

[综述]

文章编号:1003-0077(2006)02-0087-07

国际化文字处理综述*

芮建武, 吴健, 孙玉芳

(中国科学院 软件研究所 开放系统与中文信息处理中心, 北京 100080)

摘要:计算机与不同用户的交互通常必须实现通过多种文字信息的输入/输出以实现,因此操作系统对多种文字的支持程度是其功能性的一个衡量标准。各种文字特征的巨大差异导致现代操作系统的文字处理实现非常复杂。本文总结了操作系统文字处理的范围与内容,包括文本输入与存储,文本处理以及用户交互处理;归纳了通用的文字处理模型和可能采取的技术途径及其优缺点;分析了常用操作系统的文字处理实现;最后展望了文字处理仍面临的挑战。

关键词:计算机应用;中文信息处理;综述文字处理;复杂文字;字体模型

中图分类号:TP319.89 **文献标识码:**A

Survey on International Text Processing

RUI Jian-wu, WU Jian, SUN Yu-fang

(Open System & Chinese Information Processing Center, Institute of Software, Chinese Academy of Sciences, Beijing 100080, China)

Abstract: The implementation of multilingual text I/O is essential for computers to interact with all sorts of users in the world. One of the most important functionalities for a computer is, how and to which extent its operating system supports languages with multi-scripts. Owing to considerable differences amongst scripts, multilingual text processing in an a global operating system is very complicated. In this paper, firstly, the scope and the content of multilingual text processing are defined, including text input, store, processing and interactions in an internationalized manner. Secondly, models for text processing are outlined; several technical solutions are discussed; the pros and cons are listed. Thirdly, technical features of text processing implemented by current operating systems are analyzed. Finally, some challenges in the realm of internationalized text processing are presented.

Key words: computer application; Chinese information processing; overview; text processing; complex script; font model

1 文字处理范围

文字的信息处理过程需遵循相应的规范。广义的规范指一定语言社团的成员在交际中都乐于遵守使用的一套代码系统。它分为自发形式和自觉(或规定的)形式两种。其中自觉的规范是语言权威机构和语言学家对自发规范所形成的言语材料(包括文字资料)进行记录、描写、

* 收稿日期:2005-01-27 定稿日期:2005-06-01

基金项目:国家863计划资助项目(2003AA1Z2110);中国科学院知识创新工程资助项目(KGCX2-SW-504)

作者简介:芮建武(1972—),男,博士,主要研究领域为操作系统与中国民族文字信息处理。

评价、抉择后形成的一套标准文字形式。狭义的规范就表示这种自觉的规范,称为语文规范,它包括语音规范与文字规范^[1]。

用于信息处理的语文规范和通用语文规范有同有异。其中的文字规范可区分称为文字处理规范;语音规范可区分称为语音处理规范。文字处理规范讨论的基本单位为“字”、“音节”或“词”这一级别,包括字符集、键盘布局、字型规范等,这些规范通常按国家或国际标准的形式出现。文字处理还涉及到:(1)文本信息的规范,包括字符标准形式、字符序列中字符顺序、标准形式与兼容形式的相互转化等;(2)文本显现的规范,包括书写方式、字形变形、多种文字混合文本显现策略等。它们由语文规范决定。此外,文字处理还涉及非规范语文实践,例如非规范语文实践的示例等。因此,操作系统文字处理是语文规范、文字处理规范与非规范语文实践集合的子集。

2 文字处理的内容

世界文字种类繁多,其书写和字符组合结构的差异导致文字处理较为复杂。文字大致分为形意文字、意音文字、音节文字、辅音字母文字、音素字母文字等。辅音字母文字包括阿拉马字母文字(包括阿拉伯字母、蒙古字母、希伯来字母、亚美尼亚字母、乔治亚字母文字等)和印度字母文字。音素字母文字包括希腊字母、斯拉夫字母和拉丁字母文字等^[2]。各种文字都有不同于其他文字的处理特征,这些特征可总结为方向特征、字符形态变化特征、字母或符号的组合特征和排版特征。国际化文字处理要求:(1)以符合文字处理特征及用户习惯的方式显示某种文字的文本;(2)能够显示多种文字的混合文本;(3)能够输入多种文字的文本。基于 Unicode 标准^[3],本文重点来讨论国际化文字处理。

2.1 基本概念

2.1.1 字符、字形与字形簇

文字处理中区分字符、字形与字形簇非常重要。

ISO/IEC 10646 标准为每个字符指定一个数值、字符名称和示例性图形。字符是信息交换与处理的基础,这些处理包括:识别、排序、变形、格式化、归档、显现等。ISO/IEC 9541-1:1991^[4]标准把字形看作显现过程中的抽象元素,字形由字形标识符、字形描述和示例性图形来认定。标准并不定义字形的精确使用与具体外形,相同字形在不同字体中的外形可能存在较大差异,因此,字形可区分为抽象字形与实际字形。字符与字形的特征与相互关系描述如下:(1)字符体现了意义或声音的差别,字符没有内在的外形;(2)字形体现字符外形的差别,字形没有内在的意义;(3)一个或多个字符可以用一个或多个字形表示(也可能没有字形),其表示依赖于文本上下文,其映射关系表示为 $M:N$, $M>0$, $N=0$

用户通常按其认定的“字符”来完成信息处理,即以字形簇(grapheme cluster)为基本单位。Unicode 标准把字形簇定义为一个或多个字符组成的、用户默认的文字处理单元^[5],包括:组合字符序列、连体字(digraph)、带重音或者修饰符号的字符序列等。字形簇边界是文本选择、方向键移动、退格键操作、排序、规则表达式、确定文本中用户角度的字符位置与计数等操作的基础。其特点是:(1)词、行、句的边界不会出现在字形簇内部;(2)字形簇与键盘输入之间没有一一对应关系;(3)键盘按键操作可能涉及一个字形簇、其中一部分或多个字形簇序列。

如果一种文字的字符、字形与字形簇之间存在一一对应关系,则把这种文字称为简单文字;否则称为复杂文字。文字处理需要正确分清字符、字形与字形簇之间的关系。

2.1.2 源文本、显示文本与文本布局

以字符编码存储的文本通常以用户出声阅读的顺序排列,这种线性顺序的文本称为源文本。文本显示时根据文字的拼写法(orthography)来拼写字符,形成的文本称为显示文本。例如藏文文本按线性顺序存储,在显示时需要根据藏文正字法从左向右书写和从上向下叠加拼写相结合,从而得到符合藏文正字法的显示文本。这些显示文本由与定位信息相关的字形序列组成。因此,源文本转化为字形序列的过程中将使用与文字相关的处理规则,这一过程称为文本布局。显然,语文规范与文字处理规范共同决定这些处理规则。

2.2 文本输入与存储

文字处理中的数据对象必然基于某一字符集并使用某种编码。字符集中为每个字符指定一个代码点。每个字符对应于语文规范中的一个字母、音节或字。计算机中总是基于字符代码来实现文本输入与存储。

2.3 单一文字的处理

任何文字具有其自身的特征,包括:(1)方向特征;(2)字符形态变化;(3)字母或符号的组合;(4)排版特征。

方向特征。文字的固有属性使得它只能“这样”书写而不能“那样”书写。例如,阿拉伯字母文字通常从右向左书写,各行自上而下排列;传统蒙古文自上而下书写,各列从左向右排列;古代汉字自上而下书写,各列从右向左排列,等等。闪族文字(Semitic)如阿拉伯文等从右到左书写,而阿拉伯数字从左到右书写,从而形成双向文字。双向性可能形成嵌套,需要特定的处理算法来实现显示文本的正确顺序^[6,7]。

字形形态变化包括:(1)字符上下文变形形式。例如,阿拉伯文、维吾尔文、蒙古文等文字的字符都存在与其所在词中位置相关的字形变形。(2)字形的重新排序与字形分裂。例如印地文(Indic)、梵文等中的元音符号等。

字母或符号的组合。不同文字在字母或符号组合特征方面的差异包括:(1)连体字(ligature)。连体字指多个字形以某种方式布局形成一个复杂字形,典型的例子有韩文中的拼音字母(Hangul)及其预组合形式(conjoining jamo);(2)藏文按纵向叠加组合、水平方向拼写的方式形成音节;(3)重音符号、变音符号等组合字符(combining characters)与基本字符的结合。目前基本字符与组合字符的显示可采用通用的方法^[8]。存在叠加式注音符号的文字需要特殊的处理,例如,越南文中使用大量的变音符号和其它符号,用以修改字符的声音或意义等。

排版特征。文字处理涉及到文本显示时的字形定位:(1)文本字距调整,包括整行字距调整(justification)、特定字形对调整(kerning)以及特定文本片段(run)的调整(tracking);(2)文本对齐;(3)字形移位(shifting),例如字符“-”与大写字母和小写字母连着书写时将有不同的位置;(4)草书文字的字形连接。有的文字以词为单位来调整距离,有的则以字为单位,取决于文字拼写法。多数草书文字需要在字形之间插入连接标记符号以形成更好的排版效果。例如维吾尔文使用直杠符号填入字母之间,使文本充满版面里的行;蒙古文使用字符NIRUGU(U+180A)把上下文字符连接为一体,用于视觉上拉长字符。

2.4 多种文字混合文本处理

某些文字使用多种书写系统,同时,操作系统需要同时处理多种文字。这样,操作系统需要满足两方面的要求:(1)操作系统应满足任何一种文字所使用书写系统的特征,包括方向特征、字形形态变化、字母或符号的组合以及排版特征;(2)采用合理的策略来统一处理所有文字文本。通常需要一个全局文本方向,所有文字的文本布局都要与全局文本方向相协调;每种文字存在其固有的文本阅读方向,二者结合形成混合文本处理策略。

2.5 文字处理的交互性

文本交互处理是文本编辑的基础。包括:(1)插字符号(caret)定位与移动行为。插字符号表示用户编辑文本时的字符插入位置。当使用方向键时,此插字符号应移动相应位置以使用户选择插入点。这涉及到两个问题:(a)在不同全局布局方向下方向键的物理意义不同。例如,左方向键在水平布局下表示当前行向左移动一个字符单元位置;在垂直布局情况下表示移动到前一行(如果有的话)的适当位置。(b)插字符号的移动单元。例如,在汉字文本中以字为单元移动;在藏文文本中,其移动单元可能包括几个字符。(2)以字形簇为单位管理文本选择与编辑。(3)文本中插字符号的显示,如斜体插字符号、双向文本中不同形式的插字符号等。显然,最重要的问题是处理字形簇与其源文本的映射关系。

3 处理模型与技术途径

3.1 处理模型

早期文字处理仅涉及简单文字,具有最简单的模型。它基于如下假设:(1)在键盘按键、存储的字符、外设输出的字形之间有一一对应关系;(2)用于文本显示所需的所有信息来源于纯文本。因此,需要扩展文字处理模型:使用键盘处理模块来产生需要的字符;使用智能显现技术来控制字形的复杂处理。

ISO/IEC TR 15285:1998^[9]区分字符与字形,定义了字符与字形的操作模型。操作模型把对信息的处理分为两个领域:一是内容的处理,这类处理基于字符;二是显现的处理,这类处理基于字形。前者包括查找、排序、拼写检查、语法检查等操作,它们独立于字体资源。后者包括格式化、显示、打印等操作,它们依赖于字体资源。二者的关系:(1)基于字符的处理使用独立的、逻辑的信息表示;(2)必要时把这种逻辑表示转换为其他表示形式;(3)基于字形的处理通常使用一种中间形式,称为独立于设备的文档。它表示用来在某一媒介上显现文档的所有信息,包括字体识别符、字形识别符、坐标位置、对字体资源的引用等信息。

ISO/IEC TR 15285 使用三种字体模型:编码字体模型、字体资源模型和智能字体模型。

编码字体模型使用字符代码来索引字体中的字形规格信息(metrics)和字形轮廓描述信息,同时使用字符代码确定字体中独立的字形标识符。这要求:(1)对所支持的不同字汇表(repertoire),需要提供不同的字体及不同的字符—字形映射表。(2)字符代码到字体中的信息为一一映射。其缺点是:不能实现复杂的字形处理需求。

字体资源模型可定义独立于字符集与文字处理过程的字体资源,允许在文本输出的布局与显现两个过程之间的信息交换。它的特征是:(1)字体资源中的字形规格信息和字形轮廓描述信息根据字形识别符来获得。(2)字体资源中存在字符与字形识别符之间的映射。(3)文本布局过程使用预定义的字符—字形识别符映射表把字符代码序列映射为字形序列。其方法是在格式化文档中使用字形索引值来表示字符,同时建立字形索引映射表。此字形索引映射表把字形索引值映射为字形识别符,此映射可能保持不变:(a)面向某一特定字体;(b)多个字体之间共享;(c)标准映射。(4)字形索引映射表与字体资源共同形成特定输出过程的索引字体。此模型的优点在于可支持大字符集。

智能字体模型来源于Unicode标准。智能字体中增加了附加描述信息,这些描述信息主要实现对字形的特殊处理,包括字形的替换与复杂字形的定位。此外,这些描述信息实际上包含了对某种文字的文字处理规则。通过这些处理规则,文本布局与显现过程可直接正确地显示“纯文本”。

3.2 技术途径

实现复杂的文字处理需要文字处理引擎和相应字体技术相结合。

所有字体都包括字形 ID、字形轮廓描述和字符与字形 ID 之间的映射。现代字体的特点是:(1)字体是字形集合,每个字形具有一个字形 ID;(2)同一字体中一个字符可能对应多个字形;(3)字体所支持的文字需求不同,因而一个字体中可含有多个字符—字形映射;(4)不同字体的相同字符可能有不同字形 ID;(5)字体可以包含文字处理信息,如字形替换与定位等。因此,字体有时可视为一个“字形处理系统”。文字处理引擎根据文字处理特征实现文本的正确显现,通常包含许多与文字(甚至语言)相关的规则。这些规则包括:(1)文本边界的划分;(2)文字相关的断行行为(break lines);(3)双向文本的嵌套处理;(4)文本布局与页布局。

不同文字处理技术途径具有不同文本处理能力。目前主要有以下技术途径:

(1)简单字体处理,包括基于 TrueType 字体的途径以及 LaTeX + Metafont 技术途径。其特点是:(a)字体中字符与字形之间存在一一映射;(b)文字特征完全由文字处理引擎来控制。

(2)半智能字体处理,包括以 OpenType 字体为基础的技术途径。OpenType 字体基于 TrueType 字体格式,增加相应的表来实现字形替换、定位、基线对齐与文本齐行等面向字形的特征^[10~12]。其特点是:(a)OpenType 字体使用 GDEF/GSUB/GPOS 表来实现字形替换与字形定位;(b)缺乏文字相关行为的语义,需要文字处理引擎来实现文字相关处理。

(3)智能字体处理,主要指 Apple 公司的 AAT/ATSUI 处理途径。AAT 字体也是 TrueType 字体的扩展:(a)提供相应机制来使用或调入不同于 TrueType 字体格式的字形光栅化模块;(b)使用附加表来实现布局信息的控制,其中表“mort”定义了一系列的字形变形(metamorphosis)行为,可根据最终用户的需要打开/关闭。“mort”表本身是基于状态表的自包含系统,从字符到字形的映射所需要的所有信息包含于字体中,所以此途径能够自动完成整个映射过程。其特点是:(a)具有良好的可扩展性;(b)仅 Mac OS 系统上可使用;(c)具有专利权。

(4)用户定制型处理,包括基于 Windows 平台的 Graphite 技术途径等。Graphite 使用一种基于规则的编程语言 GDL 来指定面向文字的处理特征。根据用户定义的 GDL 规则,TrueType 字体可被编译为一种具有 Graphite 扩展表的字体,此字体将被 Graphite 文字处理引擎用来实现特定的文字处理特征。其优点是:(a)具有良好的可扩展性;(b)文字处理行为可定制。

4 文字处理实现现状

目前存在多种应用广泛的文字处理系统实现,主要包括:(1)Uniscribe 是 Windows 操作系统使用的文字处理引擎。它基于 OpenType 字体技术,其特点是:(a)处理完整的书写系统行为;(b)与文字相关的信息硬编码(hard-coded)在 Uniscribe 中,可扩展性较差;(c)文字特点复杂或市场份额较小的文字未被考虑。(2)Mac OS 基于 TrueType 规范相继使用 QuickDraw、AAT 和 ATSUI 实现高级文本输出^[13]。(3)Solaris 采用 STSF 体系结构^[14],使用 UMLE^[15]进行复杂文本输出,提供了 Open Group 制定的可移植布局服务^[6]的 API 接口。(4)Graphite 是 SIL 公司基于 Windows 平台上用于复杂文字的可扩展文本输出引擎^[16]。(5)Tex/LaTeX/Omega 文字排版系统具有如下特点:(a)在处理重音符号方面有很强的扩展能力,能够把重音符号扩展到两个甚至三个字母之上。(b)使用 Metafont 编程语言提供了参数化曲线的扩展能力。它输入描述字体的文本文件,并把它生成为位图字体。与 AAT 和 OpenType 相比,Metafont 提供的附加数据有限,它不支持连体字、字形替换等操作。(6)IBM 公司发起的国际化组件 ICU 项目提供了支持多平台的 Unicode 字符处理例程,实现了用于文字处理的 ICU Layout Engine^[17]。(7)开源项目

FreeType 是目前使用非常广泛的字形光栅化器。FontConfig 实现了系统级的字体匹配解决方案。通过使用 FreeType 和 FontConfig, XFree86 进一步扩展 XRE (X Render Extension), 形成 Xft^[18]。在 Xft 的基础上, GNOME 使用 Pango 来实现国际化文字处理。KDE 使用 Qt 的文字处理引擎。(8) OpenI18N 下属的 ml7n-lib 子组提供了 ml7n-lib 库来实现复杂文字处理。

5 文字处理展望

文字发展不仅有其优胜劣汰的规律,还与文化的传播相联系。随着文化与信息传播的全球化,国际化文字处理面临新的复杂性,主要包括:(1)文字处理范围逐步扩大;(2)复杂文字处理途径需要逐步完善;(3)多种文字混合处理要求统一的文字处理框架;(4)文字处理需要逐步实现标准化。

据统计,Unicode 标准 3.2 版(2002 年发布)收录了 52 种文字,已经查明仍需要指定代码空间的文字至少还有 96 种,其中近三分之一是活文字^[19]。近两年新收录了其它的部分文字,包括中国提交编码方案的新傣文(New Tai Lue)等。中国现行使用的文字,包括拉丁字母、阿拉伯字母、回鹘字母、印度字母、巴利字母、斯拉夫字母和自创字母文字等共计 50 种^[20]。此外,还有一些未能通行的文字以及古文字^[21]。收录到 Unicode 标准 4.0 版(或 ISO/IEC 10646:2003)的中国文字包括汉文、蒙古文(包括传统蒙文、托忒蒙文、锡伯文、满文)、藏文、维吾尔文(哈萨克文、柯尔克孜文)、彝文、朝鲜文、傣文。这些文字除汉文外,在计算机系统中仅实现字处理软件级别的支持,有的仍在使用自定义编码。由于文字类型多样、书写方式齐全、文字特征差异明显以及音节或字(词)的字符组合结构复杂等特点,中国民族文字的计算机处理实际上面临很大的挑战。

复杂文字的文字处理特征差异明显,目前 OpenType 字体技术及相应文字处理引擎的技术途径存在字体资源的可交换性差、文字处理平台不通用等问题。由于文字处理很强的复杂性、软件模块的高耦合等特点,提供新的途径实现对文字处理特征的自动支持应该是值得研究的课题。

现代应用需要多种文字的混合处理。目前文字处理具有以下问题:(1)现有操作系统支持的文字较少,一些文字特征较复杂或市场份额较小的文字,多数未予以考虑;(2)文字处理体系结构不完整,其原因来源于“渐进性与构造性支持”;(3)各种系统的文字处理逻辑不一致。因此,需要从宏观的角度来考虑所有文字的需求。因为文字处理涉及相关文字的专家知识,所有文字的支持都由相同研究与开发小组来完成往往是不可能的工作,所以,统一的文字处理框架应该既能适应所有文字的需求,又具有很强的可扩展性:(1)抽象出所有文字的共性,形成特定的文字处理特征;(2)区别文字相关的处理和与文字无关的处理,形成通用的软件结构;(3)为文字相关的处理模块提供通用的接口。

国际化文字处理的标准化是其逐步完善的标志。目前需要做的工作主要有:(1)提取出文字处理的一些标准过程,例如实现组合字符显示的通用过程(UTN # 2^[8])等;(2)面向特定目标的算法,例如双向算法(UAX # 9^[7])、排序算法(UTS # 10,见 Unicode 标准)等;(3)文字处理标准,包括字符集标准与键盘布局、字型标准的制订与修订等。由于以下原因,文字处理标准可能出现反复修订的现象:(1)对文字固有的字母表、词法与语法的系统整理需要文字学的专家知识,需要多方的协商与合作;(2)出于信息表示的歧义与实现技术的局限,字符集中往往增加一些特定的字符。在中国民族文字信息处理中,文字处理标准的制订是一项非常重大的、国际性的工程^[22],需要充分结合最新的文字处理技术来达到最终目标。

参 考 文 献:

- [1] 戴昭铭. 规范语言学探索[M]. 上海三联书店, 1998 年 7 月.
- [2] 周有光. 比较文字学初探[M]. 北京: 语文出版社, 1998 年 11 月.
- [3] Joan Aliprand, Julie Allen, Joe Becker, Mark Davis, Michael Everson, Asmus Freytag, John Jenkins, Mike Ksar, Rick McGowan, Eric Muller, Lisa Moore, Michel Suignard, Ken Whistler. The Unicode Standard Version 4.0.0[M]. Addison-Wesley, August 2003.
- [4] ISO/IEC JTC 1/SC 34. ISO/IEC 9541 - 1 Information technology-Font information interchange- Part 1: Architecture, 1991.
- [5] The Unicode Consortium. The Unicode Standard Version 4.0.0[M], chapter UAX # 29 Text Boundaries. Available at <http://www.unicode.org/>, 2003.
- [6] The Open Group. CAE Specification³ Portable Layout Services: Context-Dependent and Directional Text. Number C616. Available at <http://www.opengroup.org/onlinepubs/9638399/toc.htm>, February 1997.
- [7] The Unicode Consortium. The Unicode Standard Version 4.0.0[M], chapter UAX # 9 The Bidirectional Algorithm. Available at <http://www.unicode.org/>, 2003.
- [8] The Unicode Consortium. The Unicode Standard Version 4.0.0[M], chapter UTN # 2 A General Method for Rendering Combining Marks. Available at <http://www.unicode.org/>, 2003.
- [9] ISO/IEC JTC 1/SC 2. ISO/IEC TR 15285:1998 Information technology-An operational model for characters and glyphs, 1998.
- [10] 肖明, 胡金柱, 赵慧. 字形技术及 OpenType 字体文件格式研究[J]. 中文信息学报, 1999, 13(6): 53 - 60.
- [11] John H. Jenkins. The Unicode character-glyph model: Case studies [A]. In: 15th International Unicode Conference [C], San Jose, California, USA, September 1999. The Unicode Consortium.
- [12] Joshua Hadley, Dirk Meyer. Script-specific font features [A]. In: Eighteenth International Unicode Conference [C], Hong Kong, April 2001. The Unicode Consortium.
- [13] Apple Computer, Inc.. Inside Mac OS X-Rendering Unicode Text with ATSUI. Available at <http://developer.apple.com/documentation/>, December 2002.
- [14] Alexander Gelfenbain. Standard type service framework-Unicode-based framework for rendering typographically sophisticated text [A]. In: 22th International Unicode Conference [C], San Jose, California, September 2002. The Unicode Consortium.
- [15] Ienup Sung. Universal multi-script layout engine for complex text layout scripts[A]. In: 14th International Unicode Conference, Boston, Massachusetts, USA, March 1999. The Unicode Consortium.
- [16] Sharon Correll. Graphite: An extensible rendering engine for complex writing systems[A]. In: Seventeenth International Unicode Conference (IUC17), San Jose, California, September 2000. The Unicode Consortium.
- [17] Eric Mader. An ICU open source library supporting the display of complex scripts [A]. In: 19th International Unicode Conference [C], San Jose, California, September 2001. The Unicode Consortium.
- [18] Keith Packard. The Xft font library: Architecture and users guide [A]. In: USENIX XFree86 Conference Proceedings [C], Oakland, California, USA, November 2001. The USENIX Association.
- [19] Michael Everson. Leaks in the Unicode pipeline: Script, script, script ... [A]. In: 21th International Unicode Conference [C], Dublin, Ireland, May 2002. The Unicode Consortium.
- [20] 黄行. 我国少数民族文字的类型、功能和规划工作 [A]. 见: 赵丽明、黄国营编, 汉字的应用与传播——99 汉字应用与传播国际学术研讨会论文集 [C]. 北京: 华语教学出版社, 2000 年, 220 - 238.
- [21] 道布, 谭克让. 中国少数民族文字 [M]. 中国藏学出版社, 1991 年.
- [22] 李宇明. 搭建中华字符集大平台 [J]. 中文信息学报, 2003, 17(2): 1 - 6.