Appendix C Implementing Korean Jamos

Because many people are not acquainted with the features of Korean text, it is worth discussing some of the characteristics of the two different coding methods: decomposed jamos vs. precomposed syllables.

Modern Hangul syllable blocks can be expressed with either two or three jamos, either in the form *consonant + vowel* or the form *consonant + vowel + consonant*. There are 19 possible leading (initial) consonants, 21 vowels, and 28 trailing (final) consonants. This results in 399 possible two-jamo syllable blocks and 10,773 possible three-jamo syllable blocks, for a total of 11,172 modern Hangul syllable blocks.

C.1 Coverage

The conjoining jamos from U+1100 to U+11FF provide for full coverage of modern Korean syllable blocks, as well as for all old Korean syllable blocks.

The 2,350 precomposed syllable blocks in Unicode 1.0 provide for the most common syllable blocks in modern Hangul. The Supplementary Set A and B provide an additional set of precomposed characters. The Korean national standards organizations intended these sets to be used together with the private use zone to supply the additional characters needed to complete the 11,172 modern Korean syllable blocks.

C.2 Collation

Korean text is normally collated syllable block by syllable block. Because of the arrangement of the conjoining jamos, sequences of them can be collated with a binary comparison. For example, in comparing (a) LVTLV against (b) LVLV, the first syllable block (LVT) should be compared against the second (LV). Supposing the first two characters are identical—since all trailing consonants have binary values greater than all leading consonants—the T would compare as greater than the second L in (b). This produces the correct ordering between the strings. The positions of the fillers in the code charts were also chosen with this in mind.

☞ *As with any coded characters, collation cannot just depend on a binary comparison. Odd sequences such as superfluous fillers will produce an incorrect sort, as will cases where a non-jamos character follows a sequence (such as comparing LVT against LVx, where x is a Unicode character above U+11FF, such as IDEOGRAPHIC SPACE).*

Collating precomposed syllable blocks requires a table, since the syllable blocks are coded in several sets. If mixtures of precomposed syllable blocks and jamos are collated, the easiest approach is to decompose the precomposed syllable blocks before comparing.

C.3 Rendering

There are generally two different methods of rendering modern Korean text, both of which can be easily used with conjoining jamos.

C.3.1 Fully-formed glyphs.

With this system, there are 11,172 separate glyphs, one corresponding to each syllable block. One can algorithmically map from the sequences of modern jamos to a glyph number for a syllable block using the following:

$$\text{glyphNumber} = \text{leadingNumber} \times 588 + \text{vowelNumber} \times 28 + \text{trailingNumber}$$

Since the mapping between the jamos and the 11,172 possible values can be done with simple arithmetic calculations, there is no significant performance impact compared to precomposed syllable blocks. For example, the cost in copying the bits in the glyph to the screen or paper vastly overwhelms a few arithmetic operations.

C.3.2 Glyph components.

With this system, there are separate glyph components which are superimposed to get the proper appearance for the syllable block. Although the components correspond to the jamos, each jamo may need more than one corresponding glyph, since the exact shape of the glyph will depend on the context. For example, some systems use about 500 glyphs to get reasonably good shapes (though not up to the same quality of a full set of 11,172 glyphs). Once again, since there is a simple algorithmic mapping between sequences of modern jamos and the possible syllable block values (see above), there is essentially no difference in overhead in using jamos versus precomposed syllable blocks.

☞ *Rendering is independent of character representation: both methods of rendering can be used with both methods of representing syllable blocks.*

C.4 Keyboard Input

Representing Hangul text via conjoining jamos or via precomposed syllable blocks produces different requirements on the method for keyboard input. In the case of stored syllable blocks, the keyboard input method necessarily involves some system of conjoining jamos plus software for converting the sequence of input jamos into a sequence of stored syllable blocks.

When text is stored directly in the form of conjoining jamos, there are two main keyboarding systems for modern Korean syllable blocks: two-stroke and three-stroke methods. A three-stroke method maps directly onto the conjoining jamos, having leading consonants, vowels, and trailing consonants. A two-stroke method has keys for consonants and keys for vowels, but does not distinguish leading consonants from trailing consonants. Instead, it depends on the structure of modern Korean, in which syllable blocks are either of the form LV or LVT. As the user types, consonants are changed to L or T according to context with the following rules:

C ➡ L

LL ➡ TL

For example, the sequence CVCCVCV becomes LVTLVLV.

Either keyboarding method can be used with precomposed syllable blocks: effectively, the preceding and following syllable blocks are decomposed to jamos, the character is inserted, consonants are changed to leading or trailing, and syllable blocks are recomposed.

☞ *Keyboarding is independent of character representation: both methods of keyboarding can be used with both methods of representing syllable blocks.*

C.5 Application Compatibility

For example, if an application program supporting Unicode works in English, it should be easily localizable into Korean, given operating system support. The main areas that cause problems in localization are input, manipulation and rendering:

a. Input is easier with jamos, since the keyboard input can exactly match the characters in the data stream. There is no requirement for application programs to support input methods, which removes a significant burden.

b. Manipulation includes cases such as concatenation or truncation of text. Conjoining jamos must not be confused with a double-byte character set (DBCS) such as shift-JIS, where there is a mixture of codes with different lengths. A major problem with DBCS is that if bytes are treated in isolation (or misinterpreted as a single-byte character set [SBCS]), then the text will be misparsed. For example, if a random byte is misinterpreted as a single byte and removed from a text stream, the meaning of the surrounding bytes can be completely corrupted.

The individual jamos maintain their independent identity: if a character is removed from a text stream, for example, the surrounding characters maintain their correct interpretation. However, programs may want to preserve syllable block boundaries, which does require some analysis of the text.

c. Rendering is not generally a problem for application compatibility. In modern systems it is handled by the the operating system, and does not require any additional work on the part of the application program.

d. The storage of Korean text using conjoining jamos differs takes about 2.2 times as many bytes as when stored as precomposed Hangul syllables. (The exact figure depends on the particular composition of the text: the factor of 2.2 is based upon samples with half of the Korean syllables having two jamos, the others having three, and 20% of the text consisting of other characters such as space, punctuation, etc.) The number of characters in Korean text expressed in jamos is roughly equivalent to the corresponding English text: systems and application programs that can handle the volume of data necessary for English Unicode will easily handle that of Korea.