L2/08-329

Title: Suggested Comments on Korean Annex in Ballot for ISO/IEC 14651 Amd 1

Source: Ken Whistler

Date: August 15, 2008

I suggest the following text as a ballot comment regarding the Korean Annex C.4 currently under ballot for ISO/IEC 14651 Amd 1.

> The new text for an Annex C.4 "A proposed method of preprocessing Hangul (or Hangeul)" is unnecessarily long and presents many pages regarding BNF and a syntax-directed translation abstraction which are completely unneeded in order to make the main points regarding preprocessing of Hangul for collation.
>
> The U.S. requests that the ballot text for C.4 be removed completely. Instead, the following text could be added to explain the suggested preprocessing method for Hangul.
>
> C.4 Preprocessing Hangul
>
> The CTT does not formally include weights for Hangul syllables. As a result, some tailoring of the table and/or preprocessing of strings is required in order to collate Hangul data according to the algorithm specified in this standard.
>
> One method is simply to give modern Hangul syllable characters primary weights in increasing sequence and to normalize any input strings containing conjoining jamos so that all input strings contain only preformed Hangul syllable characters. This method is quick and efficient for data containing only modern Hangul, because the Hangul syllables are already encoded in the correct collation order for Korean.
>
> For data containing Old Hangul, the situation is more complicated, because no Unicode normalization form provides input strings that can simply be weighted element-by-element to produce appropriate keys for collation. Further preprocessing may be necessary to produce keys that can be used for the desired collation behavior.
>
> The essential issue is that the desired Hangul collation order is a *syllabic* order, but Hangul syllables are built up from a sequence of three jamos: a syllable-initial, a syllable-peak (= vowel), and an optional syllable-final. Each of those jamos, in turn, may consist of one to three subparts, particularly in Old Hangul, which has numerous consonant or vowel clusters represented by single jamo characters.
>
> The basic strategy for handling such Hangul data is to first ensure that it is represented entirely in jamos, so that there is no mixture in the input of conjoining jamos with preformed Hangul syllable characters. This step can be accomplished using Unicode Normalization Form NFD to

decompose any preformed Hangul syllable characters. Then a Korean syllable boundary determination algorithm is used to identify all syllabic boundaries in the data.

Once all the Korean syllabic boundaries are determined in the input data, the initial, peak, and final jamos for each syllable can be weighted so as to provide keys which, when compared, give the desired results for syllabic ordering of the strings.

Ideally, Old Hangul data preprocessed to decompose it and make syllabic-boundary determinations will contain exactly one initial jamo, one peak jamo, and optionally, one final jamo for each syllable. However, certain kinds of input data might result, when decomposed, in sequences containing more than one initial conjoining jamo, and so on. In such cases, Hangul preprocessing may involve an additional mapping step that ensures that any such sequence of jamos is first mapped to the corresponding single initial jamo intended to represent that consonant cluster (of two or three subparts) for Old Hangul. The same considerations apply for any sequences of peak jamos or final jamos in the data.

There are various strategies for weighting the initial, peak, and final jamos of the preprocessed Korean text to produce the desired syllabic order. One approach is to expand each initial, peak, and final jamo into a sequence of three weights, based on the internal composition of each jamo. Simple jamos get one weight; two-part jamos get two weights; three-part jamos get three weights. Any remaining positions in the nine weights for each syllable are filled with EMPTY (U0000) weights.

For example, for the preformed Hangul syllable U+AC01, the data would first be preprocessed into the jamo sequence <1100, 1161, 11A8> and then be weighted as the following key:

U1100 U0000 U0000 U1161 U0000 U0000 U11A8 U0000 U0000

For a sequence containing a multi-part Old Hangul jamo representing a cluster, the keys would have multiple values. For example, for the input sequence <1123, 1161, 11A8>, the U+1123 would be given weights by its three subparts as follows:

U1107 U1109 U1103 U1161 U0000 U0000 U11A8 U0000 U0000

The same kind of weight expansion would be done for any multi-part peak or final jamo as well. When a Korean syllable contains no final jamo, the last three weights are all set to EMPTY (U0000).

With each Korean syllable expanded to nine weights by this preprocessing and weighting scheme, weights for each syllable are lined up correctly on syllabic boundaries, and direct comparison of the resulting keys produces the correct collation results.

Although this weighting strategy works for all Hangul data, including modern Hangul and Old Hangul, it produces much-expanded keys which are not efficient for production applications of collation. Once syllabic boundaries are determined by preprocessing of Korean data, alternative

approaches to weighting the jamos can produce much more compact keys which also produce the same end results for collation.

It is also possible for a collation implementation to do the equivalent of this syllabic preprocessing for Hangul data on the fly while weighting an input string, so it is not technically required to have a formally separate preprocessing step for Hangul which converts all of the input data into a preprocessing form first before weighting it for comparison. This is particularly important for incremental comparison algorithms, which are very performance-sensitive, and which typically cannot afford to preprocess entire strings before starting to do incremental comparison of them.