The Cochineal Font Package

Michael Sharpe

November 19, 2023

Cochineal is a fork of Crimson, a remarkable creation of Sebastian Kosch inspired by oldstyle font designers. The name Cochineal is intended to suggest that, while it is crimson, there may be bugs involved. More than 1500 glyphs were added to Crimson, allowing a more than modest chance that some poor spacing or kerning or accent placement might have been introduced. Such problems will occur in the less frequented parts of the fonts, and I would appreciate reports of problems you may discover. You are unlikely to trip over issues if you stay within the bounds of, say, the T1 encoding. I did correct problems in a number of glyphs from Crimson that FontForge uncovered and that could have led to poor rendition on some platforms, especially MAC OS. I also changed the emsize to 1000 (the standard for PostScript-flavored Opentype) from the Crimson value of 1024 and rescaled accordingly.

Cochineal provides Roman, Bold, Italic and BoldItalic. (Crimson provided semibold weights but with only limited coverage, which I did not wish to extend as this would have required the creation of about 2000 additional glyphs. Bob Tennent’s Crimson package does support semibold and the other styles, though without my glyph additions.)

**Package Features:**

In addition to the encodings OT1, T1, TS1, LY1 in general use by Western European (and some Eastern European) scripts, the package also offers LGR with full support for monotonic, polytonic and some ancient Greek, and T2A and OT2 Cyrillic support. All allow a choice from four figure styles—TLF (tabular lining figures, monospaced and uppercase), LF (proportional lining figures, uppercase), TOSF (tabular oldstyle figures, monospaced and lowercase) and OSF (proportional oldstyle figures, lowercase). The encodings OT1, T1, TS1, LY1 offer **Small Caps**, even **ITALICS SMALL CAPS**, and additional figure styles—superiors, inferiors and denominators. These features are available from either [fontspec](http://fontspec.sourceforge.net) or from [pdf](https://www.latex-project.org/). In [pdf](https://www.latex-project.org/), you access these through the macros \textsu, \textinf and \textde, or through their font-switching equivalents \sufigures, \infigures and \defigures. For example:

- \textsu{lle} Dupont and \textsufigures lle Dupont both produce Mlle Dupont.
- \textinf{12345} and \textinf{12345} render as ₁₂₃₄₅, dipping noticeably below the baseline, while \textdefigures 12345 and \textdefigures 12345 render as 12₃₄₅, aligned with the baseline.

As of version 1.062 (2020), there is a new \textfrac macro that works as in the following examples:

- (two arguments) \textfrac{31}{32} renders as ³¹⁄₃₂;
- (three arguments) \textfrac{63}{64} renders as 2₆³⁄₆₄.

In addition, there is a \textcircled macro that makes \textcircled{W} render as ⊙. Numerals can also be circled: \textcircled{2} renders as ⊡.

If you load the [cochineal](http://cochineal.sourceforge.net) package by means of option cochineal to the [newtx](http://www.latex-project.org/) package, version 1.71 or higher,
there is a stacked fraction construction available using the macro \textscfrac which behaves like \textfrac except with the fractional part stacked vertically rather than diagonally. For example, \textscfrac[2]{5}{16} produces $2\frac{5}{16}$.

**Package Options and Macros:**
The package defines two macros, \useosf and \useproportional, useable only in the preamble, which determine the default figure style in text. A typical invocation would be something like

\usepackage{cochineal} % default figure style is tabular, lining
% load sans and typewriter fonts
% load a math font---it will use tabular lining figures in math
\useosf % switch from lining figures to oldstyle figures
\useproportional % switch from tabular to proportional

There is a simpler way to achieve essentially the same result, but with the advantage that the figure styles are not loaded until after the math package (if any) is loaded, so that math always uses the default tabular lining figures.

% If you use babel, load it here, before cochineal
\usepackage[p,osf]{cochineal} % default figure style is proportional, oldstyle
% load sans and typewriter fonts
% load a math font---it will use tabular lining figures in math
\useosf % switch from lining figures to oldstyle figures
\useproportional % switch from tabular to proportional

As of version 1.71 of newtx, it may be more convenient to use its mechanism for loading cochineal as text and newtxmath as math.

% If you use babel, load it here, before newtx
\usepackage[p,osf,cochineal]{newtx} % text figure style is proportional, oldstyle, math will use tabular, lining figures when loaded
% if you use unicode latex with

No matter what the default figure style in text, the package provides switches and macros to use any available figure style.

- \textlf{} and \lfstyle give proportional lining figures; \textttlf{} and \ttlfstyle give tabular lining figures; \texttosf{} and \tosfstyle give proportional oldstyle figures; \texttttosf{} and \ttosfstyle give tabular oldstyle figures; \textfrac{3}{4} uses superior and denominator figures to make the fraction $\frac{3}{4}$.

The options that can be passed to cochineal.sty are the following:

- scale or scaled: a magnification factor—e.g., scaled=1.02 enlarges all text controlled by the package by 2%;
- p, or proportional: make proportional figures the default rather than tabular; lf, or lining: make lining figures the default (this is already the default); osf, or oldstyle: make oldstyle figures the default rather than lining;
- sups: use superior figures to make footnote markers, rather than the \LaTeX’s default markers;
- swashQ: use Cochineal's swash Q instead of its tamer default version, Q;
- scosf: always use oldstyle figures within a small caps block;
- **theoremfont**: for theorem statements in the plain style, use a doctored version of italics that has upright figures, braces, brackets, parentheses, exclamation mark, colon and semicolon. It is implemented as the slanted shape, and there are two options you can add to the call to the cochineal package to modify the way it outputs figures.

  The default (neither of the following options is set) is to use upright figures within theorem text, but in the same alignment (proportional or tabular) and the same style (lining or oldstyle) as in general text.

  Option **thmtabular** causes figures in theorem text to render in tabular alignment, while option **thmlining** causes figure styles to render in lining rather than oldstyle. The two may be used in conjunction, forcing figures in theorem text to render as tabular lining figures.

**Usage with unicode LaTeX**

The cochineal package works with all latex engines, and has some extra tricks available only in unicode LaTeX. In addition, newtx also works with all latex engines and may be used to make a simplified call to use cochineal for text and newtxmath (with cochineal letters) for math. Two crucial options—**type1text** and **nofontspec**—each of which may be used with both newtx and cochineal, determine how unicode latex will process your tex document.

  - Neither **type1text** (nor the equivalent **type1**) nor **nofontspec** specified: load fontspec and use Cochineal opentype fonts.
  - **type1text** specified but not **nofontspec**: load fontspec but process text as pdflatex would. (Other subsidiary opentype fonts could be loaded using fontspec arguments.)
  - **nofontspec** specified: fontspec not loaded, text processed as under pdflatex.

There are some more decorative glyph shapes as alternatives for Q and J in version 1.077 that are available only using unicode latex via **StylisticSet=3** and **StylisticSet=2** respectively. (The latter is available only in italic shapes.) They may be turned on as global options to newtx or cochineal by **altQ**, **altJ**.

% preamble should include, in this order:
\usepackage[T1]{fontenc}
\usepackage[p,osf]{cochineal}
% load babel here
\usepackage[varqu,varl,var0]{inconsolata}
\usepackage[compact,line]{cabin}

\def\decorativeQ{{\addfontfeature{RawFeature=+ss03}Q}}
\def\decorativeJ{{\addfontfeature{RawFeature=+ss02}J}}

Q: \decorativeQ: Q
J: \decorativeJ: J

There is also an option **oldSS** you may specify to use the older form of `\SS` rather than the newer German Sharp S glyph.

**Mathematical accompaniment**

The package contains fonts for use as math letters that are derived from Cochineal Roman and Greek glyphs and the newtxmath family. Note that \( \nu \) and \( \gamma \) (Greek nu) are quite distinct. Here’s a sample.

\% preamble should include, in this order:
\usepackage[T1]{fontenc}
% load babel here
\usepackage[p,osf]{cochineal}
\usepackage[varqu,varl,var0]{inconsolata}
\usepackage[compact,line]{cabin}
The typeset math below follows the ISO recommendations that only variables be set in italic. Note the use of upright shapes for $d$, $e$ and $\pi$. (The first two are entered as \texttt{\textit{d}} and \texttt{\textit{e}}, and in fonts derived from \texttt{mtpro2} or \texttt{newtxmath}, the latter is entered as \texttt{\textup{pi}}.)

**Simplest form of the Central Limit Theorem:** Let $X_1, X_2, \ldots$ be a sequence of iid random variables with mean $0$ and variance $1$ on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$. Then

$$
\mathbb{P}\left(\frac{X_1 + \cdots + X_n}{\sqrt{n}} \leq \nu\right) \to \mathcal{N}(\nu) := \int_{-\infty}^{\nu} \frac{e^{-t^2/2}}{\sqrt{2\pi}} \, dt \quad \text{as} \ n \to \infty, \text{ for every } \nu \in \mathbb{R},
$$

or, equivalently, letting $S_n := \sum_1^n X_k$,

$$
\mathbb{E}f\left(\frac{S_n}{\sqrt{n}}\right) \to \int_{-\infty}^{\infty} f(t) \frac{e^{-t^2/2}}{\sqrt{2\pi}} \, dt \quad \text{as} \ n \to \infty, \text{ for every } f \in \mathcal{C}(\mathbb{R}).
$$
Cochineal’s TS1 (textcomp)

Coverage of TS1 has been enriched as of version 1.062 so that it now qualifies as sub-encoding 0, with essentially all TS1 features available.

<table>
<thead>
<tr>
<th></th>
<th>'0'</th>
<th>'1'</th>
<th>'2'</th>
<th>'3'</th>
<th>'4'</th>
<th>'5'</th>
<th>'6'</th>
<th>'7'</th>
</tr>
</thead>
<tbody>
<tr>
<td>'00x'</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>'01x'</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
</tr>
<tr>
<td>'02x'</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
<td>21</td>
<td>22</td>
<td>23</td>
</tr>
<tr>
<td>'03x'</td>
<td>←24</td>
<td>→25</td>
<td>←26</td>
<td>←27</td>
<td>←28</td>
<td>←29</td>
<td>←30</td>
<td>←31</td>
</tr>
<tr>
<td>'04x'</td>
<td>b</td>
<td>32</td>
<td>33</td>
<td>34</td>
<td>35</td>
<td>$36$</td>
<td>37</td>
<td>38</td>
</tr>
<tr>
<td>'05x'</td>
<td>40</td>
<td>41</td>
<td>42</td>
<td>43</td>
<td>$44$</td>
<td>45</td>
<td>46</td>
<td>47</td>
</tr>
<tr>
<td>'06x'</td>
<td>o</td>
<td>48</td>
<td>1</td>
<td>49</td>
<td>2</td>
<td>50</td>
<td>3</td>
<td>51</td>
</tr>
<tr>
<td>'07x'</td>
<td>8</td>
<td>56</td>
<td>9</td>
<td>57</td>
<td>Q</td>
<td>58</td>
<td>Q</td>
<td>59</td>
</tr>
<tr>
<td>'08x'</td>
<td>d</td>
<td>64</td>
<td>65</td>
<td>66</td>
<td>67</td>
<td>68</td>
<td>69</td>
<td>70</td>
</tr>
<tr>
<td>'09x'</td>
<td>0</td>
<td>100</td>
<td>101</td>
<td>102</td>
<td>103</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>'10x'</td>
<td>104</td>
<td>105</td>
<td>106</td>
<td>107</td>
<td>108</td>
<td>109</td>
<td>110</td>
<td>111</td>
</tr>
<tr>
<td>'11x'</td>
<td>112</td>
<td>113</td>
<td>114</td>
<td>115</td>
<td>116</td>
<td>117</td>
<td>118</td>
<td>119</td>
</tr>
<tr>
<td>'12x'</td>
<td>120</td>
<td>121</td>
<td>122</td>
<td>123</td>
<td>124</td>
<td>125</td>
<td>126</td>
<td>127</td>
</tr>
<tr>
<td>'13x'</td>
<td>128</td>
<td>129</td>
<td>130</td>
<td>131</td>
<td>$\hat{1}$</td>
<td>132</td>
<td>$\hat{1}$</td>
<td>133</td>
</tr>
<tr>
<td>'14x'</td>
<td>336</td>
<td>137</td>
<td>138</td>
<td>139</td>
<td>$\in$</td>
<td>140</td>
<td>$\subset$</td>
<td>141</td>
</tr>
<tr>
<td>'15x'</td>
<td>G</td>
<td>144</td>
<td>P</td>
<td>145</td>
<td>£</td>
<td>146</td>
<td>147</td>
<td>$\lambda$</td>
</tr>
<tr>
<td>'16x'</td>
<td>162</td>
<td>163</td>
<td>164</td>
<td>165</td>
<td>166</td>
<td>167</td>
<td>168</td>
<td>169</td>
</tr>
<tr>
<td>'17x'</td>
<td>a</td>
<td>170</td>
<td>a</td>
<td>171</td>
<td>a</td>
<td>172</td>
<td>a</td>
<td>173</td>
</tr>
<tr>
<td>'18x'</td>
<td>176</td>
<td>$\pm$</td>
<td>177</td>
<td>$\pm$</td>
<td>178</td>
<td>$\pm$</td>
<td>179</td>
<td>$\pm$</td>
</tr>
<tr>
<td>'19x'</td>
<td>$\star$</td>
<td>184</td>
<td>1</td>
<td>185</td>
<td>$\circ$</td>
<td>186</td>
<td>$\sqrt{187}$</td>
<td>$\frac{1}{4}$</td>
</tr>
<tr>
<td>'20x'</td>
<td>192</td>
<td>193</td>
<td>194</td>
<td>195</td>
<td>196</td>
<td>197</td>
<td>198</td>
<td>199</td>
</tr>
<tr>
<td>'21x'</td>
<td>200</td>
<td>201</td>
<td>202</td>
<td>203</td>
<td>204</td>
<td>205</td>
<td>206</td>
<td>207</td>
</tr>
<tr>
<td>'22x'</td>
<td>208</td>
<td>209</td>
<td>210</td>
<td>211</td>
<td>212</td>
<td>213</td>
<td>$\times$</td>
<td>214</td>
</tr>
<tr>
<td>'23x'</td>
<td>216</td>
<td>217</td>
<td>218</td>
<td>219</td>
<td>220</td>
<td>221</td>
<td>222</td>
<td>223</td>
</tr>
<tr>
<td>'24x'</td>
<td>224</td>
<td>225</td>
<td>226</td>
<td>227</td>
<td>228</td>
<td>229</td>
<td>230</td>
<td>231</td>
</tr>
<tr>
<td>'25x'</td>
<td>232</td>
<td>233</td>
<td>234</td>
<td>235</td>
<td>236</td>
<td>237</td>
<td>238</td>
<td>239</td>
</tr>
<tr>
<td>'26x'</td>
<td>240</td>
<td>241</td>
<td>242</td>
<td>243</td>
<td>244</td>
<td>245</td>
<td>246</td>
<td>247</td>
</tr>
<tr>
<td>'27x'</td>
<td>248</td>
<td>249</td>
<td>250</td>
<td>251</td>
<td>252</td>
<td>253</td>
<td>254</td>
<td>255</td>
</tr>
</tbody>
</table>
Typesetting Greek with LaTeX

Cochineal offers a rather complete LGR-encoded glyph collection, lacking just a few ancient symbols.
The file lgrenc.def, which is loaded whenever LGR is first loaded, specifies the above and more transcodings from Latin to Greek. See the usage.pdf file in the babel-greek documentation.

The tfm files associated with the Cochineal LGR encoding were constructed with ligatures defined so that most Greek characters could be constructed using Latin input.

Note that you do not have to load the babel package in order for the transcoding and ligatures to function properly.

**Using Cochineal LGR Greek in other LaTeX packages**

There is nothing particular about Cochineal here that would not apply to any other text font package with a working LGR setup, as described above. There was for a number of years a small package substitutefont that allowed the substitution of, say, LGR Cochineal Greek as the LGR Greek font in another text font package that lacked a suitable one of its own. This functioned less well as LaTeX evolved and its functionality was absorbed in 2020 by the LaTeX command \DeclareFontFamilySubstitution, which shares some of the problems of the now obsolete \substitutefont. I've added this short section to describe some work-arounds those who wish to have access to this functionality.

Here are the problems I encountered:

- If you try to add LGR Cochineal Greek to another package that has no support for LGR Greek, the attempt would fail. The work-around is to add in your preamble lines like

  \usepackage[\textit{LGR,T1}]\{fontenc\}
  \DeclareFontFamily\{LGR\}\{\textit{rmdefault}\}\{}

  after loading the text font package.

- Many text font packages do not settle on the final definition of \textit{\textit{rmdefault}} until just before processing \begin{document}. You may take care of all possibilities by adding to your preamble (near the end), imagining we are adding Cochineal LGR Greek to newtx:

  \DeclareFontFamilySubstitution\{LGR\}\{ntxtlf\}\{Cochineal-TLF\}
  \DeclareFontFamilySubstitution\{LGR\}\{ntxlf\}\{Cochineal-LF\}
  \DeclareFontFamilySubstitution\{LGR\}\{ntxtosf\}\{Cochineal-TOSF\}
  \DeclareFontFamilySubstitution\{LGR\}\{ntxosf\}\{Cochineal-OsF\}

  To reconcile the difference in size between newtx and Cochineal, with respective x-heights of 450 and 432, the former is 1.042 times taller than the latter, and so we have to set the scale for Cochineal, \texttt{\textbackslash Cochineal\textbackslash scale} to 1.042 times that of newtx, \texttt{\textbackslash ntx\textbackslash scale}. So, the next line in the preamble should be something like

  \makeatletter
  \@tmpdima=\ntx\textbackslash scale\p@ \@tmpdima=1.042\@tmpdima
Typesetting Russian

With T2A encoding, the process is the same as with other T2A-encoded fonts, though the gaps in coverage may affect users of a number of non-Russian Cyrillic scripts. The only figure style provided is tabular lining (TLF.)
The OT2 encoding (supposedly obsolete, but still useful) is intended for limited use in producing Russian characters with a Western keyboard, making by means of \TeX a transliteration of ASCII for most characters in the range 33–122, and providing ligatures to generate the rest, just as in the Greek case. See the documentation of \texttt{nimbus15} for further details.

<table>
<thead>
<tr>
<th>\texttt{00x}</th>
<th>\texttt{01x}</th>
<th>\texttt{02x}</th>
<th>\texttt{03x}</th>
<th>\texttt{04x}</th>
<th>\texttt{05x}</th>
<th>\texttt{06x}</th>
<th>\texttt{07x}</th>
<th>\texttt{08x}</th>
<th>\texttt{09x}</th>
<th>\texttt{0Ax}</th>
<th>\texttt{0Bx}</th>
<th>\texttt{0Cx}</th>
<th>\texttt{0Dx}</th>
<th>\texttt{0Ex}</th>
<th>\texttt{0Fx}</th>
</tr>
</thead>
<tbody>
<tr>
<td>'0'</td>
<td>'1'</td>
<td>'2'</td>
<td>'3'</td>
<td>'4'</td>
<td>'5'</td>
<td>'6'</td>
<td>'7'</td>
<td>'0x'</td>
<td>'1x'</td>
<td>'2x'</td>
<td>'3x'</td>
<td>'4x'</td>
<td>'5x'</td>
<td>'6x'</td>
<td>'7x'</td>
</tr>
<tr>
<td>ऎ</td>
<td>ऍ</td>
<td>ऎ</td>
<td>ऎ</td>
<td>ऎ</td>
<td>ऎ</td>
<td>ऎ</td>
<td>ऎ</td>
<td>ऎ</td>
<td>ऍ</td>
<td>ऎ</td>
<td>ऎ</td>
<td>ऎ</td>
<td>ऎ</td>
<td>ऎ</td>
<td>ऎ</td>
</tr>
<tr>
<td>ऎ</td>
<td>ऍ</td>
<td>ऎ</td>
<td>ऎ</td>
<td>ऎ</td>
<td>ऎ</td>
<td>ऎ</td>
<td>ऎ</td>
<td>ऎ</td>
<td>ऍ</td>
<td>ऎ</td>
<td>ऎ</td>
<td>ऎ</td>
<td>ऎ</td>
<td>ऎ</td>
<td>ऎ</td>
</tr>
<tr>
<td>औ</td>
<td>ख</td>
<td>औ</td>
<td>औ</td>
<td>औ</td>
<td>औ</td>
<td>औ</td>
<td>औ</td>
<td>औ</td>
<td>ऍ</td>
<td>औ</td>
<td>औ</td>
<td>औ</td>
<td>औ</td>
<td>औ</td>
<td>औ</td>
</tr>
<tr>
<td>औ</td>
<td>ख</td>
<td>औ</td>
<td>औ</td>
<td>औ</td>
<td>औ</td>
<td>औ</td>
<td>औ</td>
<td>औ</td>
<td>ऍ</td>
<td>औ</td>
<td>औ</td>
<td>औ</td>
<td>औ</td>
<td>औ</td>
<td>औ</td>
</tr>
<tr>
<td>व</td>
<td>व</td>
<td>व</td>
<td>व</td>
<td>व</td>
<td>व</td>
<td>व</td>
<td>व</td>
<td>व</td>
<td>व</td>
<td>व</td>
<td>व</td>
<td>व</td>
<td>व</td>
<td>व</td>
<td>व</td>
</tr>
<tr>
<td>व</td>
<td>व</td>
<td>व</td>
<td>व</td>
<td>व</td>
<td>व</td>
<td>व</td>
<td>व</td>
<td>व</td>
<td>व</td>
<td>व</td>
<td>व</td>
<td>व</td>
<td>व</td>
<td>व</td>
<td>व</td>
</tr>
<tr>
<td>व</td>
<td>व</td>
<td>व</td>
<td>व</td>
<td>व</td>
<td>व</td>
<td>व</td>
<td>व</td>
<td>व</td>
<td>व</td>
<td>व</td>
<td>व</td>
<td>व</td>
<td>व</td>
<td>व</td>
<td>व</td>
</tr>
<tr>
<td>व</td>
<td>व</td>
<td>व</td>
<td>व</td>
<td>व</td>
<td>व</td>
<td>व</td>
<td>व</td>
<td>व</td>
<td>व</td>
<td>व</td>
<td>व</td>
<td>व</td>
<td>व</td>
<td>व</td>
<td>व</td>
</tr>
</tbody>
</table>

**Additional glyphs for use in German orthography**

Prior to version 1.050, \texttt{cochineal} offered basic support for German orthography, having all required accented glyphs and the lower case ꞎ, as well as a small caps ꞃ. Under \texttt{LaTeX}, the T1 encoding
contained S_S, but only as a synthesized character in the tfm. Unicode users could not make use of S_S as it was not present in the otf. So, with unicode tex processing:

```
{\addfontfeature{StylisticSet=1}\ss \textsc{\ss}}
```

typesets, as in LaTeX processing, to

ß ß

Note also that in unicode processing, in order to obtain the expected case change behavior, it may be necessary to add in your preamble:

```
\uccode`ß="1E9E
```

As of version 1.050 of cochineal, there are now glyphs in each style for U+1E9E and for its small caps version, as well as S_S as a real character, accessible under unicode TeX. The glyphs may be used as the uppercase and small caps versions of germanbls. Currently, the new glyphs are not available in any of the LaTeX encodings and must be used via unicode TeX.

The following tables show how to access the new glyphs in unicode TeX. Note that you will need to set StylisticSet=1 if you wish not to use the new sharp-s glyphs.

### New symbols:

<table>
<thead>
<tr>
<th>Glyph name</th>
<th>glyph</th>
<th>macro</th>
</tr>
</thead>
<tbody>
<tr>
<td>uni1E9E</td>
<td>ß</td>
<td>\symbol{&quot;1E9E}</td>
</tr>
<tr>
<td>uni1E9E.ss01</td>
<td>SS</td>
<td>\addfontfeature{StylisticSet=1}\symbol{&quot;1E9E}</td>
</tr>
<tr>
<td>germanbls.sc</td>
<td>ss</td>
<td>\textsc{\ss}</td>
</tr>
<tr>
<td>germanbls.sc.ss01</td>
<td>ss</td>
<td>\addfontfeature{StylisticSet=1}\textsc{\ss}</td>
</tr>
</tbody>
</table>

### Effect of choice of StylisticSet:

<table>
<thead>
<tr>
<th>StylisticSet</th>
<th>\ss</th>
<th>SS</th>
<th>MakeUppercase{\ss}</th>
<th>\textsc{\ss}</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>ß</td>
<td>ß ß</td>
<td>ß ß</td>
<td>ß ß</td>
</tr>
<tr>
<td>=1</td>
<td>ß</td>
<td>SS</td>
<td>SS</td>
<td>ss</td>
</tr>
</tbody>
</table>