G2P Sorting: An Automated Natural Sorting Method for Graphetically Encoded Mongolian | G2P 排序——形码蒙文的自动化自然排序法

Author | 作者: SHEN Yilei | 沈逸磊 (shenyileirob 于 qq.com)

Date | 日期: 2018-04-01

0 Abstract | 提要

G2P sorting, as an automated natural sorting method for graphetically encoded Mongolian,[1] is presented in this document.

本文介绍一种形码蒙文[2]的自动化自然排序法——G2P 排序。

Section 1 introduces Mongolian sorting conventions including unfolded and folded sorting, setting the stage for the upcoming discussion. Section 2 evaluates the feasibility of various automated sorting methods for the phonetic and graphetic encoding approaches, revealing the significance of G2P sorting for the graphetic encoding approach as an automated natural sorting method. Implementation details of G2P sorting are addressed in Section 3, showing its technical possibility, while the results of applying G2P sorting to a dictionary database demonstrate its naturalness in Section 4. Section 5 is a short summary of the key points.

第 1 节介绍了蒙文的排序惯例，包括展开式排序和折叠式排序，为下文的讨论作准备。第 2 节评估了音码形码的各种自动化排序法，展现 G2P 排序作为自动排序法对形码的重大意义。第 3 节描述了 G2P 排序的实现细节，显示其技术上的可能性；第 4 节中 G2P 应用于词典数据库的结果展现了其自然度。第 5 节是要点的简短总结。

1 Mongolian sorting conventions | 蒙文排序惯例

1.1 Unfolded sorting | 展开式排序

Although Mongolian is generally taught to native Mongolians in a syllabic or semisyllabic way, modern Mongolian dictionaries are sorted alphabetically. The commonly assumed alphabet (unfolded alphabet) used in sorting is

a ·e ·i ·o ·u ·ö ·ü ·ë ·n ·ŋ ·b ·p ·x ·g ·m ·l ·s ·š ·t ·d ·č ·ǰ ·y ·r ·w ·f ·k ·c ·z ·h ·ř ·ł ·ẑ ·ĉ,

with only a few minor variations:

- *ë*(ᠶ) is treated either as an independent letter or as a variant of *e* (ᠡ);
- *ŋ* (ᠩ) is treated either as an independent letter or as a letter sequence *ng* (<ᠨ, ᠭ/ᠺ>).

尽管蒙文一般是以音节或半音节的方式教授给蒙古人的，现代蒙文词典的排列按照的却是字母序。排序时通用的字母表（展开字母表）是

a ·e ·i ·o ·u ·ö ·ü ·ë ·n ·ŋ ·b ·p ·x ·g ·m ·l ·s ·š ·t ·d ·č ·ǰ ·y ·r ·w ·f ·k ·c ·z ·h ·ř ·ł ·ẑ ·ĉ,

还有几处次要的变异：

[1] In this document, the *Mongolian* script is confined to Modern Hudum Mongolian unless otherwise specified.
[2] 本文中「蒙文」限定指现代胡都木蒙古文，除非另有指明。

- *ë*（᠊᠋）是当成独立字母还是 *e*（᠎）的变体；
- *ŋ*（᠋）是当成独立字母还是字母序列 *ng*（<᠊, ᠋/᠌>）。

This Mongolian alphabet is characterized by its opaque identification of letters. As is known, there are several groups of letters in this alphabet which share the same written forms in certain cases (*a*/*e*/*n*, *ëw*, *o*/*u*/*öü*, *x*/*g*, *t*/*d*, etc.), and the distinctions of letters are established generally on the contrasts in colloquial pronunciations. However, individual cases of identifying letters can be problematic:

- Due to drastic phonological changes in history, identifying letters relying merely on colloquial pronunciation is not always possible. Besides phonological knowledge, one's orthographical, etymological, and morphological knowledge of the language/script also take part. Therefore it is quite often that one's identification may differ widely from another's as these criteria do not always converge. For example, the Mongolian word ᠴᠢᠨᠣ᠋ "wolf" may be identified as *čino_a*, *činu_a*, or *činw_a* according to different criteria that contradict each other.
- As for phonological knowledge alone, people of various dialectal backgrounds tend to identify letters according to their local pronunciations. Even if confining our discussion to standard pronunciations, it should be noted that the standard pronunciations of China and Mongolia differ in many aspects.[3] Worse still, the standard Mongolian pronunciation of China is an idealized phonology which is native to nowhere,[4] and is poorly popularized in China today.

这个蒙文字母表有个特点，字母的认定很不透明。都知道这个字母表里有几组字母在某些情形里写法一样（*a*/*e*/*n*、*ëw*、*o*/*u*/*öü*、*x*/*g*、*t*/*d*，等等），字母的区别一般是建立在口语读音的对立上的。但具体例子里的字母认定就可能很成问题：

- 由于历史上激烈的音系变化，只靠口语读音来认定字母不总能行得通。除了音系知识，一个人对蒙古语文的正字法、词源、形态等知识都会起作用。于是某个人的字母认定跟另一个人大相迳庭的情况比比皆是，因为这几条判据并不总会指向同一点。比如说，蒙文词 ᠴᠢᠨᠣ᠋「狼」根据相互冲突的不同判据可能被认定为 *čino_a*、*činu_a* 或 *činw_a*。
- 单就音系知识来说，方言背景各异的人会倾向于根据本地的读音来认定字母。即使把讨论限定在标准音之内，应当注意中蒙两国的标准音在很多方面都不一样[5]。不止于此，中国标准音是一个理想音系而并非一地之音[6]，并且在今天的中国还远远没有推广开。

As a consequence, the usability of unfolded collation is questionable:

- One word may be identified as various letter sequences and thus sorted variously among dictionaries, and one will not be able to find the entry when they has a different identification of the underlying letters from that of the dictionary compiler.
- When encountering a word they does not know how to pronounce correctly, one will have to go over every orthographically possible reading in order to find it in a dictionary. However, it is quite common for a single written form to have no less than four possible readings like *talai*, *telei*, *dalai*, and *delei*.

所以说，展开式排序的应用价值很值得怀疑：

- 一个词可能被认定成很多种字母序列，从而在不同的词典里排序也千差万别，进而一个人如果跟词典编纂者对底层字母的认定不一样就查不到那个词条。

[3] *čagan* "white" is pronounced as /**ʧ**aga:**n**/ in the standard pronunciation of China and /**ʦ**aga:**ŋ**/ in the standard pronunciations of Mongolia.

[4] *xöxexota* "Hohhot" is pronounced as /**x**ox**x**ɔt/ in the standard pronunciation of China but /**g**ox**g**ɔt/ in Zhenglan Banner (whose phonology is taken as the archetype of the standard pronunciation).

[5] *čagan*「白」在中国标准音中读作 /**ʧ**aga:**n**/，而在蒙古国标准音中读作 /**ʦ**aga:**ŋ**/。

[6] *xöxexota*「呼和浩特」在中国标准音中读作 /**x**ox**x**ɔt/，但在正蓝旗（其音系被视作标准音的原型）读作 /**g**ox**g**ɔt/。

- 碰到不知道正确读音的词的话，就得把所有正字法上可能的读法都查一遍才能在词典里查到。但一个写法有不下四种可能的读音的例子俯拾即是，比如 *talai*、*telei*、*dalai*、*delei*。

1.2 Folded sorting | 折叠式排序

Nevertheless, there has been another approach to Mongolian sorting dating back to pre-modern times which can evade these above-mentioned shortcomings, whose principle is quite intuitive: for each component of the syllable (onset, nucleus, or coda), one shape is subsumed under only one letter, which means the sorting behavior for a specific written form is fully determined as long as the syllable structure is known. This approach is termed here as “folded” because a syllabary/alphabet that *folds* homographic written forms is adopted in sorting. In this folded approach,

- *telei* and *dalai* (ᠲᠠᠯᠠᠢ) are sorted like **talai* because they are written alike, which has maximally eliminated ambiguous reading;
- *bey_e* (ᠪᠡᠶ᠎ᠡ) is sorted like **bay_a*, not **bai_a* (cf. *bai* (ᠪᠠᠢ)), where the syllable structure is retained to make the sorting linguistically significant.

不过早在近代之前蒙文排序还有另一种方法能避免上面提到的这些缺点，原理并不难懂：对音节的每一个成分（首音、核音、尾音），一个字形只归在一个字母下。这就是说，只要音节结构已知，一个特定的写法的排序行为就是完全确定的。这种方法这里叫作「折叠式」，因为排序中采用的音节表/字母表把同形的写法折叠了起来。在这种折叠式排序中，

- *telei* 和 *dalai*（ᠲᠠᠯᠠᠢ）排得如同 **talai*，因为写法一样；这样尽可能消除了歧义的读法；
- *bey_e*（ᠪᠡᠶ᠎ᠡ）排得如同 **bay_a* 而不是 **bai_a*（对照 *bai*（ᠪᠠᠢ））；这样保持了音节结构，让排序语言学上有意义。

Arguments against folded sorting may come in two aspects:

- It goes against the common practice of unfolded sorting which has been established for several decades.
- The folded letters are difficult for users to memorize because they are not based on colloquial pronunciations.

反对折叠式排序的论点可能来自两个方面：

- 与几十年来建立起来的通行做法展开式排序相悖。
- 用户难以记忆折叠字母，因为折叠字母不是基于口语读音的。

My defense of folded sorting also comes in two aspects:

- The unfolded sorting widely adopted today itself is not a well-established tradition.
  - The shift from syllabic sorting to alphabetical sorting took place only no more than a century ago.
  - There are still discrepancies in unfolded sorting like how to arrange *ŋ* in dictionaries published in China alone.
  - The Chinese standard of Mongolian sorting (GB/T 30851-2014) contradicts widely adopted sorting practices and other Chinese standards of the script (e.g., GB/T 32912-2016) in many aspects. More ironically, the text description of the sorting standard contradicts with its collation table.
  - Users have long been accustomed to the sorting instability resulting from fickle letter identification.
- Folded sorting is much easier for users to master.
  - Folded letters are not memorized but derived directly from the written form.
  - Unlike unfolded sorting, etymological or morphological knowledge is no longer necessary in folded sorting. Basic orthography alone is almost enough to determine the folded letter sequence from a given shape, though colloquial pronunciation might be marginally helpful in case that syllabification ambiguity arises.
  - Spelling variations in digital text contributed by the public show that users have partially assumed a folded alphabet in typing.

我为折叠式排序的辩护同样分两方面：

- 今天普遍在用的展开式排序本身不是什么根深叶茂的传统。
  - 从音节排序转移到字母表排序发生的时间距今未满一世纪。
  - 光中国出版的词典里还有诸如 *ŋ* 怎么安排的分歧。
  - 蒙文排序的中国标准（GB/T 30851-2014）与普遍采用的排序做法以及蒙文的其他中国标准（比如 GB/T 32912-2016）在好些方面都有矛盾。更讽刺的是，这部排序国标的文字描述与其排序表都矛盾。
  - 善变的字母认定导致排序的不稳定，用户早就对此习以为常了。
- 折叠式排序更容易让用户掌握。
  - 折叠字母不是靠记忆的，而是直接从写法里推出来的。
  - 不像展开式排序，词源、形态上的知识对折叠式排序都不再必要。只需基础的正字法就足以确定给定字形的折叠字母序列，尽管口语读音可能在音节结构有歧义的边缘情况下有些用。
  - 大众贡献的数字文本里的错误拼写显示，用户在打字时已经部分地采用了折叠字母表。

In the end, the conclusion is that in no definite aspect is folded sorting inferior to unfolded sorting.

最后结论是，折叠式排序没有哪个方面会输给展开式排序。

2 Sorting automation and encoding schemes | 排序自动化与编码方案

As we turn back to the encoding issue, problems that arise in sorting automation is what we are concerned about. Although it is impossible to talk of implementation of automated sorting without addressing a specific encoding scheme, encoding schemes and automated sorting are essentially two dimensions and intersect each other.

回到编码的议题上来，排序自动化中出现的问题是我们所关心的。尽管自动排序的实现脱不开具体的编码方案来讲，编码方案和自动化排序本质上是互相交叉的两个维度。

Remarks on various automated sorting methods for phonetic and graphetic approaches are summarized as in Table 1. Here *character sorting* refers to sorting encoded characters without any contextual sensitivity (expansion or contraction). Except for a few diacritical characters (such as FVSes in phonetic encoding), character sorting treats encoded characters as letters.

对针对音码和形码的几种自动化排序法的评价总结见表 1。这里「字符排序」指的是不含任何语境相关性（扩展或者缩合）来排列编码字符。除掉一些辨音符类的字符（譬如音码里的 FVS），字符排序把编码字符当成字母。

Table 1 Automated sorting methods × encoding approaches | 表 1 自动化排序法 × 编码取径

	Character sorting	Natural sorting	
		Unfolded sorting	Folded sorting
Phonetic encoding	**Unreliable**		Feasible
Graphetic encoding	Unusable	Impossible	**Feasible**

	字符排序	自然排序	
		展开排序	折叠排序
音码	**不可靠**		可行
形码	不可用	不可能	**可行**

For phonetic encoding schemes:

- Character sorting and unfolded sorting yield identical results in most of the cases. Being the most widely used automated sorting of Mongolian today, however, the sorting results are unreliable, because one entry can be represented with different character sequences by different users. There are several reasons for this typing inconsistency:

    - Typo: Typists may not always be able to notice their typing errors.
    - Misuse: Typists may identify letters differently.
    - Abuse: Typists may favor some incorrect keystrokes deliberately for the sake of expediency.

  Therefore, when text from different sources is aggregated, one entry can appear in several places of the sorting result.
- Folded sorting is feasible for phonetic encoding because it is also feasible for graphetic encoding (see below), and phonetic encoding contains all information graphetic encoding conveys and beyond.

对音码来说：

- 字符排序和展开排序给出的结果多数情形下是一样的。虽然是今天蒙文自动化排序里用得最广的方法，其排序结果并不可靠，因为一个条目可能被不同的用户以不同的字符序列表示。打字打得不一致有好几种原因：
    - 打错：打字人不总能注意到自己打错了字。
    - 误用：打字人可能把字母认定得不一样。
    - 滥用：打字人可能为了方便故意喜欢打不正确的键。

  于是，当各种来源的文本汇集到一起，一个条目可能出现在在排序结果的好几个地方。
- 折叠排序对音码是可行的，因为其对形码也可行（见下），并且音码在包含了形码里表示的信息之外还包含了别的信息。

For graphetic encoding schemes:

- Direct graphetic character sorting, being the least costly method, is not linguistically significant, and is incomprehensible to users.
- Unfolded sorting is impossible, because the distinctions between homographic letterforms are missing from the encoded text.
- Folded sorting of graphetic encoding is feasible.
    - It is comprehensible to users and thus usable, as has been discussed in Section 1.2.
    - It is technically implementable, because the information of folded letters is still largely reconstructible from graphetic strings through a series of contextual transformations, which I will demonstrate shortly.

对形码来说：

- 直接排形码字符是最不费事的，但语言学上没有意义，于是对用户来说不可理解。
- 展开排序不可能，因为编码文本中已经没有同形字母的区别了。
- 形码的折叠排序是可行的。
    - 用户能理解，所以是可用的，这在第 1.2 节已经讨论过。
    - 技术上是能实现的，因为折叠字母的信息从形码字符串里通过一系列语境变换后大体可以重构得，下面就来展示这一点。

Automated folded sorting of graphetic encoding, or **G2P sorting**, is particularly of great significance because it seems to be the only practical solution to automated sorting for graphetically encoded Mongolian.

针对形码自动化的折叠排序，或称 **G2P 排序**，意义尤其重大，因为对形码蒙文来说可行的自动化排序可能就这么一种。

3 Implementation of G2P sorting | G2P 排序的实现

G2P sorting is implemented in two steps:

- Step 1: Reconstruct folded letter sequences from graphetic character strings.
- Step 2: Sort the entries by their folded letter sequences.

G2P 排序分两步进行：

- 第一步：从形码字符串中重构出折叠字母。
- 第二步：根据条目的折叠字母序列进行排序。

G2P sorting cannot be accommodated within the main algorithm of the Unicode Collation Algorithm, as only limited contextual sensitivity (to the extent that adding a few contractions to the collation element table will work, like Thai/Lao CV inversion) can be handled thereby. However, the mapping from graphetic characters to folded character is highly context-sensitive. As a result, a separate step of transformation is needed to reconstruct folded strings before the main procedure of sorting. Luckily this additional step is quite similar to the preprocessing mentioned in the Unicode Collation Algorithm, the only difference being that the results of reconstruction here are not Unicode character strings but imaginary sequences of folded letters.

G2P 排序不能被 Unicode 排序算法的主算法容纳，因为这种框架只能应付很有限的语境相关性（在排序元素表里加一些缩合项就能解决的那种程度，像泰文/老挝文的元辅音倒置）。但是，从形码字符到折叠字母的映射是高度语境相关的。所以在排序的主程序之前需要有一步单独的转换。幸运的是多的这一步跟 Unicode 排序算法中提及的预处理很像，唯一的差别是这里重构的结果不是 Unicode 字符而是折叠字母的假想序列。

The present discussion is aimed at showing that there are always practical solutions to the reconstruction of folded letters, however graphetically radical the encoding scheme is (as long as the medial double tooth (ᠷ) representing *x* or *g* is not broken into two single teeth (ᠷ)). In order to cover all practical variations within the graphetic encoding approach, only the implementation for the most radical graphetic encoding scheme is addressed here, and the readers should find it easy to adjust it to the rest. It should be emphasized that the present discussion shall *not* be construed as favoring any specific variant in the spectrum of graphetic encoding.

本文的讨论意在展示，无论字形编码方案有多彻底（只要表示 *x* 或 *g* 的中形双牙（ᠷ）不断成两个单牙（ᠷ）），折叠字母的重构总有可行的解决方案。为了涵盖形码之下所有可行的变异，这里给出的实现法只针对其中最激进的形码方案；读者应该不难对其进行调整以适用于其他。值得强调的是，这里的讨论并不应该解读为偏向形码旗下的任一种特定变体。

3.1 Collation element table | 排序元素表

The collation element table of G2P sorting is given as Table 2, where folded letters are listed by their sorting order, and corresponding glyphs, graphic characters, and unfolded letters along with their orders are given together. Folded letters and graphetic characters in biunique mapping with them are transliterated with small letters, while graphetic characters not in biunique correspondence with folded letters are transliterated with capital letters. As can be seen from the table, there might be up to 9 underdetermined graphetic characters, namely A, I, O, U, X, G, L, W, and H, depending on the radicalness of the graphetic encoding scheme. Secondary weights of collation elements in the table are largely arbitrarily given, as there is no widely accepted secondary weighting convention for either unfolded or folded sorting.

表 2 给出了 G2P 排序的排序元素表，其中排序元素依其次序列出，对应的图形、形码字符、展开式字母及其次序一并给出。一一对应的折叠式字母和形码字符用小写字母表示，不与折叠式字母一一对应的形码字符用大写字母表示。表中可以看出，根据形码方案激进度的不同，至多可能有 9 个待定的字母，即 A、I、O、U、X、G、L、W、H。表中排序元素的次级权重大体是任意取的，因为无论是展开式还是折叠式排序都没有一套通行的次级权重惯例。

Table 2 Collation element table | 表 2 排序元素表

Unfolded order	Unfolded letter	Collation element	Folded letter	Graphetic character	Glyph | 图形			
					IS	I	M	F
展开音序	展开字母	排序元素	折叠字母	形码字符	单	上	中	下
—	—	[.00.3]	ʔ	A	□	꠰	꠰	□
1 2_1	*a e*	[.01.1]	*a*	A	□	꠰	꠰	꠰/ꠇ
		[.01.2]	*ɑ* = ɑ		ꠇ	■	■	■
2_2	*e*	[.02.1]	*ɇ*	—	■	■	·	ꠇ
		[.02.2]	*ë*	W	■	□	ꠉ	ꠉ
3	*i*	[.03.1]	*ij*	II	□	□	ꠋ	□
		[.03.2]	*i*	I	ꠉ	ꠊ	ꠊ	ꠉ/ꠊ
4 5 6_1 7_1	*o u* $ö_1$ $ü_1$	[.04.1]	*o*	O	■	ꠌ	ꠌ/ꠍ	ꠌ/ꠍ
		[.04.2]	*u*	U	ꠍ	■	■	ꠍ
6_2 7_2	$ö_2$ $ü_2$	[.05.1]	*ö*	OI	■	■	ꠏ/ꠐ	□
		[.05.2]	*ü* = ü		■	■	■	ꠑ/ꠒ
11	*n*	[.11.1]	*n* = n		■	ꠓ	ꠓ	ꠓ/ꠔ
		[.11.2]	*ń*	A	□	□	ꠔ	ꠔ
12	*ŋ*	[.12.1]	*ŋ*	AG	□	□	ꠕ	ꠖ
13	*b*	[.13.1]	*b* = b		■	ꠗ	ꠗ	ꠗ
14	*p*	[.14.1]	*p* = p		■	ꠘ	ꠘ	ꠘ
15_1	*x*	[.15.1]	*x*	X	■	ꠙ	ꠚ	ꠚ
16_1	*g*	[.16.1]	*ġ* = ġ		■	ꠛ	ꠜ	ꠚ/ꠝ
		[.16.2]	*ǵ*	X	■	□	ꠚ	ꠚ
15_2 16_2	x_2 g_2	[.17.1]	*g*	G	■	ꠞ	ꠞ	ꠟ
17	*m*	[.18.1]	*m* = m		■	ꠠ	ꠡ	ꠢ
18	*l*	[.19.1]	*l*	L	■	ꠣ	ꠤ	ꠥ
19	*s*	[.20.1]	*s* = s		■	ꠦ	ꠧ	꠨
20	*š*	[.21.1]	*š* = š		■	꠩	꠪	꠫
21_1 22_1	t_1 d_1	[.22.1]	*t* = t		■	꠬	꠬	꠭
21_2 22_2	t_2 d_2	[.23.1]	*d* = d		■	꠮	꠮	꠯
		[.23.2]	*ð*	OA	□	■	꠯	꠯/꠰
23	*č*	[.24.1]	*č* = č		■	꠱	꠱	꠱
24	*ǰ*	[.25.1]	*ǰ* = ǰ		■	■	꠲	꠲
		[.25.2]	*ɟ*	I	ꠉ	ꠊ	□	□
25	*y*	[.26.1]	*y* = y		■	꠳	꠳	■
		[.26.2]	*ǐ*	I	□	ꠊ	ꠊ	ꠉ
26	*r*	[.27.1]	*r* = r		■	꠴	꠴	ꠉ
27	*w*	[.28.1]	*w*	W	■	꠵	꠵	꠵
		[.28.2]	*ŭ*	U	□	■	■	ꠍ
28	*f*	[.29.1]	*f* = f		■	꠶	꠶	꠶
29	*k*	[.30.1]	*k* = k		■	꠷	꠷	꠷
30	*c*	[.31.1]	*c* = c		■	꠸	꠸	꠸
31	*z*	[.32.1]	*z* = z		■	꠹	꠹	꠹
32	*h*	[.33.1]	*h*	H	■	□	꠺	꠻
		[.33.2]	*ɦ*	AH	■	꠼	□	□
33	*ř*	[.34.1]	*ř* = ř		■	꠽	■	■
34	*ɬ*	[.35.1]	*ɬ*	LH	■	꠾	꠾	□
35	*ẑ*	[.36.1]	*ẑ*	H	■	꠿	□	□
36	*ĉ*	[.37.1]	*ĉ*	OO	■	ꡀ	□	■

3.2 Reconstructing folded letters from graphetic characters | 从形码字符中重构折叠字母

In fact, finite unilinear (no branching, no looping) application of regular expression substitutions will suffice for our purposes of folded letter reconstruction. A set of substitutions containing no dictionary items for content words is adopted, as is listed in Table 3. Catch-all substitutions that ensure completeness of transformation are highlighted. It is worth mentioning that devices like groups (`(…)`) or quantifiers (`*`, `+`, `?`, `{m,n}`, etc.) are not even resorted to here.

事实上，有限、单线性（无分支无循环）地运用正则表达式替换就可以满足我们重构折叠字母的需要。这里采用的是一组不含实词词典项的替换，如表 3 所列。用于保证转换完备性的兜底替换被标为高亮。值得注意的是诸如分组（`(…)`）、量词（`*`、`+`、`?`、`{m,n}`，等等）的手段都没有用到。

Table 3 List of regex substitutions employed in folded letter reconstruction | 表 3 折叠字母重构中采用的正则替换列表

Search 查找式	Subs. 替换式
\bOOA\b	oð
\bI\b	i
\bIIAA\b	iĭań
\bIIAr\b	iĭar
\bOOI	ĉi
\bH	ẑ
\bAH	ɦ
\bLH	ɬ
\bW	w
\bI	ɟ
I(?=[ɑ])	ĭ
U(?=[ɑ])	ŭ
U	u
H	h
L	l
OI\b	oi
OII\b	öi
(?<=\bG)OII	öi
OII	oï̈
OI	ö
(?<=\BA)II\b	ĭi
(?<!I)I(?!I)	i
\bO	o
O(?!A)	o
(?<!A)G	g
(?<=\B[AWO])II	ï̈
I	i
OAG(?![AIOɑüaɐ̈ëiïjouö])	oŋ
(?<=\B[AIOɑüaɐ̈ëiïjouö])W	w
(?<=\B[AWIɑüaɐ̈ëiïjouö])OA	ð
(?<!A)G	g
O	o
X(?=[ŭAIɑüaɐ̈ëiïjouö])	x
X	ǵ
GW(?=[nbpmsštdčǰrfkczglɟwhġyř?xĭŭẑĉɦɬ])	gë
G(?=[AIɑüaɐ̈ëiïjouö])	g
(?<=[nbpmsštdčǰrfkczglɟwhġyř?xĭŭẑĉɦɬ])AAG	aŋ
WAAG	waŋ
(?<=[nbpmsštdčǰrfkczglɟwhġyř?xĭŭẑĉɦɬ])AG	ag
(?<=[Iɑüaɐ̈ëiïjouö])AG	ŋ
WAG	ëŋ

(?<=[nbpmsštdčǰrfkczglɉwhġyř?xǐŭẑĉɦɬ])AA	ań
WAA	wań
(?<=[nbpmsštdčǰrfkczglɉwhġyř?xǐŭẑĉɦɬ])A	a
(?<=[Iɑüaɐ̈ëiɨjouö])A	ń
(?<=\bA)WA(?![ńŋǵðɑüaɐ̈ëiɨjouö])	ëń
(?<=[nbpmsštdčǰrfkczglɉwhġyř?xǐŭẑĉɦɬ])WA(?![ńŋǵðɑüaɐ̈ëiɨjouö])	ëń
WA	wa
AAAG	ʔaŋ
AAG	ʔɐ̈ŋ
AG	ʔɐ̈g
G	g
AAA	ʔań
AA	ʔa
A(?=[Wɑüaɐ̈ëiouö])	ʔ
A	ʔɐ̈
W(?=[i]\B)	w
W(?=[ɨjńŋǵð])	ë
W(?=[u])	ë
(?<=[ʔɦɬ])W	ë
(?<=[Iɑüaɐ̈ëiɨjouöńŋǵð])W	w
WW	ëw
W	ë

4 Testing G2P sorting | 测试 G2P 排序

G2P sorting method specified in the previous section is tested against a 26433-word spelling database. The results are as follows.

前一节所述的 G2P 排序法被应用到一个 26433 词的数据库来进行测试，结果如下。

4.1 Misreconstruction of folded letters | 折叠字母的误重构

The folded letters reconstructed from graphetically encoded data matched the original unfolded letters quite well in the correspondences defined in Table 2, except for the four major types of exceptions listed in Table 4 below:

- Type A: All words beginning with *ei* or *eü*, and all words beginning with *en* which is not followed by a vowel.
- Type B: All words containing *on*/*un*/*ön*/*ün* which is preceded by a vowel but not followed by a vowel. Mostly in genitive stems of fugitive-*n* native words (e.g., EREÜ: nom. *ereü*, gen. *ereün=ü*), but may also occur in loan words.
- Type C: A few words containing *w* or *ë*. The list of affected words may vary, depending on the specific transformational rule set chosen.
- Type D: A few words with ill-formed spelling.

从形码数据里重构而来的折叠字符按照表 2 里定义的对应关系与原始的展开字母对应得很好，除了以下表 4 中所列的四大类例外：

- A 类：所有以下述结构起首的词：*ei*、*eü*、后面不是元音的 *en*。
- B 类：所有包含下述结构的词：前面是元音而后面不是元音的 *on*/*un*/*ön*/*ün*。多数是隐现 *n* 固有词的属格词干（如 EREÜ：主格 *ereü*，属格 *ereün=ü*），但借词里也会有。
- C 类：一小部分包含 *w* 或 *ë* 的词。影响到的词范围并不固定，取决于具体选定的转换规则。
- D 类：一小部分拼写异常的词。

Table 4 Types of misreconstruction | 表 4 误重构的分类

Type	Unfolded letter	Folded letter	Reconstructed folded letter	Graphetic character	Major etymological class	Count
类	展开字母	折叠字母	重构折叠字母	形码字符	主要词源类	词数
A	*e*	*ʔɛ́*	*ʔ*	A	Native | 固有	27
B	*Un*	*oń*	*ð*	OA	Native/Loan | 固有・外来	12
C	*w* *ë*	*w* *ë*	*ë* *w*	W	Loan | 外来	6
D	Ill-formed spelling | 异常拼写				Native | 固有	2

As is shown, all cases of misreconstruction are syllabification errors, which are predictable in general. Table 5 gives some examples of misreconstruction of each type.

可见所有误重构的例子都是音节划分的错误，大体上可以预测。表 5 给每一类误重构举了一些例子。

Table 5 Examples of misreconstruction | 表 5 误重构举例

Type	Unfolded letter	Folded letter	Reconstructed folded letter	Graphetic character
类	展开字母	折叠字母	重构折叠字母	形码字符
A	*ei*	*ʔɛ́i*	*ʔi*	AI
	ein	*ʔɛ́ijń*	*ʔiiń*	AIIA
	e ü	*ʔɛ́u*	*ʔu*	AU
	ende	*ʔɛ́ńda*	*ʔada*	AAdA
B	*ereïn*	*ʔɛ́raoń*	*ʔɛ́rað*	ArAOA
	ondoon	*ʔońdooń*	*ʔońdoð*	AOAdOOA
C	*niswanis*	*niswanis*	*nisëńnis*	nIsWAnIs
	nirwalaxu	*nirwalaxu*	*nirëńlaxu*	nIrWALAXU
	burwasad	*borwasað*	*borëńsað*	bOrWAsAOA
	y ẅi	*yëlwi*	*yëlëi*	yWLWI
D	*tŋri*	*tŋri*	*tagri*	tAGrI
	tŋrilig	*tŋrilig*	*tagrilig*	tAGrILIG

In the 26433-word database, only 47 entries of misreconstruction occur, making up 0.17% of the total.

在 26433 词的数据库里只出现了 47 条误重构，占总数 0.17%。

4.2 Comparison of G2P sorting with ideal unfolded sorting | G2P 排序与理想展开式排序的比较

The result of the G2P sorting test is then compared with ideal unfolded sorting. The results are shown in Figure 1 and Figure 2:

G2P 排序测试的结果与理想展开排序比较，结果如图 1 及图 2 所示：

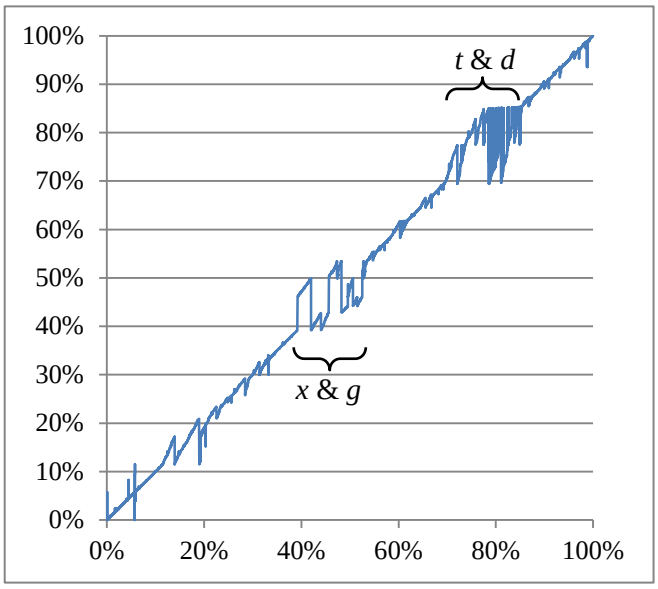

Figure 1 G2P sorting ~ ideal unfolded sorting

图 1 G2P 排序 ~ 理想展开排序

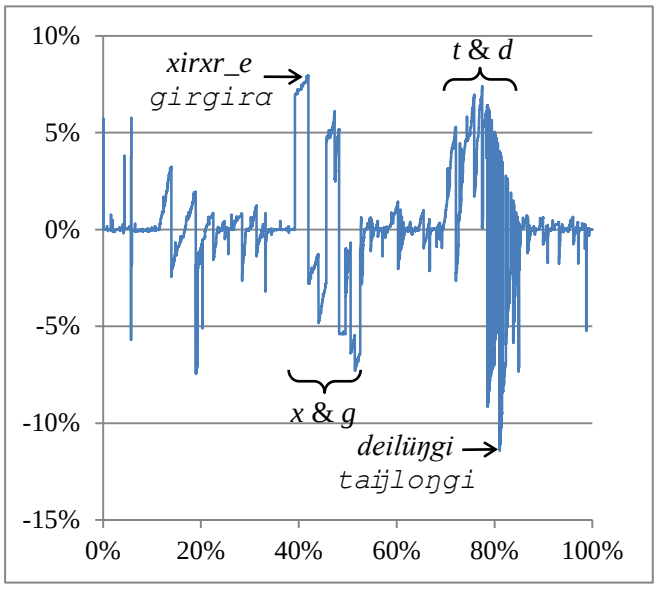

Figure 2 Lead and lags of G2P sorting relative to ideal unfolded sorting

图 2 G2P 排序相对于理想展开排序的超前和滞后

Figure 1 shows the relative order of G2P sorting against ideal unfolded sorting in percentage, and Figure 2 shows their difference. The maximum lead of G2P sorting to ideal unfolded sorting is −11.41%, occurring at *deilüŋgi* sorted as `taijloŋgi`, while the maximum lag is +7.96%, occurring at *xirxir_e* sorted as `girgirɑ`. The standard deviation (σ) of the sorting order differences of all 26433 entries is 2.84%.

图 1 显示的是以百分数计算的 G2P 排序的相对顺序和理想展开式排序的比较，图 2 显示的是两者的差。G2P 排序相对于理想展开式排序的最大超前是 −11.41%，出现在 *deilüŋgi* 排列作 `taijloŋgi`；最大滞后是 +7.96%，出现在 *xirxir_e* 排列作 `girgirɑ`。总 26433 条目序差的标准差（σ）是 2.84%。

The major contributions to the sorting differences come from *x*/*g* and *t*/*d* pairs of the first letter, and less significantly from *a*/*e*, *o*/*u* and *ö*/*ü* pairs of the first vowel. A user accustomed to ideal unfolded sorting should find it easy to adapt to G2P sorting without much training, just to remember to make some adjustment to these tangled pairs according to their written shapes.

排序差异主要是首字母的 *x*/*g*、*t*/*d* 两对，其次是第一元音的 *a*/*e*、*o*/*u*、*ö*/*ü* 几对。用惯了理想展开排序的用户应该会觉得 G2P 排序不用怎么训练就很容易适应，只需记住要对这几对纠缠的字母按照其书写形状作些调整。

To give an intuitive view of the sorting similarity, an extract of the results of both sorting methods are given in Table 6, where the items are extracted in a uniformly spaced manner from the G2P-sorted database. The results show that only 5 entries (highlighted) out of total 52 differ. In conclusion, G2P sorting does not deviate as much from ideal unfolded sorting as one might expect.

为了直观地展示排序的相似性，表 6 中列出了两种排序法的节选，其中条目是从 G2P 排序的数据库中用等间隔的方法抽取出来的。结果显示，总 52 条目中只有 5 条（已高亮）有差异。结论是，G2P 排序与理想的展开式排序并没有想像中差得那么多。

Table 6 Extract of G2P sorting and ideal unfolded sorting | 表 6 G2P 排序与理想展开排序的节录

Rel. delta sort. ord.	Unfolded sort. ord.	Unfolded letter	G2P sort. ord.	Reconstructed folded letter	Graphetic character
相对序差	展开排序序号	展开字母	G2P 排序序号	重构形码字符	形码字符
0.06%	517	*agawa*	500	`ʔaġawa`	`AAġAWA`
0.00%	1000	*asagtuxu*	1000	`ʔasaǵdoxu`	`AAsAXdOXU`

−0.05%	1488	*arčigur*	1500	*ʔarčiġor*	AArčIġOr
0.06%	2015	*elegdexü*	2000	*ʔɐ́lagdago*	ALAGdAGO
0.04%	2511	*iin*	2500	*ʔiiń*	AIIA
0.01%	3003	*irgagalaxu*	3000	*ʔirġaġalaxu*	AIrġAġALAXU
−0.91%	3259	*obosxixü*	3500	*ʔobosgigo*	AObOsGIGO
0.62%	4164	*usadxagči*	4000	*ʔosaðxaǵči*	AOsAOAXAXčI
−0.14%	4462	*ursigtai*	4500	*ʔorsiǵdai*	AOrsIXdAI
	(5000	*örgedxel*)			
1.04%	5276	*ülemǰidexü*	5000	*ʔölamǰidago*	AOILAmǰIdAGO
−1.89%	5000	*örgedxel*	5500	*ʔörgaðgal*	AOIrGAOAGAL
−0.63%	5834	*namči*	6000	*namči*	nAmčI
−0.09%	6475	*noxaituxu*	6500	*noxaïjdoxu*	nOXAIIdOXU
−0.01%	6997	*bagaturčud*	7000	*baġadorčoð*	bAġAdOrčOOA
0.71%	7689	*beyeči*	7500	*bayači*	bAyAčI
1.23%	8326	*buurǰi*	8000	*boorǰi*	bOOrǰI
0.25%	8565	*buǰigirtuxu*	8500	*boǰigirdoxu*	bOǰIGIrdOXU
0.02%	9005	*bürxüixü*	9000	*börgoïjgo*	bOIrGOIIGO
0.00%	9500	*xabursil*	9500	*xaborsil*	XAbOrsIL
0.00%	10000	*xasumal*	10000	*xasomal*	XAsOmAL
2.56%	11179	*xoordalg_a*	10500	*xoordalġɑ*	XOOrdALġɑ
1.82%	11482	*xošoŋčilaxu*	11000	*xošoŋčilaxu*	XOšOAGčILAXU
	(12250	*xög*)			
5.39%	12928	*gagaglaxu*	11500	*ġaġaǵlaxu*	ġAġAXLAXU
6.75%	13788	*gutugaxu*	12000	*ġodoġaxu*	ġOdOġAXU
2.63%	13198	*gemsixü*	12500	*gamsigo*	GAmsIGO
1.20%	13319	*gilalǰam_a*	13000	*gilalǰamɑ*	GILALǰAmɑ
	(13788	*gutugaxu*)			
−4.72%	12250	*xög*	13500	*gög*	GOIG
0.45%	14120	*güyüldüxü*	14000	*göyoldogo*	GOIyOLdOGO
0.34%	14590	*mesil*	14500	*masil*	mAsIL
−0.01%	14997	*möŋxelexü*	15000	*möŋgalago*	mOIAGGALAGO
−0.11%	15471	*sanagatai*	15500	*sanaġadai*	sAnAġAdAI
−0.80%	15789	*salbarxai*	16000	*salbarxai*	sALbArXAI
0.05%	16514	*sibeg*	16500	*sibag*	sIbAG
−0.03%	16993	*siratuxu*	17000	*siradoxu*	sIrAdOXU
−0.82%	17283	*soyoxai*	17500	*soyoxai*	sOyOXAI
−0.02%	17995	*šaliyatuxu*	18000	*šaliyadoxu*	šALIyAdOXU
−0.23%	18440	*taitagar*	18500	*taïjdaġar*	tAIIdAġAr
0.86%	19227	*tebxe*	19000	*tabga*	tAbGA
6.16%	21133	*dagir*	19500	*tagir*	tAGIr
−4.21%	18884	*tasixu*	20000	*tasixu*	tAsIXU
−5.30%	19097	*tawarčilaxu*	20500	*tawarčilaxu*	tAWArčILAXU
	(19227	*tebxe*)			
−4.68%	19761	*togonočaxu*	21000	*toġonočaxu*	tOġOnOčAXU
−4.62%	20275	*tulgagurtai*	21500	*tolġaġordai*	tOLġAġOrdAI
	(21133	*dagir*)			
2.03%	22537	*düŋsüixü*	22000	*töŋsoïjgo*	tOIAGsOIIGO
0.39%	22604	*dürsütei*	22500	*törsodai*	tOIrsOdAI
−0.48%	22874	*časuraxu*	23000	*časoraxu*	čAsOrAXU
0.00%	23500	*čisurxau*	23500	*čisorxau*	čIsOrXAU
0.35%	24092	*čüürelǰexü*	24000	*čööralǰago*	čOIOrALǰAGO

0.87%	24730	*ǰegüüte*	24500	*ǰagolda*	IAGOLdA
−0.01%	24998	*ǰilabči*	25000	*ǰilabči*	IILAbčI
−0.46%	25378	*ǰoltai*	25500	*ǰoldai*	IOLdAI
0.43%	26114	*yexemsüg*	26000	*yagamsog*	yAGAmsOG

5 Summary | 总结

The key points of this document are:

- Folded sorting is not a new invention, nor is it inferior to unfolded sorting.
- G2P sorting, as an automated variant of folded sorting, can be a good solution when automated sorting of graphetically encoded Mongolian is needed.
- Technically G2P can be implemented with an additional step of reconstructing the folded letters from the graphetic characters.
- The results of G2P sorting are not dissimilar to those of widely used unfolded sorting, and users will easily get used to it.

本文的要点有：

- 折叠式排序不是新发明，也不并不输给展开式排序。
- G2P 排序是折叠式排序的一个自动化变种，在需要自动化排序形码蒙文的场合是个不错的解决方案。
- 技术上 G2P 可以用加一步从形码字符重构折叠字母来实现。
- G2P 排序的结果与广泛使用的展开式排序结果比较相似，用户容易适应。

A Orderings of dictionary headings in ideal unfolded sorting and G2P sorting | 理想展开排序和 G2P 排序下的词典标目顺序

Table 7 shows an ideal unfolded sorting of dictionary headings, which is the most widely accepted version in modern China. Headings in adjacent cells with the same underline patterns are homographic.

表 7 给出的是词典标目的理想展开式排序，也是现代中国用得最多的一个版本。相邻格子里下划线样式相同的标目是同形的。

Table 7 Ideal unfolded sorting of dictionary headings | 表 7 词典标目的理想展开排序

	a	$<_1$	*e* $<_2$ *ë*	$<_1$	*i*	$<_1$	*o*	$<_1$	*u*	$<_1$	*ö*	$<_1$	*ü*
$<_1$	*na*	$<_1$	*ne* $<_2$ *në*	$<_1$	*ni*	$<_1$	*no*	$<_1$	*nu*	$<_1$	*nö*	$<_1$	*nü*
$<_1$	*ba*	$<_1$	*be* $<_2$ *bë*	$<_1$	*bi*	$<_1$	*bo*	$<_1$	*bu*	$<_1$	*bö*	$<_1$	*bü*
						⋮							
$<_1$	*xa*	$<_1$	*xe* $<_2$ *xë*	$<_1$	*xi*	$<_1$	*xo*	$<_1$	*xu*	$<_1$	*xö*	$<_1$	*xü*
$<_1$	*ga*	$<_1$	*ge* $<_2$ *gë*	$<_1$	*gi*	$<_1$	*go*	$<_1$	*gu*	$<_1$	*gö*	$<_1$	*gü*
						⋮							
$<_1$	*ta*	$<_1$	*te* $<_2$ *të*	$<_1$	*ti*	$<_1$	*to*	$<_1$	*tu*	$<_1$	*tö*	$<_1$	*tü*
$<_1$	*d'a* $<_2$ *da*	$<_1$	*d'e* $<_2$ *d'ë* $<_2$ *de* $<_2$ *dë*	$<_1$	*d'i* $<_2$ *di*	$<_1$	*d'o* $<_2$ *do*	$<_1$	*d'u* $<_2$ *du*	$<_1$	*d'ö* $<_2$ *dö*	$<_1$	*d'ü* $<_2$ *dü*
						⋮							

Table 8 shows G2P sorting of dictionary headings. Headings on both sides of "≡"s are homographic. "⊃"s indicate additional subsumptions, and the parenthesized terms should be in their normal places in ideal folded sorting. For example, *en* should be sorted under the heading of *e* in ideal folded sorting, instead of under *a* in G2P sorting.

表 8 给出的是词典标目的 G2P 排序。「≡」两边的标目是同形的。「⊃」表示额外的归并，括号括起的项在理想折叠排序中应当回归到其正常位置上去。譬如，*en* 在理想折叠式排序中应该排在标目 *e* 之下，而不像 G2P 排序一样排在 *a* 之下。

Table 8 G2P sorting of dictionary headings | 表 8 词典标目的 G2P 排序

	a (⊃ *en*)	$<_1$	*e* $<_2$ *ë*	$<_1$	*i* (⊃ *ei*)	$<_1$	*o* ≡ *u* (⊃ *üi*)	$<_1$	*ö* ≡ *ü*
$<_1$	*na* ≡ *ne*	$<_1$	*në*	$<_1$	*ni*	$<_1$	*no* ≡ *nu* (⊃ *nüi*)	$<_1$	*nö* ≡ *nü*
$<_1$	*ba* ≡ *be*	$<_1$	*bë*	$<_1$	*bi*	$<_1$	*bo* ≡ *bu* (⊃ *büi*)	$<_1$	*bö* ≡ *bü*
					⋮				
$<_1$	*xa*					$<_1$	*xo* ≡ *xu*		
$<_1$	*ga*					$<_1$	*go* ≡ *gu*		
$<_1$	*xe* ≡ *ge*	$<_1$	*gë*	$<_1$	*xi* ≡ *gi*			$<_1$	*xö* ≡ *xü* ≡ *gö* ≡ *gü*
					⋮				
$<_1$	*ta* ≡ *te* ≡ *da* ≡ *de*	$<_1$	*të* ≡ *dë*	$<_1$	*ti* ≡ *di*	$<_1$	*to* ≡ *tu* (⊃ *tüi*) ≡ *do* ≡ *du* (⊃ *düi*)	$<_1$	*tö* ≡ *tü* ≡ *dö* ≡ *dü*
$<_1$	*d'a* ≡ *d'e*	$<_1$	*d'ë*	$<_1$	*d'i*	$<_1$	*d'o* ≡ *d'u* (⊃ *d'üi*)	$<_1$	*d'ö* ≡ *d'ü*
					⋮				

Tables of dictionary headings in Hudum listed by both unfolded and folded letters are given here for the convenience of reference.

依展开式字母和折叠式字母排序的胡都木文词典标目表这里一并给出，以便参考。

Table 9 Dictionary headings in Hudum | 表 9 词典标目的胡都木文

(a) Listed by unfolded letters | 依展开字母排列

	-a	-e	-ë	-i	-o/u	-ö/ü
-	𐴀	𐴁	𐴂	𐴃	𐴄	𐴅
n-	𐴆		𐴇	𐴈	𐴉	𐴊
b-	𐴋		𐴌	𐴍	𐴎	𐴏
⋮	⋮					
x-	𐴐	𐴑	𐴒	𐴓	𐴔	𐴕
g-	𐴖				𐴗	
⋮	⋮					
t/d-	𐴘		𐴙	𐴚	𐴛	𐴜
d'-	𐴝		𐴞	𐴟	𐴠	𐴡
⋮	⋮					

(b) Listed by folded letters | 依折叠字母排列

	-a	-ɸ	-ë	-i	-o	-ö
ʔ-	𐴀	𐴁	𐴂	𐴃	𐴄	𐴅
n-	𐴆	■	𐴇	𐴈	𐴉	𐴊
b-	𐴋	■	𐴌	𐴍	𐴎	𐴏
⋮	⋮					
x-	𐴐	■	■	■	𐴔	■
ǵ-	𐴖	■	■	■	𐴗	■
g-	𐴑	■	𐴒	𐴓	■	𐴕
⋮	⋮					
t-	𐴘	■	𐴙	𐴚	𐴛	𐴜
d-	𐴝	■	𐴞	𐴟	𐴠	𐴡
⋮	⋮					

B Correspondence between graphetic characters and folded letters | 形码字符－折叠字母对应表

Correspondence between graphetic characters and folded letters is given as Table 10 for the convenience of reference. Single characters and character combinations are aligned in the table.

表 10 中给出了形码字符和折叠式字母之间的对应关系以便参考。表中的单字母和字母组合是对齐的。

Table 10 Correspondence between graphetic characters and folded letters | 表 10 形码字符–折叠字母对应表

Glyph | 图形				Graphetic character	Folded letter	Unfolded letter
IS	I	M	F			
单	上	中	下	形码字符	折叠字母	语音字母
𐴀	𐴁	𐴂	𐴃/𐴀	**A**	ʔ	—
					ʔɸ	e
					a	a
					ń	n
𐴆	𐴇	𐴈	𐴉/𐴆	**I**	i	i
					j	j
					ǐ	y
■	𐴋	𐴌/𐴋	𐴍/𐴎	**O**	o	o/u/ö/ü
𐴏	■	■	𐴏	**U**	u	o/u/ö/ü
					ŭ	w
■	𐴐	𐴑	𐴒	**X**	x	x
					ǵ	g
■	𐴖	𐴖	𐴗	**G**	g	x/g
■	𐴘	𐴙	𐴚	**L**	l	l
■	𐴛	𐴛	𐴜	**W**	ë	e
					w	w
■	𐴝	𐴝	𐴞	**H**	h	h
					ẑ	ẑ

Glyph | 图形				Graphetic character	Folded letter	Unfolded letter
IS	I	M	F			
单	上	中	下	形码字符	折叠字母	语音字母
□	□	𐴟	𐴠	AG	ŋ	ŋ
■	𐴡	□	□	AH	ɦ	h
□	■	𐴢	𐴣/𐴤	OA	ð	d
□	□	𐴥	□	II	ɨj	i
■	■	𐴦/𐴧	□	OI	ö	ö/ü
□	■	𐴢	𐴣/𐴤	OA	ð	d
■	■	𐴦/𐴧	□	OI	ö	ö/ü
■	𐴨	□	■	OO	ĉ	ĉ
□	□	𐴟	𐴠	AG	ŋ	ŋ
■	𐴩	𐴩	□	LH	ɬ	ɬ
■	𐴡	□	□	AH	ɦ	h
■	𐴩	𐴩	□	LH	ɬ	ɬ

(End of document | 文档结尾)