引入排序码实现藏文字符的排序

黄鹤鸣[1],赵晨星[2]
(1.青海师范大学 物理系,青海 西宁 810008;
2.青海藏文信息技术研究所,青海 西宁 810008)

摘 要:字符排序一般要遵循字典顺序并且需要为每个参与排序的字符赋予特定的排序码。藏文字符有两种编码方式:动态组合方式和静态组合方式,对于动态组合方式编码的藏文组合字符而言,仅仅根据构成它的字母来排序,则排序结果与字典顺序有较大的差异。文中分析了藏文字符的字典顺序,总结了藏文的字典顺序规律,提出了引入藏文字符集扩展集A中的UNICODE编码为排序码对藏文组合字符进行排序的观点,使得藏文组合字符的排序符合藏文字典顺序。

关键词:藏文;字符;排序;排序码
中图分类号:TP391.1 **文献标识码**:A **文章编号**:1673-629X(2008)10-0068-03

Introducing Sort Code to Realize Tibetan Characters' Sort

HUANG He-ming[1], ZHAO Chen-xing[2]
(1. Physics Department, Qinghai Normal University, Xining 810008, China;
2. Qinghai Institute of Tibetan Information and Technology, Xining 810008, China)

Abstract: The character sorting should follow the sort order of dictionary, and need the specially appointed sort codes for each character participated to sort. Sorting Tibetan dynamic composing character by just using the letters that composing it will lead to too much difference to the character order of Tibetan dictionary. This paper proposed to sort the composite character by introducing the sort code. This sort code is the very code of the character in the extended set of Tibetan character set. Thus, the sort order of the composing character will consist with that of Tibetan dictionary.

Key words: Tibetan; character; sort; sort code

0 引 言

排序是字符串查询的基础,但藏文字符串的排序不同于英文或者汉文字符串的排序。主要原因是藏文中字的含义不同于汉文中的字也不同于英语中的字母。

一个藏文字的顺序由构成该字的各个字母依次决定,藏文字符排序的关键就是将各个字母对一个字顺序的影响讨论清楚,如果这个问题解决了藏文字符排序问题就解决了,因为在字层面上的排序,可以利用汉文或者英文字符串的排序码算法[1~3]。首先简要介绍了藏文中字母、字符和字(音节)之间的区别和联系;一般来说,字符排序的标准是字符的字典顺序,因此分析了藏文的字典顺序,总结了藏文字典顺序的特点;介绍了藏文的两种编码方式:动态组合方式和静态组合方式;最后提出了引入排序码对藏文组合字符进行排序的观点。

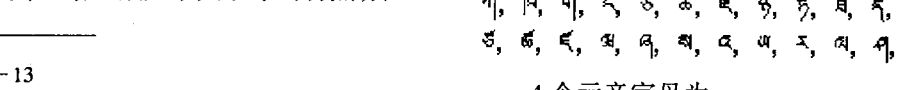
1 藏文中的字母、字符和字

藏文是一种拼音文字,但藏文中的字的含义既不同于英文中的字母也不同于汉文中的一个字。藏文中字母、字符和字之间有严格的区分。

1.1 藏文字母

一般意义上的藏文是指第三次厘定后的藏文,又称为现代藏文。藏文有30个辅音字母和4个元音。

30个辅音字母为:



4个元音字母为:

1.2 藏文字符

。

一般意义上的藏文字符指由多个辅音和元音的垂直叠加体。其中,辅音的个数为一到三个,元音的个数

收稿日期:2008-01-13
基金项目:信息产业部电子信息产业发展基金(信部运[2002]393号)
作者简介:黄鹤鸣(1969-),男(藏族),青海乐都人,副教授,研究方向为藏文信息技术。

为零到一个。根据辅音的个数将藏文组合字符可以分为三种:

(1)一个辅音加元音的形式,例如,组合字符 ཀ;仅仅一个辅音也可以看作字符,例如: ཀ;原因是该字符含有零元音。

(2)两个辅音加元音的形式,例如,字符 སྒ;

(3)三个辅音加元音的形式,例如,字符 སྒྲ。

组合字符中有一个辅音对整个字符的发音影响较大被称为基本辅音,其他辅音叠加在基本辅音的上方或下方。出现在基本辅音上方的辅音称为上加辅音,上加辅音有: ར(ར的变形) ལ和 ས。出现在基本辅音下方的辅音称为下加辅音,下加辅音有: ྱ(ཡ的变形)、ྭ(ཝ的变形)和 ྲ(ར的变形)。因此一个完整的藏文字符的结构是:

上加辅音(可能变形)+基本辅音+下加辅音(可能变形)+元音

图 1 是一个典型的藏文字符,它依次由基本辅音 ཀ、上加辅音 ས、下加辅音 ྱ 和元音 ི 组成。

1.3　藏文字

一个藏文字(见图 2)就是一个藏文音节,一个音节中最多有一个组合字符,被称为基本字符,基本字符前面可以有一个辅音称为前加辅音,能作前加辅音的字母有: ཀ、ག、ང、མ、འ;在基本字符后最多可以有两个辅音,分别称为后加辅音和又后加辅音,能作后加辅音的有: ག、ང、ད、ན、བ、མ、འ、ར、ལ、ས,能作又后加辅音只有两个: ད 和 ས。

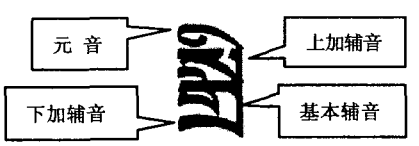


图 1　藏文字符的构成

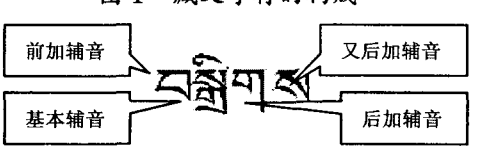


图 2　藏文字的构成

2　字典中藏文字符顺序

藏文字的顺序则由构成字的字母依次决定,这些字母在决定字符的顺序时的重要性次序是(如图 3 所示):基本辅音、上加辅音、前加辅音、下加辅音、元音、后加辅音和又后加辅音。例如,字 སྒྲུབ་པ་ 的顺序依次由基本辅音 ག、上加辅音 ས、前加辅音 ད、下加辅音 ྲ、元音 ུ、后加辅音 བ 和又后加辅音 ས 来决定。字典中的所有字按照基本辅音分为 30 部分,分别称为 ཀ 部、ཁ 部、……ཨ 部,例如字 སྒྲུབ་པ་ 中的基本辅音是 ག,因此它属于 ག 部。这类似于将所有以 a 开始的英文单词集合称为 a 部,将所有以 b 开始的英文单词称为 b 部。从字的顺序可以看出,组合字符的字典顺序依次由基本辅音、上加辅音、下加辅音和元音来决定。例如,字符 སྒྲ 的顺序依次由基本辅音 ག、上加辅音 ས、下加辅音 ྱ 和元音 ི 来决定。

3　藏文字符的两种编码方式

藏文字符集有三个标准:《信息技术 信息交换用藏文编码字符集 基本集》(以下简称基本集),《信息技术 藏文编码字符集 扩充集 A》(以下简称扩充集 A),《信息技术 藏文编码字符集 扩充集 B》(以下简称扩充集 B)。有了这三个标准可以采用两种方式对藏文字符编码:动态组合方式和预组合方式。

3.1　动态组合编码方式

基本集[4]于 1997 年 9 月由国家技术监督局批准,共收集了 193 个字符,其中包括:辅音字母(包括变形辅音字母)、不占位的辅音字母(包括变形辅音字母)、单元音、语音符、藏文标点符号、藏文数字(包括 10 个藏文中独有的半值数字)、图形符号等。基本集也是藏文字符集的国际标准。

由于藏文字符是字母的垂直叠加,因此,仅利用基本集中的字母就可以动态地组合出所有组合字符。这样组合出的字符本身没有编码,与之对应的是构成字符的所有字母的编码。例如,与字符 སྒྲ 对应编码序列是:0F66 OF90 0FB2 0F74(分别是字母 ས、ག、ྲ 和 ི 的 Unicode 编码)。

这种编码方式的优点:

1)符合藏文是拼音文字的特点;

2)字符集所需要的 Unicode 编码较少;

3)符合藏文的国际标准。

3.2　静态组合编码方式

扩充集 A 是基本集的扩充,共收录了 1536 个常用组合字符,每个组合字符都有编码,编码范围:U+F300－U+F8FF。扩充集 A 中字符的 Unicode 编码本身反映了字符的字典顺序。

扩充集 B 是扩充集 A 的补充,共收录了 5669 个次常用组合字符,编码范围:U+0F0000－U+0F1625。扩充集 B 中字符的 Unicode 编码同样反映了该字符的字典顺序。

因此,对于扩展集 A 和 B 而言,每一个字符都有 Unicode 编码,并且该编码反映了字符的字典顺序。对于这种方式编码的字符串,可以利用较高版本的 Microsof Word® 来完成排序,并且排序结果和字典顺序一致。

4 动态组合编码方式中字符的排序

无法利用现有的软件对动态组合方式编码的字符串来排序，开发相应的排序软件有两种方式可供选择：一是完全由构成字符的字母来决定排序；二是引入排序码。

4.1 由字母决定字符排序的方式

在完全由字母决定字符顺序的方式中，字符顺序完全由构成字符的字母来决定，并且每个字母在决定字符顺序时的重要性与字符的书写顺序有关。藏文组合字符在书写时遵循的原则是：1)先辅音后元音；2)辅音部分如果有叠加则按照从上到下的次序书写。因此决定字符顺序的各个字母的重要性依次是：上加辅音、基本辅音、下加辅音，最后是元音，如图 3 所示[5]。

这也是一般的动态组合输入法对藏文组合字符的编码处理方式：辅音部分从上到下编码，并且第一个辅音为占位字符，其余辅音以及元音都是不占位字符。

为了讨论方便将字符中的辅音从上到下依次称为第一层辅音、第二层辅音、第三层辅音。第一层辅音可能是上加辅音也可能是基本辅音：对于没有上加辅音的字符，第一层辅音就是基本字符；而有上加辅音的字符，第一层辅音是上加辅音。第二层辅音也有两种可能：有上加辅音时基本辅音为第二层而没有上加辅音时下加辅音为第二层。当组合字符同时有上加辅音和有下加辅音时才有三层辅音，因此第三层只有一种可能，那就是字符的下加辅音。字符中各个字母的书写顺序如图 4 所示。

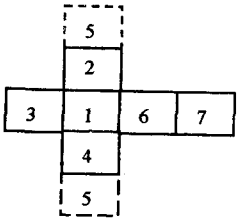


图 3 各个部件决定字的顺序时的重要性

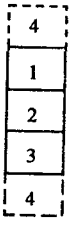


图 4 字符中各个字母的书写顺序

定义藏文字符结构 TibetanCharacter 如下：

```
typedef struct TibetanCharacter
{  TCHAR firstConsonant;//第一层辅音，可能是上加辅音也可能是基本辅音
   TCHAR secondConsonant;//第二层辅音，可能是基本辅音也可能是下加辅音
   TCHAR thirdConsonant;//第三层辅音下加辅音
   TCHAR vowel; //元音
} TibetanCharacter;
```

比较两个组合字符时，先比较辅音部分后比较元音：如果两个字符的前 $n(n=0,1,2)$ 层辅音都相同则字符顺序由第 $n+1$ 层辅音决定；如果各层辅音都相同则由元音决定。即依次比较两个字符对应的结构中成员 firstConsonant、secondConsonant、thirdConsonant 和 vowel 的值，如果两个字符对应的结构中各个成员都相同则表示它们是相同字符，否则由第一个不同的成员来决定字符的顺序。

这种排序的缺点是带上加辅音的组合字符的顺序与字典中的顺序差别较大，例如：字典中字符 的顺序依次由 (占位字符)、(不占位)、(不占位)和 (不占位)决定，应该归到 部[6]；但在这种排序中字符 顺序却依次由 (占位)、(不占位)、(不占位)和 (不占位)决定，应该将其归到 部。对于其它有上加辅音的组合字符情况类似。

4.2 引入排序码的排序方式

为使组合字符的顺序与字典顺序一致，引入排序码，使得任何一个字符由它的排序码来决定顺序。因此，需要建立组合字符与排序码之间的联系。为了将组合字符对应的编码序列与它的排序码也就是组合字符在扩展集中的 Unicode 编码建立一一对应关系，建立了如下的表格(见表 1)：

表 1 建立字符的编码序列与排序码之间的一一对应关系

组合字符对应的的编码序列				排序码
第一层辅音	第二层辅音	第三层辅音	元 音	
0F68	NULL	NULL	0F72	F300
	…			…
0F66	0F90	0FB2	0F74	F33F
…				…
0F67	NULL	NULL	0FAD	F8FF

这样仅用排序码就可以确定字符的顺序而不必考虑构成字符的字母，因此在表示藏文字符的中添加一个新的成员 sortCode，用来存放排序码。

```
typedef struct TibetanCharacter
{  TCHAR firstConsonant;//第一层辅音，可能是上加辅音也可能是基本辅音
   TCHAR secondConsonant;//第二层辅音，可能是基本辅音也可能是下加辅音
   TCHAR thirdConsonant;//第三层辅音下加辅音
   TCHAR vowel;//元音
   TCHAR sortCode;//排序码
} TibetanCharacter;
```

(下转第 74 页)

表 2 SAttribute 表

ID	ParentID	AttributeName	DataType	NodeType	Use	Represent	Represent_value
1	2	No	int	null	required	null	null
2	1	name	string	null	required	null	null

表 4 Element_value 表

ID	Value
3	Tom
4	M
5	1982-9-09-04

表 5 Attribute 表

ID	ParentID	AttributeName	Value
1	1	name	CS
2	2	No	200520000

3 结束语

主要讨论了如何在关系数据库中存储 XML Schema 和 XML 文档。

通过简化 XML Schema 文档，将其定义的关于 XML 文档元素和属性的信息保存到关系数据库中，可以用来验证 XML 文档和数据库中数据的正确性。将 XML 文档元素和属性的信息保存到 Element_Info、Element_value 和 Attribute 三个表中，可以实现 XML 文档到关系数据库的存储无损映射，有利于减少了数据冗余。对于包含混合内容的 XML 文档，可以建立 MixedTable 保存 XML 文档的文本内容。

参考文献：

[1] W3C. Extensible Markup Language (XML)1.0 (Fourth Edition) [EB/OL]. 2006. http://www.w3.org/TR/REC-xml/.

[2] Florescu D, Kossmann D. A Performance Evaluation of Alternative Mapping Schemes for Storing XML Data in a Relational Database[R]. France: INRIA, 1999.

[3] Florescu D, Kossmann D. Storing and Querying XML Data using an RDMBS[J]. IEEE Data Engineering Bulletin, 1999, 22(3):27-34.

[4] Deutsch A, Fernandez M, Suciu D. Storing Semistructured Data with STORED[C]// In Proc of ACM SIGMOD Conf. on Management of Data. Philadephia, PA, USA: ACM Press, 1999:431-442.

[5] Shanmugasundaram J, Tufte K. Relational Databases for Querying XML Documents Limitations and Opportunities[C]// Proceedings of the 25th VLDB Conference. Edinburgh, Scotland, UK: Morgan Kaufmann Publishers, 1999.

[6] W3C. XML Schema 1.1 Part 1: Structures[EB/OL]. 2006. http://www.w3.org/TR/xmlschema11-1/.

(上接第 70 页)

排序过程为：首先，将组合字符中各个字母的 Unicode 编码读到结构 TibetanCharacter 的成员 firstConsonant、secondConsonant、thirdConsonant 和 vowel 中；其次，从表中查出该字符对应的排序码并且将排序码存放到结构的成员 sortCode 中；最后，根据两个字符的排序码的大小来决定两个字符的顺序[4]。

5 实验结果

由于在完全由字母决定字符顺序的方式中，主要是有上加辅音的组合字符顺序与字典中的顺序差别较大，而引入排序码后这种差别就不存在了。表 2 是实验结果(文中只列举了几个有上加辅音的组合字符的排序结果)。从表中可以看出引入排序码后组合字符的顺序与它的字典顺序一致了。

综上所述，引入排序码进行排序的优点有：

1)对于组合字符，无论采取动态组合编码方式还是采取预组合编码方式，排序的结果都是一样的；

2)排序结果与组合字符在字典中的顺序是一致的。

表 2 两种排序结果对比

组合字符	利用字母排序	利用排序码排序	字典顺序
	属于  部	属于  部	属于部
	属于  部	属于  部	属于部
	属于  部	属于  部	属于部

参考文献：

[1] 黄福员，聂瑞华. 冒泡排序算法的改进[J]. 微机发展(现名：计算机技术与发展)，2003，13(11)：29-30.

[2] 张 磊. 基于链表的对分排序算法及实现[J]. 微机发展(现名：计算机技术与发展)，2002，12(2)：56-58.

[3] 张南平. 一种新型单循环排序算法[J]. 微机发展(现名：计算机技术与发展)，2005，15(5)：115-116.

[4] ISO/IEO 10646-1. Tibetan Character Collection[S]. ISO/IEC JTC1/SC2/WG2，1997.

[5] 国家技术监督局. 信息技术 信息交换用藏文编码字符集(基本集)[S]. 北京：中国标准出版社，1998.

[6] 新编藏文字典编写组. 新编藏文字典[M]. 西宁：青海民族出版社，1989.