L2/09-052

Proposed Draft Unicode Technical Report #47

KOREAN PROCESSING TRANSFORMATIONS

Version	1 (draft 1)
Authors	Mark Davis (markdavis@google.com), Jungshik Shin
Date	2009-01-27
This Version	http://www.unicode.org/reports/tr47/tr47-1.html
Previous Version	n/a
Latest Version	http://www.unicode.org/reports/tr47/
Revision	1

Summary

Korean text can be represented in Unicode in different useful ways, including the customary NFC and NFD forms (see UAX #15, “Unicode Normalization Forms” [UAX15]). This document provides background information on the nature of Korean text, including clarifications of the atomic units of encoding at a very basic level, and defines three additional transformations that can be useful in processing Korean text. It also discusses transformations of compatibility characters.

Status

Contents

[Ed note: make images for the syllables that are not precomosed.]

1 Introduction

Korean text can be represented in Unicode in different useful ways, including the customary NFC and NFD forms (see UAX #15, “Unicode Normalization Forms” [UAX15]). This document provides background information on the nature of Korean text, including clarifications of the atomic units of encoding at a very basic level, and defines three additional transformations that can be useful in processing Korean text. It also discusses transformations of compatibility characters.

A maximal decompounding transformation goes further than NFD in reducing Korean text to completely atomic units. A maximal compounding transformation provides the reverse operation, going further than NFC in forming compounds out of the completely atomic units. These are used by a wide variety of software, especially in open-source, and also correspond to certain keyboarding methods. Finally, the transformation favored by the new Korean Standard KS X 1026-1:2007 is described, wherein a syllable is either a single character or a sequence of exactly two or three characters as described below.

These transformations should not be mistaken for Unicode Normalization Forms; in particular, for NFC, the recommended format for interchange. However, it is straightforward to convert to and from NFC as needed.

2 Character Types

2.1 Atomic Jamo Characters

At its heart, Korean can be represented by a sequence of atomic jamo characters. The atomic jamo characters are listed in the table below, broken out between modern and archaic.

Abbr.	Property Value Name	Comments
L	Leading_Jamo	*leading consonants; also called choseong*
V	Vowel_Jamo	*vowels; also called jungseong*
T	Trailing_Jamo	*trailing consonants; also called jongseong*
LV	LV_Syllable	
LVT	LVT_Syllable	
NA	Not_Applicable	*including some Hangul compatibility characters: see below.*

Korean syllables normally consist of one lead character, followed by one vowel character, followed by zero or one trail character. The jamo characters are further divided into modern versus archaic, and atomic versus compound.

Atomic Jamo Characters

14 Modern Atomic L Characters [ᄀᄂᄃᄅ-ᄇᄉᄋᄌᄎ-ᄒ]	10 Archaic Atomic L Characters [ᄼᄾᅀᅌᅎᅐᅔᅕᅙ]
U+1100 (ᄀ) HANGUL CHOSEONG KIYEOK U+1102 (ᄂ) HANGUL CHOSEONG NIEUN U+1103 (ᄃ) HANGUL CHOSEONG TIKEUT	U+1140 (ᅀ) HANGUL CHOSEONG PANSIOS U+114C (ᅌ) HANGUL CHOSEONG YESIEUNG U+1159 (ᅙ) HANGUL CHOSEONG YEORINHIEUH

U+1105 (ᄅ) HANGUL CHOSEONG RIEUL	
U+1106 (ᄆ) HANGUL CHOSEONG MIEUM	(for Chinese phonetics)
U+1107 (ᄇ) HANGUL CHOSEONG PIEUP	U+113C (ᄼ) HANGUL CHOSEONG CHITUEUMSIOS
U+1109 (ᄉ) HANGUL CHOSEONG SIOS	U+113E (ᄾ) HANGUL CHOSEONG CEONGCHIEUMSIOS
U+110B (ᄋ) HANGUL CHOSEONG IEUNG	U+114E (ᅎ) HANGUL CHOSEONG CHITUEUMCIEUC
U+110C (ᄌ) HANGUL CHOSEONG CIEUC	U+1150 (ᅐ) HANGUL CHOSEONG CEONGCHIEUMCIEUC
U+110E (ᄎ) HANGUL CHOSEONG CHIEUCH	U+1154 (ᅔ) HANGUL CHOSEONG CHITUEUMCHIEUCH
U+110F (ᄏ) HANGUL CHOSEONG KHIEUKH	U+1155 (ᅕ) HANGUL CHOSEONG CEONGCHIEUMCHIEUCH
U+1110 (ᄐ) HANGUL CHOSEONG THIEUTH	
U+1111 (ᄑ) HANGUL CHOSEONG PHIEUPH	
U+1112 (ᄒ) HANGUL CHOSEONG HIEUH	
10 Modern Atomic V Characters [ᅡᅣᅥᅧᅩᅭᅮᅲᅳᅵ]	**1 Archaic Atomic V Characters** [ᆞ]
U+1161 (ᅡ) HANGUL JUNGSEONG A	U+119E (ᆞ) HANGUL JUNGSEONG ARAEA
U+1163 (ᅣ) HANGUL JUNGSEONG YA	
U+1165 (ᅥ) HANGUL JUNGSEONG EO	
U+1167 (ᅧ) HANGUL JUNGSEONG YEO	
U+1169 (ᅩ) HANGUL JUNGSEONG O	
U+116D (ᅭ) HANGUL JUNGSEONG YO	
U+116E (ᅮ) HANGUL JUNGSEONG U	
U+1172 (ᅲ) HANGUL JUNGSEONG YU	
U+1173 (ᅳ) HANGUL JUNGSEONG EU	
U+1175 (ᅵ) HANGUL JUNGSEONG I	
14 Modern Atomic T Characters [ᆨᆫᆮᆯᆷᆸᆺᆼ-ᇂ]	**3 Archaic Atomic T Characters** [ᇫᇰᇹ]
U+11A8 (ᆨ) HANGUL JONGSEONG KIYEOK	U+11EB (ᇫ) HANGUL JONGSEONG PANSIOS
U+11AB (ᆫ) HANGUL JONGSEONG NIEUN	U+11F0 (ᇰ) HANGUL JONGSEONG YESIEUNG
U+11AE (ᆮ) HANGUL JONGSEONG TIKEUT	U+11F9 (ᇹ) HANGUL JONGSEONG YEORINHIEUH
U+11AF (ᆯ) HANGUL JONGSEONG RIEUL	
U+11B7 (ᆷ) HANGUL JONGSEONG MIEUM	
U+11B8 (ᆸ) HANGUL JONGSEONG PIEUP	
U+11BA (ᆺ) HANGUL JONGSEONG SIOS	
U+11BC (ᆼ) HANGUL JONGSEONG IEUNG	
U+11BD (ᆽ) HANGUL JONGSEONG CIEUC	
U+11BE (ᆾ) HANGUL JONGSEONG CHIEUCH	
U+11BF (ᆿ) HANGUL JONGSEONG KHIEUKH	
U+11C0 (ᇀ) HANGUL JONGSEONG THIEUTH	
U+11C1 (ᇁ) HANGUL JONGSEONG PHIEUPH	
U+11C2 (ᇂ) HANGUL JONGSEONG HIEUH	

2.2 Compound Jamo Characters

Many of the jamo characters are compound, combining two or more characters together. These cases only include "like" characters, such as two L characters or two V characters. These compound characters are, for historic reasons, neither compatibility nor canonical variants of their decompositions. The compound jamo characters are listed in the table below, broken out between modern and archaic. Because the archaic characters are so numerous, they are linked to rather than listed explicitly.

Note: even though the column on the right is titled Archaic, some of these characters may be used -- but just not part of Compound Syllables (see below). For more information, see http://i18nl10n.com/korean/uyeo.html.

[Ed note: the Unicode 5.2 characters need to be added to these lists.]

Compound Jamo Characters

5 Modern Compound L Characters [ᄁ ᄄ ᄈ ᄊ ᄍ]	62 Archaic Compound L Characters
U+1101 (ᄁ) HANGUL CHOSEONG SSANGKIYEOK U+1104 (ᄄ) HANGUL CHOSEONG SSANGTIKEUT U+1108 (ᄈ) HANGUL CHOSEONG SSANGPIEUP U+110A (ᄊ) HANGUL CHOSEONG SSANGSIOS U+110D (ᄍ) HANGUL CHOSEONG SSANGCIEUC	[ᄓ–ᄻ ᄽ ᄿ ᅁ–ᅋ ᅍ ᅏ ᅑ–ᅓ ᅖ–ᅘ]
11 Modern Compound V Characters [ᅢ ᅤ ᅦ ᅨ ᅪ–ᅬ ᅯ–ᅱ ᅴ]	**44 Archaic Compound V Characters**
U+1162 (ᅢ) HANGUL JUNGSEONG AE U+1164 (ᅤ) HANGUL JUNGSEONG YAE U+1166 (ᅦ) HANGUL JUNGSEONG E U+1168 (ᅨ) HANGUL JUNGSEONG YE U+116A (ᅪ) HANGUL JUNGSEONG WA U+116B (ᅫ) HANGUL JUNGSEONG WAE U+116C (ᅬ) HANGUL JUNGSEONG OE U+116F (ᅯ) HANGUL JUNGSEONG WEO U+1170 (ᅰ) HANGUL JUNGSEONG WE U+1171 (ᅱ) HANGUL JUNGSEONG WI U+1174 (ᅴ) HANGUL JUNGSEONG YI	[ᅶ–ᆝ ᆟ–ᆢ]
13 Modern Compound T Characters [ᆩ ᆪ ᆬ ᆭ ᆰ–ᆶ ᆹ ᆻ]	**52 Archaic Compound T Characters**
U+11A9 (ᆩ) HANGUL JONGSEONG SSANGKIYEOK U+11AA (ᆪ) HANGUL JONGSEONG KIYEOK-SIOS U+11AC (ᆬ) HANGUL JONGSEONG NIEUN-CIEUC U+11AD (ᆭ) HANGUL JONGSEONG NIEUN-HIEUH U+11B0 (ᆰ) HANGUL JONGSEONG RIEUL-KIYEOK U+11B1 (ᆱ) HANGUL JONGSEONG RIEUL-MIEUM U+11B2 (ᆲ) HANGUL JONGSEONG RIEUL-PIEUP U+11B3 (ᆳ) HANGUL JONGSEONG RIEUL-SIOS U+11B4 (ᆴ) HANGUL JONGSEONG RIEUL-THIEUTH U+11B5 (ᆵ) HANGUL JONGSEONG RIEUL-PHIEUPH U+11B6 (ᆶ) HANGUL JONGSEONG RIEUL-HIEUH U+11B9 (ᆹ) HANGUL JONGSEONG PIEUP-SIOS U+11BB (ᆻ) HANGUL JONGSEONG SSANGSIOS	[ᇃ–ᇩ ᇬ–ᇯ ᇱ–ᇸ]

2.3 Compound Syllables

For modern Korean, all 11,172 combinations of modern L, V, or T characters that form syllables are present in Unicode in compound forms, from AC00 to D7A3.

U+AC00 (가) HANGUL SYLLABLE GA
U+AC01 (각) HANGUL SYLLABLE GAG
U+AC02 (갂) HANGUL SYLLABLE GAGG
U+AC03 (갃) HANGUL SYLLABLE GAGS
...
U+D7A0 (힠) HANGUL SYLLABLE HIK
U+D7A1 (힡) HANGUL SYLLABLE HIT

U+D7A2 (힢) HANGUL SYLLABLE HIP
U+D7A3 (힣) HANGUL SYLLABLE HIH

Every one of these compound syllables are of the form LV or LVT, where some of the L, V, or T characters may be compound jamo. Of these, 399 are LV syllables and 10,773 are LVT syllables.

Example:

Compound Syllable	Component Compound Jamo	Component Atomic Jamo
U+AF51 (꽑) HANGUL SYLLABLE GGWALG	U+1101 (ᄁ) HANGUL CHOSEONG SSANGKIYEOK, U+116A (ᅪ) HANGUL JUNGSEONG WA, U+11B0 (ᆰ) HANGUL JONGSEONG RIEUL-KIYEOK	U+1100 (ᄀ) HANGUL CHOSEONG KIYEOK, U+1100 (ᄀ) HANGUL CHOSEONG KIYEOK, U+1169 (ᅩ) HANGUL JUNGSEONG O, U+1161 (ᅡ) HANGUL JUNGSEONG A, U+11AF (ᆯ) HANGUL JONGSEONG RIEUL, U+11A8 (ᆨ) HANGUL JONGSEONG KIYEOK,

3 Unicode Normalization Forms

Unicode supplies several normalization forms, which are described in [Unicode Normalization Forms]. The most common of these, and the one recommended for use on the web, is called NFC: where C stands for Composition. In terms of Korean characters, in this form a string is first put into a uniform decomposed form (NFD), and then for any modern L, V, and T characters, a V joins with a preceding L to form an LV compound syllable, and a T joins with a preceding LV to form an LVT compound syllable.

Examples where NFC composes:

U+1101 (ᄁ) HANGUL CHOSEONG SSANGKIYEOK,
U+116A (ᅪ) HANGUL JUNGSEONG WA
→
U+AF48 (꽈) HANGUL SYLLABLE GGWA

U+AF48 (꽈) HANGUL SYLLABLE GGWA,
U+11B0 (ᆰ) HANGUL JONGSEONG RIEUL-KIYEOK
→
U+AF51 (꽑) HANGUL SYLLABLE GGWALG

Normalization Form C only composes pairwise, in a single pass. It does not compose series of jamo of the same type.

Example where NFC does not compose:

The following sequence will ***not*** produce U+AF51 (꽑) HANGUL SYLLABLE GGWALG under NFC:

U+1100 (ᄀ) HANGUL CHOSEONG KIYEOK,
U+1100 (ᄀ) HANGUL CHOSEONG KIYEOK,
U+1169 (ᅩ) HANGUL JUNGSEONG O,

U+1161 (ᅡ) HANGUL JUNGSEONG A,
U+11AF (ᆯ) HANGUL JONGSEONG RIEUL,
U+11A8 (ᆨ) HANGUL JONGSEONG KIYEOK

Unicode Normalization Forms are widely used, and have strict stability requirements that prevent any changes. However, other transformations can be specified for particular processing purposes.

Note that the visual appearance of NFC and NFD forms should be identical even though the underlying codes may be different.

4 Maximal Korean Decompounding and Compounding

The Maximal Korean Compounding and Decompounding transformations are parallel to the NFC and NFD transformations, but are based on a finer-grained use of atomic jamo characters. These forms are defined by the data tables in [MKD] and [MKC], using Unicode Locales (CLDR) transform rules. The following are textual descriptions of how they are formed.

- The Maximal Korean Decompounding transformation first puts an input string into NFD format, then applies an additional set of rules to break down jamo characters of the same type.
- The Maximal Korean Compounding transformation first maximally decomposes an input string according to Maximal Korean Decompounding, then applies an additional set of rules to combine like jamo where possible, and finally applies NFC.

Note that once a string is transformed by Maximal Korean Decompounding, it is stable under NFD; that is, any application of NFD to the string will make no further changes. Similarly, once a string is transformed by Maximal Korean Compounding, it is stable under NFC; that is, any application of NFC to the string will make no further changes.

[Ed Note: review the transform rules with regard to IEUNG vs. YESIEUNG.]

Example of Maximal Korean Decompounding:

Action	Text
Input	U+AF51 (꽑) HANGUL SYLLABLE GGWALG
NFD	U+1101 (ᄁ) HANGUL CHOSEONG SSANGKIYEOK, U+116A (ᅪ) HANGUL JUNGSEONG WA, U+11B0 (ᆰ) HANGUL JONGSEONG RIEUL-KIYEOK
Add. Rules	U+1100 (ᄀ) HANGUL CHOSEONG KIYEOK, U+1100 (ᄀ) HANGUL CHOSEONG KIYEOK, U+1169 (ᅩ) HANGUL JUNGSEONG O, U+1161 (ᅡ) HANGUL JUNGSEONG A, U+11AF (ᆯ) HANGUL JONGSEONG RIEUL, U+11A8 (ᆨ) HANGUL JONGSEONG KIYEOK,

Example of Maximal Korean Compounding:

Action	Text
Input	U+1100 (ᄀ) HANGUL CHOSEONG KIYEOK, U+1100 (ᄀ) HANGUL CHOSEONG KIYEOK, U+1169 (ᅩ) HANGUL JUNGSEONG O, U+1161 (ᅡ) HANGUL JUNGSEONG A,

	U+11AF (ᆯ) HANGUL JONGSEONG RIEUL, U+11A8 (ᆨ) HANGUL JONGSEONG KIYEOK
Add. Rules	U+1101 (ᄁ) HANGUL CHOSEONG SSANGKIYEOK, U+116A (ᅪ) HANGUL JUNGSEONG WA, U+11B0 (ᆰ) HANGUL JONGSEONG RIEUL-KIYEOK
NFC	U+AF51 (꽑) HANGUL SYLLABLE GGWALG

Note that the visual appearance of NFC, NFD, Maximal Korean Decompounding and Maximal Korean Compounding should be identical even though the underlying codes may be different

5 Common Korean Compounding

[Ed note: probably want some name that references the Korean standard.]

The Korean Standard KS X 1026-1:2007 defines a format whereby the intent is for compound syllables to occur only where all of the characters in a syllable can be included, and otherwise syllables are represented using two or three jamo characters. The Common Korean Compounding transformation is intended to match that format, and is defined by the data tables in [CKC] using the Unicode Locales (CLDR) transform rules. The following is a textual description of how it is formed.

The Common Korean Compounding transformation first transforms the input string into NFC, and then applies NFD to exactly the following Compound Syllables in that result:

1. any LV Compound Syllable that is preceded by an L or followed by a V or T.
2. any LVT Compound Syllable that is preceded by an L or fosllowed by a T.

Importantly, a string transformed by Common Korean Compounding is ***not*** stable under NFC; if NFC is applied to the string, it may change. However, this will only happen if the Common Korean Compounding string contains archaic syllables. The transformation back from NFC is simple - just applying the rules given aboves.

[Ed Note: describe how to insert FILLER characters to match KS X 1026 recommendations.]

Example:

The following illustrates the difference between these formats.

[Ed Note: In the following table, add lines between each of the rows to make the following clearer, and use images for the syllables.

Also break into three tables: one that shows the normal cases, where NFC is identical to Common Korean Compounding. The second shows cases where they differ, but no fillers are necessary. And the third shows how fillers would be inserted to make 'normal' syllable boundaries, as described in KS X 1026.]

NFC	NFC code points	→	Common Korean Compounding	Common Korean Compounding codepoints
꽑	U+AF51		꽑	U+AF51
ᄀ꽑	U+1100 U+AF51		ᄀ꽑	U+1100 U+1101 U+116A U+11B0
꽑ᅩ	U+AF51 U+1169		꽑ᅩ	U+AF51 U+1169
꽑ᆯ	U+AF51 U+11AF		꽑ᆯ	U+1101 U+116A U+11B0 U+11AF
꽈	U+AF48		꽈	U+AF48
ᄀ꽈	U+1100 U+AF48		ᄀ꽈	U+1100 U+1101 U+116A
꽈ᅩ			꽈ᅩ	
꽐			꽐	

	U+AF48 U+1169 U+AF50			U+1101 U+116A U+1169 U+AF50

Note that the visual appearance of NFC, NFD, Maximal Korean Decompounding, Maximal Korean Compounding, and Common Korean Compounding should be identical even though the underlying codes may be different.

6 Transforming Compatibility Characters

There are a number of Korean characters defined for compatibility in Unicode, split among three blocks. Disregarding unassigned characters, these are:

Hangul_Compatibility_Jamo
  U+3131 (ㄱ) HANGUL LETTER KIYEOK
  ...U+318E (ㆎ) HANGUL LETTER ARAEAE

Enclosed_CJK_Letters_And_Months
  U+3200 (㈀) PARENTHESIZED HANGUL KIYEOK
  ...U+321E (┌) PARENTHESIZED KOREAN CHARACTER O HU
  U+3260 (㉠) CIRCLED HANGUL KIYEOK
  ...U+327E (┌) CIRCLED HANGUL IEUNG U

Halfwidth_And_Fullwidth_Forms
  U+FFA0 ( ) HALFWIDTH HANGUL FILLER
  ...U+FFDC (▪) HALFWIDTH HANGUL LETTER I

Most of these compatibility Korean characters are jamo characters, except for the following four (which are also the only ones to contain the word "KOREAN" instead of "HANGUL"):

  U+321D (┌) PARENTHESIZED KOREAN CHARACTER OJEON
  U+321E (┌) PARENTHESIZED KOREAN CHARACTER O HU
  U+327C (┌) CIRCLED KOREAN CHARACTER CHAMKO
  U+327D (┌) CIRCLED KOREAN CHARACTER JUEUI

The compatibility jamo characters corresponding to V characters are unproblematic; they are mapped to V characters by the Unicode compatibility normalizations (NFKC, NFKD). However, most other compatibility characters do not distinguish between the L and T forms for the consonants. They are mapped to L forms by the Unicode compatibility normalizations except where they cannot be represented by a modern L characters. In that latter case they are mapped to T characters. Thus the Unicode compatibility normalizations do not normally result in well-formed syllables.

There are two better approaches to take. One is to use the Unicode compatibility normalizations, but insert FILLER characters so as to represent separate elements. This results in a format that is stable under all Unicode normalizations, and represented the compatibility characters as independent elements. However, for many purposes it is useful to transform the Hangul Letters and Halfwidth Hangul Letters into complete syllables, with L, V, ***and optionally T*** characters. This is common in keyboarding, for example.

The Maximal Korean Compatibility Compounding transformation is defined by the data tables in [MKKC], using the Unicode Locales transformation rules. The following is a textual descriptions of how it is formed.

> The Maximal Korean Compatibility Compounding transformation first converts any Compatibility or Halfwidth Hangul Letter L into a T if it is before an L form, then applies NFKD, then finally applies Maximal Korean Compounding.

Once a string has be transformed by Maximal Korean Compatibility Compounding, it is stable under NFKC; that is, any application of NFKC to the string will make no further changes.

[Ed Note: The transformation rules are just a stub at this point, and need to be filled out.]

Note that the visual appearance of Maximal Korean Compatibility Compounding will be different than its source if the source contained Hangul Compatibility Jamo or Halfwidth Jamo.

7 Optimization

The specifications provided above are *logical* specifications; they can and should be optimized in production software. In particular, the Maximal Korean Compounding form can be generated in a single pass: without first doing a decompounding. Such an optimization is even easier to perform for Korean characters than for NFC, because compounding of Korean characters only takes place between adjacent characters.

This is done by generating a (logical) table for all pairs of character that interact. That table can be used to make a single pass through the text, producing Maximal Korean Compounding as it goes. The pairs that interact fall into two categories:

7.1 <X,Y> → <Z>

Merge the pair into a single character. Example:

- U+1169 (ᅩ) HANGUL JUNGSEONG O, U+1161 (ᅡ) HANGUL JUNGSEONG A
  → U+116A (ᅪ) HANGUL JUNGSEONG WA

7.2 <X,Y> → <Z,W...>

Replace the pair by a different string. This only happens in degenerate cases. Example:

- U+11B1 (ᆱ) HANGUL JONGSEONG RIEUL-MIEUM, U+1101 (ᄁ) HANGUL CHOSEONG SSANGKIYEOK
  → U+11D1 (ᇑ) HANGUL JONGSEONG RIEUL-MIEUM-KIYEOK, U+1100 (ᄀ) HANGUL CHOSEONG KIYEOK

Example of Maximal Korean Compounding using pairwise combination:

Action	Text
Input	U+1100 (ᄀ) HANGUL CHOSEONG KIYEOK, U+1100 (ᄀ) HANGUL CHOSEONG KIYEOK, U+1169 (ᅩ) HANGUL JUNGSEONG O, U+1161 (ᅡ) HANGUL JUNGSEONG A, U+11AF (ᆯ) HANGUL JONGSEONG RIEUL, U+11A8 (ᆨ) HANGUL JONGSEONG KIYEOK, U+1103 (ᄃ) HANGUL CHOSEONG TIKEUT, U+1103 (ᄃ) HANGUL CHOSEONG TIKEUT, ...
Combine pair	→ U+1101 (ᄁ) HANGUL CHOSEONG SSANGKIYEOK, U+1169 (ᅩ) HANGUL JUNGSEONG O, U+1161 (ᅡ) HANGUL JUNGSEONG A, U+11AF (ᆯ) HANGUL JONGSEONG RIEUL, U+11A8 (ᆨ) HANGUL JONGSEONG KIYEOK, U+1103 (ᄃ) HANGUL CHOSEONG TIKEUT,

	U+1103 (ᄃ) HANGUL CHOSEONG TIKEUT, ...
Combine pair	→ U+AF2C (꼬) HANGUL SYLLABLE GGO, U+1161 (ᅡ) HANGUL JUNGSEONG A, U+11AF (ᆯ) HANGUL JONGSEONG RIEUL, U+11A8 (ᆨ) HANGUL JONGSEONG KIYEOK, U+1103 (ᄃ) HANGUL CHOSEONG TIKEUT, U+1103 (ᄃ) HANGUL CHOSEONG TIKEUT, ...
Combine pair	→ U+AF48 (꽈) HANGUL SYLLABLE GGWA, U+11AF (ᆯ) HANGUL JONGSEONG RIEUL, U+11A8 (ᆨ) HANGUL JONGSEONG KIYEOK, U+1103 (ᄃ) HANGUL CHOSEONG TIKEUT, U+1103 (ᄃ) HANGUL CHOSEONG TIKEUT, ...
Combine pair	→ U+AF50 (꽐) HANGUL SYLLABLE GGWAL, U+11A8 (ᆨ) HANGUL JONGSEONG KIYEOK, U+1103 (ᄃ) HANGUL CHOSEONG TIKEUT, U+1103 (ᄃ) HANGUL CHOSEONG TIKEUT, ...
No combination, move to next	U+AF51 (꽑) HANGUL SYLLABLE GGWALG, U+1103 (ᄃ) HANGUL CHOSEONG TIKEUT, U+1103 (ᄃ) HANGUL CHOSEONG TIKEUT, ...
Combine pair	U+AF51 (꽑) HANGUL SYLLABLE GGWALG, → U+1104 (ᄄ) HANGUL CHOSEONG SSANGTIKEUT ...

Such a table, if it were expressed literally, would be huge: here are the counts for the replacement strings:

Length	Count of instances
1	19,892
2	88,650
3	222,510
4	3,192

So in practice such a pairwise table would not be stored literally, but instead, because of the regularities in Korean compounding, would be primarily expressed via small tables and coded algorithms.

8 Compatibility and Security

[Ed note: Describe the relations between the above forms, and give scenarios as to how they would be transformed back and forth. Note the security issues.]

9 Acknowledgements

[Ed note: Add thanks ...]

References

[CKC]	Common Korean Compounding Data *For the latest version, see:* http://www.unicode.org/reports/tr47/ckc-1.txt
[MKC]	Maximal Korean Compounding Data *For the latest version, see:* http://www.unicode.org/reports/tr47/mkc-1.txt
[MKD]	Maximal Korean Decompounding Data *For the latest version, see:* http://www.unicode.org/reports/tr47/mkd-1.txt
[MKKC]	Maximal Korean Compatibility Compounding Data *For the latest version, see:* http://www.unicode.org/reports/tr47/mkkc-1.txt

Modifications

This section indicates the changes introduced by each revision.

Revision 1

- First version