RGI_Emoji_Qualification

From: Mark Davis
Date: 2022-06-14

Proposal 1: Add an additional property of strings, RGI_Emoji_Qualification, with property values defined by the corresponding status values in emoji-test.txt, namely:

   Fully_Qualified, Minimally_Qualified, Unqualified

Action: Make the appropriate changes to UTS #51 listed in this document for RGI_Emoji_Qualification

Action: Make the appropriate changes to UTS #18 listed in this document for RGI_Emoji_Qualification

Proposal 2: Add additional sequences to emoji-test.txt with the new status value ‘over-qualified’, and add the corresponding property value Over_Qualified to RGI_Emoji_Qualification.

The sequences in question are characters with Emoji_Presentation=Yes that start a sequence in emoji-variation-sequences.txt with the second field = “emoji style”.

Action: Make the appropriate changes to UTS #51 and emoji-test.txt listed in this document for the new property value Over_Qualified

Detailed Changes

For proposal 1, no data changes need to be made; the property values can be derived from the existing emoji-test.txt file. So the only changes would be some additions to 2 specs.

1a. Add the following definition to UTS #51 in the next version after 15.0.

ED-28. RGI emoji qualification — the status of emoji sequences

1. This is an enumerated property of strings, defined by the emoji-test.txt file [emoji-data]. It assigns one of the three values in ED-18, ED-18a, and ED-19 to each emoji in ED-27
**RGI emoji set** and related sequences with missing variation selectors. The property value names and short aliases are:

- Fully_Qualified, FQE
- Minimally_Qualified, MQE
- Unqualified, UQE

1b. Add **RGI emoji qualification** to [https://unicode.org/reports/tr18/#Full_Properties](https://unicode.org/reports/tr18/#Full_Properties) in the next version of UTS #18, version 24.

Proposal 2 would involve changes to the emoji-test.txt file, plus 2 additions to UTS #51

2a. Add the following definition to [UTS #51](https://unicode.org/reports/tr51) in version 15.1

**ED-19a. over-qualified emoji** — characters with Emoji_Presentation=Yes that start a sequence in `emoji-variation-sequences.txt` with the second field = “emoji style”.

2b. Add Overqualified, OQE to the property values of **RGI emoji qualification** above.

2c. Change the generation program for emoji-test.txt to add the overqualified emoji.

---

**Background**

Why add RGI_Emoji_Qualification?

There has been a longstanding effort in ECMAScript to improve Unicode capabilities. We have responded to that need by extending UTS #18 *Unicode Regular Expressions* very significantly over the past few years.

One valuable outcome is that each web application can avoid downloading significant amounts of data to perform basic tasks with emoji and other Unicode constructs. For example, one developer noted that leveraging the “v flag + ‘RGI_Emoji’ in emoji-test-regex-pattern reduces pattern size from 11,631 to 3,146 bytes, saving 8,485 bytes or -73%”.

Yet what products have to deal with is not just the formal RGI_Emoji. In the wild, emoji are emitted with various combinations of variation selectors: sometimes too few, sometimes too many. Part of that is due to history. Variation sequences were not introduced at the same time as emoji were, and
implementations have only slowly picked up on them. Even now, keyboard implementations are emitting emoji that have missing variation selectors, or too many variation selectors.

By leveraging the data that we already have in emoji-test.txt, we can help address this problem.

Why add Over_Qualified?

These are characters with Emoji_Presentation = true and but that also (for historical reasons) have an emoji presentation sequence. That is, they have an ED-9a, emoji presentation sequence in emoji-variation-sequences.txt. That list is found in unicode.org/emoji/charts/text-style.html in the +EP +EPSq column.

These are valid sequences, and should have been listed in the file. Moreover, they do occur in the wild, so implementations need to be able to easily identify them.

What is the list of Over_Qualified emoji?

The list needs to be generated by the tooling that produces emoji-test.txt. It includes all and only those sequences in emoji-variation-sequences.txt that have a first character with Emoji_Presentation=YES, and a second field = “emoji style”.

Here is a snapshot of what it would look like:

```
231A FE0F ; over-qualified ; # (1.1) WATCH
231B FE0F ; over-qualified ; # (1.1) HOURGLASS
23CF FE0F ; over-qualified ; # (4.0) EJECT SYMBOL
23EA FE0F ; over-qualified ; # (6.0) BLACK LEFT-POINTING DOUBLE TRIANGLE
23ED FE0F ; over-qualified ; # (6.0) BLACK RIGHT-POINTING DOUBLE TRIANGLE WITH VERTICAL BAR
23EE FE0F ; over-qualified ; # (6.0) BLACK LEFT-POINTING DOUBLE TRIANGLE WITH VERTICAL BAR
23EF FE0F ; over-qualified ; # (6.0) BLACK RIGHT-POINTING TRIANGLE WITH DOUBLE VERTICAL BAR
23F1 FE0F ; over-qualified ; # (6.0) STOPWATCH
23F2 FE0F ; over-qualified ; # (6.0) TIMER CLOCK
23F3 FE0F ; over-qualified ; # (6.0) HOURGLASS WITH FLOWING SAND
23F8 FE0F ; over-qualified ; # (7.0) DOUBLE VERTICAL BAR
23FA FE0F ; over-qualified ; # (7.0) BLACK SQUARE FOR STOP
23FA FE0F ; over-qualified ; # (7.0) BLACK CIRCLE FOR RECORD
24C2 FE0F ; over-qualified ; # (1.1) CIRCLED LATIN CAPITAL LETTER M
25AA FE0F ; over-qualified ; # (1.1) BLACK SMALL SQUARE
25AB FE0F ; over-qualified ; # (1.1) WHITE SMALL SQUARE
25FB FE0F ; over-qualified ; # (3.2) WHITE MEDIUM SQUARE
25FC FE0F ; over-qualified ; # (3.2) BLACK MEDIUM SQUARE
25FD FE0F ; over-qualified ; # (3.2) WHITE MEDIUM SMALL SQUARE
25FE FE0F ; over-qualified ; # (3.2) BLACK MEDIUM SMALL SQUARE
260E FE0F ; over-qualified ; # (1.1) BLACK TELEPHONE
261D FE0F ; over-qualified ; # (1.1) WHITE UP POINTING INDEX
262A FE0F ; over-qualified ; # (1.1) STAR AND CRESCENT
262E FE0F ; over-qualified ; # (1.1) PEACE SYMBOL
262F FE0F ; over-qualified ; # (1.1) YIN YANG
263A FE0F ; over-qualified ; # (1.1) WHITE SMILING FACE
264A FE0F ; over-qualified ; # (1.1) GEMINI
264B FE0F ; over-qualified ; # (1.1) CANCER
264C FE0F ; over-qualified ; # (1.1) LEO
264D FE0F ; over-qualified ; # (1.1) VIRGO
264E FE0F ; over-qualified ; # (1.1) LIBRA
```