L2/16-315

Re: **Handling interaction between UCD and emoji properties**
To: **UTC**
From: **Mark Davis, Peter Edberg**
Date: **2016-10-31**

We have the following action from UTC #148:

148	**A006**	**Mark Davis, Peter Edberg, Emoji Subcommittee**		**Produce a proposal for handling the interaction between segmentation and emoji properties. Either (1) move some properties into the UCD, or (2) decouple properties from segmentation. Spell out the preferred option(s) and alternatives.**

There are two areas where the emoji properties interact with the UCD properties, in particular, where changing the value of the property Emoji=No to Emoji=Yes should cause other properties to change.

1. Segmentation
    - The Grapheme Break, Word, and Linebreak properties have values that depend on the Emoji property
2. Variation Selectors
    - The UTC has committed to maintaining the invariant that if a character has the property value Emoji_Presentation=No, then it has Emoji and Text variation selectors (VS16 and VS15). Those are established by the data in StandardizedVariants.txt

There are two alternatives for dealing with the interaction.

A. Move some emoji properties into the UCD.
    - This is the simplest approach, but constrains those emoji properties to be only changed with yearly (June) Unicode releases, when (for now) emoji are developing on a faster pace. It also makes the Unicode release yet larger and more complicated.

B. Decouple emoji properties from UCD properties
    - If feasible, this allows development on a different schedule and pace.

Docoupling

The following presents a possible approach to decoupling.

Grapheme Break, Word, and Linebreak

The UTC had already decided to “future proof” the segmentation of emoji, by providing data that removes segmentation boundaries between “emoji-like” characters that could possibly get the value Emoji=Yes in the future, and be part of ZWJ sequences.

Because there was not enough time before the Unicode 9.0 release (June 2016), this was done in CLDR 30 (Oct 2016). It takes the form of a property, plus customized segmentation rules.

- Property: ExtendedPictographic.txt
- Customized Rules: LDML: Extended Pictographic

The relevant text from the from the LDML spec for customizing the rules is:

> Let Extended_Pictographic be defined as in ExtendedPictographic.txt
> Let EmojiRK = [\p{GCB=Regional_Indicator}[*#0-9©®™〰〽]]
> Let EmojiNRK = [\p{Emoji=Yes}-EmojiRK]
>
> The customized rules replacing GB11, WB3c, and LB8a are:

GB11′ **(Extended_Pictographic | EmojiNRK)** ZWJ × **(Extended_Pictographic | EmojiNRK)**
WB3c′ ZWJ × **(Extended_Pictographic | EmojiNRK)**
LB8a′ ZWJ × (ID | **Extended_Pictographic | EmojiNRK**)

The future-proofing of the segmentation rules handles the change from Emoji=No to Emoji=Yes by having an expanded set of characters that could possibly have their Emoji status changed in that way, so it effectively decouples the two properties. That is, as long as characters changed from Emoji=No to Emoji=Yes are in Extended_Pictographic, the UCD does not need to be changed in order for segmentation to still work. Of course, if any characters outside of Extended_Pictographic would need to be changed, that would just have to wait for the next version of Unicode.

That means that Extended_Pictographic needs to (a) include all the prospective characters, and (b) not include extraneous characters (that are neither pictographic symbols nor emoji-like).

We could decouple the Emoji property from segmentation by creating a new **UCD** property called **Extended_Pictographic (EP)** with the contents being the CLDR values for (Extended_Pictographic | EmojiNRK). The amended segmentation rules in UAX 29 and 14 would become:

GB11′ Extended_Pictographic ZWJ × Extended_Pictographic
WB3c′ ZWJ × Extended_Pictographic
LB8a′ ZWJ × (ID | Extended_Pictographic)

That allows any of the Extended_Pictographic characters (like the MALE SIGN) to be changed to have Emoji=Yes without affecting segmentation.

TR51

A parallel change could be made to the definition in TR#51: ED-16. emoji zwj sequence.

This is **not** required for decoupling, but would have the advantage of making the the emoji zwj sequence more stable with respect to changes in the Emoji property. It would do this by allowing a broader set of **trailing characters** in an emoji zwj sequence.

emoji_zwj_sequence := emoji_zwj_element (ZWJ emoji_zwj_element)+
→
emoji_zwj_sequence := emoji_zwj_element (ZWJ (emoji_zwj_element **| *Extended_Pictographic*** *))+*

Variation Selectors

The other problem for changes to emoji properties are the StandardizedVariants.txt. There are a few ways to decouple these. The simplest way is to define that the valid sequences with VS15 and VS16 are no longer established by the presence of those sequences in StandardizedVariants.txt. Instead, in the Unicode Standard (and in the header of StandardizedVariants.txt) we define the valid sequences with VS15 and VS16 to also include any sequences matching the following:

\p{Extended_Pictographic} [\x{FE0E}\x{FE0F}]