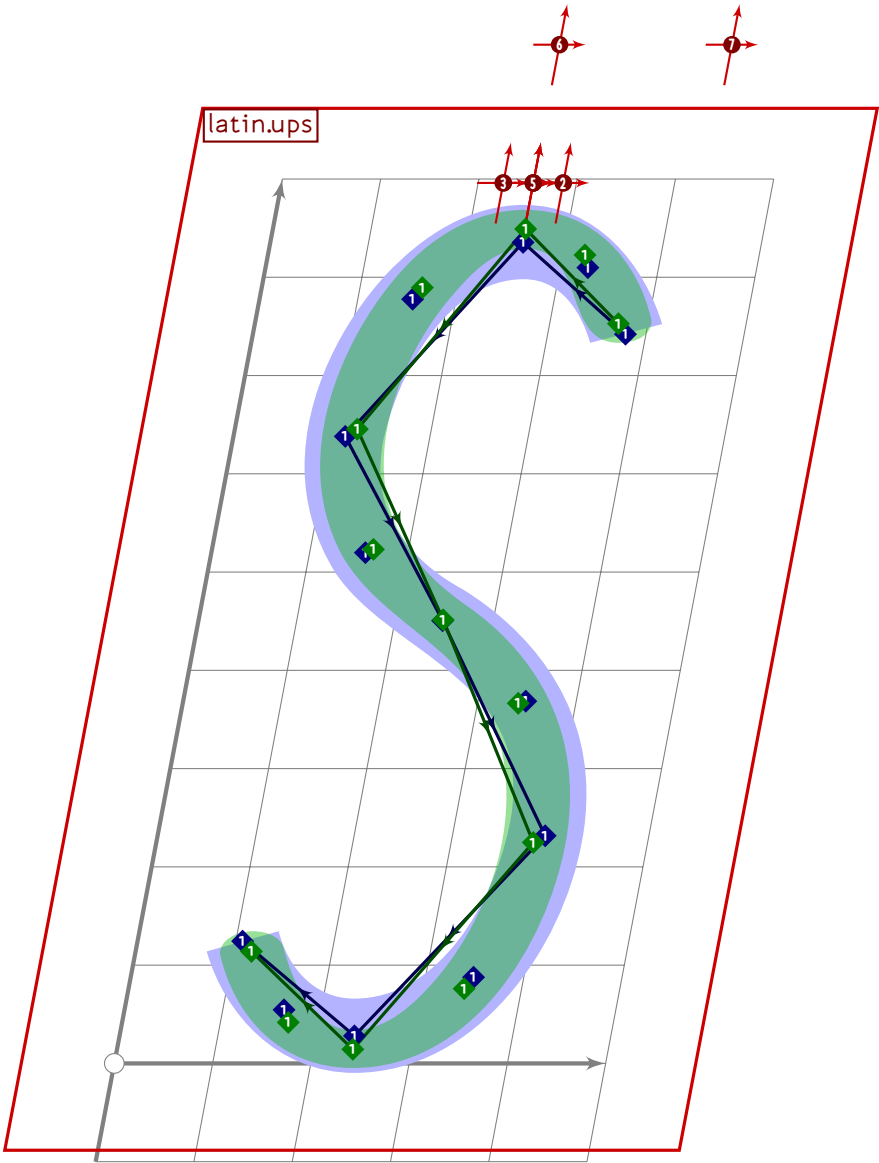
PROO

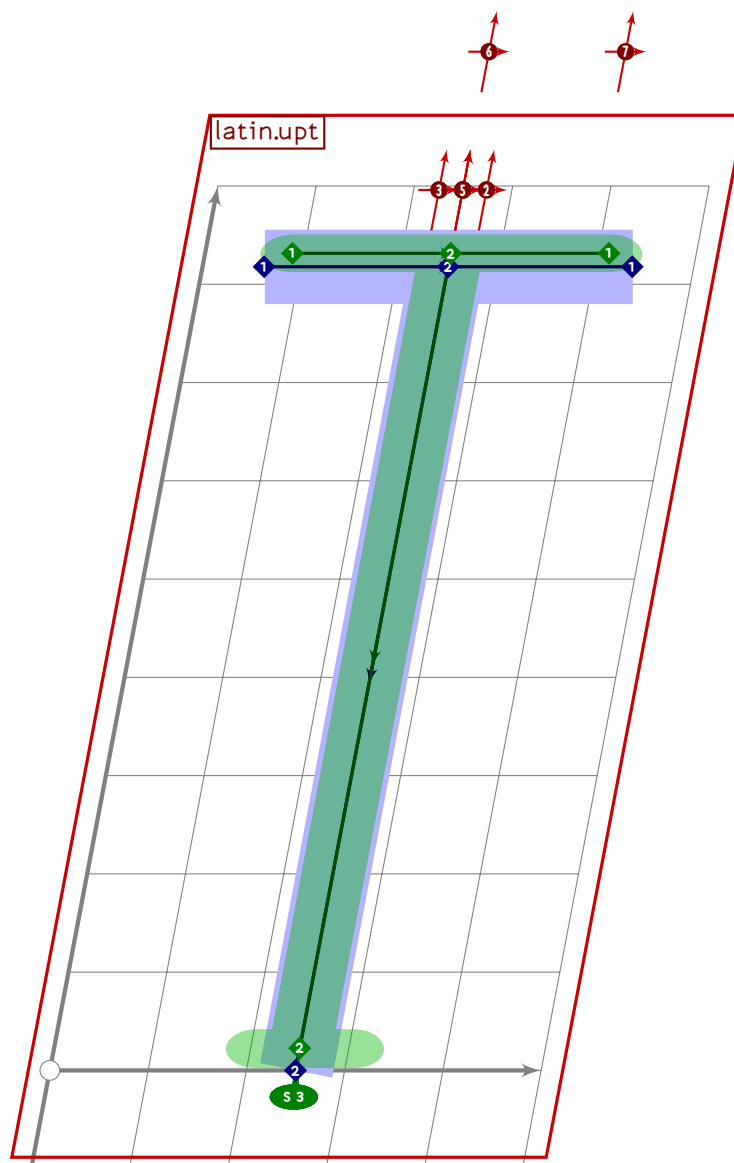




PROO

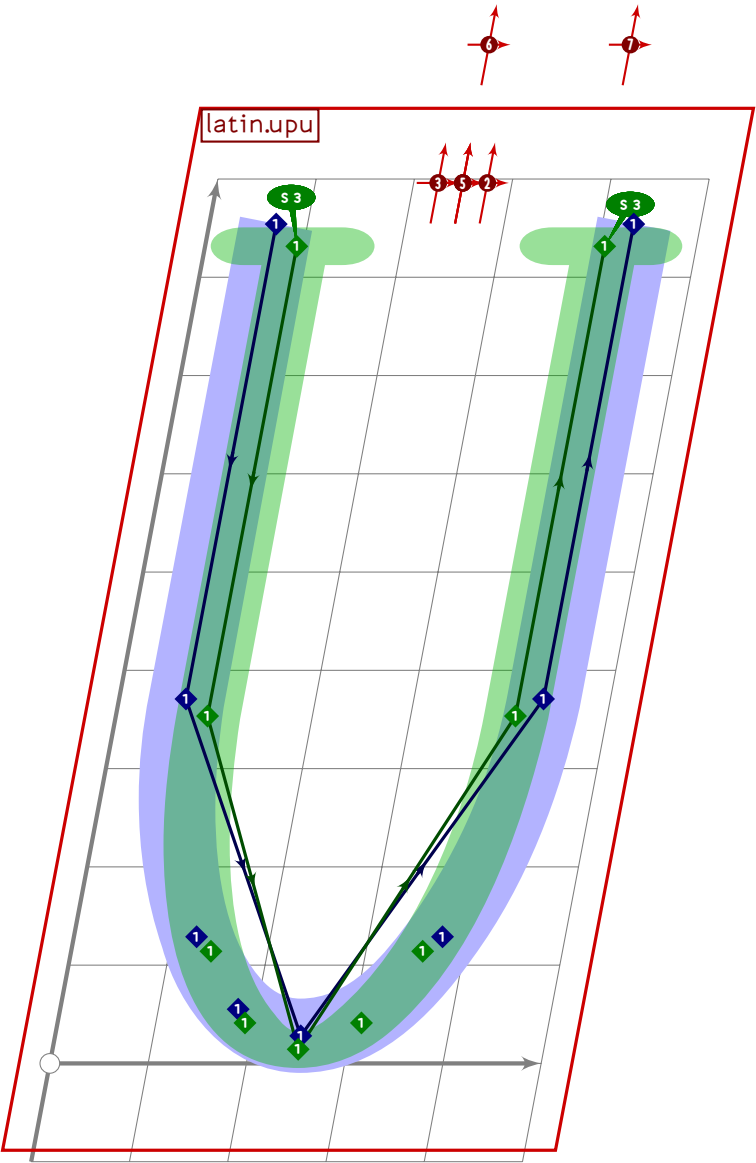
PROO

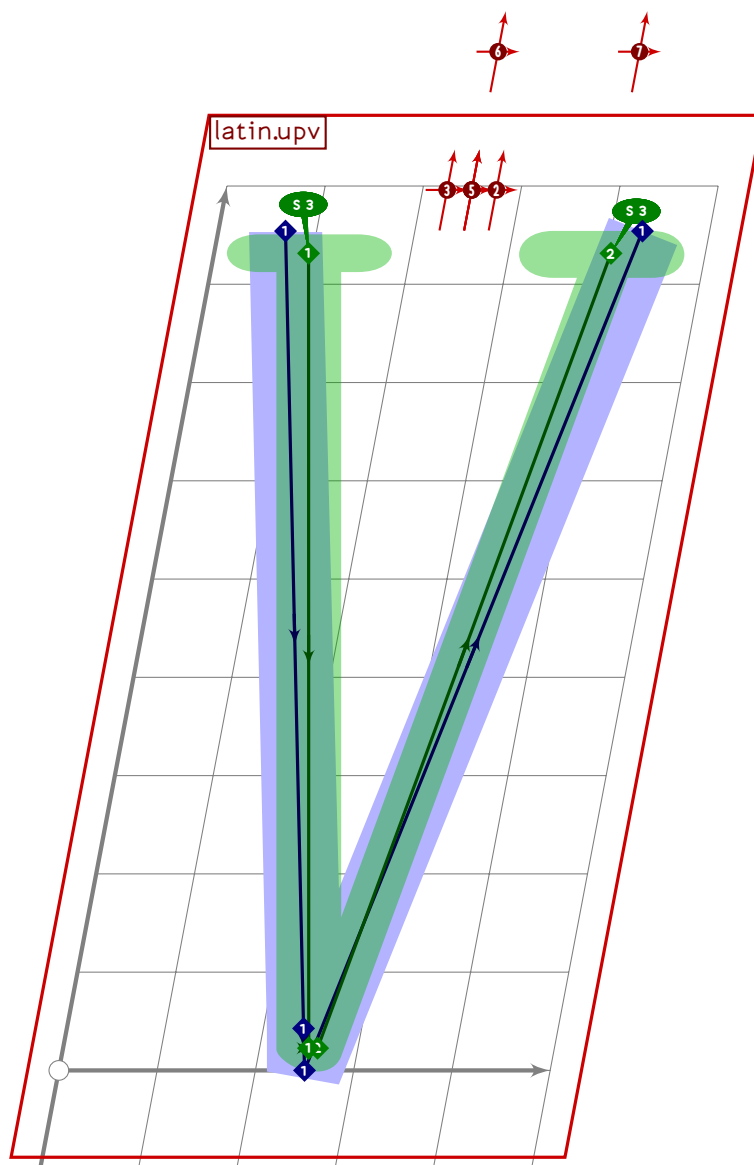




PROO

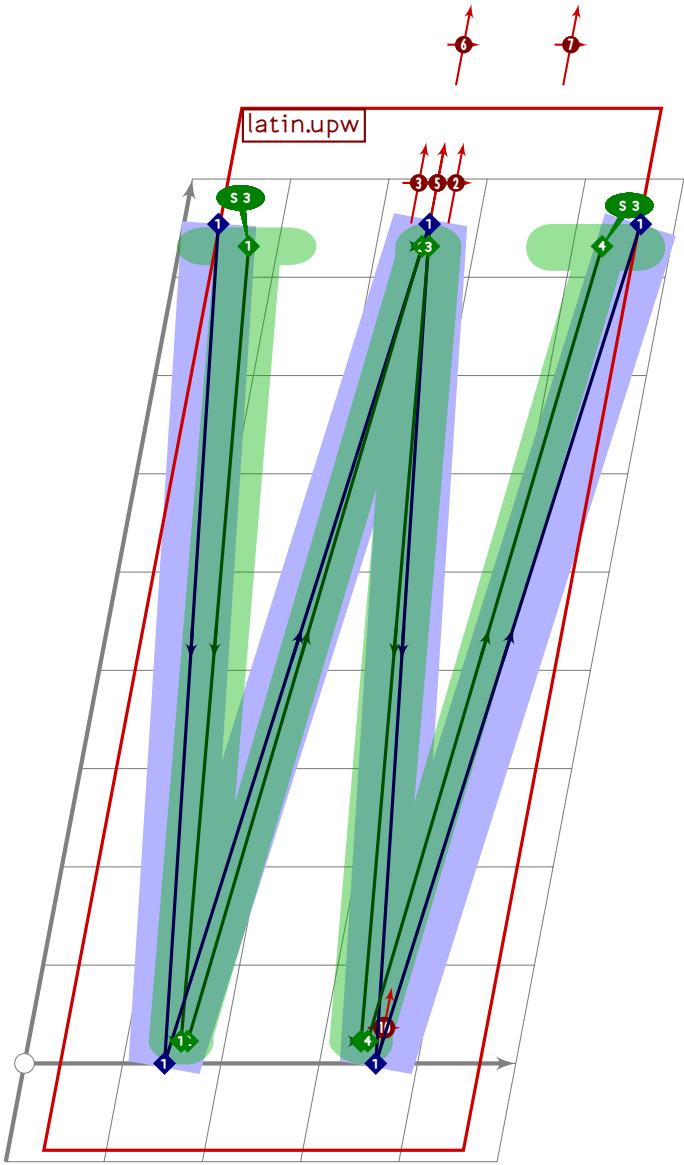
PROO

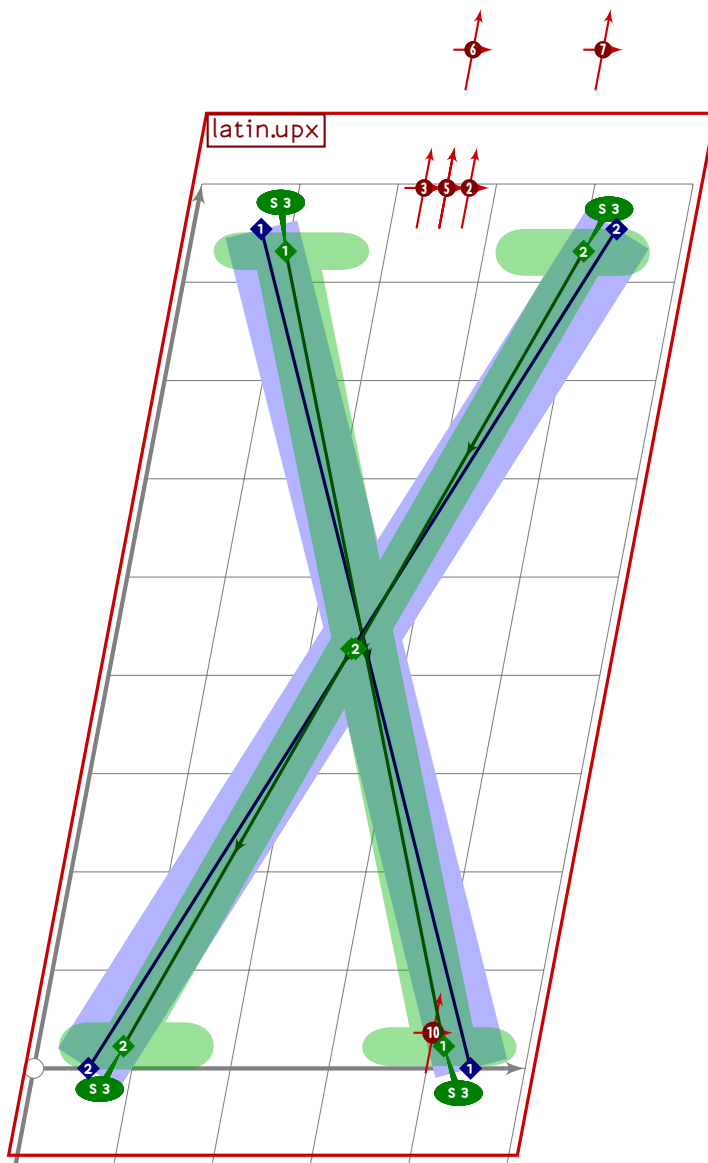




PROO

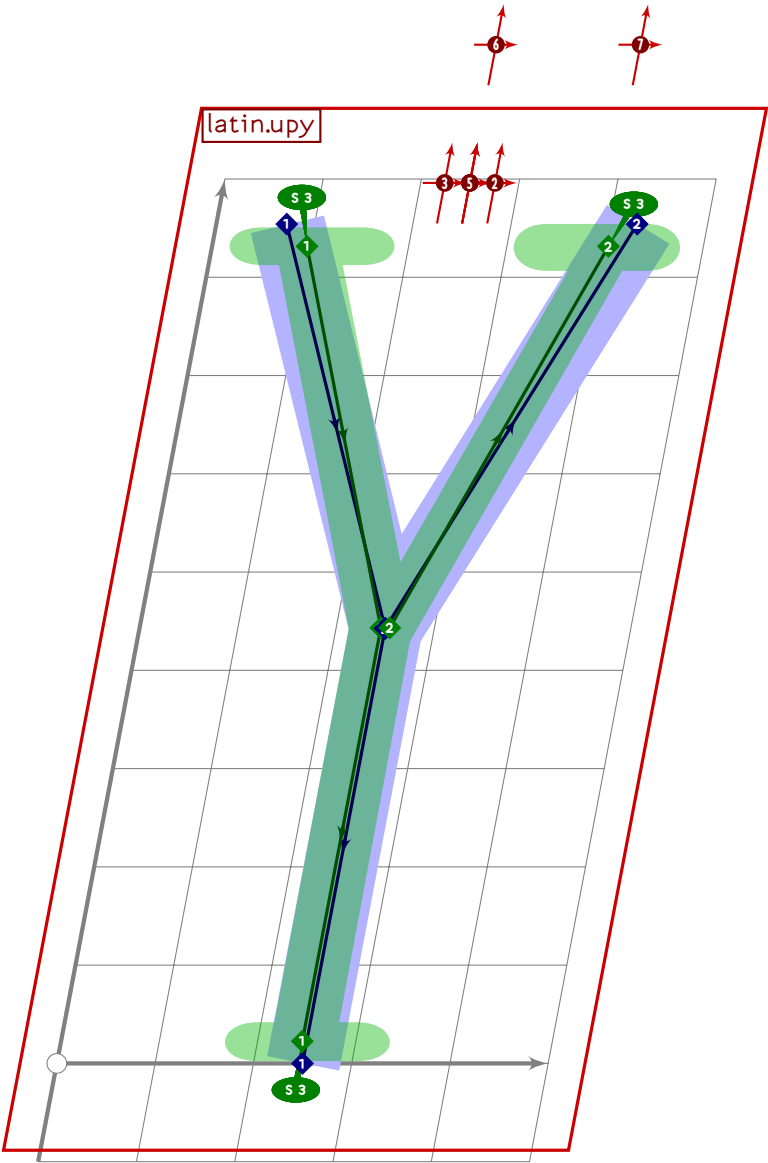
PROO

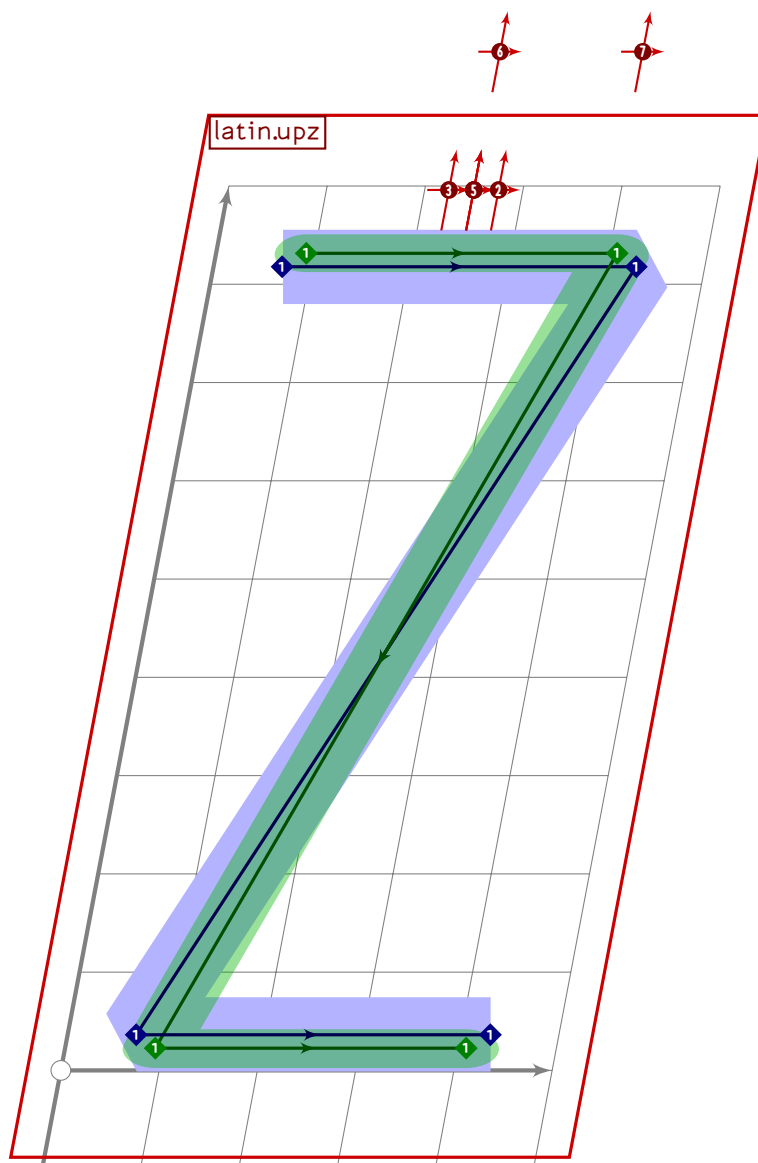




PROO

PROO

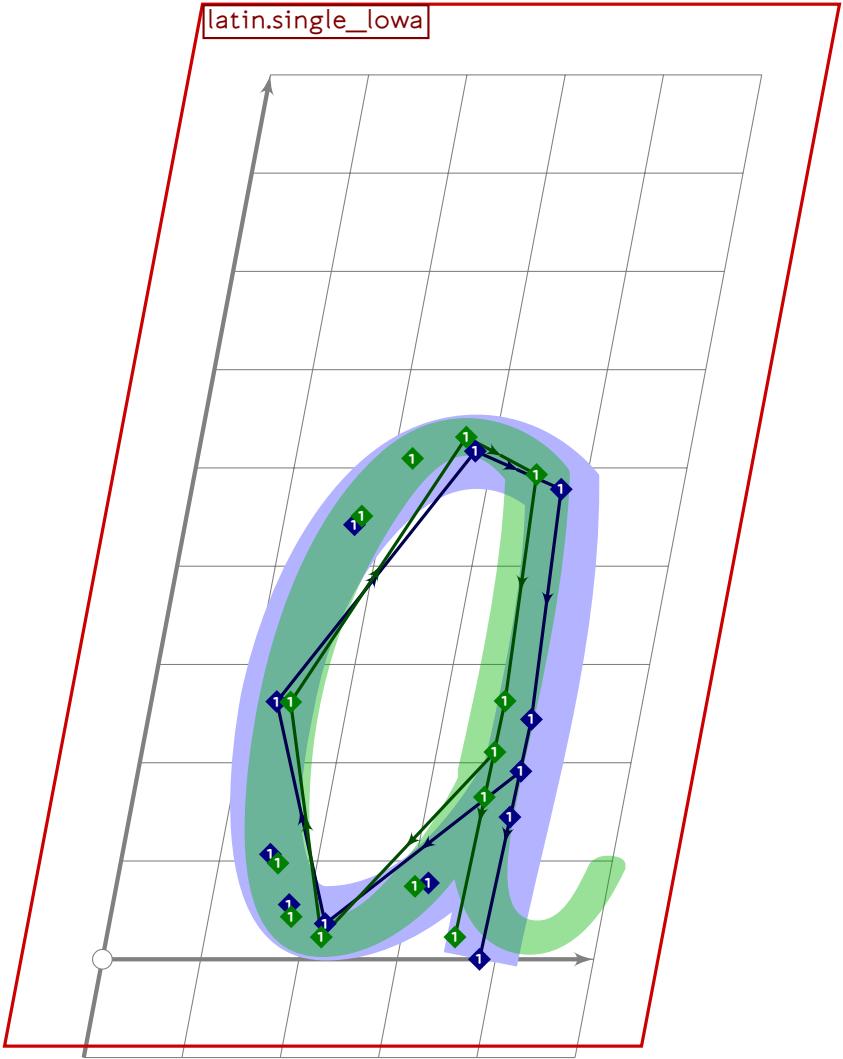


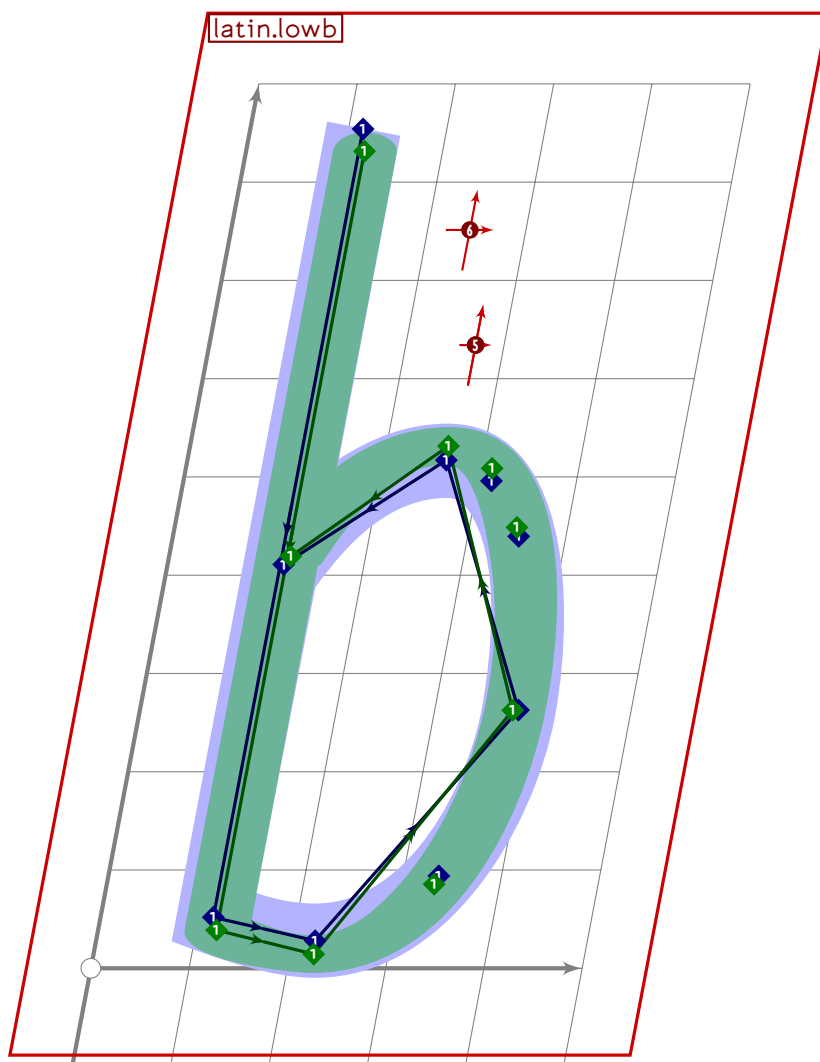


PROO

latin.single_lowa

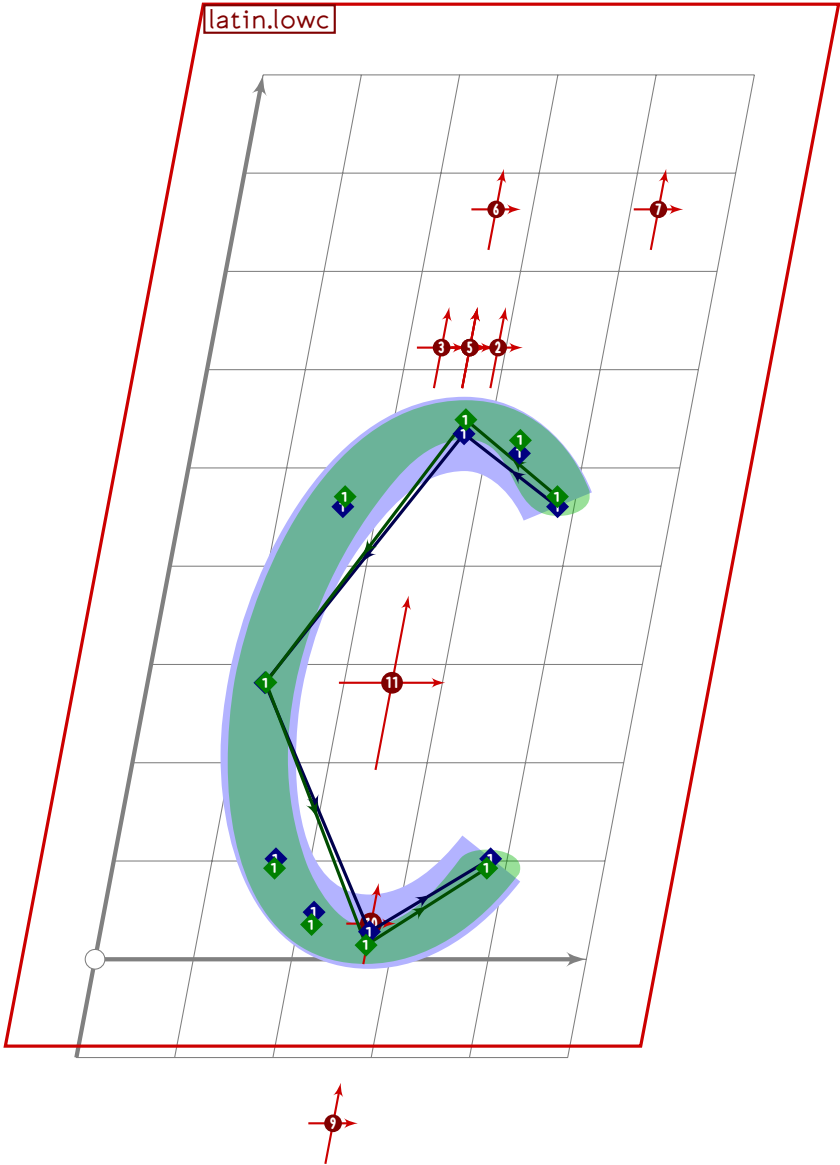
PROO

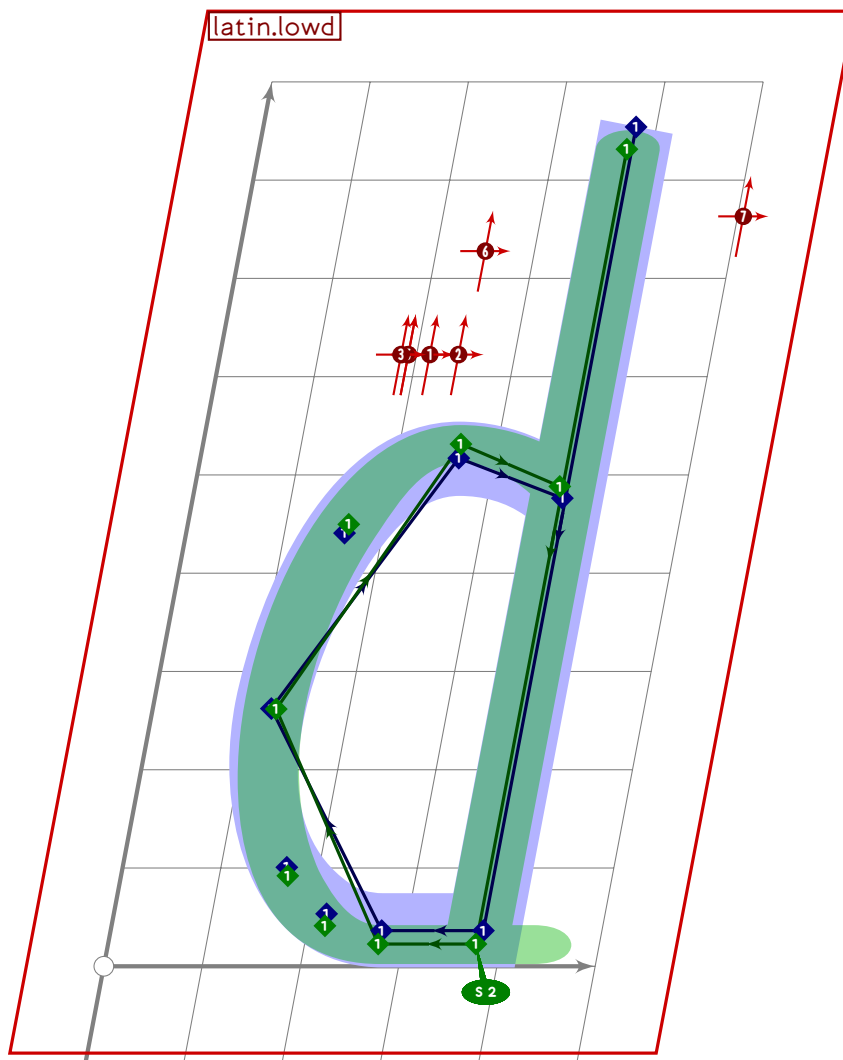




PROO

PROO

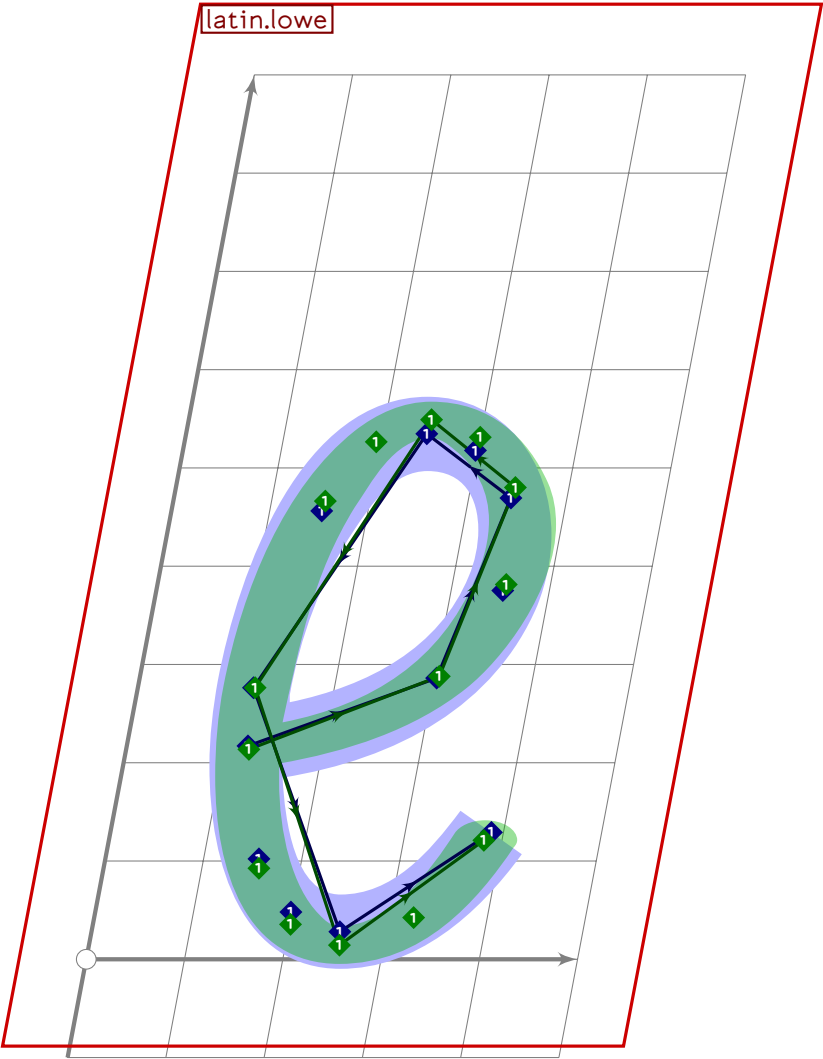


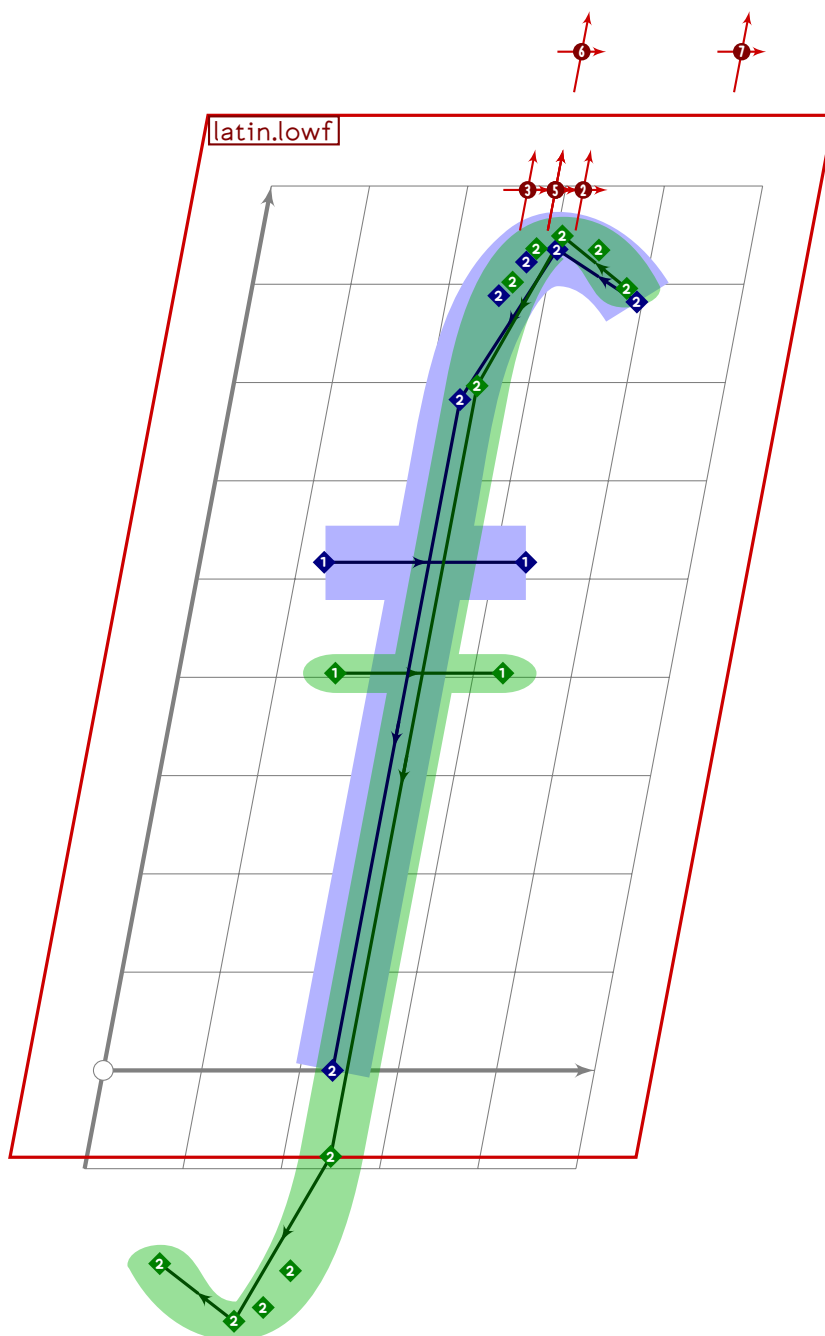


PROO

U+0065
tsuita.e

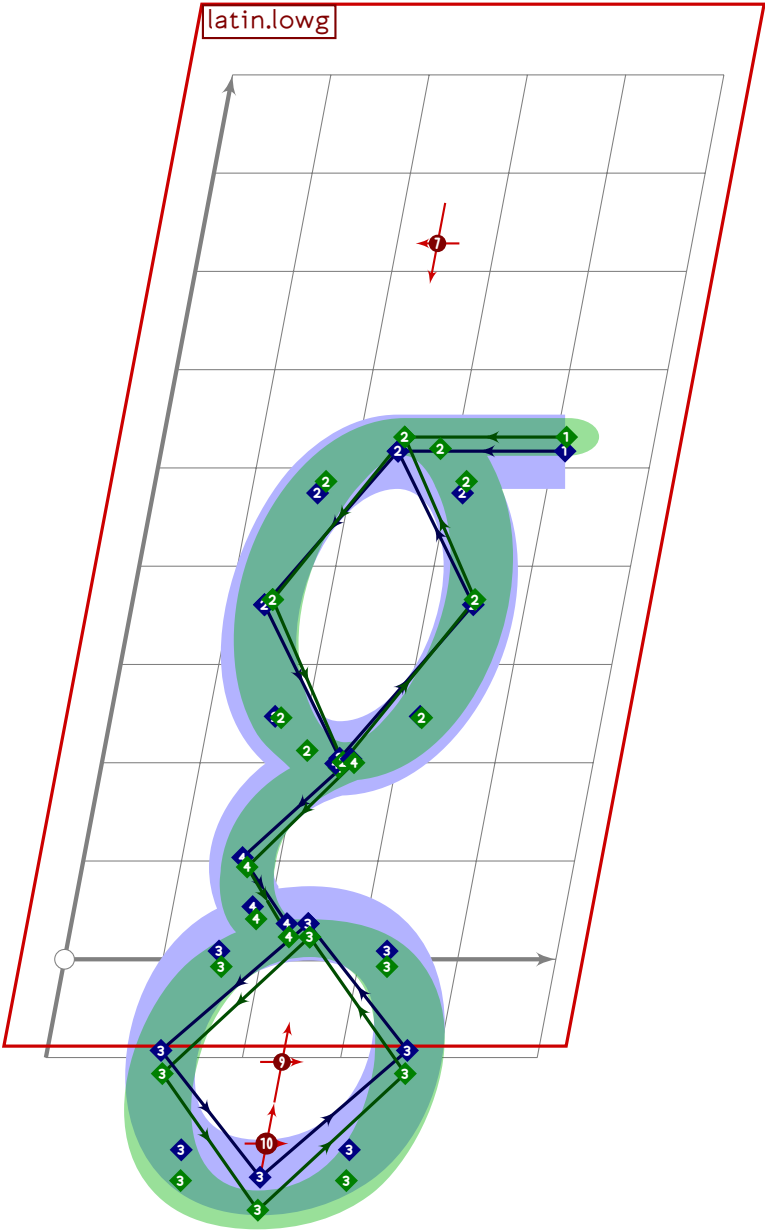
PROO

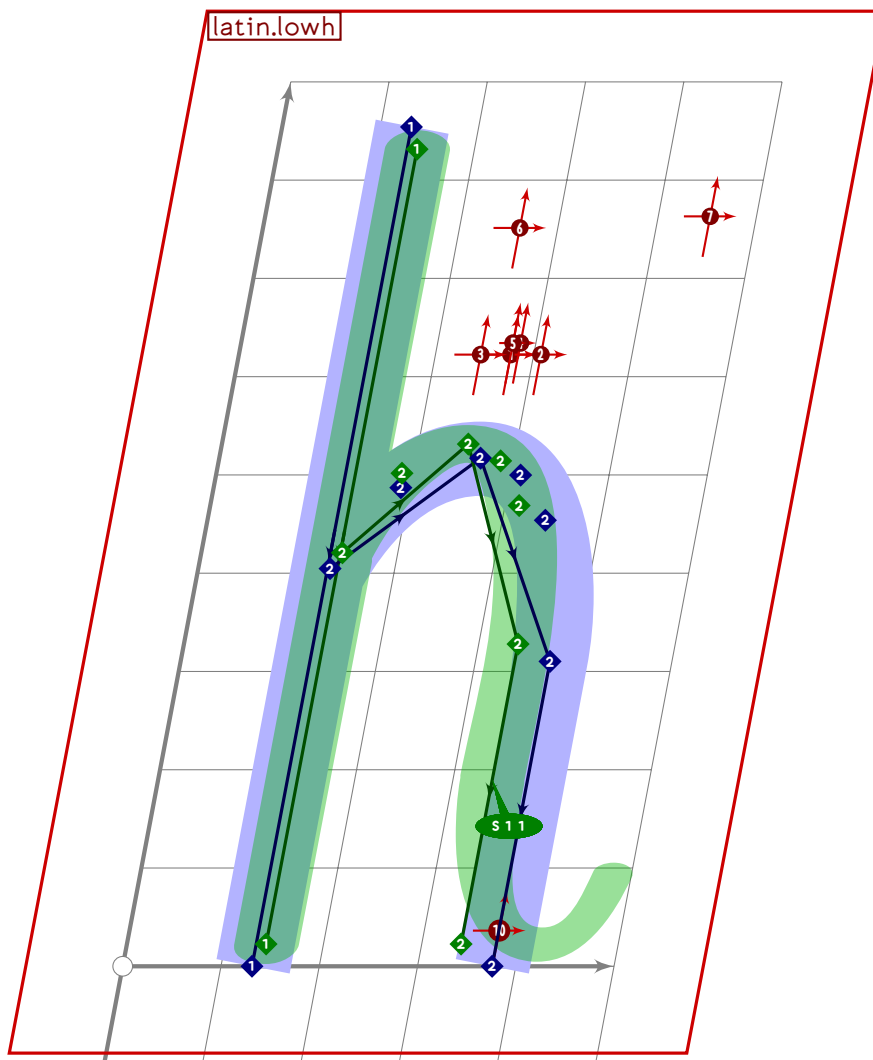




PROO

PROO

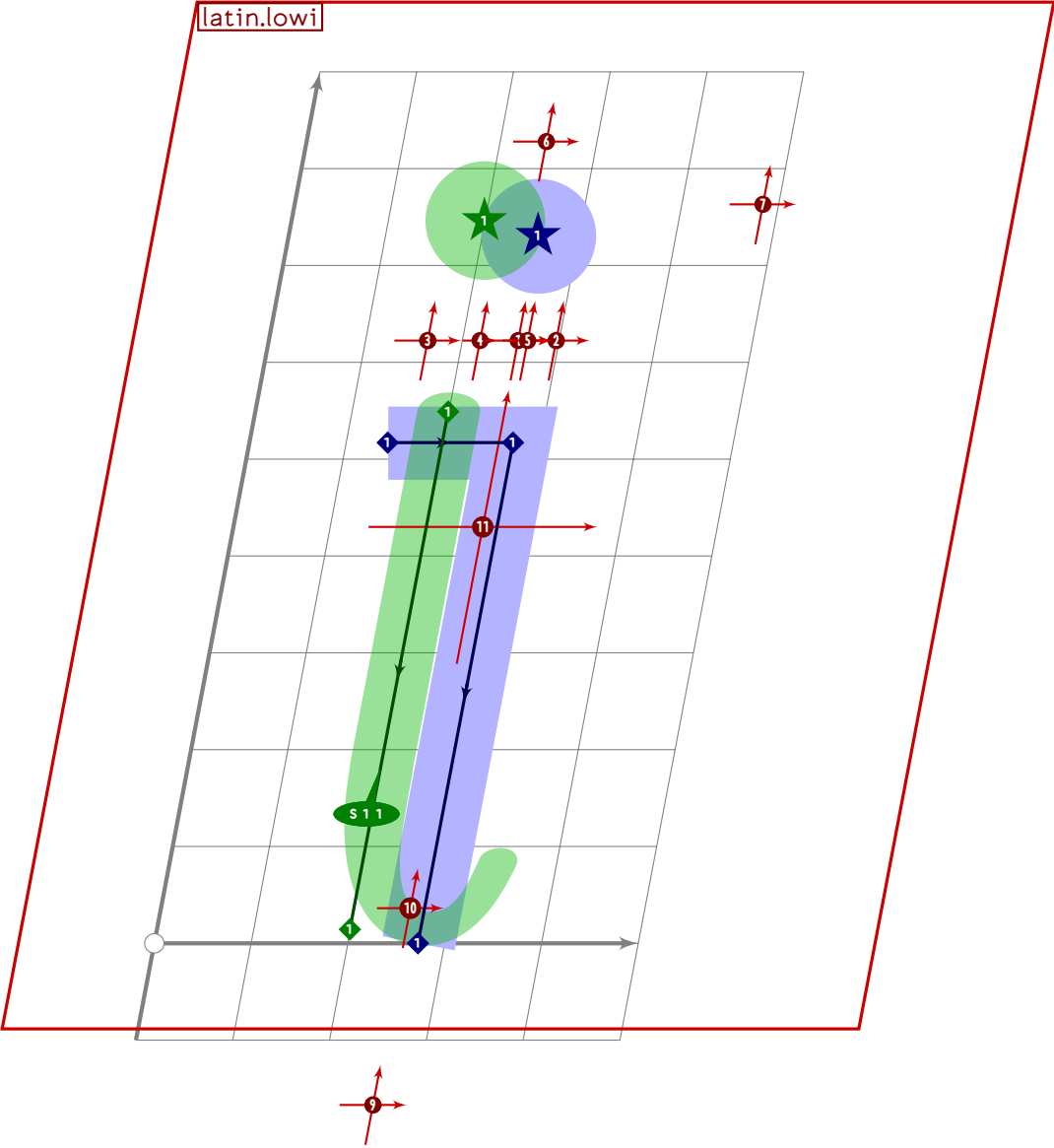


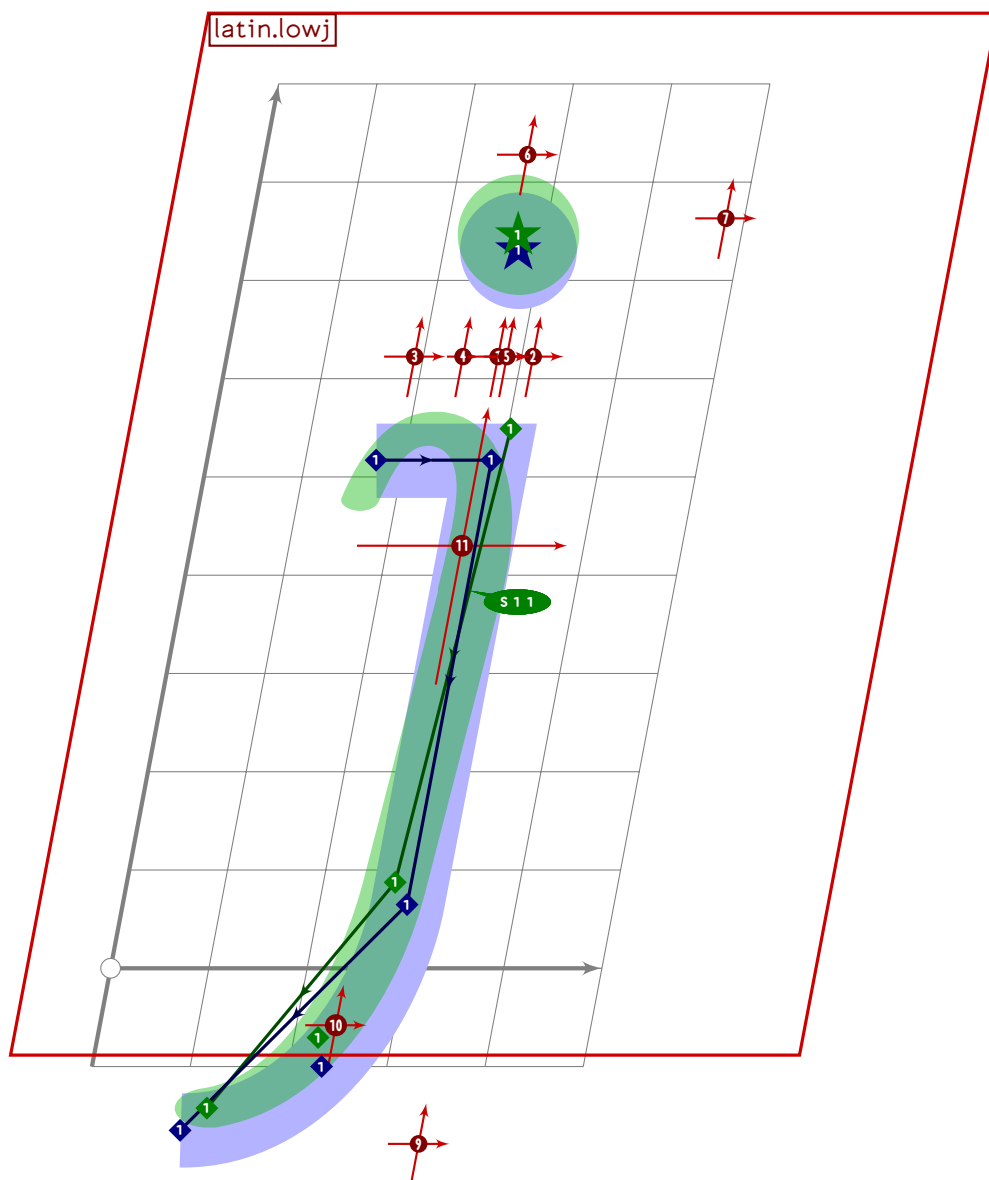


PROO

U+0069
tsuita.i

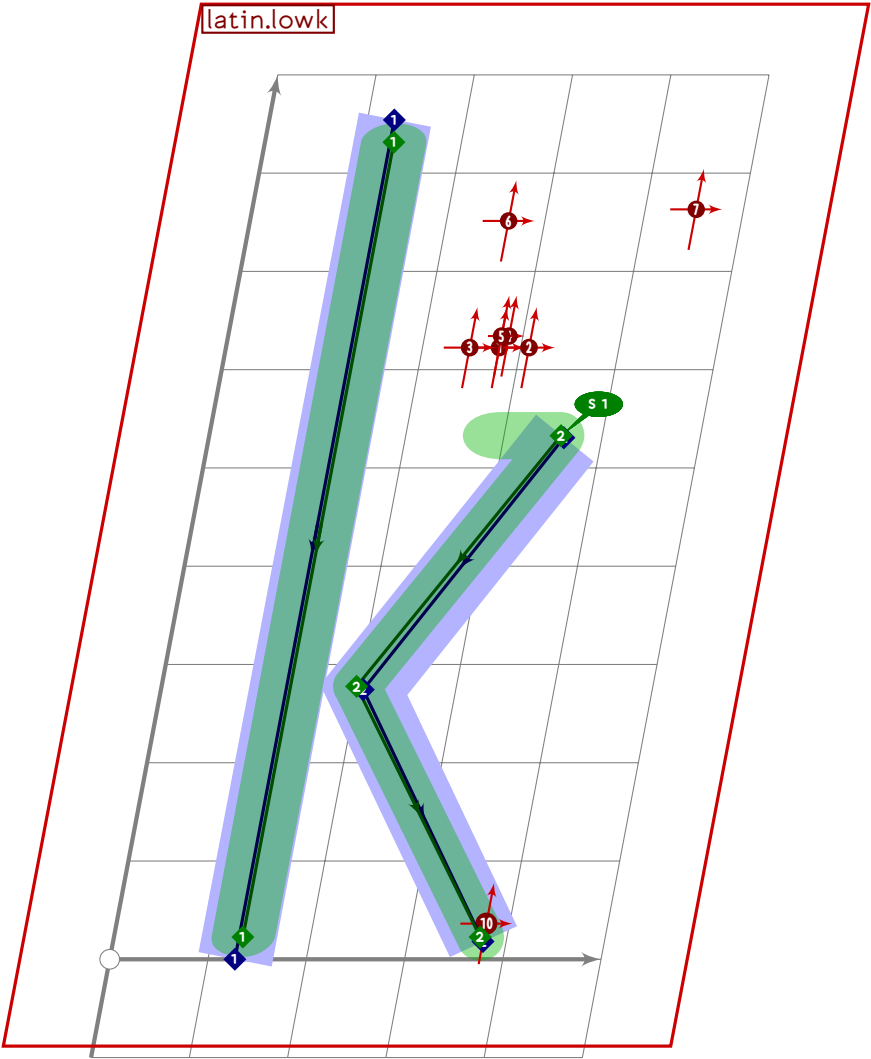
PROO

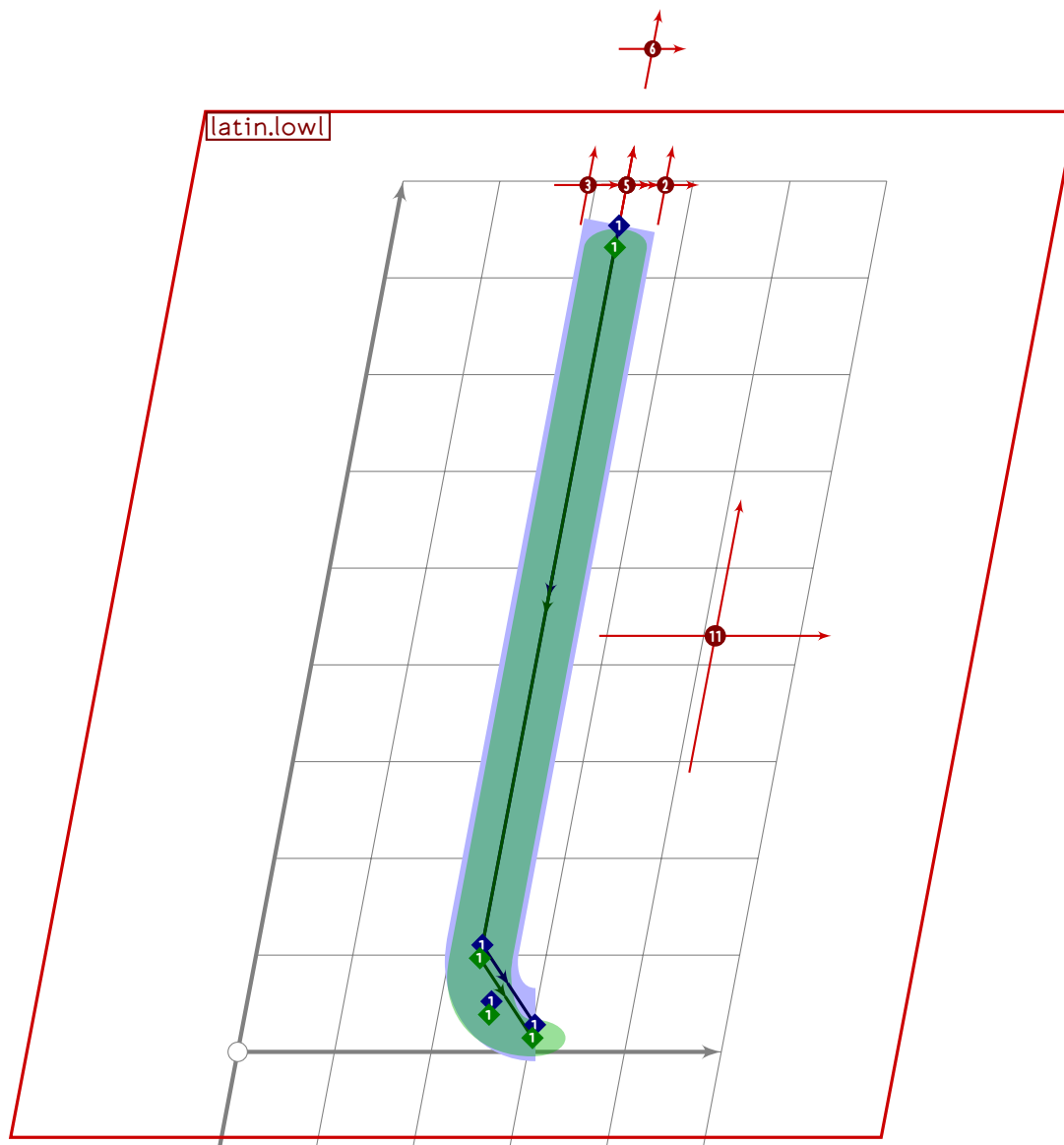




PROO

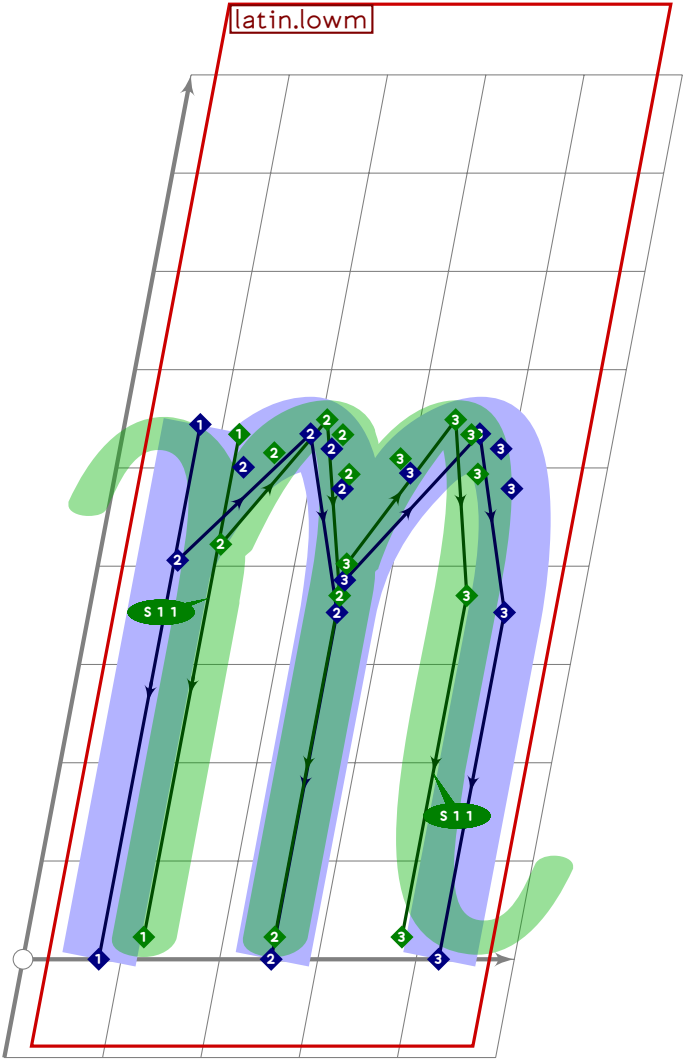
PROO

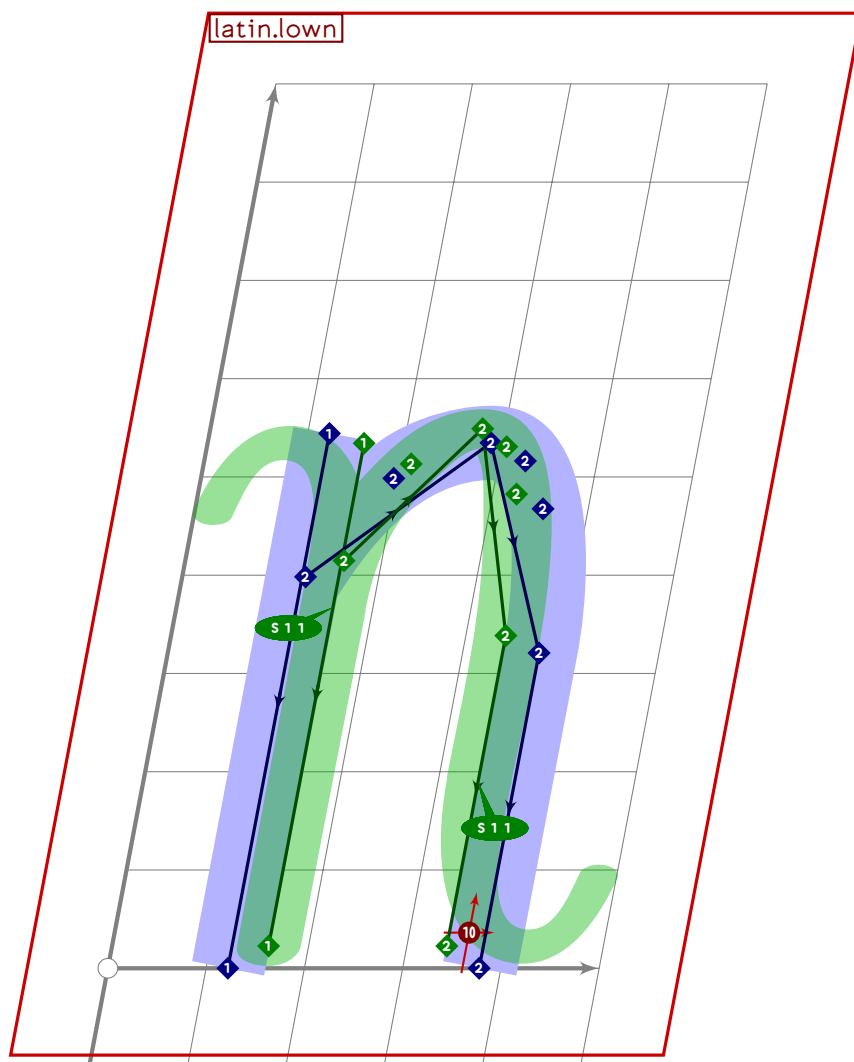




PROO

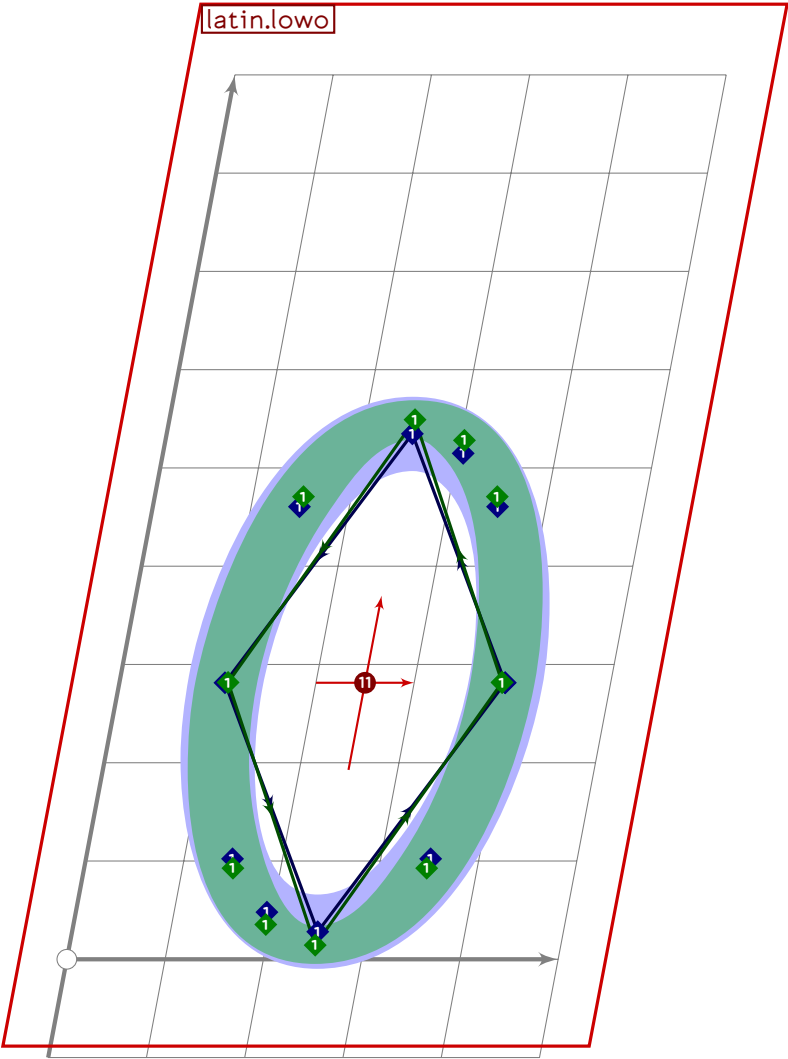
PROO

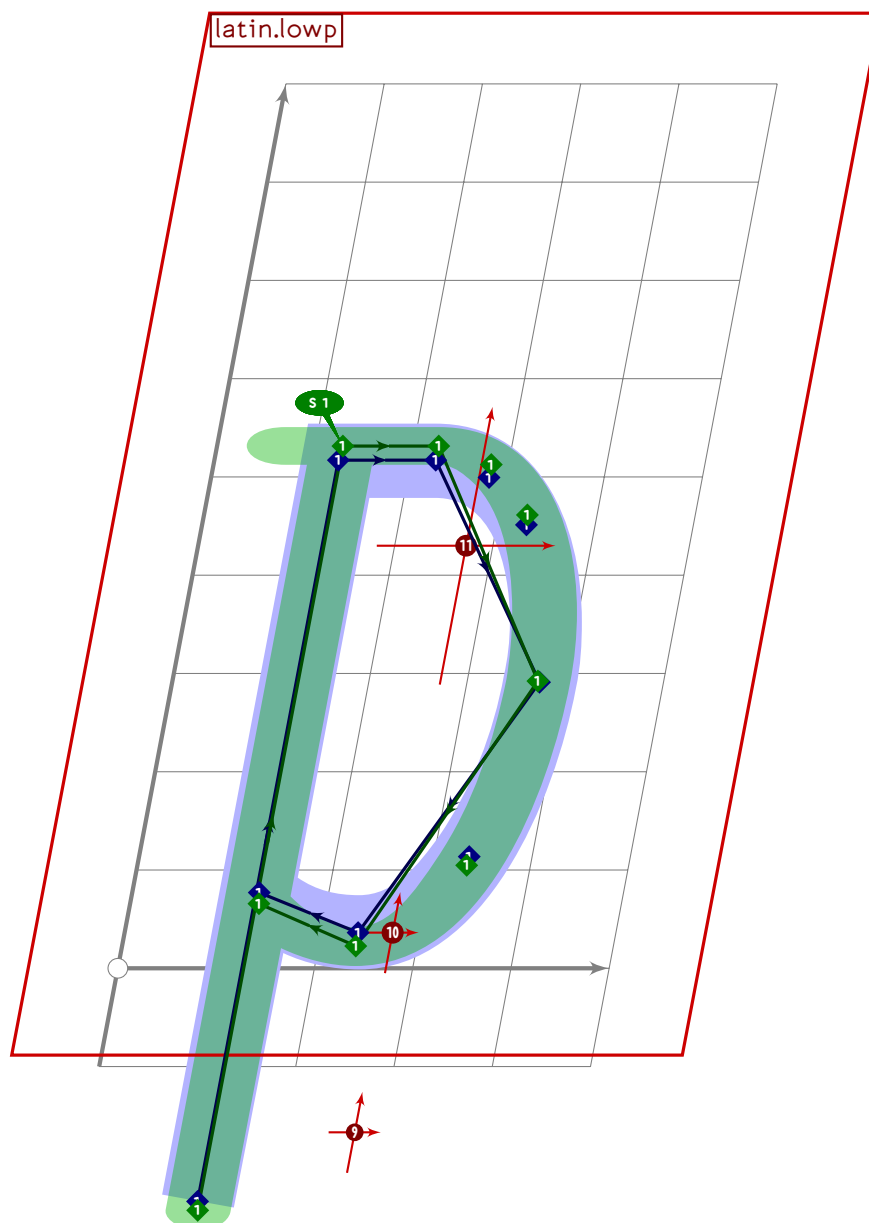




PROO

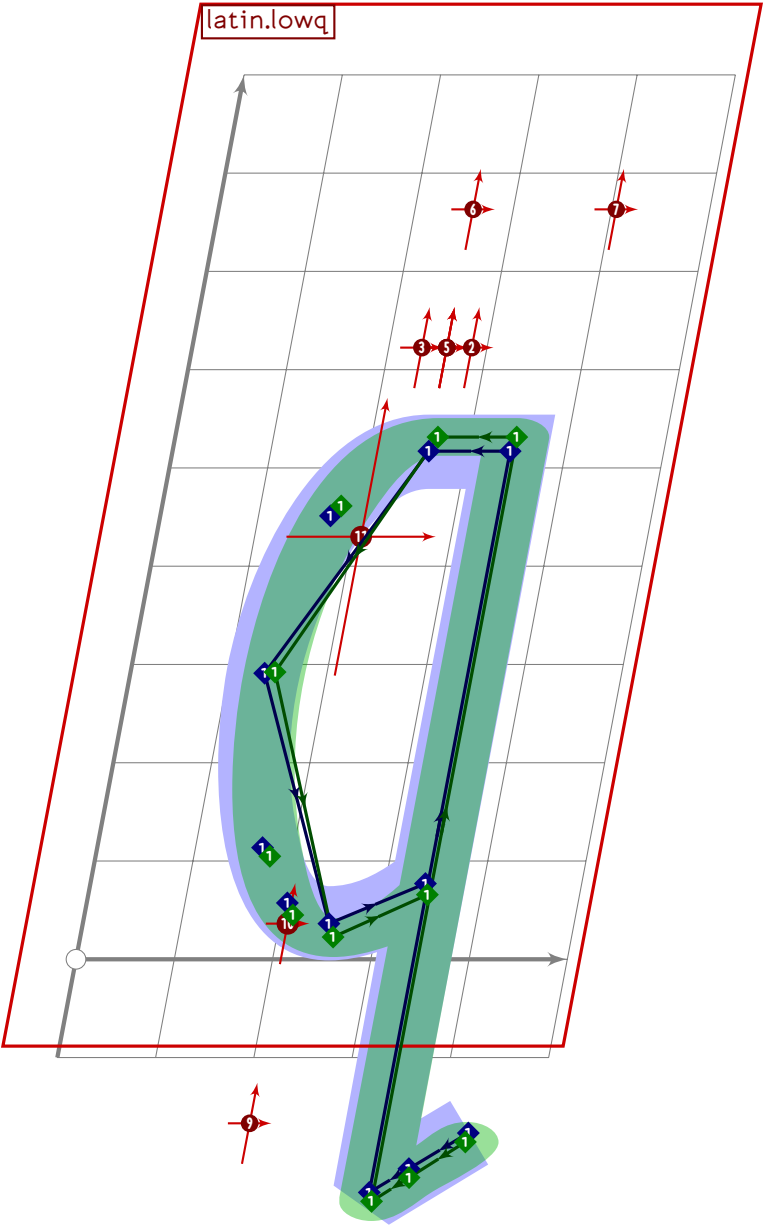
PROO

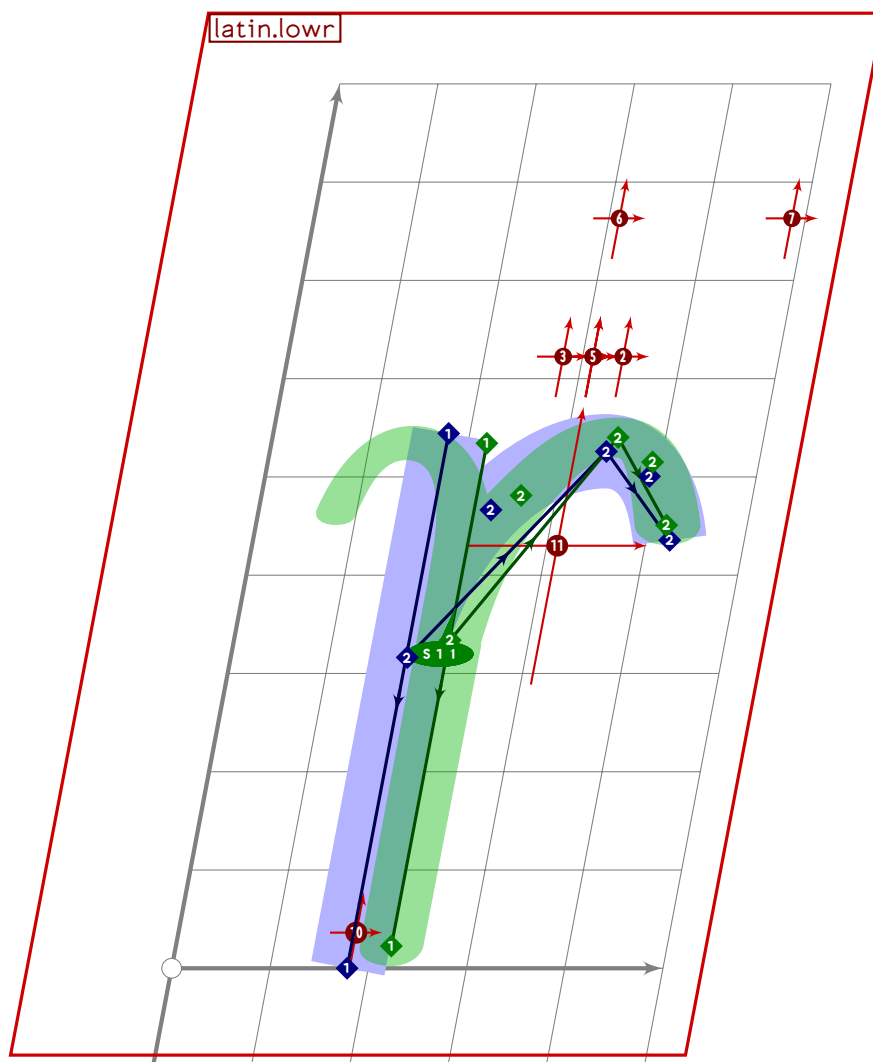




PROO

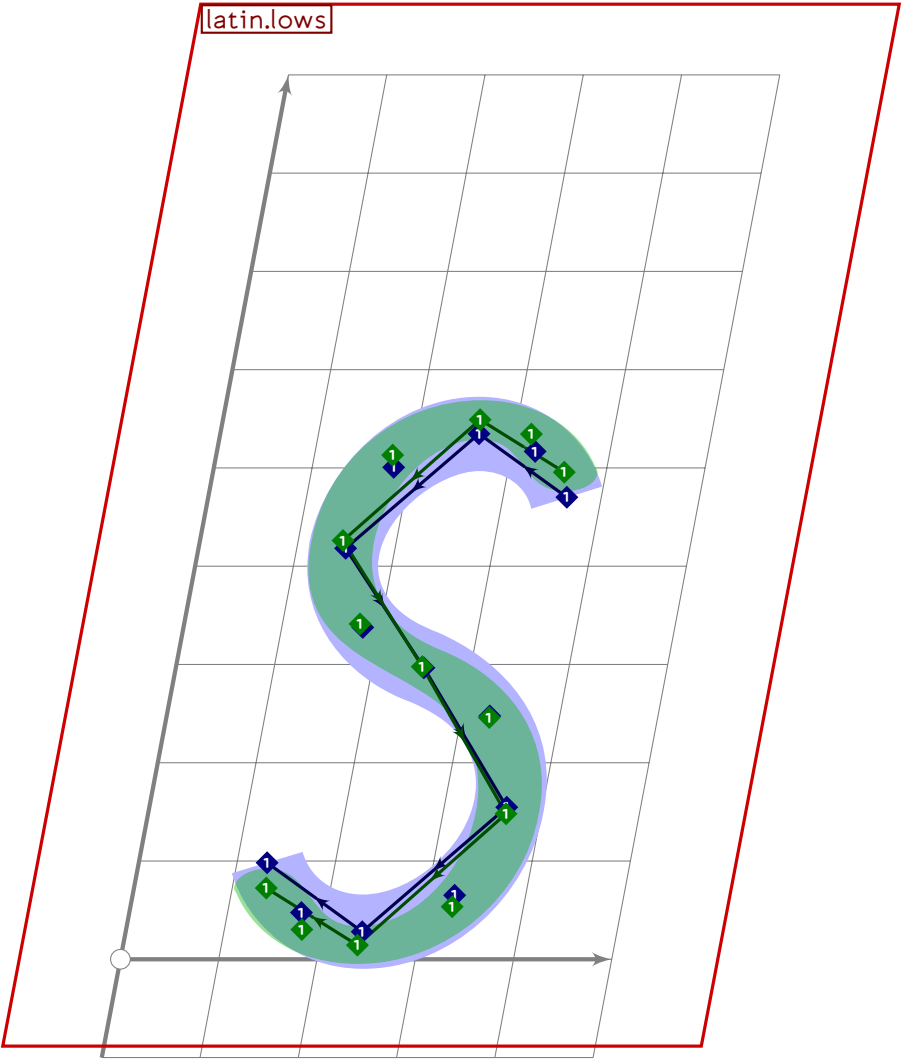
PROO

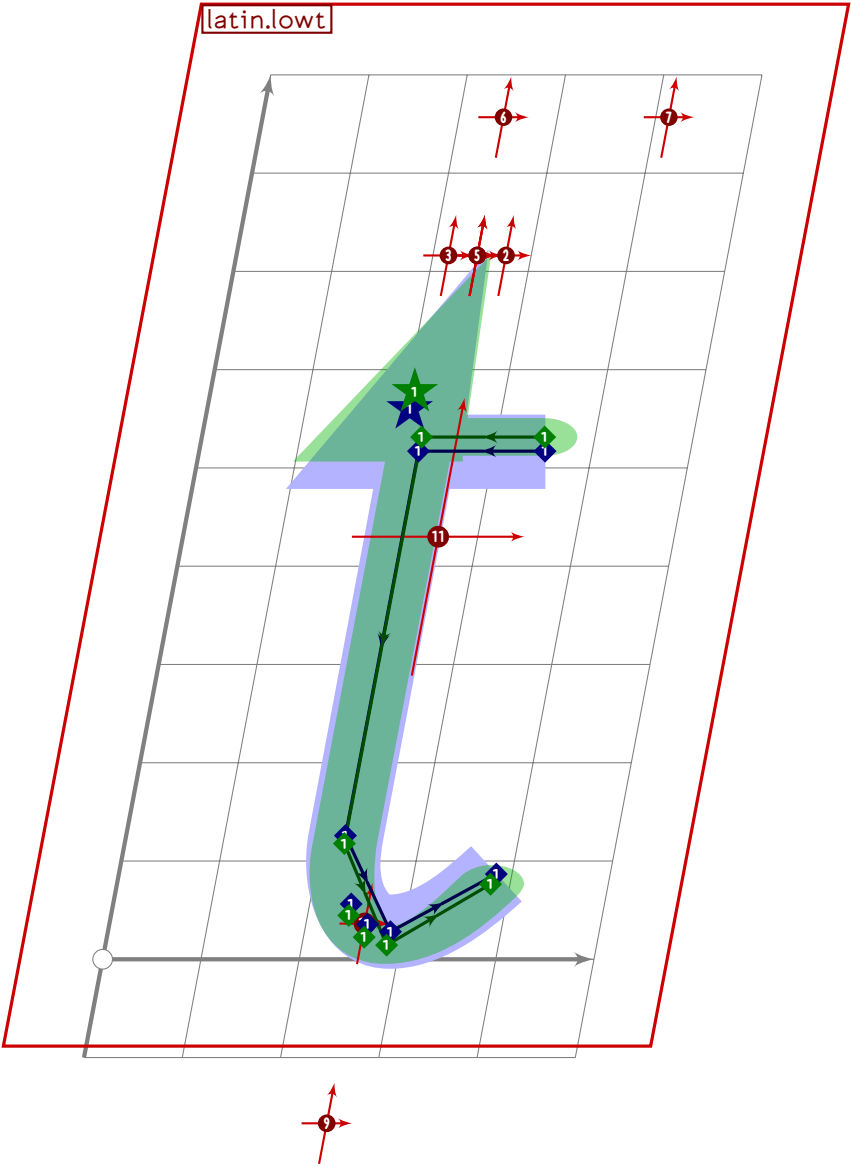


U+0072
tsuita.r

PROO

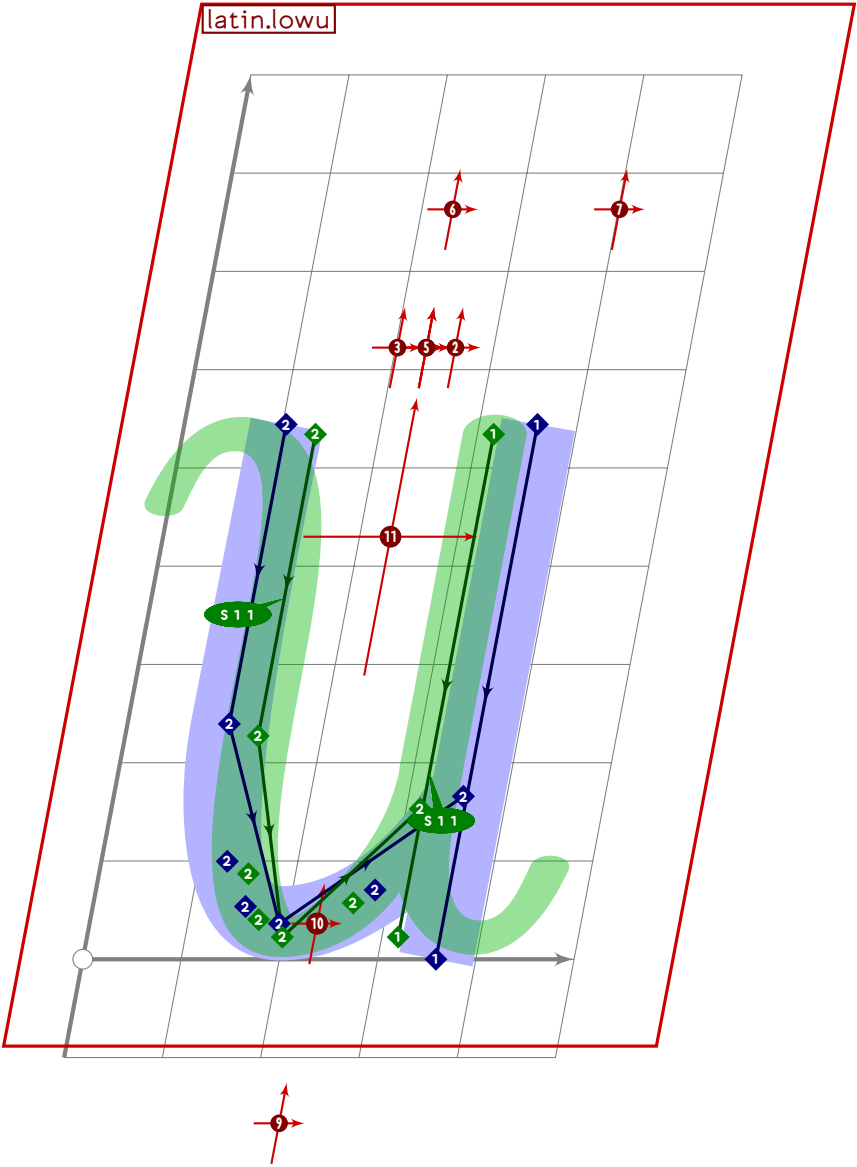
PROO

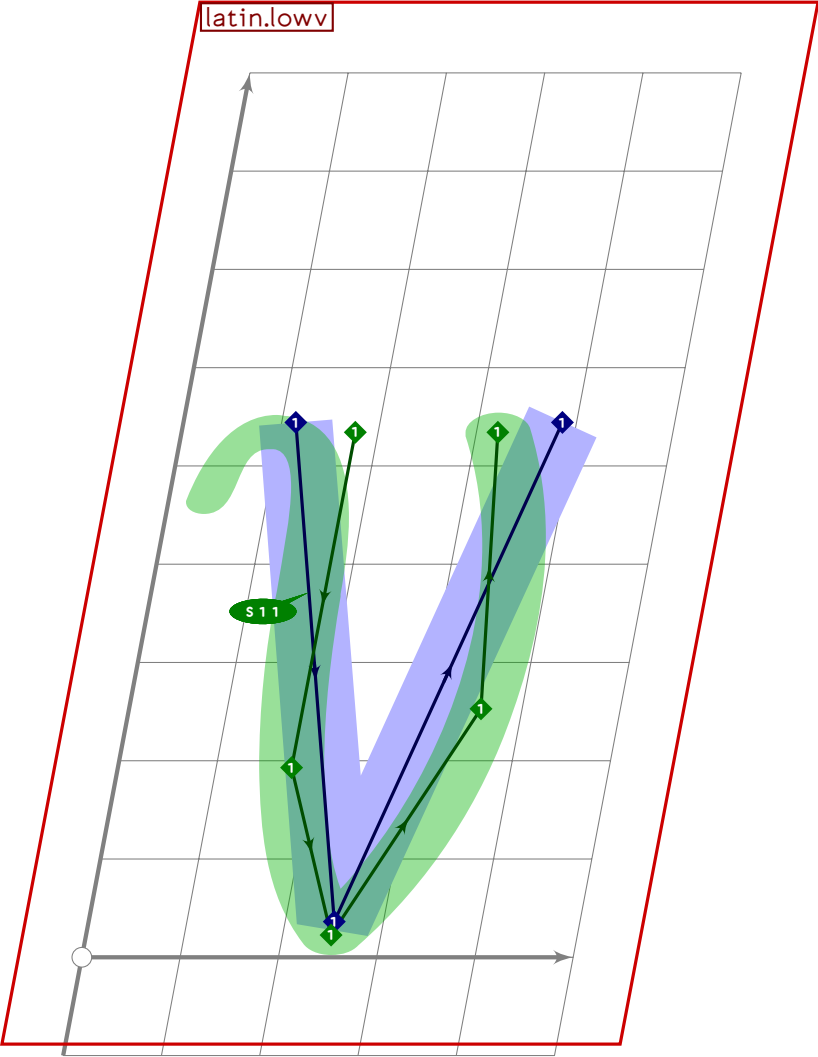




PROO

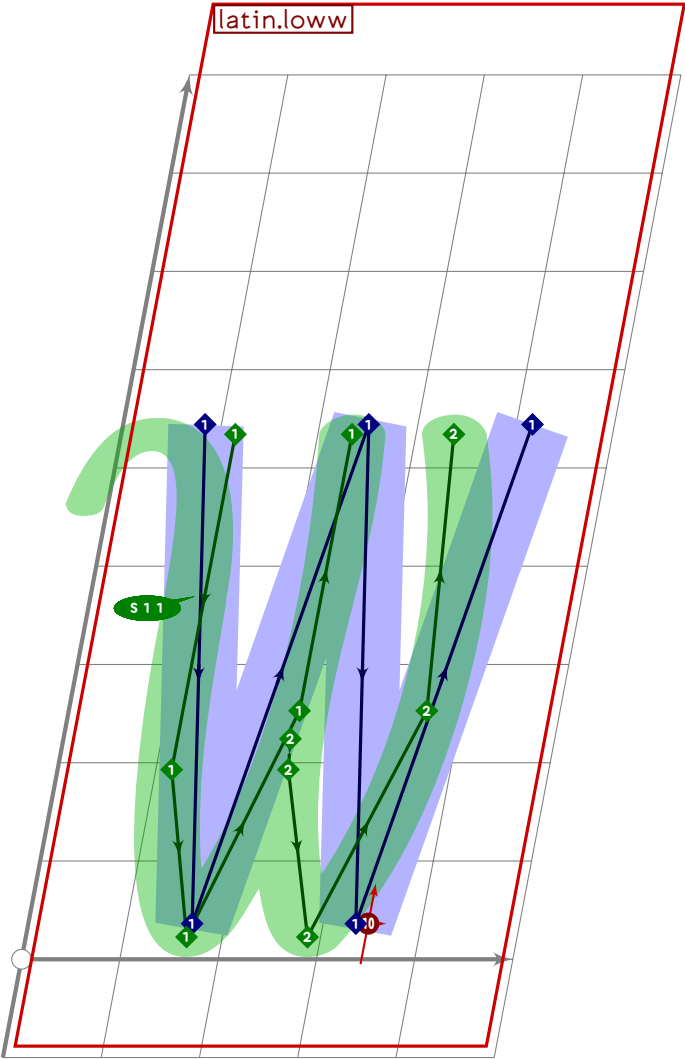
PROO

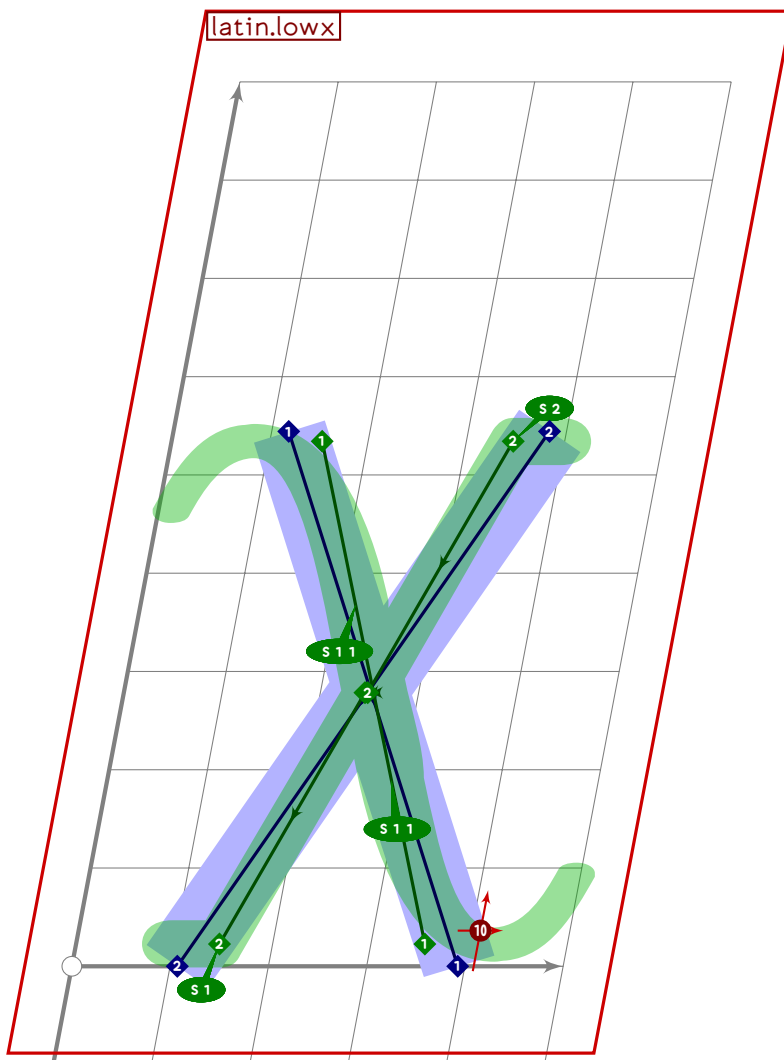




PROO

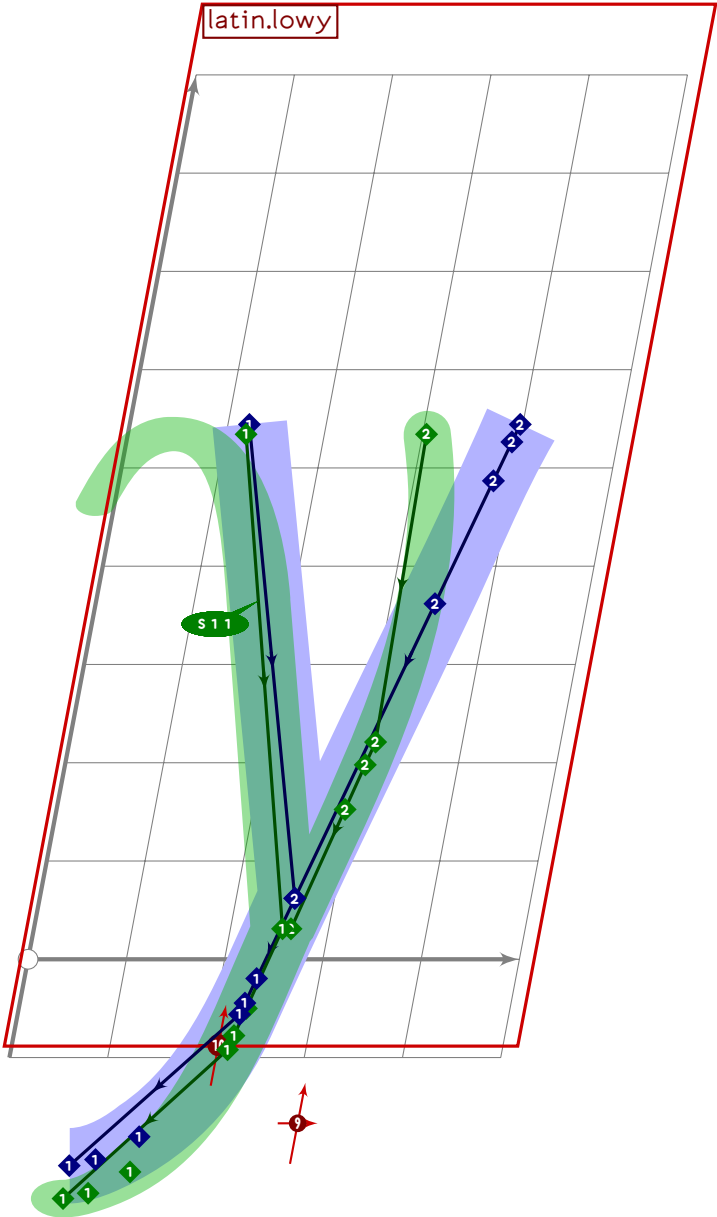
PROO

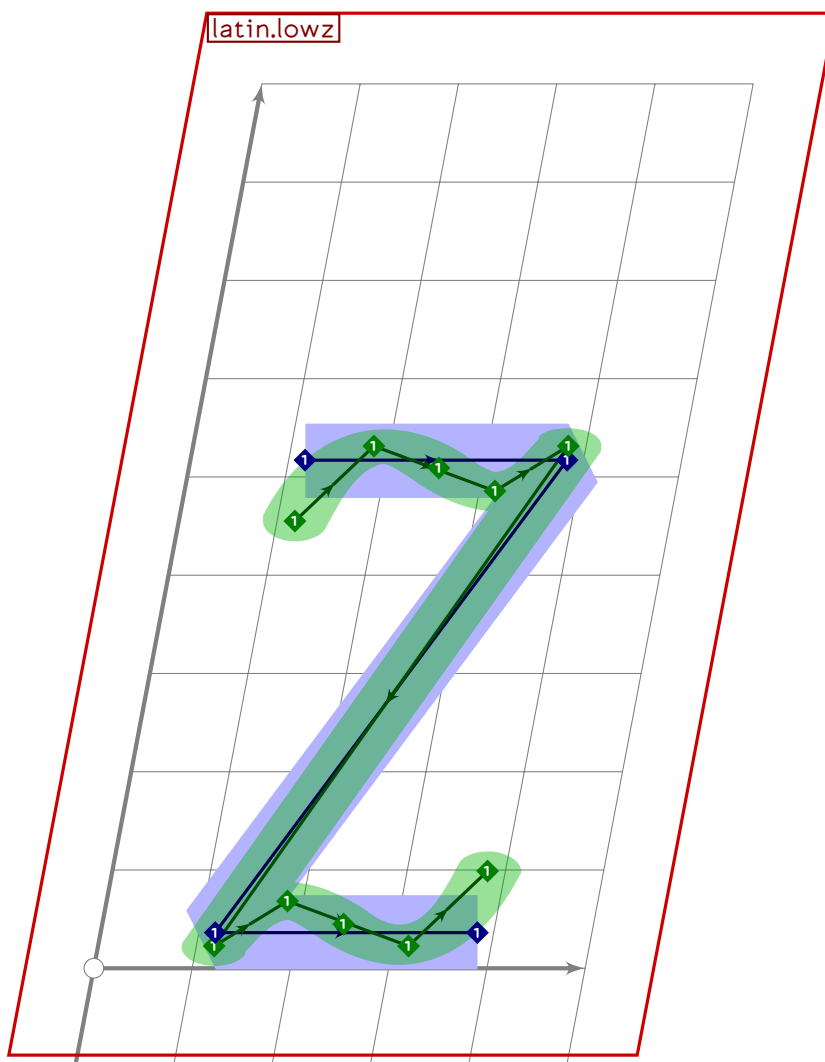




PROO

PROO





PROO

latin.lowaogonek
latin.single_lowa

PROO

