Variation sequences for combining marks

Norbert Lindenberg
2020-09-16

Proposal

I propose to replace the restriction against variation sequences for nonspacing combining marks in the Unicode Standard with a restriction against variation sequences for combining marks with combining classes other than 0.

Variation sequences and normalization

The current restriction against variation sequences for nonspacing combining marks is described as "necessary to prevent problems in the interpretation of such sequences in normalized text". The problems that such variation sequences might cause are related to the *Canonical Ordering Algorithm* in section 3.11 *Normalization Forms* of The Unicode Standard. This phase of the Unicode normalization algorithm looks at uninterrupted sequences of marks with combining classes other than 0 and reorders these marks by their combining classes in ascending order. This is commonly used to erase irrelevant differences in mark sequences in non-Brahmic scripts, e.g., when a base character has one mark on top and another at the bottom. It is less common in Brahmic scripts, whose Unicode encoding usually orders according to a linguistic model, not the visual model underlying canonical ordering.

For the following discussion, we use a few sample characters:

- U+1037 ◌့ MYANMAR SIGN DOT BELOW, ccc = 7
- U+103A ◌် MYANMAR SIGN ASAT, ccc = 9
- U+108D ◌ွ MYANMAR SIGN SHAN COUNCIL EMPHATIC TONE, ccc = 220
- U+FE00 VARIATION SELECTOR-1, ccc = 0

The first problem arising from variation sequences in canonical ordering is that variation selectors have ccc = 0, and so would interrupt sequences of marks that otherwise would be treated as a reorderable in canonical ordering. For example, the sequence 103A 1037 is reordered to 1037 103A, but if 103A were the initial character in a variation sequence, followed by FE00, then 103A FE00 1037 could no longer be reordered.

The second problem is that canonical ordering is based on code points and does not take variation sequences into consideration. For example, the sequence 108D 103A FE00 would be reordered to 103A 108D FE00, separating the variation selector from the ASAT it belongs to and attaching it to SHAN COUNCIL EMPHATIC TONE, creating an invalid variation sequence.

In the context of these problems, however, the restriction against nonspacing combining marks is both too loose and too restrictive to meet its goal. It's too loose because there are marks that have ccc ≠ 0 but also gc ≠ Mn, such as several viramas (ccc = 9), Hangul tone marks (ccc = 224), and musical symbols (ccc = 216). And it's too restrictive because many nonspacing marks, especially in Brahmic scripts, have ccc = 0, so that they would block reordering anyway and adding a variation selector wouldn't have any impact on canonical ordering. Hence the proposal to base the restriction on the combining class, not the general category, so that it correctly addresses the two problems.

Variation sequences and text segmentation

Variation selectors have gc = Mn, and Unicode text segmentation never separates nonspacing combining marks from preceding characters. Allowing other nonspacing marks to start variation sequences has no impact on text segmentation.

Variation sequences and line breaking

Variation selectors have Line_Break value CM (combining mark), and Unicode line breaking never separates characters of this line break class from preceding characters. Allowing other nonspacing marks to start variation sequences has no impact on line breaking.

Variation sequences and rendering

Any new variation sequence requires updates to fonts for the script it belongs to, to map the new sequence to an appropriate glyph – there's no difference between base characters and combining marks in this respect. However, OpenType shaping engines and fonts using Graphite or Apple Advanced Typography technologies for Brahmic scripts also implement validation, checking incoming text against cluster models and inserting dotted circles into character sequences that are malformed according to their cluster models.

The cluster model used by OpenType shaping engine do not currently allow variation sequences for combining marks, so they would have to be updated to accommodate such sequences if any were actually defined. The cluster models implemented by Graphite- or AAT-based fonts are generally not documented, but would likely have to be updated as well.

Suggested wording

- In section 23.4 *Variation Selectors* of The Unicode Standard, change the paragraph introducing variation sequences to remove the discussion of restrictions:

> ***Variation Sequence***. A variation sequence always consists of a base character or a ~~spacing~~ combining mark ~~(gc = Mc)~~ followed by a single variation selector character. That two-element sequence is referred to as a variant of the base character or ~~spacing~~ combining mark. For simplicity of exposition, the following discussion only mentions base characters; variation sequences involving ~~spacing~~ combining marks are uncommon, but otherwise behave similarly.

- In the same section, change the paragraph discussing restrictions on initial characters in variation sequences:

> The initial character in a variation sequence is never a ~~nonspacing~~ combining mark ~~(gc = Mn)~~ with a combining class other than 0 or a canonical decomposable character. These restrictions on the initial character of a variation sequence are necessary to prevent problems in the interpretation of such sequences in normalized text.