L2/11-030

Case-insensitive matching in regular expressions in UTS-18

There are a number of semantic issues regarding caseless (case-insensitive) regular expression matching that are not addressed in UTS-18, and which make it hard for implementors. I have a proposal for one issue, but am not really qualified to advance a proposal for others, but wish to bring it to the attention of the UTC to discuss.

I'm trying to avoid using notation, because I'm told that the notations are variable and may not be familiar to the UTC members.

The first issue is what to do about matching properties caselessly. My proposal is that properties should in general match the same set of code points under both cased and case-insensitive matching. The reason is that a property applies to a particular code point, and may not apply to the fold case of that code point (which may consist of multiple code points), and lead to unexpected results.. For example, the block, age, changes when case folded, etc. all may be different for the case fold of a code point than for the code point itself. , Since LATIN SMALL LIGATURE FF folds to 'ff', it could caselessly match a regular expression which includes two ASCII Hex Digit property matches in a row; a common paradigm. This is likely to be totally unexpected, and may lead to security holes; hence the proposal to specify that this shouldn't happen.

However, in some languages, such as Perl 5 (which I am working to bring more into compliance with TUS), traditionally the Posix classes, such as Upper would match Lower (or Alpha) under caseless matching. For these, I believe it should be mentioned that it's acceptable for the following:
Gc=Lu, Gc=Ll, and Gc=Lt all could match Gc=LC case insensitively; and Uppercase and Lowercase both could match 'Cased ' case-insensitively.

The other issue has to do with multi-character folds, including those generated by using NFD and NFKD semantics under the definition of caseless matching in the standard. The case fold of U+00DF LATIN SMALL LETTER SHARP S for example is the two character string 'ss'. If one wants to match caselessly the complement of a set that includes U+00DF, does that mean it matches any two-character string that isn't “ss”. That doesn't make sense to me. Also regular expressions in Perl and ICU, for example, allow capturing parts of the string that was matched. One can get in the position that a partial character is matched; but then what should be captured? There has been recent discussion of this with examples on the unicode.org forum, at http://www.unicode.org/forum/viewtopic.php?f=20&t=114.
Asmus Freytag proposed the following definition to deal with the problem. I have not studied it enough to comment.

“A pattern is case insensitive if and only if it matches the same strings as a pattern in which all literals and set members x are replaced by the full set or characters or sequences y for which toCaseFold(y) = toCaseFold(x)

“The matching with such a re-written pattern would then be against the original string S (not the case folded S).”

I'm bringing this up to stimulate UTC discussion.

Karl Williamson