Sweden's comments on FCD2 of 14651 (International String Ordering – Method for Comparing Character Strings and Description of the Common Template Tailorable Ordering)

Sweden votes NO with the following comments:

(Where the heading says "major" all points, except where otherwise noted initially, are "major".)

1 Definitions (major comment)

The definitions (section 4) are not always to the point, and sometimes unclear. **Please change the definitions to something very close to the following** (and alter subsequent text accordingly):

abstract glyph	a recognizable abstract graphic symbol which is independent of any specific design.
character string	a sequence of (coded) characters (((considered as a single object?)))
collation	ordering of elements based on ordering of character strings.
collation delta	list of differences for a specific collation table relative to one of its ancestor template collation tables. Each collation table can have only one immediate ancestor.
collation element	sequence of *n* weight strings, where *n* is the number of levels in the collation table. The weights may be given as symbolic weights.
collation item	non-empty sequence of characters that has an entry in the collation table.
(collation) key	a real value (strictly) between 0 and 1, formed by concatenating the collation subkeys for a given string after an initial '0.', and regarding the result as a fractional numeral (in the radix of the digits used). The reference method puts a level separator weight between each pair of the concatenated subkeys. The collation keys 0 and 1 can be used as special collation keys, respectively strictly less than and strictly greater than any collation key formed from any character string by the reference method. (Note that hardware supported floating point datatypes are not suited for representing these values, since these datatypes rarely will have sufficient precision, unless the strings compared are limited to two or three, maybe four, characters.)
(collation) level	whenever used without qualification in this International Standard, *level* stands for the number of the 'pass' done over a string to compute its reference collation key.
collation subkey	a sequence of weights computed for a character string for a particular level.
(collation) preparation	a process in which character strings are mapped to (other) character strings logically before using the key calculation specified in the reference method of this International Standard.
(collation) weight	length *b* digit sequence. For the reference method, the value of *b* must be fixed for each level (but may be different for different levels) and the radix of the digits must be the same for all levels.
graphic character	a character that has a visual representation normally handwritten, printed, or displayed.

(level) separator weight

a (non-zero) collation weight smaller (when regarded as an integer) than all weights used in collation elements at the preceding level, and with the same number of digits as used for the weights in the preceding level. A level separator weight is inserted by the reference method between each collation subkey.

ordering a process in which a set of strings are assigned a lexicographic order

symbolic weight name bound to a weight. Each symbolic weight is defined for a particular level.

symbolic collation item

a name bound to a non-empty character string. The name may be used in specifying collation items.

2 Table well-formedness (major)

1. Currently, each collation element that has a non-empty string of weights at level *i* also has a non-empty string of weights at level *i*+1 (The empty string of (symbolic) weights is called IGNORE in the balloted table). **This rule seems to be of no purpose.** Instead the well-formedness rules expressed in N639, and as comments in N641, should apply. These allow, or rather mandate, that level 2 items, combining accents mostly, have empty weight strings also at level 3 and 4.
2. In N641 **all modifier weights at levels 2 and 3 are heavier than any base weight at that level. This is in order to avoid edge case anomalies that will result if this is not followed**. In order to implement a check on this criterion, it facilitates if base and **modifier weights are declared as such** for each level. The current POSIX based syntax does not allow for that, but N639 does.

3 Key construction description in main text (major)

1. The key construction in the main text loosely refers to computing the 'numeric key', **but does not explain in sufficient detail how that numeric key is formed**. Some text is given in the above definitions, but this may need to be moved and/or expanded.
2. **Please delete section 6.2.2.2.** The main text (in section 6.2.2.2) suggests that level 4 (or in general the last level) should be treated differently from the other levels. This is both unnecessary and confusing, and the net effect (or, preferably, better!) should be produced by other means. Make a normative change of level 4 in the template table (see below, point 8, and level 4 as given in document N641) and the addition of an informative annex on key reduction (see document N642).
3. **N642 is a suggested annex** giving detail for two alternative methods to reduce the length of a subkey, without changing the ordering of strings as given by the collation keys as computed by the reference method. They are similar in spirit and internal key structure to what current section 6.2.2.2 would produce, but does correct a number of details. We strongly suggest instating into this standard this informative annex as part of the replacement of flawed section 6.2.2.2.

4 Table format (major)

Though there is no formal link from 14641 to 14652, there are still strong formal and informal links from (CD of) 14652 to 14651. Though we hope that 14652 will be very substantially revised before turning into a standard, the existing link will taint the interpretation of the current table in 14651. Since these interpretations are greatly dissimilar, it would be highly preferable to use a table format in 14651 that **cannot be directly referenced** by (current) 14652, nor by the POSIX standards.

In order not to invent a completely new syntax for this, we suggest **basing the new table format on XML** (or SGML). At the same time one can address some of the **shortcomings** of the current table format (like that symbolic weights are not associated with a particular level, that well-formedness criteria are not enforceable at the syntactic level, that the 'auto-weighting' of symbolic weights is not explained, nor eliminable).

Document N639 gives a draft XML DTD for such a new table format (this has been updated, and the updated version can be supplied by the Swedish delegate). Document N641 gives a draft XML data file for the template table (some modifications has been done to this to follow the updated DTD).

Changing the table format should not incur significant additional delay in passing 14651 as a standard, considering that major changes need be done to level 2, 3, and 4 of the data in the table, whatever the format.

5 Level 1 in table (major)

1. The US delegate has done some changes to level 1. Some additional changes for Indic scripts may be needed. Though the Swedish representative has no expertise in Indic scripts, Jeoren Hellingman has been asked to supply comments on this point, and has done so. These comments have been forwarded to the US delegate for change in the data table. (See also N641, where these changes have been done by moving the entries to the suggested order; note however, that the symbolic weights have not been corrected accordingly).
2. Some generation errors afflict the balloted table. They occur when a punctuation character is at the beginning of a decomposition, but there is a letter (or digit) thereafter (degrees-C, degrees-F, parenthesised numbers and letters). (This has been fixed in a later version of the table; it is *partially* fixed also in N641.)
3. (minor) While handling of numeric order collation of digit sequences is to be taken care of in the preparation stage in general, it seems unnecessary to do so for certain pre-isolated numbers, e.g. parenthesised numbers, and month numbers, where the parentheses (etc) and digits are made into a *single* character. Here it is known that there will be at most two digits, so we can easily have a "virtual" 0 as the initial digit for the one-digit isolated numbers (see N641, where this has been carried out).
4. Again for numbers, annex C gives informative details on how to handle numerical order collation of numerals in general, it also needs to have PLUS and MINUS as first level significant characters. We see no reason not to have it that way in the template, in order to avoid additional special tailorings to take care of this (see N641).
5. (unclear) It is unclear to this reviewer if the Greek lowercase letters with ypogrammeni (and the combining ypogrammeni) should include a level 1 weight corresponding to *iota*. But since the uppercasing of combining ypogrammeni is an uppercase *iota*, it seems plausible that this combining character should have a level 1 weight the same as that for *iota* (with corresponding changes for the precomposed forms with ypogrammeni), and a level 2 weight of VRNT1.

6 Level 2 in table (major)

1. There is a systematic error in the balloted version of the template table at level 2 (missing BLANK; or as it is renamed BASE). This has been corrected in later versions of the table, including in N641).
2. (unclear) TONOS and AIGUT are mixed up at level 2 in the balloted table (tentatively fixed in N641).

3. (minor) The symbolic weights at level 2 for the accents are often in French, while the name of that accent in the 10646 character names are in English. It may better to take the accent name used in the character name as the level 2 symbolic weight of an accent.
4. All base weights at level 2 MUST be smaller than any level 2 modifier weight (as in N641).
5. (minor) More base weights at level 2: for tailorings it would be helpful to have a number of predeclared lighter and heavier variant weights at level 2 (see N641). This would relieve tailoring from declaring them.
6. Some ligatures have orthographic significance, like the oe ligature (tentative list below). Level 2-4 should consider these as single characters, even though they are collated as two letters at level 1. This makes the table more logical, since these letters are considered to be single letters, rather than two letters. (See COMB2 and COMB2L in N641.)

```
<ci1 mtc="0133" v1="L79D L7B1" v2="COMB2" v3="MIN"   cmt="LATIN SMALL LIGATURE IJ"/>
<ci1 mtc="0132" v1="L79D L7B1" v2="COMB2" v3="CAP"   cmt="LATIN CAPITAL LIGATURE IJ"/>
<ci1 mtc="0153" v1="L815 L72F" v2="COMB2" v3="MIN"   cmt="LATIN SMALL LIGATURE OE; COMB2L?"/>
<ci1 mtc="0152" v1="L815 L72F" v2="COMB2" v3="CAP"   cmt="LATIN CAPITAL LIGATURE OE; COMB2L?"/>
<ci1 mtc="00DF" v1="L86D L86D" v2="COMB2" v3="MIN"   cmt="LATIN SMALL LETTER SHARP S"/>
<ci1 mtc="FB4F" v1="LB21 LB2C" v2="COMB2" v3="MIN"   cmt="HEBREW LIGATURE ALEF LAMED"/>
<ci1 mtc="05F0" v1="LB26 LB26" v2="COMB2" v3="MIN"   cmt="HEBREW LIGATURE YIDDISH DOUBLE VAV"/>
<ci1 mtc="05F1" v1="LB26 LB2A" v2="COMB2" v3="MIN"   cmt="HEBREW LIGATURE YIDDISH VAV YOD"/>
<ci1 mtc="05F2" v1="LB2A LB2A" v2="COMB2" v3="MIN"   cmt="HEBREW LIGATURE YIDDISH DOUBLE YOD"/>
<ci1 mtc="FB1F" v1="LB2A LB2A" v2="COMB2 PATAH" v3="MIN"   cmt="HEBREW LIGATURE YIDDISH YOD YOD PATAH"/>
<ci1 mtc="0950" v1="LBD0 LBBD" v2="COMB2" v3="MIN"   cmt="DEVANAGARI OM"/>
<ci1 mtc="0AD0" v1="LC90 LC81" v2="COMB2" v3="MIN"   cmt="GUJARATI OM"/>
```

7 Level 3 in table (major)

1. In the balloted version of the table, Arabic ligature characters wrongly get the same weights at levels 1-3 as sequences of shaped Arabic letters, *of the wrong shape*. This is fixed in N641.
2. In the balloted version of the table, single characters with two digits in a circle wrongly get the same weights at levels 1-3 as two circled digits with a circle each. This is fixed in N641.
3. For simplicity, squared ligatures should be treated in the same way as other ligatures. (See N641.)
4. In order to make tailoring to get capital letters before minuscule letters easier, it is preferable to have only two weights indicating capital and miniscule status at level 3. (See N641.)
5. (minor) in order to ease tailoring for such things as Danish "Aa" and Spanish "Ch", it would be helpful to predeclare a CAP-MIN weight (see N641).
6. (minor) The NOBREAK and VERTICAL weights are not used, since they apply only to punctuation, which only have a level 4 weight anyway. These two weights may be deleted.
7. The balloted version of the table has only one weight for FONT, whereas there are sometimes **multiple font variations of the same character**. To remedy that N641 uses several different 'FONT' weights (ITALIC, SCRIPT, BLACK_LETTER, BOLD, DOUBLE_STRUCK, SANS_SERIF). This should be done also for the final version of the template table.
8. In order not to get a large number of possible combinations weights for **level 3**, N641 uses an approach similar to that used on level 2: **base weight and a sequence of modifier weights**.
9. In the balloted version of the table, some of the **square ligatures get the wrong level 1-3 weights, where Katakana or punctuation occurs** in the expansion of the square ligature. This is fixed in N641, and should be likewise fixed in the final version of the template table.

8 Level 4 in table (major)

1. C0 and C1 control characters (except tab/nl/cr) should be **ignored at all levels**; they should NOT affect even level 4. Similarly for BiDi control characters.

2. Currently level 4 consist of the 10646 character code (or a string of such). This leads to very strange behaviour if used right off. E.g. "it's" and "its" get ordered in the given order if the apostrophe is the ASCII one (a vertical glyph with mixed usage), but if one uses 02BC (modifier letter apostrophe, preferred character for this usage, the order becomes "its" followed by "it's". Former section 6.2.2.2 tried to fix this with a hack (including some edge case anomalies), but it is much preferable to use a proper solution: **give all letters and digits a level 4 weight called PLAIN that is heavier than all level 4 weights for symbols and punctuation**. Then we get a consistent and explainable order, also when punctuation is involved.
3. Weights of symbols/punctuation should **NOT be their 10646 code point**. Indeed, the "Canadian specials" hack in the balloted table indicate that a code point weight approach is unacceptable. All of the symbols and punctuation (that is ignored at levels 1-3) should have a level 4 weight such that they are grouped fairly logically together, which may give the "Canadian specials" weights such that their ordering is conforming with the Canadian standard, **but still groups similar symbols/punctuation together considering all of 10646.**

9 Example tailorings (minor)

There are two example tailorings of the template table given in an annex. However, neither of them is a "full" tailoring based on the template table. This makes them nearly useless as examples. N640 is a, in some sense, **"full" tailoring based on the template table (in XML format)**. (This has been updated to follow the updated DTD.)

In addition the two tailorings already present should be made "full", and in particular be made to be based on the template, and it would also be helpful to have a tailoring for Japanese where the length marks are collated as a variant of the vowel each represent (depending on the preceding letter). (N641 has, in comments, so tailored 3 (of about 80*2) kana letters with length marks.)

10 Editorial comments

We have a number of editorial comments that can most easily be found by a difference-annotated version of the 14651 text. **(to be supplied)**