
**Information technology — International
string ordering and comparison — Method
for comparing character strings and
description of the common template
tailorable ordering**

iTeh STANDARD PREVIEW
*Technologies de l'information — Classement international et comparaison
de chaînes de caractères — Méthode de comparaison de chaînes de
caractères et description du modèle commun d'ordre de classement*

[ISO/IEC 14651:2001](https://standards.iso.org/standards/catalog/standards/sist/9734992d-9cd1-43c1-bd7c-817ff109a93e/iso-iec-14651-2001)

<https://standards.iteh.ai/catalog/standards/sist/9734992d-9cd1-43c1-bd7c-817ff109a93e/iso-iec-14651-2001>

Reference number
ISO/IEC 14651:2001(E)



PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

iTeh STANDARD PREVIEW
(standards.iteh.ai)

[ISO/IEC 14651:2001](https://standards.iteh.ai/catalog/standards/sist/9734992d-9cd1-43c1-bd7c-817ff109a93e/iso-iec-14651-2001)

<https://standards.iteh.ai/catalog/standards/sist/9734992d-9cd1-43c1-bd7c-817ff109a93e/iso-iec-14651-2001>

© ISO/IEC 2001

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.ch
Web www.iso.ch

Printed in Switzerland

Contents

Foreword	iv
Introduction	v
1. Scope	1
2. Conformance	2
3. Normative references	2
4. Terms and definitions	3
5. Symbols and abbreviations	4
6. String comparison	4
6.1. Preparation of character strings prior to comparison	4
6.2. Key building and comparison	5
6.2.1. Preliminary considerations	5
6.2.2. Reference ordering key formation	6
6.2.3. Reference comparison method for ordering character strings	7
6.3. Common Template Table: formation and interpretation	8
6.3.1. BNF syntax rules for the Common Template Table in Annex A	9
6.3.2. Well-formedness conditions	12
6.3.3. Interpretation of tailored tables	13
6.3.4. Evaluation of weight tables	14
6.3.5. Conditions for considering specific table equivalences	15
6.3.6. Conditions for results to be considered equivalent	15
6.4. Declaration of a delta	16
6.5. Name of the Common Template Table and name declaration	17
Annex A (normative) Common Template Table	18
Annex B (informative) Example tailoring deltas	19
B.1. Example 1 – Minimal tailoring	19
B.2. Example 2 – Reversing the order of lowercase and uppercase letters	19
B.3. Example 3 – Canadian delta and benchmark	19
B.4. Example 4 – Danish delta and benchmark	22
Annex C (informative) Preparation	25
C.1. General considerations	25
C.2. Thai string ordering – a case involving preparation required to get the proper ordering ..	25
C.2.1. Thai Ordering Principle	25
C.2.2. Algorithmic Aspect	27
C.3. Handling of numeral substrings in collation	29
C.3.1. Handling of 'ordinary' numerals for natural numbers	29
C.3.2. Handling of positional numerals in other scripts	32
C.3.3. Handling of other non-pure positional system numerals or non-positional system numerals (e.g. Roman numerals)	32
C.3.4. Handling of numerals for whole numbers	32
C.3.5. Handling of positive positional numerals with fractional parts	34
C.3.6. Handling of positive positional numerals with fraction parts and exponent parts	35
C.3.7. Handling of date and time of day indications	35
C.3.8. Making numbers less significant than letters	37
C.3.9. Maintaining determinacy	37
Annex D (informative) Tutorial on solutions brought by this standard to problems of lexical ordering	38
D.1. Problems	38
D.2. Solution	39
D.3. Tailoring	41
Bibliography	42

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 3.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this International Standard may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

International Standard ISO/IEC 14651 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 22, *Programming languages, their environments and system software interfaces*.

Annex A forms a normative part of this International Standard. Annexes B, C and D are for information only.

iTeh STANDARD PREVIEW

(standards.iteh.ai)

ISO/IEC 14651:2001

<https://standards.iteh.ai/catalog/standards/sis/9734902d19ed1e13e1bd7817ff109a93e/iso-iec-14651-2001>

Introduction

This International Standard provides a method, applicable around the world, for ordering text data, and provides a Common Template Table which, when tailored, can meet a given language's ordering requirements while retaining reasonable ordering for other scripts.

The Common Template Table requires some tailoring in different local environments. Conformance to this International Standard requires that all deviations from the Template, called "deltas", be declared to document resultant discrepancies.

This International Standard describes a method to order text data independently of context.

The Draft Technical Report ISO/IEC DTR 14652 (under development) has specifications for ordering that informatively complements the specifications in this International Standard, and where additional information may be sought on ordering keywords defined in this International Standard.

iTeh STANDARD PREVIEW (standards.iteh.ai)

[ISO/IEC 14651:2001](https://standards.iteh.ai/catalog/standards/sist/9734992d-9cd1-43c1-bd7c-817ff109a93e/iso-iec-14651-2001)

<https://standards.iteh.ai/catalog/standards/sist/9734992d-9cd1-43c1-bd7c-817ff109a93e/iso-iec-14651-2001>

iTeh STANDARD PREVIEW
(standards.iteh.ai)

ISO/IEC 14651:2001

<https://standards.iteh.ai/catalog/standards/sist/9734992d-9cd1-43c1-bd7c-817ff109a93e/iso-iec-14651-2001>

Information technology — International string ordering and comparison — Method for comparing character strings and description of the common template tailorable ordering

1. Scope

This International Standard defines:

- A reference comparison method. This method is applicable to two character strings to determine their relative order. The method can be applied to strings containing characters from the full repertoire of ISO/IEC 10646-1. This method is also applicable to subsets of that repertoire, such as those of the different ISO/IEC 8-bit standard character sets, or any other character set, standardised or not, to produce ordering results valid (after tailoring) for a given set of languages for each script. This method uses collation tables derived either from the Common Template Table defined in this International Standard or from one of its tailorings.
- A reference format. The format is described using the Backus-Naur Form (BNF). This format is used to describe the Common Template Table. The format is used normatively *within* this International Standard.
- A Common Template Table. A given tailoring of the Common Template Table is used by the reference comparison method. The Common Template Table describes an order for all characters encoded in the first edition of ISO/IEC 10646-1 up to Amendment 7. It allows for a specification of a fully deterministic ordering. This table enables the specification of a string ordering adapted to local ordering rules, without requiring an implementer to have knowledge of all the different scripts already encoded in the UCS.

<https://standards.iteh.ai/catalog/standards/sist/9734992d-9cd1-43c1-bd7c-14651-2001>
NOTE 1: This Common Template Table is to be modified to suit the needs of a local environment. The main benefit, worldwide, is that for other scripts, often no modification may be required and that the order will remain as consistent as possible and predictable from an international point of view.

NOTE 2: The character repertoire used in this International Standard is equivalent to that of the Unicode Standard version 2.1.

- A reference name. The reference name refers to this particular version of the Common Template Table, for use as a reference when tailoring. In particular, this name implies that the table is linked to a particular stage of development of the ISO/IEC 10646 Universal multiple-octet coded character set.
- Requirements for a declaration of the differences (delta) between the collation table and the Common Template Table.

This International Standard does *not* mandate:

- A specific comparison method; any equivalent method giving the same results is acceptable.
- A specific format for describing or tailoring tables in a given implementation.
- Specific symbols to be used by implementations except for the name of the Common Template Table.
- Any specific user interface for choosing options.
- Any specific internal format for intermediate keys used when comparing, nor for the table used. The use of numeric keys is not mandated either.

ISO/IEC 14651:2001 (E)

- A context-dependent ordering.
- Any particular preparation of character strings prior to comparison.

NOTE 1: It is normally necessary to do preparation of character strings prior to comparison even if it is not prescribed by this International standard (see informative Annex C).

NOTE 2: Although no user interface is required to choose options or to specify tailoring of the Common Template Table, conformance requires always declaring the applicable delta, a declaration of differences with this table. It is recommended that processes present available tailoring options to users.

2. Conformance

A process is conformant to this International Standard if it meets the requirements prescribed in subclauses 6.2 to 6.5.

A declaration of conformity to this International Standard shall be accompanied by a statement, either directly or by reference, of the following:

- The number of levels that the process supports; this number shall be at least three.
- Whether the process supports the `forward,position` processing parameter.
- Whether the process supports the `backward` processing parameter and at which level.
- The tailoring *delta* described in clause 6.4 and how many levels are defined in the delta.

It is the responsibility of implementers to show how their delta declaration is related to the table syntax described in 6.3, and how the comparison method they use, if different from the one mentioned in clause 6, can be considered as giving the same results as those prescribed by the method specified in clause 6.

[https://standards.iteh.ai/catalog/standards/sist/9734992d-9cd1-43c1-bd7c-](https://standards.iteh.ai/catalog/standards/sist/9734992d-9cd1-43c1-bd7c-817ff109a93e/iso-iec-14651-2001)

[817ff109a93e/iso-iec-14651-2001](https://standards.iteh.ai/catalog/standards/sist/9734992d-9cd1-43c1-bd7c-817ff109a93e/iso-iec-14651-2001)

3. Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this International Standard. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this International Standard are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies. Members of ISO and IEC maintain registers of currently valid International Standards.

- ISO/IEC 10646-1:1993 *Information technology – Universal Multiple-Octet Coded Character Set (UCS) – Part 1: Architecture and Basic Multilingual Plane.*
- ISO/IEC 10646-1:1993/Amd.1:1996 *Information technology – Universal Multiple-Octet Coded Character Set (UCS) – Part 1: Architecture and Basic Multilingual Plane Amendment 1: Transformation Format for 16 planes of group 00 (UTF-16).*
- ISO/IEC 10646-1:1993/Amd.2:1996 *Information technology – Universal Multiple-Octet Coded Character Set (UCS) – Part 1: Architecture and Basic Multilingual Plane Amendment 2: UCS Transformation Format 8 (UTF-8).*
- ISO/IEC 10646-1:1993/Amd.4:1996 *Information technology – Universal Multiple-Octet Coded Character Set (UCS) – Part 1: Architecture and Basic Multilingual Plane Amendment 4.*

- ISO/IEC 10646-1:1993/Amd.5:1998 *Information technology – Universal Multiple-Octet Coded Character Set (UCS) – Part 1: Architecture and Basic Multilingual Plane Amendment 5: Hangul syllables.*
- ISO/IEC 10646-1:1993/Amd.6:1997 *Information technology – Universal Multiple-Octet Coded Character Set (UCS) – Part 1: Architecture and Basic Multilingual Plane Amendment 6: Tibetan.*
- ISO/IEC 10646-1:1993/Amd.7:1997 *Information technology – Universal Multiple-Octet Coded Character Set (UCS) – Part 1: Architecture and Basic Multilingual Plane Amendment 7: 33 additional characters.*
- ISO/IEC 10646-1:1993/Amd.9:1997 *Information technology – Universal Multiple-Octet Coded Character Set (UCS) – Part 1: Architecture and Basic Multilingual Plane Amendment 9: Identifiers for characters.*
- ISO/IEC 10646-1:1993/Amd.18:1999 *Information technology – Universal Multiple-Octet Coded Character Set (UCS) – Part 1: Architecture and Basic Multilingual Plane Amendment 18: Symbols and other characters.*

NOTE: Only the EURO SIGN and the OBJECT REPLACEMENT CHARACTER from Amendment 18 are accounted for in Annex A of this International Standard at this time.

4. Terms and definitions

For the purposes of this International Standard, the following terms and definitions apply:

- 4.1 character string** a sequence of characters
- 4.2 collation** equivalent to the term "ordering"
- 4.3 collating symbol** a symbol used to specify weights assigned to a collating element
- 4.4 collation (weighting) table** a mapping from collating elements to weighting elements
- 4.5 collating element** a sequence of one or more characters that are considered a single element for ordering
- 4.6 delta** differences from a given collation table. The given collation table, together with a given delta, forms a new collation table. Unless otherwise specified in this International Standard, the term "delta" always refers to differences from the Common Template Table as defined in this International Standard
- 4.7 (collation) level** the sequence number for a subkey
- 4.8 ordering** a process by which two strings are determined to be in exactly one of the relationships of **less than**, **greater than**, or **equal** to one another
- 4.9 ordering key** a sequence of subkeys used to determine an order
- 4.10 (collation) preparation** a process in which given character strings are mapped to (other) character strings logically before the calculation of the ordering key for each of the strings
- 4.11 reference comparison method** the method for establishing an order between two ordering keys (see clause 6)

ISO/IEC 14651:2001 (E)

- 4.12 subkey** a sequence of weights computed for a character string for a particular level.
- 4.13 symbol** collating element
- 4.14 (collation) weight** a positive integer value, used in subkeys, reflecting the relative order of collating elements
- 4.15 weighting element**
a list of a given number of weights sequentially ordered by level

5. Symbols and abbreviations

Following ISO/IEC 10646-1:1993, Amendment 9, characters are referenced as UXXXX where X stands for any hexadecimal digit (using upper case letters where applicable) and refers to the value of that character in ISO/IEC 10646-1. This convention is used throughout this International Standard.

In the Common Template Table arbitrary symbols representing weights are used according to the BNF notation description in clause 6.3.1.

6. String comparison

6.1. Preparation of character strings prior to comparison

It may be necessary to transform character strings before the reference comparison method is applied to them (see annex C for an example of such preparation). Although not part of the scope of this International Standard, preparation may be an important part of the ordering process. See Annex C for some examples of preparation.

Where applicable, it can be an important part of the preparation phase to map characters from a non-UCS encoding scheme to the UCS for input to the comparison method. This task can amongst other things encompass the correct handling of escape sequences in the originating encoding scheme, the mapping of characters without an allocated UCS codepoint to an application-defined codepoint in the private zone area and change the sequence of characters in strings that are not stored in logical order. For example, for visual order Arabic code sets, input strings must be put into logical order; and for some bibliographic code sets, strings with combining accents stored before their respective base character require that the combining accents be put after their base character. The resulting string sequence may then have to be remapped into its original encoding scheme.

NOTE 1: The Common Template Table is designed so that combining sequences and corresponding single characters (precomposed) will have precisely the same ordering. To avoid inadvertently breaking this invariant (and in the process breaking Unicode conformance), tailoring should reorder combining sequences when corresponding characters are reordered. For example, if Å is reordered after Z, then the sequence <A>+<combining diaeresis> should also be reordered. To avoid exposing encoding differences that may be invisible to the end-user, it is recommended that strings be normalized according to Unicode normalization to achieve this equivalence – see Bibliography, Unicode Technical Report no. 15.

NOTE 2: Escape sequences and control characters constitute very sensitive data to interpret, and it is highly recommended that preparation should filter out or transform these sequences.

NOTE 3: Since the reference method is a logical statement for the mechanism for string comparison, it does not preclude an implementation from using a non-UCS character encoding only, as long as it produces results as if it were using the reference comparison method.

6.2. Key building and comparison

6.2.1. Preliminary considerations

6.2.1.1. Assumptions

The collation table is a mapping from collating elements to weighting elements. In each weighting element, four levels are described in the Common Template Table. This number of levels can be extended or reduced, but not below 3 levels, in tailoring.

NOTE: *In the Common Template Table, levels generally have the following characteristics, although the purpose of each level is not absolute:*

Level 1: *This level generally corresponds to the set of common letters of the alphabets for that script, if the script is alphabetic, and to the set of common characters of the script if the script is ideographic or syllabic.*

Level 2: *This level generally corresponds to diacritical marks affecting each basic character of the script. For some languages, letters with diacritics are always considered an integral part of the basic letters of the alphabet, and are not considered at this second level, but rather at the first. For example, in Spanish, N TILDE is considered a basic letter of the Latin script. Therefore, tailoring for Spanish will change the definition of N TILDE from "the weight of an N in the first level and the weight of a TILDE in the second level" to "the weight of an N TILDE (placed after N and before O) in the first level, and indication of the absence of a diacritic in the second level". For some characters, variant letter shapes are also dealt with on level 2. An example of this is ß, the LATIN LETTER SHARP S, which is treated as equivalent to ss on level 1, but traditionally distinguished from it on level 2.*

Level 3: *This level generally corresponds to case distinctions or to distinctions based on variant letter shapes (like the distinction between Hiragana and Katakana).*

Level 4: *This level generally corresponds to weighting differences that are less significant than those at the other levels. Often the last level (level 4 in the Common Template Table) is intended to specify additional weighting for "special" characters, i.e., characters normally not part of the spelling of words of a language (such as dingbats, punctuation, etc.), sometimes called "ignorable" characters in the context of computerized ordering.*

6.2.1.2. Processing properties

A given tailored table has specific scanning and ordering properties. These properties may have been changed by the tailoring.

A scanning direction (forward or backward) for each level is used to indicate how to process the string. The scanning direction is a global property of each level defined in the tailored table.

An optional property of the last level of comparison, if that level is greater than three, is that before comparing weights of each "ignorable" character, a comparison on the numeric position of each such character in the two strings may be effected. This processing property is known as the 'position' option. In other words, for two strings equivalent at all levels except the last one, the string having an ignorable in the lowest position comes before the other one. In case corresponding "ignorables" are at the same position, then their weights are considered, until a difference is found. *Support* for this kind of processing is *optional* and is not necessary to claim conformance.

NOTE: *The scanning direction (forward or backward) is not normally related to the natural writing direction of scripts. The scanning direction applies to the logical sequence of the coded character string.*

ISO/IEC 14651:2001 (E)

According to ISO/IEC 10646, for scripts written right to left, such as Arabic, the lowest positions in the logical sequence of characters correspond to the rightmost characters of a string (from the point of view of their natural presentation sequence). Conversely, for the Latin script, written left to right, the lowest positions in the logical sequence of characters correspond to the leftmost characters of the string (from the point of view of their natural presentation sequence).

Scanning forward starts with the lowest position in the logical sequence, while scanning backward starts from the highest position, independently of the presentation sequence. The scanning direction for ordering purposes is a global property of each level described in the table.

In ISO/IEC 10646-1, the Arabic script is artificially separated into two pseudo-scripts: 1) the logical, intrinsic Arabic, coded independently of shapes, and 2) the Arabic presentation forms. Both allow the complete coding of Arabic, but intrinsic Arabic is normally preferred for better processing, while presentation-form Arabic is preferred by some presentation-oriented applications. ISO/IEC 10646-1 does not prescribe that the presentation forms be stored in any specific order, and in some implementations, the storage order for the latter is the reverse of the storage order used for intrinsic Arabic. It is therefore advisable that prehandling be used to make sure that Arabic presentation forms and other Arabic characters be fed to the comparison method in logical order.

A tailored table may be separated into sections for ease of tailoring. Each section is then assigned a name consistent with the specification in subclause 6.3.1. One of the tailoring possibilities is to assign a given order to each section and to change the relative order of an entire section relative to other sections.

6.2.2. Reference ordering key formation

When two strings are to be compared to determine their relative order, the two strings are first broken up into a sequence of collating elements taking into account the multi-character “collating-element” statements declared and used in a tailored table (if the syntax of clause 6.3.1 is used). For the syntax used for expressing the Common Template Table, the name of a collating element consisting of a single character, is formed by the UCS value of the character, expressed as a hexadecimal string, prefixed with “U”. For multi-character collating elements, the name and association to characters can be found via the collating elements declarations.

Then a sequence of m intermediary subkeys is formed out of a character string, where m is the number of levels described in a tailored collation weighting table.

Each ordering key is a sequence of subkeys. Each subkey is a list of numeric weights. A subkey is formed by successively appending the list of the weights assigned, at the level of the subkey, to each collating element of the string. The keyword “IGNORE” in the Common Template Table at the place of a sequence of collating symbols at a level, indicates that the sequence of weights at that level for that collating element is an empty sequence of weights.

There are three ways of forming subkeys: subkeys formed using the “forward” processing parameter; subkeys formed using the “backward” processing parameter; and subkeys formed using the “forward,position” processing parameter. Subkeys that use the “position” option can only occur at the last level, and only if that level is greater than three. Support of the “position” option is not required for conformance. If the processing parameter “forward,position” is not supported, “forward,position” shall be interpreted as if the processing parameter had been “forward”.

If there is no entry in the tailored table for a character of the input string, then the character’s weights are undefined. Characters with undefined weights should be ordered, with respect to characters that have defined weights, as if the undefined ones were given the weight named “UNDEFINED” at the first level. If there is no weight assignment to the symbol “UNDEFINED” before the symbol <SFFFF>’s weight assignment in a given tailored table, then the table shall be interpreted as if “UNDEFINED” was weighted just before <SFFFF> (the maximal level 1 weight). The ordering of characters with undefined weights with respect to other characters with undefined weights is not specified in this standard.

NOTE: A possible way to order characters with undefined weights is as if there were tailoring lines like this one added to the table, in UCS code point order (call the maximal level 4 weight <PLAIN> here):

```
<UXXXX> "<UNDEFINED><UXXXX>";<BASE>;<MIN>;<PLAIN>
```

6.2.2.1. Formation of a subkey with the “forward” level processing parameter

Subkeys, at a particular level, formed with the “forward” level processing parameter, are built in the following way:

During forward scanning of each collating element of the input character string, one or more weights are obtained. These weights are obtained by matching the collating element in the given tailored collation weighting table, obtaining the list of weights assigned to the collating element at the particular level. The obtained weight list is appended to the end of the subkey.

6.2.2.2. Formation of a subkey with the “backward” level processing parameter

Subkeys, at a particular level, formed with the “backward” level processing parameter are built by forming a subkey as with the “forward” parameter, then reversing that subkey weight by weight.

6.2.2.3. Formation of a subkey with the “forward,position” level processing parameter

Subkeys, at the last level, formed with the “forward,position” level processing parameter are formed by forming a subkey as with the “forward” parameter, but for collating elements that are not “IGNORE”d at all levels but the last one, their last level weighting (list of weights) is replaced by a single weight (call it <PLAIN> here) that is larger than all other weights at the last level in the given tailored table. Collating elements that are “IGNORE”d at all levels but the last one, retain their weighting according to the given tailored table. Finally, any trailing sequence of the maximal weight (<PLAIN>) is removed from the subkey, effectively replacing each trailing maximal weight with a zero weight.

NOTE: For any level, implementations are allowed to apply an order-preserving reduction of all subkeys at that level. Such an order-preserving subkey reduction is useful for levels 2, 3, and 4. Level 2 often has long stretches of the weight named <BASE> in Annex A. Level 3 often has long stretches of the weight named <MIN> in Annex A. Level 4 often has long stretches of the weight named <PLAIN> here. One such ordering preserving subkey reduction technique effectively encodes, in the last level subkey, the (relative) position, as single number each, of each otherwise ignored character; hence the name of the “position” option.

6.2.3. Reference comparison method for ordering character strings

The reference comparison method for ordering two given character strings (*after* collation preparation, which is not part of the reference comparison method itself) is to compare ordering keys generated by the reference key formation method described in subclause 6.2.2 of this International Standard:

- Begin by building an ordering key, using a given tailored collation weighting table, for each of the two given character strings being compared.
- Then compare the resulting keys according to the key ordering definition below in this subclause. Keys can be compared either up to a given level, or up to the last level of the given tailored collation weighting table.

NOTE 1: The comparison may be made while generating the ordering keys for two strings to be compared, stopping the key generation when the order of the strings can be determined. Such a technique is sometimes termed *lazy evaluation*, and some systems support it by default. This avoids generating the full ordering key, when an ordering difference may be found early in the keys. When a bigger set of strings are to be ordered, it may be advisable to generate the ordering keys, and store each key or an initial segment of each, before comparing the keys.

Weights for different levels should not be compared, which implies that subkeys at different levels should not be compared. Nor should keys generated from different tailored tables be compared.

ISO/IEC 14651:2001 (E)

NOTE 2: This allows implementations to assign weightings at each level independently of the other levels, and independently of other tailorings.

m is the maximal level of a given tailored table. Recall that a key is a list, of length m , of subkeys; a subkey is a list of weights; and a weight is a positive integer. Other notations used below are:

- L_z is the length of the subkey z , i.e., the number of weights in the subkey.
- $z_{wt}(a)$, where $1 \leq a \leq L_z$, is the weight at index position a (an integer > 0) of the subkey z .
- $u_{sk}(b)$, where $1 \leq b \leq m$, is the subkey at level b (an integer > 0) of the key u .

The orderings of weights, subkeys, and ordering keys (up to a given level, or up to the last level) are total orders, defined for a given tailored collation table as follows:

1. Weights are positive integer values (in the reference method), and are compared as such for the purposes of collation.
2. A subkey v is *less than* a subkey w (written $v < w$) **if and only if** there exists an integer i , where $1 \leq i \leq L_v + 1$ and $i \leq L_w$, such that for all integers j , where $1 \leq j < i$, the equality $v_{wt}(j) = w_{wt}(j)$ holds; and either $i \leq L_v$ and $v_{wt}(i) < w_{wt}(i)$, or $i = L_v + 1$ and $0 < w_{wt}(i)$.

A subkey v is *greater than* a subkey w (written $v > w$) **if and only if** w is less than v . A subkey v is *equal to* a subkey w (written $v = w$) **if and only if** neither v is less than w , nor w is less than v .

3. An ordering key x is *less than* an ordering key y at level s (written $x <_s y$) **if and only if** there exists an integer i , where $1 \leq i \leq s$ and $i \leq m$, such that for all integers j , where $1 \leq j < i$, the equality $x_{sk}(j) = y_{sk}(j)$ holds, and $x_{sk}(i) < y_{sk}(i)$.

An ordering key x is *greater than* an ordering key y at level s (written $x >_s y$) **if and only if** y is less than x at level s . An ordering key x is *equal to* an ordering key y at level s (written $x =_s y$) **if and only if** neither x is less than y at level s , nor y is less than x at level s .

4. For ordering keys, $<$, $>$, and $=$ are defined as $<_m$, $>_m$, and $=_m$ respectively.

NOTE 3: For ordering keys, $x <_t y$ implies $x <_{t+1} y$, $x >_t y$ implies $x >_{t+1} y$, $x =_t y$ implies $x =_{t-1} y$, $x <_0 y$ is false, $x >_0 y$ is false, and $x =_0 y$ is true. Above level m , for a given tailored table, there are no further ordering distinctions. Note that this definition implies that if two ordering keys are in the 'less than' relationship at level 1, they will also be in the 'less than' relationship at levels 2, 3, 4, etc. In general, whenever two ordering keys are less than at a given level, they will also automatically be less than at all subsequent, higher levels. Conversely, if two ordering keys are equal at a given level, they will also automatically be equal at all preceding, lower levels.

6.3. Common Template Table: formation and interpretation

This clause specifies:

- The syntax used to form the Common Template Table in Annex A of this International Standard or a tailored table based upon the Common Template Table as expressed in Annex A.
- Conditions of well-formedness of a table using this syntax.
- Interpretation of tailoring statements in deltas for tables formed using this syntax.
- Evaluation from symbols to weights of tailored tables formed using this syntax.
- Conditions for considering two tables as equivalent.

- Conditions for considering comparison results as equivalent.

6.3.1. BNF syntax rules for the Common Template Table in Annex A

Definitions between <angle brackets> make use of terms not defined in this BNF syntax, and assume general English usage.

Other conventions:

- * indicates 0 or more repetitions of a token or a group of tokens;
- + indicates 1 or more repetitions of a token or a group of tokens;
- ? indicates optional occurrence of a token or a group of tokens (0 or 1 occurrences);
- parentheses are used to group tokens;
- production rules are terminated by a semicolon;

Define collation tables as sequences of lines:

```
weight_table = common_template_table | tailored_table ;

common_template_table =
    simple_line+ ;

tailored_table = table_line+ ;
```

Define the line types:

```
simple_line = (symbol_definition | collating_element |
              weight_assignment | order_end)? line_completion ;

table_line = simple_line | tailoring_line ;

tailoring_line = (reorder_after | order_start | reorder_end |
                 section_definition | reorder_section_after)
                 line_completion ;
```

Define the basic syntax for collation weighting:

```
symbol_definition =
    'collating-symbol' space+ symbol_element ;

symbol_element = symbol | symbol_range ;

symbol_range = symbol '..' symbol ;

symbol = simple_symbol | ucs_symbol ;

ucs_symbol = ('<U' four_digit_hex_string '>') |
             ('<U-' eight_digit_hex_string '>') ;

simple_symbol = '<' identifier '>' ;

collating_element =
    'collating-element' space+ symbol space+
    'from' space+ quoted_symbol_sequence ;

quoted_symbol_sequence =
    '"' symbol+ '"' ;

weight_assignment =
    simple_weight | symbol_weight ;
```