L2/12-223

Anomalous Level 4 Weights in Tables for UCA

Author: Richard Wordingham

Date: 5 July 2012

Introduction

This is submitted as an error report, for treatment as though submitted through <http://www.unicode.org/reporting.html>.

The Unicode Collation Algorithm (UCA) UTS#10 Version 6.1.0 Section 3.6.1 states, à propos the Default Unicode Collation Element Table (DUCET), 'the file contains a fourth level (as in `[.0712.0020.0008.0044]`), which is computable'. Section 7.3 gives the rules for calculating the fourth level weight for characters.

When checking the corresponding values in DUCET (file `allkeys.txt`) and the CLDR root locale (as defined by `allkeys_CLDR.txt`), many discrepancies were found. The same discrepancies are present in drafts for Unicode 6.2.0 (e.g. `allkeys-6.2.0d2.txt`).

While the 4th level is primarily presented as a short cut for semistable sorting, it does have the merit of distinguishing distinct characters with primary or secondary weights that are not distinguished at the third level, while (in general) ignoring completely ignorable characters such as `U+00AD SOFT HYPHEN`. Characters distinguished include not only compatibly variants of the same letter, such as Mathematical symbols with very different connotations like `U+210C BLACK-LETTER CAPITAL H` and `U+210D DOUBLE-STRUCK CAPITAL H`, but also IPA diacritics with opposite meanings, such as `U+031D COMBINING UP TACK BELOW` and `U+031E COMBINING DOWN TACK BELOW`.

The first five anomalies indisputably affect quaternary collation and search. It is arguable that the fifth anomaly is a deficiency in the UCA rather than in DUCET or the CLDR root locale. The first four anomalies are definite breaches of the UCA.

The status of the next four anomalies depends on the meaning of the entries in the published tables for strings that are not in NFD. It may be that they should have no meaning for compliant implementation of the UCA, and are mere disinformation, e.g. giving spurious agreement with some versions of the ISO 14651 baseline collation. Alternatively, they may be meant as a convenience for implementations that attempt to avoid unnecessary normalisation, in which case they are errors. In this case, all nine anomalies affect quaternary collation and search and the significance of the fifth anomaly is greatly increased.

1. False Canonical Decomposition

This anomaly applies to DUCET only.

The following points receive 4th weights as though their canonical decompositions were:

```
00D8;LATIN CAPITAL LETTER O WITH STROKE;004F 0338
00F8;LATIN SMALL LETTER O WITH STROKE;006F 0338
0110;LATIN CAPITAL LETTER D WITH STROKE;0044 0335
0111;LATIN SMALL LETTER D WITH STROKE;0064 0335
0126;LATIN CAPITAL LETTER H WITH STROKE;0048 0335
0127;LATIN SMALL LETTER H WITH STROKE;0068 0335
0141;LATIN CAPITAL LETTER L WITH STROKE;004C 0335
0142;LATIN SMALL LETTER L WITH STROKE;006C 0335
3032;VERTICAL KANA REPEAT WITH VOICED SOUND MARK;3031 3099
3034;VERTICAL KANA REPEAT WITH VOICED SOUND MARK UPPER HALF;3033 3099
```

For example, U+00D8 has weight `[.1756.0020.0008.004F][.0000.0054.0002.0338]`, whereas by Section 7.3 Item 2, it should have weight `[.1756.0020.0008.00D8][.0000.0054.0002.00D8]`. Note further that the weight of the string `<U+004F, U+0338>` is `[.1756.0020.0008.004F][.0000.0054.0002.0338]`, so this does affect the relative sorting of

strings.

2. Non-Zero 4th Weights for Format Controls

This applies to DUCET only.

The following format controls (gc=Cf) with zero level 3 weights are give non-zero 4th weights, contrary to Section 7.3 Item 1:

```
0600 ARABIC NUMBER SIGN
0601 ARABIC SIGN SANAH
0602 ARABIC FOOTNOTE MARKER
0603 ARABIC SIGN SAFHA
0604 ARABIC SIGN SAMVAT
06DD ARABIC END OF AYAH
2061 FUNCTION APPLICATION
2062 INVISIBLE TIMES
2063 INVISIBLE SEPARATOR
2064 INVISIBLE PLUS
110BD KAITHI NUMBER SIGN
```

3. CJK Weights

This applies to DUCET only.

If the primary weight is the implicit weight for a character in a CJK block, then the 4th weight is the codepoint of the corresponding CJK element, rather than the codepoint of the actual character. For example:

```
2F17  ; [.FB40.0020.0004.5341][.D341.0000.0000.5341] # KANGXI RADICAL TEN
3038  ; [.FB40.0020.0004.5341][.D341.0000.0000.5341] # HANGZHOU NUMERAL TEN
3229  ; [*02FB.0020.0004.3229][.FB40.0020.0004.5341][.D341.0000.0000.5341]
        [*02FC.0020.001F.3229] # PARENTHESIZED IDEOGRAPH TEN
3289  ; [.FB40.0020.0006.5341][.D341.0000.0000.5341] # CIRCLED IDEOGRAPH TEN
```

By contrast, the CLDR root follows the rules given in the UCA, and the fourth weight distinguishes U+2F17 and U+3038.

4. No Quaternary Elements

This applies to the CLDR root only.

Whenever the first three weights are zero, so is the fourth, contrary to Section 7.3 Item 1.

5. Weights for Contractions

This applies to the CLDR root only.

There is no formal basis for the apparent DUCET rule that when NFKC decomposition of a character yields the same weights to three levels as those assigned to the character itself, the contraction should then have the same level 4 weights. However, it does seem odd that the CLDR root locale should have discarded this rule.

The 4th weights for contractions appear to be derived by taking the NFD decomposition of the string, and applying the weights in turn to the otherwise non zero collation elements of the contraction. This has some unwanted effects even when collations runs purely in NFD. For example, deprecated `U+0F77 TIBETAN VOWEL SIGN VOCALIC RR` and its preferred, compatibility decomposition <U+0FB2, U+0F81> have the same weight, `[.2578.0020.0002.0F77]`, in DUCET. The

same, mutandis mutatis, applies to U+0F79 TIBETAN VOWEL SIGN VOCALIC LL. In the CLDR root, the weights are:

```
0F77  ;          [.2578.0020.0002.0F77]
0FB2 0F71 0F80 ; [.2578.0020.0002.0FB2]
0FB2 0F81 ;      [.2578.0020.0002.0FB2]

0F79  ;          [.257A.0020.0002.0F79]
0FB3 0F71 0F80 ; [.257A.0020.0002.0FB3]
0FB3 0F81 ;      [.257A.0020.0002.0FB3]
```

Similarly, though less seriously, we now get different weights for malformed <U+0E4D THAI CHARACTER NIKHAHIT, U+0E32 THAI CHARACTER SARA AA > and U+0E33 THAI CHARACTER SARA AM, and also for the Lao equivalents

```
0E33      ; [.249F.0020.0002.0E33]
0E4D 0E32 ; [.249F.0020.0002.0E4D]

0EB3      ; [.24CE.0020.0002.0EB3]
0ECD 0EB2 ; [.24CE.0020.0002.0ECD]
```

Similar unwanted distinctions now occur between U+013F LATIN CAPITAL LETTER L WITH MIDDLE DOT on one hand and <U+004C LATIN CAPITAL LETTER L, U+00B7 MIDDLE DOT> and <U+004C, U+0387 GREEK ANO TELEIA> on the other, and similarly for their lower casings.

```
004C 00B7 ; [.16F6.0020.0008.004C][.0000.0139.0002.00B7]
004C 0387 ; [.16F6.0020.0008.004C][.0000.0139.0002.00B7]
013F      ; [.16F6.0020.0008.013F][.0000.0139.0002.013F]


0140      ; [.16F6.0020.0002.0140][.0000.0139.0002.0140]
006C 00B7 ; [.16F6.0020.0002.006C][.0000.0139.0002.00B7]
006C 0387 ; [.16F6.0020.0002.006C][.0000.0139.0002.00B7]
```

What had been deliberately made indistinguishable are now distinguishable!

This method of assigning weights leads to many new failures to preserve collation element mappings under canonical equivalence. Apart from music symbols, this is only a problem if an application uses NFC weights for NFC strings and NFD weights for NFD strings.

Many music symbols are tertiary ignorables, and under the CLDR root they become quaternary ignorables. Through severable possible mechanisms, U+1D15F MUSICAL SYMBOL QUARTER NOTE and U+1D160 MUSICAL SYMBOL EIGHTH NOTE then get different weights if looked up in NFC, and the same weights if looked up via their NFD decompositions.

Related to this anomaly is a difference in the 4th weights for the Thai etc. reversing contractions for consonant-vowel combinations. In DUCET the 4th weights are the codepoint of the consonant and the vowel; in the CLDR root the 4th weights are of the vowel and consonant. However, this has no effect on collation or UCA-compliant searching. (It would defeat a non-compliant quaternary search that converted a string to collation elements and then searched for matching collation elements, but such a search would not be compliant with UCA Version 6.1.0.)

6. Canonical Singleton Decompositions

This anomaly applies to both DUCET and the CLDR root locale.

Codepoints and their singleton decompositions, if their collation mapping is to a single collation element, are given different weights. This may be in accordance with the literal wording of Section 7.3 Item , but violates the principle of canonical equivalence. The codepoints affected are:

0340 COMBINING GRAVE TONE MARK v. 0300 COMBINING GRAVE ACCENT
0341 COMBINING ACUTE TONE MARK v. 0301 COMBINING ACUTE ACCENT

0343 COMBINING GREEK KORONIS v. 0313 COMBINING COMMA ABOVE
0374 GREEK NUMERAL SIGN v. 02B9 MODIFIER LETTER PRIME
037E GREEK QUESTION MARK v. 003B SEMICOLON
0387 GREEK ANO TELEIA v. 00B7 MIDDLE DOT
1FBE GREEK PROSGEGRAMMENI v. 03B9 GREEK SMALL LETTER IOTA
1FEF GREEK VARIA v. 0060 GRAVE ACCENT
1FFD GREEK OXIA v. 00B4 ACUTE ACCENT
2000 EN QUAD v. 2002 EN SPACE
2001 EM QUAD v. 2003 EM SPACE
2126 OHM SIGN v. 03A9 GREEK CAPITAL LETTER OMEGA
212A KELVIN SIGN v. 004B LATIN CAPITAL LETTER K
2329 LEFT-POINTING ANGLE BRACKET v. 3008 LEFT ANGLE BRACKET
232A RIGHT-POINTING ANGLE BRACKET v. 3009 RIGHT ANGLE BRACKET

Curiously, implicit weights count as two collation elements for this anomaly – CJK compatibility characters receive the same weights as the characters they decompose to.

I suggest that Section 7.3 Item 3 be changed from

> If a character is weighted as an expansion based on a canonical decomposition, then assign the code point of each character in the decomposition as the fourth-level weight for the corresponding element of the expansion.

to

> If a character is weighted based on a canonical decomposition, then the fourth weights shall be those assigned by looking up the collation elements for the decomposition.

We still need a rule for contractions that are not related to single characters.

7. Other Weights Conflicting with Canonical Equivalence

This anomaly affects only DUCET.

The following are given 4th weights equal to their codepoint although by Section 7.3 Item 3 their weights have to be derived via their canonical expansions:

01E2;LATIN CAPITAL LETTER AE WITH MACRON;00C6 0304
01E3;LATIN SMALL LETTER AE WITH MACRON;00E6 0304
01FC;LATIN CAPITAL LETTER AE WITH ACUTE;00C6 0301
01FD;LATIN SMALL LETTER AE WITH ACUTE;00E6 0301
1E9B;LATIN SMALL LETTER LONG S WITH DOT ABOVE;017F 0307
03D3;GREEK UPSILON WITH ACUTE AND HOOK SYMBOL;03D2 0301
03D4;GREEK UPSILON WITH DIAERESIS AND HOOK SYMBOL;03D2 0308
FB1F;HEBREW LIGATURE YIDDISH YOD YOD PATAH;05F2 05B7
FB3A;HEBREW LETTER FINAL KAF WITH DAGESH;05DA 05BC
FB43;HEBREW LETTER FINAL PE WITH DAGESH;05E3 05BC

For example, the weights of U+01E2 are [.15D4.0020.000A.01E2][.0000.0139.0004.01E2] [.1631.0020.001F.01E2][.0000.005B.0002.01E2] whereas the weights of <U+00C6, U+0304>, to which it is canonically equivalent, shall be calculated as [.15D4.0020.000A.00C6][.0000.0139.0004.00C6] [.1631.0020.001F.00C6][.0000.005B.0002.0304]. A simpler example is that U+03D3 has weight [.1931.0020.000A.03D3][.0000.0032.0002.03D3] whereas <U+03D2, U+0301> shall have weights [.1931.0020.000A.03D2][.1931.0020.000A.03D2].

If the NFC weight is used, there will be the result that <U+00C6, U+0954> sorts before <U+01FC>, whereas the opposite result is obtained if the NFD weights are used!

8. Wrong Nukta

This applies to DUCET only.

Within weights otherwise derived from canonical decompositions, all nuktas are given fourth weights as though they were `U+093C DEVANAGARI SIGN NUKTA`. This error affects:

```
09DC BENGALI LETTER RRA
09DD BENGALI LETTER RHA
09DF BENGALI LETTER YYA
0A33 GURMUKHI LETTER LLA
0A36 GURMUKHI LETTER SHA
0A59 GURMUKHI LETTER KHHA
0A5A GURMUKHI LETTER GHHA
0A5B GURMUKHI LETTER ZA
0A5E GURMUKHI LETTER FA
0B5C ORIYA LETTER RRA
0B5D ORIYA LETTER RHA
1109A KAITHI LETTER DDDHA
1109C KAITHI LETTER RHA
110AB KAITHI LETTER VA
```

There are no contractions for these elements, so different orderings could result depending on whether the text is in NFC or NFD.

9. Phasing Error in Weight Assignments

This affects the CLDR root only.

Where a character gets its weights as a result of canonical decomposition, if *m* elements come from the first character and *n* from the second, the first 4th weight is applied to the first *n* elements and the second weight is applied to the last *m* elements.

For example, `<00C6, 0304>` shall get the weights `[.15D4.0020.000A.00C6][.0000.0139.0004.00C6][.1631.0020.001F.00C6][.0000.005B.0002.0304]` but U+01E2 is assigned the weight `[.15D4.0020.000A.00C6][.0000.0139.0004.0304][.1631.0020.001F.0304][.0000.005B.0002.0304]`.

The characters affected, listed along with their *canonical* decompositions:

```
01E2;LATIN CAPITAL LETTER AE WITH MACRON;00C6 0304
01E3;LATIN SMALL LETTER AE WITH MACRON;00E6 0304
01FC;LATIN CAPITAL LETTER AE WITH ACUTE;00C6 0301
01FD;LATIN SMALL LETTER AE WITH ACUTE;00E6 0301
01FE;LATIN CAPITAL LETTER O WITH STROKE AND ACUTE;00D8 0301
01FF;LATIN SMALL LETTER O WITH STROKE AND ACUTE;00F8 0301
1E9B;LATIN SMALL LETTER LONG S WITH DOT ABOVE;017F 0307
FB1F;HEBREW LIGATURE YIDDISH YOD YOD PATAH;05F2 05B7
```

If the NFC weight is used, there will be the result that <U+00C6, U+0954> sorts before <U+01FC>, whereas the opposite result is obtained if the NFD weights are used!