Problems for Indic typography in current OpenType Layout implementations

John Hudson, Tiro Typeworks Ltd., 18 January 2013

In August 2012, I wrote to the OpenType developer email discussion list regarding observed issues with cross-cluster context lookups in both Microsoft and Adobe Indic shaping engines (all tested). These shaping engines not only treat the cluster (sometimes referred to as the syllable, but this is linguistically not always the case) as the basic unit of shaping for Indic script and language processing, but appear to limit the application of OpenType Layout features, lookup input and context strings to individual clusters. While this makes sense for what Microsoft's Indic font specifications refer to as 'basic shaping forms' features, *e.g.* <akhn>, <rphf>, <blwf> etc., even when one moves into 'mandatory presentation forms' features handling GSUB for *e.g.* above- or below-base substitutions cross-cluster contexts might be *crucial* for correct handling of adjacency issues (collisions or near-collisions, spacing problems). Typography is more than the default shaping of a script or language: it consists of levels of refinement and at the microtypographic level centres on the relationship of glyphs in visual proximity *regardless of their linguistic or script-systemic relationship.* In this, typography often follows text creation strategies and tactics found in manuscript traditions, and one has only to examine the range of techniques employed by Indian scribes to deal with adjacency issues to realise that cross-cluster interaction is presumed in the writing of these scripts. It should be also in OpenType Layout.

This document illustrates problems arising from the current OpenType Layout implementations as they show in two scripts, Gurmukhi and Bengali. The two typefaces used to illustrate the problem are Murty Gurmukhi (John Hudson and Fiona Ross) and Sarkar (Tim Holloway, Neelakash Kshetrimayum and Fiona Ross), both built by Tiro Typeworks Ltd.. Murty Gurmukhi is one of a suite of custom fonts being developed for Harvard University Press, and Sarkar was commissioned by Anandabazar Patrika, the largest newspaper publisher in India. The fonts contain contextual lookups in both the GSUB and GPOS tables to handle adjacency problems both within and across cluster boundaries; only the lookups limited to individual clusters work in applications using the Windows and Adobe Indic layout engines. All the lookups function correctly in the VOLT development proofing tool, indicating that the limitation is applied at the shaping engine level. In general, I've built actual GSUB substitutions to occur within cluster limits, with only context strings looking forward or backwards across syllable boundaries; some GPOS kerning lookups require pair adjustment across syllable boundaries, with or without context.

ILLUSTRATION 1

Above-line proximity problems in Gurmukhi script. In these situations marks and other above-line glyphs are either too close or actually colliding. In order to correct these problems, lookups in the font insert head-line extending shims and then contextually kern these to achieve correct distances. The red examples show the results with existing shaping engines; the green show how the same character sequences would appear if cross-syllable lookups were applied.

ILLUSTRATION 2

Above-line proximity problems in Bengali script. In the words of Aveek Sarkar, publisher of Anandabazar Patrika, *the Bengali script is a house with too many people in the upper storey: there are a variety of marks and above-line 'flags' that can interact both within and across syllable boundaries, and these produce common collisions that scribes developed various techniques to resolve. These and new techniques are possible with OpenType Layout contextual behaviour, as explored by Tim Holloway in this new newspaper typeface, but presently only those within syllables are working correctly in applications; the following are not. In this illustration, a variant short flag on the second cluster should be contextually substituted to avoid collision with the mark on the preceding cluster.*

ILLUSTRATION 3

Final form substitution problems in Bengali script. In addition to the two common initial form substitutions handled in Bengali layout using the <init> feature, the Sarkar typeface includes a number of final form variant letters with shaved head-line connection; these are stylistically more elegant and also improve spacing between words. These are contextually triggered by the absence of a following letter (ignoring marks); however, we had to disable these lookups in the current fonts because the Indic shaping engines were interpreting the end of any cluster as contextually identical to the end of a word, ignoring the presence of the adjacent letter beginning the next cluster. In this illustration, the red example shows how the text incorrectly displays with a broken head-line if the final form lookup is applied by the shaping engines; the green shows how it should apply, only to word-final glyphs.

Note that this shaping behaviour could also be correctly implemented by adding <fina> feature analysis and subsitution to Indic script shaping.