Re: Aksara Support in UTS #29
From: Mark Davis
Date: 2017-05-11
Draft: link

Document L2/17-094 “Background of Indic segmentation” is directed towards changes to UAX 29 grapheme cluster boundaries, to match “Indic Orthographic syllables”. See also the previous L2/16-016.

However, most of the discussion of orthographic syllables is not relevant to the current UAX #29, since the only part of the #29 that would need to be changed to accommodate the behavior documented in that document would be to disallow breaks between a virama and a following entity X.

Should we decide to make such a change, the table below outlines how that could be done. The yellow highlights the substantive additions; the rest is rewriting for clarity. Note: the new rule could go anywhere after GB5 and before GB999, since they are all "gluing", but it seems convenient to put it at (c).

Issues:

1. What is the exact set of X? Some values are proposed in L2/16-016.
2. It might well be that implementations only want to disallow breaks after virama in those cases where the characters on either side “merge” and the visible virama disappears. Such visual merger cannot be determined simply from the characters; it depends on the font and rendering system. In such a case, the most that UAX 29 could do is reflect the behavior of most fonts, or at least the most common-denominator visual behavior.

OLD	NEW
Grapheme_Cluster_Boundaries	
... UI interactions (such as mouse selection, arrow key movement, backspacing)...	... UI interactions (backspacing)... **[new paragraph]** Grapheme clusters can only provide an approximate answer to the question "Where to put cursors". That really should be supported by the text editing framework, depending on the lower level text rendering engine. That is the only entity that knows where the edges of glyphs are, and how they correspond to the underlying characters. It is the entity that knows that X+Y are represented as a single glyph, and cannot have a cursor between them. Or that in the representation of X+Y, the glyph for Y overlaps with the one for X (true generally when GC(Y)=Mn, but in complex scripts there are edge cases). For cursoring, the most that grapheme clusters can supply is an approximation to LCD fonts for the script.
Grapheme_Cluster_Break_Property_Values	***[Add two new categories with initial contents as follows, over time extending to different scripts and refining the contents]***
	Virama U+094D DEVANAGARI SIGN VIRAMA LinkingConsonant U+0915..U+0939 DEVANAGARI LETTER KA..DEVANAGARI LETTER HA U+0958..U+095F DEVANAGARI LETTER QA..DEVANAGARI LETTER YYA U+0978..U+097F DEVANAGARI LETTER MARWARI DDA
Grapheme_Cluster_Boundary_Rules	

<table>
<tr>
<td>The same rules are used for the Unicode specification of boundaries for both legacy grapheme clusters and extended grapheme clusters, with one exception. The extended grapheme clusters add rules GB9a and GB9b, while the legacy grapheme clusters omit them.
...</td>
<td>The same rules are used for the two variants of grapheme clusters, except the rules GB9a, GB9b, and GB9c. The following table shows the differences, which are also marked on the rules themselves.
Among the variants, the extended rules are recommended, except where the legacy variant is required for a specific environment. These are general rules: language-specific rules can be requested in CLDR.

<table>
<tr><th>Grapheme Cluster Variant</th><th>Includes</th><th>Excludes</th></tr>
<tr><td>LG: legacy grapheme clusters</td><td></td><td>GB9a, GB9b, GB</td></tr>
<tr><td>EG: extended grapheme clusters</td><td>GB9a, GB9b, GB9a</td><td></td></tr>
</table>
</td>
</tr>
<tr>
<td>Only for extended grapheme clusters:
Do not break before SpacingMarks, or after Prepend characters.

GB9a × SpacingMark
GB9b Prepend ×</td>
<td>The following rule only applies to extended grapheme clusters:
Do not break before SpacingMarks, or after Prepend characters, or between certain viramas and following consonants.

GB9a × SpacingMark
GB9b Prepend ×
GB9c Virama × LinkingConsonant</td>
</tr>
</table>