L2/20-051

Title: Response to feedback on UTS18
From: Mark Davis
Date: 2020 Jan 14

See http://www.unicode.org/reports/tr18/tr18-20.html

Feedback on Public Review

http://www.unicode.org/review/pri404/feedback.html
(below the red line: **Feedback above this line was reviewed during UTS #161 in October, 2019.**)

Date/Time: Thu Oct 10 17:11:17 CDT 2019
Name: Richard Wordingham
Report Type: Error Report
Opt Subject: Wrong Section Reference from UTS#18 to UTS#10
There is a wrong section reference following the definition of RL3.2. Instead of "Section 6.9" in "See Section 6.9, Handling Collation Graphemes in UTS #10", it should be "Section 9.9".

Recommended: add typo to list of items to fix.

Date/Time: Sat Oct 19 08:45:43 CDT 2019
Name: Richard Wordingham
Report Type: Public Review Issue
Opt Subject: PRI 404: RL3.2
Order

I do not believe that it is intended to restrict this concepts to collations used for ordering; collations used for searching make sense. The word 'order' needs to be removed or de-emphasised. For example, the visual order Tai scripts have contractions of preposed vowel plus consonant in DUCET, but these are removed for search collations.

Canonical Equivalence

The mathematically-natural extension of strings of 8-bit characters is traces of Unicode strings under canonical equivalence. However RL3.2 is completely inappropriate for this extension, for collation grapheme clusters are not closed under canonical equivalence. (Example: The common-enough misspelling รูู้ <U+0E23, U+0E39, U+0E39, U+0E49> of รู้. Under DUCET, the normalised form consists of three collation grapheme clusters, but one of the canonical equivalents consists of two collation grapheme clusters.)

Possible Solution

It might be possible to make the removal of RL3.2 for Level 3 conformance dependent on compliance with a re-instated RL2.1 with teeth. For example, I would expect "óớ" to match "(ó)+(\u031b)+".

Recommended: Moot if Level 3 is dropped (which is recommended below)

Date/Time: Sat Oct 19 19:31:51 CDT 2019
Name: Richard Wordingham
Report Type: Public Review Issue
Opt Subject: PRI 404 - Update to UTS #18
For Section 2.8, \p{isNFD} is useful as a *character* property in contexts such as [\p{L}&\p{isNFD}]\p{Mn}*, where one is handling all letters, regardless of whether they're explicitly encoded in Unicode.

Recommended: Add example of how to achieve that with NFD_QC.

Date/Time: Sun Jan 5 23:41:36 CST 2020
Name: Karl Williamson
Report Type: Public Review Issue
Opt Subject: PRI 404 UTS #18

I am using a single form to submit all my comments about this PRI. I hope that's most convenient for you.

Recommended: most of these agree with current state; others called out.

I am used to seeing the document organized by level. And retaining that is fine with me. Maybe newcomers would be better off the other way; I don't know

Character class is a better term, so I agree with that change

I'm fine with removing level 3 conformance.

I think the additions to 1.2 are good. Note that string matching has long been an issue under case-insensitive matching, when a string may be case folded to by a single code point. The most common example is the LATIN SHARP S, both upper and lower, which matches strings like 'SS', 'Ss', \x{17f}\x{17f} caselessly.

Regarding \m vs \p, I support your option 2, to use \m in this document for the most clarity.

Recommended: Take \m choice into consideration during UTC discussion

Perl does support symmetric difference, and will continue to support it even if you remove it. It corresponds to exclusive or. I do not know how much actual use it has gotten. An argument for keeping it is that it creates a complete set of operations on sets that correspond to non-set operators.

FYI, I originally implemented in Perl union, etc., all at the same precedence level, but user feedback that this was confusing given that the language otherwise emulates C precedence, forced me to change it. I would expect that the precedence chosen should mirror that of the containing language.

You say "it far is better to write \u{1F44E) rather than \uD83D\uDC4E (using UTF-16) or \xF0\x9F\x91\x8E (using UTF-8)." I completely agree with that, but I think it should be phrased "it is far better ..."

Recommended: add typo to list of items to fix.

Section 2.7:

I would like to see an example of a useful regex that contains the newly required Equivalent_Unified_Ideograph property.

Recommended: add new property example.

Though unchanged in this release, the earlier example "Characters with names starting with "LATIN LETTER" and ending with "P":" seems to me like no one would ever want to use this except some nerds out drinking, looking for Unicode trivia. Names are pretty arbitrary and capricious, and I don't understand what the motivation for this query would be. I could see some use for the one about names containing "variation", etc, as those names are less capricious (I hope anyway).

Recommended: add better example.

I have some problems with Identifier_Status and _Type. I'm fine with including them. I do wish that any required property files would be part of the UCD, instead of me having to go fish for them each new release. I also don't think these are ready for prime time. People writing regex patterns using them will really want to have abbreviated names for them and their property values, as these are quite long. If you actually put them into the UCD, I suspect you would think more about good abbreviations, and add those. And contrary to what it's said in TR39, these files aren't completely in the UCD format. They lack an @missing line for example, and the heading comments are in a different format, which would make me modify my parser to handle them. And it just seems sloppy for you to not bother to make things consistent, forcing extra work on those who would implement your standard. The new file RegexPropertyAliases.txt will help, but it has a completely different format that will force me to write code to parse it.

Recommended: Retain the RegexPropertyAliases.txt since it provides a needed workaround for now. In the longer term, investigate more uniform parsing formats, and either go completely decentralized (each property file contained all data for 1 property, including aliases, etc.), or completely centralized format, with a single source for all property info, whether in the UCD or not.

Date/Time: Mon Jan 6 12:48:09 CST 2020
Name: Nozomu Katō
Report Type: Public Review Issue
Opt Subject: PRI 404 UTS#18
About 1.2 Properties:

I do not think that it is a good idea to use the \p notation for expressing specific sequences of code points, i.e., properties of strings, in addition to a single code point. I have the impression that it is not kind to users that some \p{...} can be used in a character class while some \p{...} cannot.

What are called "properties of strings" in the proposed text look like aggregate versions of Named Character Sequences.

Whichever option TC39 chooses, I would not like the UTC to support or promote using of \p for both properties of characters and strings.

Recommended: Take \m choice into consideration during UTC discussion

Review Notes

See http://www.unicode.org/reports/tr18/tr18-20.html

Review note: This document is organized into main sections by Levels, not by topics.

Recommended: Add action to reorganize in v14.0

Review note: The term "character range" has been replaced by Character Class, without highlighting each instance.

Recommended: Drop review note.

Review item: remove C3 if we remove Level 3, plus references to C3. and its subclauses.

Recommended: remove along with Level 3.

Review Note:

The syntax for expressing properties of strings is not settled; feedback is welcome.

Recommended: Drop the review note, pick either \p or \m as per UTC decision. Part of the text will be retained to explain the syntax.

Review note: the point made in the above paragraph has been moved earlier in this section.

Recommended: Drop note.

Review note: this section was moved to Section 2.8 Optional Properties; other sections have not yet been renumbered.

Recommended: Renumber sections, and drop note.

Review Notes:

- The syntax might be better merged into 0.1.1. Feedback is welcome.
- Symmetric difference is not typically supported, and may just be distracting; feedback is welcome as to whether we should remove it.

Recommended:

- Postpone merging syntax into 0.1.1 (until the restructuring).
- Leave Symmetric difference, but

Fix $(A-B)\cup(B-A) = (A\cup B)-(A\cap B)$
$\Rightarrow(A\setminus B)\cup(B\setminus A) = (A\cup B)\setminus(A\cap B)$

Review item: remove the above line if we remove Level 3.

Recommended: Remove along with level 3

Review Notes:

- Some of the links under Emoji will not work yet.
- ...

Recommended: drop review note

Review Note: This section was moved from...

Recommended: drop review note

Review Note: Level 3 was originally included to consider...

Recommended: As described, drop Level 3.

Review Note: The only purpose of this section was for Level 3.

Recommended: As described, drop Level 3.

Review Note: The name, location, format, and contents of the file are subject to review/feedback/change.

Recommended: drop the note unless the UTC decides to make changes. (There is a proposal to do so.)

Review Note: The above needs careful review. Comments welcome.

Recommended: Drop the note unless there is feedback on the section.

Modification Section

Clean up for publication, removing obsolete items.

Other

Major topics

\p vs \m

- **TBD**

Data file format

- **Add metaproperty info into UTS #51 and UTS #39 data files based on the file, as header comments.**

Generate UTC Discussion

Move indic properties to 2.8 Optional Properties

Writing Systems Versus Blocks ⇒ add extension F & G

Next Rev, add symmetric diff to table in Annex E.