



中华人民共和国国家标准

GB/T 36335—2018

信息技术 藏文字符排序规范

Information technology—Specification for collation of Tibetan characters

2018-06-07 发布

2019-01-01 实施

国家市场监督管理总局
中国国家标准化管理委员会 发布

目 次

前言 I

1 范围 1

2 规范性引用文件 1

3 术语和定义 1

4 藏文字符排序原则 2

附录 A（资料性附录） 藏文音节和梵文转写藏文音节的一维展开 5

参考文献..... 6

前 言

本标准按照 GB/T 1.1—2009 给出的规则起草。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别这些专利的责任。

本标准由全国信息技术标准化技术委员会(SAC/TC 28)提出并归口。

本标准起草单位:中国电子技术标准化研究院、青海师范大学、西藏自治区藏语文工作委员会办公室、中国藏学研究中心、青海民族大学、西藏大学、中国科学院软件研究所、西北民族大学。

本标准主要起草人:黄鹤鸣、熊涛、多拉、欧珠、安见才让。

信息技术 藏文字符排序规范

1 范围

本标准规定了信息处理用藏文字符和梵文转写藏文字符排序原则。

注：包括但不限于 GB/T 13000—2010、GB 16959—1997、GB/T 20542—2006、GB/T 22238—2008 中所规定的藏文字符和梵文转写藏文字符。

本标准适用于藏文信息处理。

2 规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件，仅注日期的版本适用于本文件。凡是不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

GB/T 13000—2010 信息技术 通用多八位编码字符集(UCS)

GB 16959—1997 信息技术 信息交换用藏文编码字符集 基本集

3 术语和定义

下列术语和定义适用于本文件。

3.1

字符 character

供组织、控制或表示数据的元素集合中的一个元素。

3.2

字符串 character string

能看作一个单独对象的字符序列。

3.3

编码字符 coded character

被赋予编码的字符。

3.4

图形字符 graphic character

不同于控制功能的字符，通常具有书写、打印或显示的可视表示。

3.5

占位字符 spacing character

显示时占用宽度的字符。

3.6

不占位字符 non-spacing character

显示时不占用宽度的字符，用于和前导的字符串相组合。

注：前导的字符串可能是一个占位字符也可能由一个占位字符和几个不占位字符构成的序列。

3.7

梵文转写藏文字符 transliterating Tibetan character

GB/T 13000—2010 和 GB 16959—1997 所收集的梵文转写藏文字母以及由这些字母的垂直组合

形成的字符。

3.8

第一类元音 **type one vowel**

元音 a 、 e 、 i 、 o 、 u 、 ä 、 ö 、 y 、 ø 、 ɛ 、 ɜ 以及 ɔ 。

3.9

第二类元音 type two vowel

元音^{◌̊}和^{◌̋}。

3.10

排序 collation

给定字符或者字符串之间的先后顺序。

3.11

排序元素 **collation element**

为了实现字符间的排序而为每个字符赋予的数值。

4 藏文字符排序原则

4.1 藏文符号、数字以及字母间的顺序

藏文中符号、数字以及字母间的先后顺序为：

[illegible]

同一个辅音字母的不占位字符要排在占位字符的后面。因此,所有辅音字母的占位字符和不占位字符间的先后关系为:ྱ, སྐ, རྩ, ལྩ, ཤྩ, …, ཏྩ, ཉྩ, ཐྩ, དྩ。

字母ལ、ར、ལ、ཤ作为下加辅音时,相应的不占位字符间的顺序为:ྱ,ྲ,ླ,ྤ。

元音符号之间的顺序为：◌_a、◌_i、◌_e、◌_ɛ、◌_ʊ、◌_y、◌_ɨ、◌_ɪ、◌_ə、◌_ɐ、◌_{ɔ̃}、◌_{ɑ̃}。

图形符号分解成、、后进行排序。

符号 ཨྱ, ཨྲ, ཨླ, ཨྴ, ཨྵ, ཨྶ, ཨྷ, ཨྸ, ཨྐྵ, ཨྺ, ཨྻ, ཨྼ, ཨ྽, ཨ྿, ཨྻྱ, ཨྻྲ, ཨྻླ, ཨྻྴ, ཨྻྵ, ཨྻྶ, ཨྻྷ, ཨྻྸ, ཨྻྐྵ, ཨྻྺ, ཨྻྻ, ཨྻྼ, ཨྻ྽, ཨྻ྿ 不参与排序,也就是说这些符号的出现,不影响一个藏文字符串的序性。

[illegible]

4.2 藏文音节的排序

4.2.1 藏文音节通用结构

典型的藏文音节由基本辅音字母、前加辅音字母、上加辅音字母、下加辅音字母、元音符号、后加辅音字母以及又后加辅音字母组成。有些藏文音节有两个下加辅音,而有些梵文转写藏文字符有两个元音。因此,藏文音节的通用结构应包含基本辅音字母、前加辅音字母、上加辅音字母、第一下加辅音字母、第二下加辅音字母、第一类元音符号、后加辅音字母/第二类元音符号以及又后加辅音字母,如图 1 所示。

满足后加辅音的条件,例如:དགྲ་,གྲག་,མགྲ་,དང་,གད་,མད་,གད་,དམ་,དར་,གམ་以及དམ་,则这两个辅音分别是音节的基本辅音和后加辅音;

对于仅由三个辅音字母左右组合形成的音节,如果第二和第三个辅音分别满足后加辅音和又后加辅音的条件,而第一和第三个辅音分别满足前加辅音和后加辅音的条件,例如:གྲགམ་,则这三个辅音依次是这个音节的前加辅音、基本辅音和后加辅音。

4.3 梵文转写藏文字的排序

4.3.1 梵文转写藏文字符排序规则

一个梵文转写藏文组合字符可能与另一个梵文转写藏文组合字符相比较也可能与一个藏文音节相比较。为了实现这种比较,首先,将参与排序的每个梵文转写藏文字符分解成符合藏文通用结构的音节序列;其次,将每个藏文音节按照特定的顺序展开成一维字符串(参见附录 A);最后,比较得到的一维字符串。

如 4.2.3 所述,将藏文音节展成一维字符串的特定顺序为基本辅音字母、前加辅音字母、上加辅音字母、第一下加辅音字母、第二下加辅音字母、第一类元音符号、后加辅音字母/第二类元音符号以及又后加辅音字母。

4.3.2 梵文转写藏文音节通用结构

梵文转写藏文音节是基本辅音字母、最多两个下加辅音字母和最多两个元音字母的垂直组合。因此,梵文转写藏文音节通用结构由基本辅音字母、第一下加辅音字母、第二下加辅音字母、第一类元音和第二类元音组成,如图 2 所示。

第一/第二元音
基本辅音字母
第一下加辅音字母
第二下加辅音字母
第一元音

图 2 梵文转写藏文音节通用结构

4.3.3 梵文转写藏文字符的分解

如果一个梵文转写藏文字符不符合 4.3.2 中描述的梵文转写藏文音节的通用结构,则应分解该字符。分解算法是:将第一层的辅音字母单独看作一个音节,而将剩余字母(包括元音符号)的组合看作一个整体。如果剩余字母(包括元音符号)的组合不符合梵文转写藏文音节的通用结构,则继续前面的过程,直到剩余字母(包括元音符号)的组合符合梵文转写藏文音节的通用结构。

附录 A
(资料性附录)

藏文音节和梵文转写藏文音节的一维展开

A.1 藏文音节的一维展开

一般而言,音节是藏文字母的二维组合。为了实现藏文音节的计算机排序需要将二维的藏文音节展成一维字母串,这个展开顺序为基本辅音字母、前加辅音字母、上加辅音字母、第一下加辅音字母、第二下加辅音字母、第一类元音符号、第二类元音符号/后加辅音字母、又后加辅音字母,见图 A.1。

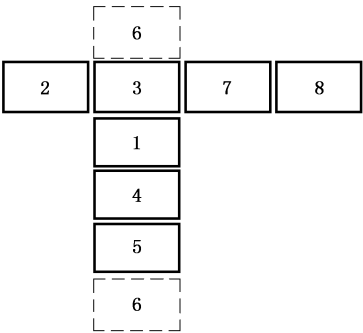


图 A.1 藏文音节的一维展开顺序

如果音节在某个位置上的字母空缺,则展开式中的相应位置用一个特殊的字符来替补。这个特殊字符的排序元素要小于任何一个藏文字母的排序元素。本标准的示例中,用符号“□”表示这个特殊字符。

- 示例 1:藏文音节ལ་按照图 A.1 所示的顺序展开为ལ་□□□□□□。
- 示例 2:藏文音节ར་按照图 A.1 所示的顺序展成一维字母串为ར་□□□□□□;藏文音节ལ་按照图 A.1 所示的顺序展成一维字母串为ལ་□□□□□□。

A.2 梵文转写藏文音节的一维展开

对分解得到的音节序列中的每一个音节按照图 A.1 所示的顺序展开成一维字母串,展开时如果某个位置上的字母空缺则用特殊字符“□”来代替。梵文转写藏文音节ལ་按照图 A.1 所示的顺序展开为ལ་□□□□□□。

示例 1:梵文转写藏文字符ལ་被分解成音节ལ་和ལ་。其中,音节ལ་按照图 A.1 所示的顺序展成一维字母串为‘ལ་□□□□□□’,而音节ལ་按照图 A.1 所示的顺序展成一维字母串为‘ལ་□□□□□□’。因此,梵文转写藏文字符ལ་对应的一维字母串为‘ལ་□□□□□□ལ་□□□□□□’。

示例 2:要比较两个梵文转写藏文字符ལ་和ལ་:需要三个步骤首先,将它们分别分解成音节序列ལ་和ལ་;其次,按照图 A.1 所示的顺序将音节序列ལ་和ལ་分别展成一维字母串,得‘ལ་□□□□□□ལ་□□□□□□’和‘ལ་□□□□□□ལ་□□□□□□’;第三,比较两个一维字母串,得到两个音节序列的先后顺序。从而得到两个梵文转写藏文组合字符间的先后顺序:字符ལ་在字符ལ་的前面。

参 考 文 献

- [1] ISO/IEC 14651:2011 Information technology—International string ordering and comparison—Method for comparing character strings and description of the common template tailorable ordering
 - [2] Unicode Technical Standard #10 Unicode Collation Algorithm
-