藏文同元码与基本集相互转换的规则与实现

武光利[1],于洪志[1],柳 春[1,2]
WU Guang-li[1],YU Hong-zhi[1],LIU Chun[1,2]

1.西北民族大学 中国民族语言文字信息技术重点实验室,兰州 730030
2.甘肃中医学院 公共课部,兰州 730000
1.State Key Lab. of National Languages Information Technology,China,Northwest University for Nationalities,Lanzhou 730030,China
2.Department of Public Course,Gansu College of Traditional Chinese Medicine,Lanzhou 730000,China
E-mail:jswgl@xbmu.edu.cn

WU Guang-li,YU Hong-zhi,LIU Chun.Regulars and realization in code transform between Tibetan Tongyuan codes and component sets.Computer Engineering and Applications,2009,45(29):134-136.

Abstract: Nowadays,in the processing course of computer information,the problem of using different codes to stand for the same characters on different characters processing platform,that is to say,the non-compatible of characters processing is a main problem to be settled.Well,in the research of Tibetan information processing,the research of Tibetan codes transforming is a hot point.Most Tibetan texts and websites use the Tongyuan codes while the Vista OS of Microsoft uses component sets.Therefore,in the field of Tibetan information processing,the codes transforming between these two is rather important.This paper mainly talks about transformational technique between Tibetan Tongyuan codes and component sets.The method of splitting Tibetan characters using Latin transliteration is taken.Tiers are taken as the bridge of Tibetan Tongyuan codes character structure and component set character structure,using a set of rules,to accomplish the transform of these two codes.
Key words: Tibetan;Latin transliteration;Tongyuan code;component set;code transform

摘 要:在当今的计算机信息处理过程中,不同文字处理平台上相同字符的不同编码问题,即文字处理的不兼容,是一个亟待解决的重要问题。而在藏文信息处理的研究中,藏文的编码转换也是一个研究热点。藏文的文本、网站大多采用同元编码方式,而微软的 Vista 操作系统采用的是基本集的编码方式,所以两种编码的转换在藏文信息处理领域是非常重要的。主要介绍了藏文同元编码与基本集的相互转换技术,采用了将藏文按照拉丁转写拆分的方法,利用层数作为藏文同元编码字符结构与基本集编码字符结构的桥梁,通过一系列规则,实现了两种编码的相互转换。
关键词:藏文;拉丁转写;同元编码;基本集;编码转换
DOI:10.3778/j.issn.1002-8331.2009.29.040 **文章编号**:1002-8331(2009)29-0134-03 **文献标识码**:A **中图分类号**:TP391

1 引言

在计算机的信息处理过程中,经常会遇到在不同的操作平台上,相同的字符采用不同的编码方案。特别是有些少数民族文字,许多地区为了本民族文字处理的需要,自定编码方案。这就造成了不同文字处理平台上相同字符的不同编码问题,即文字处理的不兼容问题。藏文的文本、网站大多采用的同元编码方式,微软的 Vista 操作系统采用的是基本集[1-2]的编码方式,所以两种编码的相互的转换在藏文信息处理中是非常重要的。

在进行编码转换时,首先将同元编码按照拉丁转写拆分的方法[3-4],拆成七元组模型。目前藏文拉丁转写的方法有:字转写法、声韵母转写法、字丁转写法[5-6]、字母转写法。该文采用的是字丁转写的方法。经过对 13 万藏文词汇的统计分析,现代藏文中出现的字丁约有 600 个。此方法首先需要对所有藏文字丁建立拉丁对照表,字丁本身带元音符号的,根据实际的元音符号进行拉丁转写,不带元音符号(内含"a"元音)的只转字丁辅音不转元音"a",详细过程见图 1。

藏文到拉丁字母的转写的规则

(1)首先进行藏文基字丁的确定。

(2)单音节藏文最多能包含 4 个字丁,对藏文进行字丁分解:藏文=前加字字丁+基字字丁+后加字字丁+再后加字字丁,其中除了基字字丁不能为空外,其余 3 个字丁都可以为空。

(3)分别对单音节藏文的 4 个字丁查阅藏文字丁-拉丁对照表(如表 1 所示),把各自的藏文字丁转换成拉丁字母串,基字字丁如果不带元音符号,则直接在其转换的拉丁字母串后加"a"。

基金项目:国家高技术研究发展计划(863)(the National High-Tech Research and Development Plan of China under Grant No.AA2006010101)。
作者简介:武光利(1981-),博士生,主要研究方向:自然语言处理、语音信息处理;于洪志(1947-),教授,博士生导师,主要研究方向:自然语言处理、语音信息处理;柳春(1976-),博士生,主要研究方向:语音信息处理。
收稿日期:2008-06-02 **修回日期**:2008-07-21

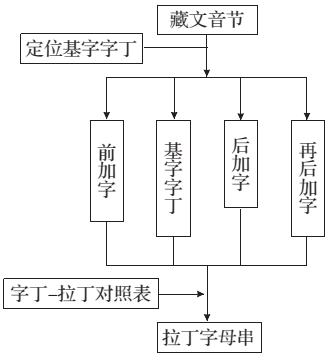

图 1　藏文字丁-拉丁转写流程图

表 1　藏文字丁-拉丁对照表

序号	藏文字丁	拉丁转写
1	總	ka
2	賭	ki
3	賁	ku
4	膕	sma
5	臺	rle
6	賢	ko
7	暢	smro

拉丁字母串=前加字拉丁转写+基字字丁+(“a”)+后加字拉丁转写+再后加字拉丁转写。按照上述规则可将藏文按照拉丁字母的转写方法进行拆分。

2　藏文术语定义和基本集字符的拆分原理

2.1　藏文相关术语的定义

藏文基本集进行信息处理时是由藏文基本字母与组合用字符构成内码复合序列，按规则迭加生成藏文整字，提供给用户使用。

藏文基本字母：由藏文的 30 个基字，5 个反体字，6 个梵音藏文迭字，共 41 个藏文字符组成。

组合用字符：是指在藏文编码字符集中，一个已标识的子集中的一种结构要素，用于与其前导的非组合用图形字符相结合，或者与一个非组合用图形字符为前导的组合用字符序列相结合[7]。

复合序列：由一个非组合用字符后随一个或多个组合用字符所组成的图形字符的序列，用于复合序列的图形符号一般由该序列中每一个字符图形符号的组合而构成的。

构字规则：

(1)藏文基本字母是复合序列的先导，标志着一个复合序列的结束，后一个复合序列的开始。

(2)当一个藏文整字由一个藏文基本字母与多重组合用字符组成时，名称中的组合用字符按下列顺序排列：

各组合用字符均相对于藏文基本字母定位，在该基本字母上面的组合用字符按向上的顺序排列，后随以基本字母下面的组合用字符，按向下的顺序排列。

2.2　藏文基本集字符的拆分原理：

从理论上讲，拼音文字的编码字符集是最容易实现的，而且，能够成为全字符集。比如英文的 26 个字母，无非有大小写之分。但是，作为拼音文字的藏文，却同时拼型，客观上存在上下迭加纵向组合的不确定性。

在确定藏文碼码字符集时，关键要进行统计意义的藏文拆分工作。

拆分，就是从藏文整字中，将排列在各层次中的藏文组合用字符分离出来，对它们的信息结构人为地重新定义。

从藏文信息结构的角度来考虑，拆分，必须保证藏文组合用字符的完整性，保证拆分后的构件能够形成一切事实存在的藏文整字。通过拆分，藏文的二维形式即平面形式已不复存在，所有的藏文编码字符以变形显现形式出现，藏文整字的信息消失了。但是，拆分形成的构件，构形对称，蕴藏着丰富的信息。

因此，藏文基本集字符的拆分原理[8]是将藏文整字细化拆分成能够充分描述全部藏文形状特征的构形稳定，数量固定，相互独立而又必要的最大的字符单位。它不仅应包含藏文基字，还应包含藏文基字与上下加变形显现和元音符、语音符的组合形式。以尽可能小的编码空间，杜绝藏文集外字的困扰。

3　结构体的定义

藏文字母信息表结构

```
struct_TIBETAN_LETTER
{
   WORD  tCode;          //同元码
   TCHAR  szLetter[3];    //字符
   WORD  bCode;          //基本集码
   TCHAR  szLatin[5];    //拉丁转写
}TIBETAN_LETTER;
```

同元藏文字符信息表结构

```
struct  _TIBETWORD
{
   TCHAR  szTibet[3];   //字符
   WORD  word;          //内码
   TCHAR  szBase[5];      //藏文基字[拉丁表示]
   TCHAR  szUp;           //上加字[拉丁表示]
   TCHAR  szDn;           //下加字[拉丁表示]
   TCHAR  szVowel;        //元音[拉丁表示]
   int  count;            //层数
   int  iAttribute;       //属性
}TIBETWORD;
```

基本集信息结构

```
struct_BASESET_INFOR
{
   WORD  wFirst;          //前导字符
   WORD  wCombine1;       //基字存放的位置，如果值为零，则表示基字为前导字符
   WORD  wCombine2;       //下加字
   WORD  wCombine3;       //元音
   int  iLayer;           //层数
}BASESET_INFOR;
```

4　同元编码转基本集编码的相互转换

4.1　同元编码转基本集的规则

判断同元藏文字符信息表结构中的层数，如果层数等于1，则直接进行编码转换。

如果层数大于 1，则规则的步骤如下：

(1)判断藏文是否有上加字，如果有上加字，则上加字的码值作为基本集信息表前导字符的值，如果没有上加字，则将基字的码值作为前导字符的值，并将第一组合字符的值设为 0。

(2)再判断是否有下加字,如果有下加字,则将下加字的码值作为第二组合字符的值,如果没有,则将第二组合字符的值设为0。

(3)再判断是否有元音,如果有元音,则将元音的码值作为第三组合字符的值,如果没有元音,则第三组合字符的值设为0。

4.2 同元编码转基本集的算法实现

算法原型:

BOOL TongToBaseset(WORD wt,BASESET_INFOR*pbi)。

输入参数:

wt 是输入的同元编码。

输出结果:

基本集信息结构的指针。

处理流程:

预处理:

首先根据藏文转拉丁转写算法将藏文转写成拉丁字符,按上述的同元藏文字符信息表结构构成藏文的拉丁转写对照表。

根据藏文字母信息表结构构建前导字符表、组合字符表、下加字字符表、元音字符表。

转换:

首先判断是否是藏文,如果是藏文,根据输入的同元编码,则采用二分查找算法,在藏文的拉丁转写对照表中查找藏文的拉丁转写,如果是1层,则直接查前导字符表得到其对应的基本集码;如果大于1层,则根据上述同元编码转基本集的规则,判断各种不同的情况,分别查找前导字符表、组合字符表、下加字字符表、元音字符表,并判断层数,分别得到其对应的码值,写入基本集信息结构,然后按照顺序依次将基本集码写出。

4.3 基本集转同元编码的规则

根据基本集结构的信息中 *iLayer* 判断其层数

(1)如果 *iLayer*=1,则直接查表得到其同元码值。

(2)如果 *iLayer*=2,其规则如下:

如果第二层是元音,则第一层是基字,第二层是元音;如果第二层不是元音,再判断第二层是不是下加字,如果第二层是下加字,则第一层是基字,第二层是下加字,如果第二层不是下加字,则第一层是上加字,第二层是基字。

(3)如果 *iLayer*=3,其规则如下:

如果第三层是元音,再判断第二层是不是下加字,如果是下加字,则第一层是基字,第二层是下加字,第三层是元音;如果第二层不是下加字,则第一层是上加字,第二层是基字,第三层是元音。如果第三层不是元音,则第一层是上加字,第二层是基字,第三层是下加字。

(4)如果 *iLayer*=4,其规则如下:

则第一层是上加字,第二层是元音,第三层是下加字,第四层是元音。

4.4 基本集编码转同元编码的算法实现

算法原型:

LPTSTR BasesetToTong(LPTSTR lpFrom,DWORD len);

int AnalyseBasesetSequence(CXWordArray*parr,WORD & code);

输入参数:

lpFrom 是藏文基本集编码(UNICODE)的宽字符流,len 是字符流长度。

*parr 是存放藏文基本集编码(UNICODE)的宽字符流的链表指针。

输出结果:

对应的同元码值。

处理流程:

根据上述所提的基本集编码转同元编码的规则,通过AnalyseBasesetSequence 函数进行分析,判断各种不同的情况,分别查找前导字符表、组合字符表、下加字字符表、元音字符表,得到各自对应的拉丁转写,再根据拉丁转写查找同元藏文字符信息表,得到藏文对应的同元编码。

5 实验结果

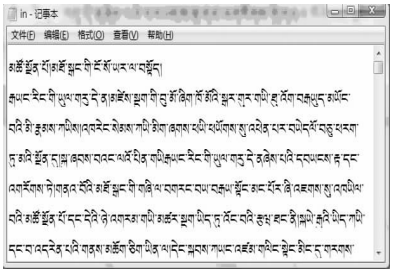

图2 基本集编码下显示的藏文

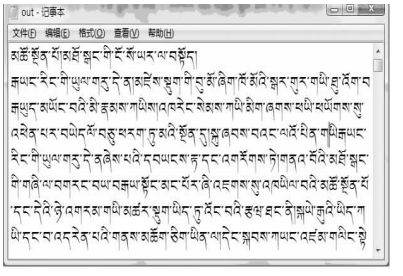

图3 转换后的同元编码下显示的藏文

6 结束语

通过对藏文同元编码与基本集编码方案的分析比较,提出了同元编码与基本集编码相互转换的规则,解决了两种编码相互转换的问题,并且软件的开发采用 COM 技术,便于算法的移植。

参考文献:

[1] 陈丽娜.藏文拉丁转写的研究与实现[J].计算机工程与设计,2006(1).

[2] 江荻.藏文的拉丁字母转写方法—兼论藏文语料的计算机转写处理[J].民族语文,2006(1).

[3] 申晓亭.少数民族文字拉丁转写的意义与方案[C]//民族语言文字信息技术研究—第十一届全国民族语言文字信息学术研讨会论文集,2007.

[4] 牛飞,德熙嘉措.藏文拉丁转写的概貌[J].青海师范大学学报:自然科学版,2006(3).

[5] 于洪志.计算机藏文编码概述[J].西北民族大学学报,1999(3).

[6] 彭寿全,黄可,张义刚.藏文综合编码方案的研究与实现[J].中文信息学报,1996(4).

[7] 扎西次仁.国际标准藏文计算机编码字符集的研究[J].中国藏学,1995(2).

[8] 国家技术监督局.GB16959-1997 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,1997.