L2/10-455

2010-11-01

Title: Of hamza and other harakat

Author: Roozbeh Pournader (HighTech Passport)

Action: For consideration by the UTC

Date: 2010-11-01

Introduction

While there has been lots of confusion and protestation about the way Unicode encodes the *hamza* and its various alternative forms and shapes, Unicode has mostly been on the right track in the way it handles the issue.

This document tries to document the cases Unicode has not done its best, and tries to suggest ways to document or fix the issues.

Requests

1. Document the special behavior of U+064A Arabic Letter Yeh when combined with U+0654 Arabic Hamza Above.
2. Consider the pros and cons of the sequence <Yeh, CGJ, Hamza Above> instead of encoding U+08A8 Yeh With Two Dots Below and Hamza Above.
3. Handle the inconsistency in the treatment of U+0681 and U+076C and provide a guideline for future encoding of characters with *hamza* above or below them.
4. Handle the inconsistency in the encoding of U+06C7 and U+06C8.
5. Provide a guideline for future encoding letter-making Arabic diacritics and Arabic letters composed from such diacritics.
6. Provide a guideline for rendering Arabic letters with multiple combining marks, especially when there is a mix of older and newer combining classes.
7. Document in more details the hardships that the combining class for Hamza Above and similar letter-making diacritics may create in processing data that is in normalization forms (NFC, NFD, etc).
8. Issue a PRI to find and document the behavior of the high *hamza* characters in the range U+0674 Arabic Letter High Hamza to U+0678 Arabic Letter High Hamza Yeh.

Background

Normalization of Hamza

Unicode 3.0 encoded three new Arabic combining characters, U+0653..0655 (two above and below *hamza*'s, and one *madda* above), and added canonical decompositions of several existing Arabic characters. Unfortunately, before there was time to understand a couple of problems in the new decompositions and fix them, the decompositions were frozen in Unicode 3.1. Later, one new letter was encoded at U+076C Arabic Letter Reh With Hamza Above (Consensus 98-C27, document L2/04-025R, encoded in Unicode 4.1) which further complicated the problem.

Here are the characters that have a *hamza* above or below form, with their decompositions:

Code	*Arabic Letter*	*Decomposition*	*Notes*
U+0623	Alef with Hamza Above	Alef+Hamza Above	
U+0624	Waw with Hamza Above	Waw+Hamza Above	
U+0625	Alef with Hamza Below	Alef+Hamza Below	
U+0626	Yeh with Hamza Above	**Yeh**+Hamza Above	Yeh is dotted, while Yeh with Hamza Above is dotless. The correct decomposition should have been “Alef Maksura+Hamza Above”.
U+0681	Hah with Hamza Above	**NONE**	Should have had a decomposition to “Hah+Hamza Above”.
U+06C0	Heh with **Yeh** Above	**Ae**+Hamza Above	The diacritic is visually similar to hamza, but is very different semantically. Some fonts distinguish them visually.
U+06C2	Heh Goal with **Hamza** Above	Heh Goal+Hamza Above	The diacritic is the same as the “Yeh Above” in U+06C0.
U+06D3	Yeh Barree with Hamza Above	Yeh Barree+Hamza Above	
U+076C	Reh with Hamza Above	**NONE**	Should not have been encoded.

In the author’s belief, the complications arise from the major omission: U+0681 Arabic Letter Hah With Hamza Above was not given a decomposition before decompositions were frozen. Then, there was the not-very-clear Arabic model in the Unicode standard, which varies from visual to semantic. It is the omission and the unclear model that resulted in the encoding of U+076C Reh With Hamza Above and the request for an Arabic Letter Beh With Hamza Above in the original version of L2/10-288.

Other missed normalizations

At the same time that decompositions were being added and getting frozen, two other existing letters were also omitted from getting decompositions:

- U+06C7 Arabic Letter U, used in Kirghiz and Azerbaijani, which should have received a decomposition to Waw+Damma;
- U+06C8 Arabic Letter Yu, used in Uighur, which should have received a decomposition to Waw+Superscript Alef.

Canonical decomposition of Yeh With Hamza Above

When a canonical decomposition was given to U+0626 Yeh With Hamza Above, Yeh+Hamza Above was chosen instead of Alef Maksura+Hamza Above.

This was probably due to two assumptions: First was perhaps that a semantic relationship was considered to exist between “Yeh With Hamza Above” and “Yeh”. The other was perhaps that Yeh was dual-joining while Alef Maksura was assumed to be right-joining. (Alef Maksura's joining type was fixed to be dual-joining shortly thereafter, although there is still software, and people, out there that treat Alef Maksura as right-joining.)

Whatever the reason, this decision made rendering decomposed data a bit harder. For example, the common sequence <Yeh, Fatha, Shadda, Hamza Above> (which is in NFD) now needed to be displayed the same way as <Yeh With Hamza Above, Fatha, Shadda> (in NFC) and <Yeh With Hamza Above, Shadda, Fatha> (canonically equivalent to the one before, in what can be called “logical order” for an Arabic script reader) . This meant that no dots should be displayed under the Yeh when an applications to render the NFD form, requiring

the application to look three characters ahead for knowing how to render the base letter.

Unfortunately, this has not been documented in the text of the standard.

Old and new combining classes

Arabic combining marks tend to have two different kinds of combining classes. There are the old combining classes, those that were given to the basic *harakat*: *fathatan*=27, *dammatan*=28, *kasratan*=29, *fatha*=30, *damma*=31, *kasra*=32, *shadda*=33, *sukun*=34, and *superscript alef*=35 (Koranic alternatives of the basic *harakat* use the same combining class as their main version).

And then there is everything else, which have combining classes of either 220, for combining marks that appear below the base letters, and 230, for those that appear above the base letters. Of the letter-making diacritics, Madda Above, Hamza Above, Hamza Below, Subscript Alef, and Wavy Hamza Below use the new (220/230) combining classes.

These lead into complicated situations with normalized text (like data in NFD and NFC). For example, a letter with a *hamza*, *shadda*, and *dammatan* above it (in near to far order) would have these marks sorted as <Dammatan, Shadda, Hamza Above>, which is actually the reverse order of how one will both read these (logical order) and render these (visual order). If that is Koranic text, it may be followed by something like U+06D9 Small High Lam Alef, resulting in a normalized order of <Dammatan, Shadda, Hamza Above, Small High Lam Alef> but a logical/visual order of <Hamza Above, Shadda, Dammatan, Small High Lam Alef>.

Applications, from text rendering engines to normal text editing procedures like those used to handle backspacing, need to come up with their own normalization for Arabic. This is to both keep the users happy and be compliant with the Unicode Standard's requirement for treating canonically equivalent sequences the same way. For example, when a text file containing the sequence mentioned above is opened, the cursor is moved after the sequence, and backspace is pressed a few times, the combing marks should be deleted in this order: Small High Lam Alef, Dammatan, Shadda, Hamza Above. (Depending on the base letter, the perceived language of the text, and other considerations, the *hamza* and the base letter may be deleted together.)

Although there is some documentation about this in the Unicode Standard, extra documentation, especially emphasizing the complex mechanics

Diacritics encoded later

Unicode 4.0 added U+0615 Arabic Small High Tah for Koranic use. This was intended only for Koranic use and was properly documented as such. The Koranic mark is similar in shape to Small Tah Above, a diacritic used in various languages of South Asia. If one also includes characters encoded in a later version of the Unicode Standard, ten characters use a diacritic that is graphically similar to this character.

Then, Unicode 4.1 added U+065A Arabic Vowel Sign Small V Above and U+065B Arabic Vowel Sign Inverted Small V Above to be used to represent vowel signs for African languages. These were identical or similar in shape to the diacritics used in various existing letters (and those to be encoded later). If one also includes characters encoded in later versions of the standard, eleven characters use a diacritic that is graphically similar to these two characters.

Finally, Unicode 6.0 encoded U+065F Arabic Wavy Hamza Below. This was done very carefully, and the encoding was synchronized with the deprecation of a precomposed letter U+0673 Arabic Letter Alef With Wavy Hamza Below that could not have any decomposition due to stability reasons.

Already representable "characters"

There exist various monolithic Arabic script "letters" that are not encoded as one character, but two. The

following are some examples:

- Beh With Heh Doachashmee, Peh With Heh Doachasmee, ..., Gaf With Heh Doachashmee: aspirated forms of Beh, Peh, etc. Easily encodable using Beh+Heh Doachashmee, Peh+Heh Doachashmee, ...
- Farsi Yeh With Hamza Above: Used as one letter in some Azerbaijani orthographies, a variant of Yeh With Hamza Above that has two dots in medial and initial positions, like Farsi Yeh. Easily represented using Farsi Yeh+Hamza Above.
- Alef Maksura With Hamza Below: Used as one letter in Koranic texts, a tooth with a hamza below, that appears in medial and initial positions only. Easily represented using Alef Maksura+Hamza Below.
- Alef Maksura With Damma Above: Used as one letter in some central Asian orthographies. Appears only in medial and initial forms. Easily represented using Alef Maksura+Damma.
- Alef Maksura With Subscript Alef: Used as one letter in some central Asian orthographies. Appears only in medial and initial forms. Easily represented using Alef Maksura+Subscript Alef.
- Waw With Madda Above: Used as one letter in some central Asian orthographies. Easily represented using Waw+Madda Above.
- Beh With Hamza Above: Used as one letter in Fulfulde (see original version of L2/10-288). Easily represented using Beh+Hamza Above.

The author believes that if proposals for such characters are presented to the committee, they should be rejected by mentioning that these can already representable in Unicode.

It could be argued that the recently approved character U+08A8 Yeh With Two Dots Below and Hamza Above has been representable as the sequence <Yeh, CGJ, Hamza Above>. How existing implementations handle that sequence and the difficulties it may create for the user community is not clear to the author.

Recommendations

1. Add a character note to U+0649 Arabic Letter Alef Maksura:
    1. “Should not be used with U+0654 Arabic Hamza Above. Use U+0626 instead.”
2. Add a character note to U+064A Arabic Letter Yeh:
    - “Always loses its dots when combined with U+0654 Arabic Hamza Above (but not with any other combining character). Use U+08A8 for cases when the dots should be kept.”
3. Add two character notes to U+0654 Arabic Hamza Above:
    - “Removes the dots from U+064A Arabic Letter Yeh when combined with it (but not any other letter).”
    - “Not restricted to hamza semantics, but also to be used for hamza-like shapes, including for creating letters that contain such a shape. Exceptions are the two characters U+0681 and U+076C.”
4. Keep the encoding of U+08A8 and document in the text of the standard that combing grapheme joiner should only be used with the Arabic script if the default order.
5. Document the security problems arising from the ability to represent U+0626 using Alef Maksura+Hamza Above, U+0681 using Hah+Hamza Above, and U+076C using Reh+Hamza Above in the various Unicode documents on security concerns.
6. Document that some Arabic combining characters **MAY NOT** be used for creating new letters (U+065A, U+065B, U+08EA, U+08EB, U+08ED, U+08EE, and the Koranic-only marks, especially U+0615 and U+06DB), while others could and **SHOULD** be used for representing unencoded letters

that contain them as diacritics, even if they are just used as modifiers to create new letters and do not have separate phonemical value (every other Arabic diacritic).

7. Document the above in the editorial text of the Unicode Standard.
8. Update the answers to the last two questions in the Unicode FAQ on “Middle Eastern Scripts and Languages” to make clear the difference between encoded letter-making diacritics (like *Hamza Above*) and non-encoded ones (like *Three Dots Above*).
9. Issue a PRI to find and document the behavior of the high *hamza* characters in the range U+0674 Arabic Letter High Hamza to U+0678 Arabic Letter High Hamza Yeh.

Acknowledgments

Lorna Priest and Martin Hosken’a proposal for encoding new characters for Arabic (L2/10-288R) helped fast-track this from the author’s to-do list to an actual document. Rick McGowan’s and Mark Davis’s encouragement and support were crucial in the development of this document. The title of the document is influenced by a novel by Gabrial García Márquez.