ISO/IEC JTC1/SC2/WG2 N3424R
L2/08-019R
2008-04-22

Universal Multiple-Octet Coded Character Set
International Organization for Standardization
Organisation Internationale de Normalisation
Международная организация по стандартизации

Doc Type: Working Group Document
Title: Proposal for encoding the Lisu script in the BMP of the UCS
Author: China
Status: Member Contribution
Replaces: L2/07-344 (N3317R2)
Action: For consideration by JTC1/SC2/WG2 and UTC
Date: 2008-04-22

0. Preamble. This revision of the document is merely to change the name of the script from *Old Lisu* to *Lisu* as per WG2's resolution at meeting #52. All script and character references are changed accordingly. Section 2 is also revised.

1. Introduction. There are 630,000 Lisu people in China, mainly distributed in the regions of Nujiang, Diqing, Lijiang, Dehong, Baoshan, Kunming and Chuxiong in the Yunnan Province. Another 350,000 Lisu live in Myanmar, Thailand and India. The population is increasing rapidly. In addition, at least 20,000 non-Lisu people in Yunnan, China, speak Lisu as their mother tongue. Many more in Yunnan and northern Myanmar speak Lisu as a second language. In Yunnan, speakers of other languages use Lisu for administration, religion, and bilingual education in schools. Lisu is considered a very vigorous language.

Somewhere between 1908 and 1914 a Karen evangelist from Myanmar by the name of Ba Thaw modified the shapes of Latin characters and created the Lisu script. Afterwards, British missionary James Outram Fraser and some Lisu pastors revised and improved the script. At present, about 200,000 Lisu in China use the Lisu script and about 160,000 in other countries are literate in it. Other user communities are mostly Christians from the Dulong, the Nu and the Bai nationalities in China.

The Lisu script is widely used in China in domains like education, publishing, the media and religion. Various schools and universities at the national, provincial and prefectural levels have been offering Lisu courses for many years (1952: Central National University; 1978: Yunnan Nationality University; 1985: Nujiang Medium Normal School). These schools have trained large groups of professionals in the Lisu language. In the publishing aspect, plenty of literature in the Lisu script has been published since 1952 by provincial and prefectural publishers (1952: Yunnan People's Publishing Agency; 1957: Yunnan Nationality Publishing House; 1981: Dehong Nationality Publishing House). These publications include dictionaries, song books, primers, readers, and textbooks. Among them, 145,000 copies of the 1994 Lisu primer edited by Yunnan Minority Language Commission and Nujiang Minority Language Commission have been distributed. As for the media, Yunnan People's Broadcasting Station launched a Lisu language broadcast in 1957. Two newspapers have been publishing sections in the Lisu script since their establishments (1954: *Dehong Tuanjie Bao*; 1983: *Nujiang Bao*). On the religious side, books published in the Lisu script includes the Bible and hymn books.

Globally, the Lisu script is also widely used in a variety of Lisu literature, including a bi-monthly published in Myanmar, some literature published in Australia, a primer published in 1922 with various revised forms still in print today, and plenty of Christian publishing such as Bibles, hymn books, and commentaries since 1921. There are also over 100 Lisu booklets in electronic form.

The Lisu script has recorded and summarised the Lisu people's rich experiences and achievements accumulated from their long-term production life. It is an extremely precious cultural heritage. Due to

the ongoing wide active use of the script, this proposal strongly recommends that the characters be encoded as part of the BMP.

2. Script Name. The Lisu script is commonly known in the West as the *Fraser* script, named after James Outram Fraser. However, such a naming scheme is not preferred for the following reasons:

(1) The name *Lao Lisu Wen*, which means 'Old Lisu writing', has been used for a long time in teaching, research, broadcasting, and relevant policies and regulations in China. Within the Lisu nationality, whenever *Lao Lisu Wen* is mentioned, it is unmistakably understood to mean the script being encoded in this proposal.

(2) The practice of naming a script after a particular originator should be avoided, as the development of a script is often a co-operative effort. The Lisu script was originally created by Ba Thaw, a Karen evangelist from Myanmar, and then British missionary James Outram Fraser and Lisu Christian clergymen amended and improved Ba Thaw's script. Therefore, it is not correct to name a script after a particular person.

(3) Many of the world's scripts are not named after a person. E.g., neither Latin nor Chinese is named after its creator despite his invention of the script.

We first proposed to call it the *Old Lisu* script to contrast specifically with the *New Lisu* writing system, which is a romanised orthography devised in the 1950s by the Chinese government and which is still in use today. However, some are concerned that the English word *old* has the connotation of being worn out or deteriorated through age. Furthermore, the consensus at meetings UTC #114 and WG2 #52 was that it is unnecessary to make a contrast in name between a script and a writing system (which does not need encoding). Therefore, we are now simply calling it the *Lisu* script in the Unicode domain.

In more recent years, an *Advanced Lisu* orthography has been proposed and used on the Internet in Thailand (Morse & Tehan, 2000). However, this is just another Latin-based writing system which does not need to be encoded. What could cause potential ambiguity in script names is a syllabic Lisu script developed in the early 1920s by Wa Renbo, a Lisu traditional priest in China. This script has now gone out of use, but should it be encoded later, a name qualifier will be needed to distinguish it from the *Lisu* script being proposed here.

3. Alphabet. There are 40 letters in the Lisu alphabet. Thirty consonants and 10 vowels were respectively written with 20 and seven Latin capital letters in upright and turned positions:

ꓐ	ꓑ	ꓒ	ꓓ	ꓔ	ꓕ	ꓖ	ꓗ	ꓘ
ꓙ	ꓚ	ꓛ	ꓜ	ꓝ	ꓞ	ꓟ	ꓠ	ꓡ
ꓢ	ꓣ	ꓤ	ꓥ	ꓦ	ꓧ	ꓨ	ꓩ	ꓪ
ꓫ	ꓬ	ꓭ	ꓮ	ꓯ	ꓰ	ꓱ	ꓲ	ꓳ
ꓴ	ꓵ	ꓶ	ꓷ					

3.1. Consonant Letters

ꓐ	[b]	ꓑ	[p]	ꓒ	[p^{h}]	ꓓ	[d]	ꓔ	[t]	ꓕ	[t^{h}]
ꓖ	[g]	ꓗ	[k]	ꓘ	[k^{h}]	ꓙ	[dʑ]	ꓚ	[tɕ]	ꓛ	[tɕ^{h}]
ꓜ	[dz]	ꓝ	[ts]	ꓞ	[ts^{h}]	ꓟ	[m]	ꓠ	[n]	ꓡ	[l]
ꓢ	[s]	ꓣ	[ʒ]	ꓤ	[z]	ꓥ	[ŋ]	ꓦ	[h]	ꓧ	[x]
ꓨ	[ɦ]	ꓩ	[f]	ꓪ	[w]	ꓫ	[ɕ]	ꓬ	[ʑ]	ꓭ	[ɣɑ]

Consonant letters have an inherent [ɑ] vowel unless followed by an explicit vowel letter. ꓱ LISU LETTER GHA sometimes represents a vowel and sometimes a consonant (e.g., ꓱ. CI ꓛU: A. ꓱ.), and so are letters W WA and Y YA. Letters ꓩ HHA and V HA represent allophones in complementary distribution: the former occurs only in a final imperative marker while the latter appears elsewhere, causing nasalisation to the whole syllable.

3.2. Vowel Letters

A	[ɑ]	Ɛ	[ɛ]	E	[e]	Ǝ	[ø]	I	[i]
O	[o]	U	[u]	Ո	[y]	ꓶ	[ɯ]	Ɑ	[ə]

With the exception of ꓶ UH and Ɑ OE, vowel letters starting a syllable have an unmarked glottal-stop onset. Letters E E, O O and U U can form diphthongs with a preceding Y YA (i.e., YE, YO and YU).

3.3. Encoding Model. It can be observed that a number of Lisu letters may look similar to certain Latin characters, yet it is best to encode the whole set separately for Lisu. This is primarily because the two scripts behave differently: Latin is bicameral while Lisu is unicameral. Section 11.1 addresses this in more detail.

4. Tone Letters. The Lisu script has six tone letters (Figures 6 and 13) that can be placed individually or in combination after the syllable to mark tones:

Orthography	Pitch	Lisu Name	English Name
.	55	MY.. TI.	MYA TI
,	35	N. PO..	NA PO
..	44	MY.. CY.	MYA CYA
.,	33	MY.. BO.,	MYA BO
;	42	MY.. N.	MYA NA
:	31	MY.. JƎ.,	MYA JEU

4.1. Simple Tones. When used individually, each of the six tone letters represents one simple tone. This set of six should be encoded separately despite resemblance to Latin punctuation marks. Again this is primarily because they have different behaviours: The tone letters are word-forming (gc=Lm) while the Latin punctuation marks are not (gc=Po). Forcing unification would create problems in determining word boundaries in text processes like word selection and whole-word searching. Section 11.2 addresses this in more detail.

Concerning TONE MYA CYA and TONE MYA BO (aka *mya po* outside China), it is theoretically possible to encode them as the following sequences:

.. *mya cya* = . MYA TI + . MYA TI

., *mya bo* = . MYA TI + , NA PO

However, this is not preferred in view of the following:

(1) Script unity: These two tones are part of a well-defined set of basic tone letters. The Lisu user community regards the set of six simple tones as foundational to their language and culture and has expressed a strong desire to keep the six together in the coded character set. Leaving these two tones out as sequences would destroy script unity.

(2) Search errors: Because MYA TI would be a sub-string of *mya cya* and *mya bo* under a sequential encoding approach, searches based on binary string comparison would yield erroneous results. E.g., a search for the string **w.** would incorrectly match occurrences of **w..** and **w.,** because they contain the search string as a sub-string. This is not acceptable to the Lisu user community.

To combat this error, one might use collation-based searching provided that the DUCET be augmented with entries mapping each sequence to a single collation element (and hence treating the sequence as a single collation grapheme; see UTS#10). However, this is both more difficult to implement and computationally more expensive than traditional binary string comparison. In addition, not all applications will implement collation-based searching. Given that the Lisu see that MYA TI should never match MYA CYA or MYA BO, this would not be a solution.

Another work-around would be to remember to set the whole-word flag for every search to circumvent the problem, but this would create unnecessary inconvenience for the user.

(3) Tone spacing: Every simple tone letter should fit into a single em square. Encoding tones as sequences would create large intra-sequence spacing in mono-space fonts. This is undesirable but can be addressed by simple kerning.

(4) Legacy implementations: Document L2/07-423 shows that all available Lisu legacy encodings have separate code points for these two tone letters. This means Lisu users have already been enjoying implementations that do not bring about any of the above problems. If *advancing* to Unicode would mean unnecessary troubles especially in searching caused by a sequential tone encoding, users would likely discard Unicode and continue to use legacy implementations.

A better approach is to encode them as units, which will solve all the above problems. The main concern here is the possibility of encoding confusion (multiple spellings). E.g., MYA CYA may be represented as a unit at one place and a sequence at another. However, this is more a fear than a problem because:

(1) Lisu users reported not knowing of anyone typing the sequence instead of the unit; every user always immediately asks where the unit key is on the keyboard.

(2) Simple keyboard rules can be implemented to forbid tone sequences of **.** MYA TI **+ .** MYA TI and **.** MYA TI **+ ,** NA PO.

Given the above relative pros and cons, it is proposed that MYA CYA and MYA BO be assigned separate code points along with the other four members in the set.

4.2. Combination Tones. The first four tone letters can be used in combination with the last two to represent tones like **.; ,; .: ,: ..:** (of which only **,;** is still in use whereas the rest are now rarely seen in China). Figure 14 lists all eight combinations.

It has been suggested that these eight combination tones be encoded as units to facilitate searching. This would not be feasible, however, because:

(1) It is also possible to obtain other permutations outside the four-by-two framework. Although so far the only attested occurrences are found in a Lisu song transcription where they are used to mark special intonations and vowel lengths as the song is sung (Figure 5), there is nothing that prevents other permutations from being used on other occasions. All 30 possible combinations would then have to be encoded.

(2) It would create a wide opportunity for multiple spellings that cannot be checked except by a large set of keyboard rules that specifically forbids each possible wrong spelling. This would cause unnecessary complications in implementations.

A more practical approach is to encode combination tones as sequences of the six simple tone letters. The following lists some example sequences:

.: = **.** MYA TI + **:** MYA JEU

,; = **,** NA PO + **;** MYA NA

..: = **..** MYA CYA + **:** MYA JEU

.,: = **.,** MYA BO + **:** MYA JEU

Concerns with this approach are similar to some of those pertaining to a sequential encoding of *mya cya* and *mya bo* mentioned in Section 4.1. However, these can be addressed as follows:

(1) Search errors: Searching for a simple tone will return matches for combination tones starting with that simple tone as well. However, this *is* acceptable to the user community and is not a problem. After all, combination tones are in fact glides going from one simple tone to another (i.e., they are compound simple tones).

(2) Tone spacing: It is desirable to have each combination tone fit into a single em square, but a sequential encoding will create large intra-sequence spacing in mono-space fonts. This can, however, be solved by simple kerning.

The above analysis suggests that encoding combination tones as sequences is a much better approach.

Note that the tone sequence **..:** coincides with the ending intonation of a question and was traditionally used to signal a question at the end of a sentence, usually followed by a = PUNCTUATION FULL STOP, as in Figure 9. Since the '80s, however, this has been replaced by the European QUESTION MARK.

5. Other Modifier Letters. Nasalised vowels are denoted by a *nasalisation mark* following the vowel, as in oʼ [ʔõ^{33}] 'goose'. This word-forming character is not encoded separately but is represented by U+02BC ʼ MODIFIER LETTER APOSTROPHE, which has the same shape and behaviour (gc=Lm) and is used in similar contexts—it already denotes glottal stop, glottalisation and ejective in other languages and is naturally appropriate for denoting nasalisation in Lisu as well.

The vowel *A glide*, pronounced [ɑ] without an initial glottal stop (and normally bearing a 31 pitch), is written after a verbal form to mark various aspects, as in NU JE.,_ ʌO., [nu^{33}dʒe^{33}ɑ^{44}ŋo^{33}] 'you will go' and GO., LƎ., ʌ_ MI., [go^{33}lø^{33}ŋɑ^{44}ɑ^{31}mi^{33}] 'but'. This word-forming character does not need to be separately encoded but can be represented by U+02CD ˍ MODIFIER LETTER LOW MACRON, which has the same behaviour (gc=Lm) and general shape—except that it is generally rendered below the baseline whereas the *A glide* sits on it (Figure 1), but this can be adjusted by font implementations such as the one used in this proposal. While it is generally used to denote a low-level tone, this does not prevent us from using it to represent the Lisu *A glide*, which is in fact a vowel contraction usually bearing a low-falling tone.

6. Digits and Separators. There are no Lisu digits. The Lisu use Arabic numerals for counting (Figure 16). The thousand separator and the decimal point are represented with the Latin comma (Figure 17) and the Latin period, respectively. To separate chapter and verse numbers, the Latin colon (Figure 1) and semi-colon are used. These can be readily distinguished from resembling tone letters by their numerical context.

7. Punctuation. The Lisu use about 10 punctuation marks. First we examine the two that need to be encoded and then we examine the rest already in the Unicode Standard.

7.1. Lisu Punctuation. **-.** LISU PUNCTUATION COMMA and **=** LISU PUNCTUATION FULL STOP are respectively used to denote a lesser and a greater degree of finality. These characters may look like (sequences of) Latin punctuation, but because they are part of a patterned set of marks in Lisu, it is best to encode them together with the other members of the set for script unity. Additional considerations specific to each character are given as follows:

U+A4FE **-.** PUNCTUATION COMMA: One possibility to encode it is to use the sequence <U+002D, U+002E>. This is not preferred in view of the following:

(1) Glyphic distinction: The representative glyph used in this proposal is only one of several possible renderings. Figure 1 and Figure 2 show an alternative rendering with the dot on the same level as the bar. Figure 4 illustrates yet another rendering option, with the dot below the bar but right justified with it. This argues for a distinct identity of PUNCTUATION COMMA.

(2) Behaviour difference: It cannot be properly processed as a unit if encoded as a sequence. Even though line-breaking can be handled correctly according to UAX #14 (LB21: × HY; Pair Table: HY ^ IS), word boundaries cannot be correctly determined. In particular, the sequence will be seen as two words instead of one according to UAX #29 (WB14: Any ÷ Any). This would be undesirable unless **all** applications can be tailored to recognise the sequence as one unit.

U+A4FF **=** PUNCTUATION FULL STOP: Though it looks like U+003D = EQUALS SIGN, they cannot be unified because of the following reasons:

(1) Glyphic distinction: The former is shorter and bolder while the latter, longer and thinner.

(2) Behaviour difference: They behave differently in relation to text processes. The former is a sentence-ending punctuation (gc=Po) that prevents a line break before (lb=EX) but allows word breaks both before and after (UAX #29, WB14) while the latter, a symbol (gc=Sm) with an alphabetic line-breaking property (lb=AL) and is word-forming (UAX #29, WB5). Unless special tailoring can be done to **all** applications, unification would not be a good solution. It should also be noted that the two characters do not occur in the same context (i.e., EQUALS SIGN is used as per its mathematical usage).

7.2. Other Punctuation. Over time various European punctuation marks have been adopted in Lisu (Figure 11). A few Chinese punctuation marks are also used in China (Figures 12 and 21). The following table lists all known adopted punctuation with respective contexts of use:

Character	Context of Use	Remarks
U+2010 HYPHEN	Syllable separation in names	Preferred to U+002D HYPHEN-MINUS, which has ambiguous semantics (TUS 5.0)
U+003F QUESTION MARK	Questions	Replaces **..:=** (see Section 4.2)
U+0021 EXCLAMATION MARK	Exclamations	
U+0022 QUOTATION MARK	Quotations	
U+0028 LEFT PARENTHESIS	Parenthetical notes	
U+0029 RIGHT PARENTHESIS	Parenthetical notes	
U+2026 HORIZONTAL ELLIPSIS	Omission of words	Always doubled in Chinese usage
U+300A LEFT DOUBLE ANGLE BRACKET	Book titles	Chinese punctuation
U+300B RIGHT DOUBLE ANGLE BRACKET	Book titles	Chinese punctuation

Since these marks are already encoded in the Standard (in the C0 Controls and Basic Latin, General Punctuation, and CJK Symbols and Punctuation blocks) and are not used in ways that their properties cannot handle, no separate encoding is needed.

8. Line-breaking. A line break is not allowed between any pair of characters in the following set:

{a letter in the alphabet, a tone letter, MODIFIER LETTER APOSTROPHE, MODIFIER LETTER LOW MACRON}

A line break is prohibited before a punctuation despite intervening spaces. There is no line-breaking hyphenation except in proper nouns, where a break is allowed after the syllable separator (HYPHEN). All these can follow normal rules when correct categories have been assigned (Section 12).

9. Word-breaking. The Lisu script separates syllables using a space or, for proper names, a hyphen. In the case of polysyllabic words, it can be ambiguous as to which syllables join together to form a word. Thus for most text processing at the character level, a syllable (starting after a space or punctuation and ending before another space or punctuation) is treated as a word except for proper names where the occurrence of a hyphen holds the word together.

10. Collating Order. The sorting order of the Lisu alphabet generally starts with sequences of voiced, voiceless unaspirated, and voiceless aspirated consonants. The order is more or less fixed before ꓩ HHA with only slight differences afterwards in the position of ꓛ FA (cf. Figures 6 through 9). This traditional order is evidenced in available literature including a primer, a dictionary and two textbooks. However, due to the fact that ꓖ GHA most often represents a consonant rather than a vowel, in China it has recently come to be placed after ꓬ YA as the last consonant (rf. Section 3). As for tones, Figure 6 shows the traditional order that is in use outside China, but in China, TONE MYA NA has been put before TONE MYA JEU for teaching purpose for over 20 years (Figure 13; rf. Section 4). Tones are followed by MODIFIER LETTER LOW MACRON and MODIFIER LETTER APOSTROPHE in that order. The collating order proposed below reflects the three aforesaid phenomena:

. *tone mya ti* U+A4F8 < , *tone na po* U+A4F9 < .. *tone mya cya* U+A4FA < ., *tone mya bo* U+A4FB < ; *tone mya na* U+A4FC < : *tone mya jeu* U+A4FD < ˍ *low macron* U+02CD < ’ *apostrophe* U+02BC < ꓐ *ba* U+A4D0 < ꓑ *pa* U+A4D1 < ꓪ *pha* U+A4D2 < ꓓ *da* U+A4D3 < ꓔ *ta* U+A4D4 < ꓕ *tha* U+A4D5 < ꓖ *ga* U+A4D6 < ꓗ *ka* U+A4D7 < ꓫ *kha* U+A4D8 < ꓙ *ja* U+A4D9 < ꓚ *ca* U+A4DA < ꓬ *cha* U+A4DB < ꓜ *dza* U+A4DC < ꓝ *tsa* U+A4DD < ꓭ *tsha* U+A4DE < ꓟ *ma* U+A4DF < ꓠ *na* U+A4E0 < ꓡ *la* U+A4E1 < ꓢ *sa* U+A4E2 < ꓣ *zha* U+A4E3 < ꓤ *za* U+A4E4 < ꓥ *nga* U+A4E5 < ꓦ *ha* U+A4E6 < ꓧ *xa* U+A4E7 < ꓩ *hha* U+A4E8 < ꓛ *fa* U+A4E9 < ꓫ *wa* U+A4EA < ꓬ *sha* U+A4EB < ꓬ *ya* U+A4EC < ꓖ *gha* U+A4ED < ꓮ *a* U+A4EE < ꓯ *ae* U+A4EF < ꓰ *e* U+A4F0 < ꓱ *eu* U+A4F1 < ꓲ *i* U+A4F2 < ꓳ *o* U+A4F3 < ꓴ *u* U+A4F4 < ꓵ *ue* U+A4F5 < ꓶ *uh* U+A4F6 < ꓷ *oe* U+A4F7

Outside China a somewhat different sort order is used in that tones follow the traditional order and letters after ꓛ FA have different placements:

… < ., *tone mya bo* U+A4FB < : *tone mya jeu* U+A4FB < ; *tone mya na* U+A4FA < ˍ low *macron* U+02CD < … < ꓛ *fa* U+A4E9 < ꓬ *sha* U+A4EB < ꓖ *gha* U+A4ED < ꓫ *wa* U+A4EA < ꓬ *ya* U+A4EC < ꓮ *a* U+A4EE < …

11. Encoding Issues. It can be observed that a number of Lisu characters may look similar to certain Latin characters. This leads some to believe they belong to the same script and should be unified. After a series of dialogue with language experts and UTC members, a number of issues have been identified around the encoding models for the alphabet and for the tone letters. These issues are addressed in the following two sections. In each section, issues pertaining to unification will be examined first followed by those concerning a separate encoding.

11.1. Issues around the Alphabet. Resemblance between certain Lisu and Latin letters naturally warns of potential confusion. It is understandable that a unification approach could avoid this problem. However, the following issues must first be considered:

(1) Script definition: According to Lyons et al (2001), a script is "a maximal collection of characters used for writing languages or for transcribing linguistic data that share common characteristics of appearance, share a common set of typical behaviours, have a common history of development, and that would be identified as being related by some community of users." In the case of Lisu vs. Latin, only the first of the four requirements is met. Whether they share a common history of development is still up to debate. What is clear is that they have different behaviours and no known user community identifies the two as being related. Therefore, they should not be considered the same script.

(2) Behaviour difference: None of the Lisu letters has case whereas all Latin ones do. Unification would mean forcing Lisu to adopt an imaginary normative property, namely, case. This would create a vulnerability to processes capable of case-folding, introducing the opportunity for lower-case Latin characters to appear in Lisu texts, which is unacceptable because these characters are meaningless and unrecognisable to Lisu readers. The immediate implication would be zero usability of any Lisu letter in IDNs, for in today's browsers, all IDNs are case-folded before being presented to the user. Another implication would be potential errors in text editing. E.g., a search for Lisu words might return lower-case Latin counterparts if such exist in the same text. The user could try to remember to set the case-sensitive flag for every search to guarantee correct matches, but this would inevitably cause some inconvenience.

Some have referred to the decisions to represent Classical Latin and Sencoten, two unicameral writing systems, with Latin capital letters and argued that the lack of case does not necessarily make Lisu a distinct script from Latin. However, these examples cannot be used as a basis of comparison with Lisu in the context of unification because:

a) Both Classical Latin and Sencoten, the latter being found around the southern tip of Vancouver Island, BC, Canada, are used in a Latin script context in that readers of these languages are probably at least semi-literate in a Latin-based language and able to recognise lower-case letters. This is not the case for Lisu readers.

b) Classical Latin is a dead language used for academic purposes only. Nobody is going to need it in IDNs or file names or do any processing with it beyond appropriate rendering in books and perhaps sorting. In these cases no tailoring will be done or truly required to be implemented and if it is, only in very particular applications which can be modified to support this particular requirement.

c) Sencoten does have a lower-case letter 's' (Harvey, 2005), and so is not a truly unicameral system.

d) Sencoten is listed as an extinct language that seems to be undergoing some revival with reportedly 185 students from nursery to Grade 9 being educated in a Sencoten curriculum (Saanich Indian School Board, 2004), but the likelihood of there ever being monolingual speakers of the language is very low indeed.

(3) No implementation: While certainly not the ideal solution, in theory it is possible to implement tailored case mappings directly in code (see TUS 5.0 Section 5.18, pp. 186–187) to guarantee that no upper-case letter will ever get mapped to lower case in matching, searching, sorting, or any text process involving Lisu texts. However, this is an immense task since every application will have to be specially tailored. Furthermore, it is highly unlikely that anyone is going to do the required implementation for a small minority, especially with such far-reaching consequences as changing the casing for **all** upper-case letters in **ASCII**. Interestingly, this is best illustrated by referring to the examples of Classical Latin and Sencoten: To date, it is clear that no implementation beyond perhaps a font and keyboard has been done since there are no special case mappings created for

either of these languages. In fact, according to available evidence, there should already be a locale-specific mapping for Classical Latin and for Sencoten—the addition of four Latin characters to cover Sencoten orthography was accepted in 2004 and rolled out in TUS 4.1 in 2005. The continued absence of these mappings even through TUS 5.0 indicates that the Unicode authority in concern failed to do its job when encoding these languages.

(4) Data corruption: Even if someone should really set out to implement tailoring for all applications, it will be unusable beyond application-level text processing. P.189 of TUS 5.0 states: "In most environments, such as in file systems, text is not and cannot be tagged with language information. In such cases, the language-specific mappings *must not* be used. Otherwise, data structures such as B-trees might be built based on one set of case foldings and used based on a different set of case foldings. This discrepancy would cause those data structures to become corrupt. For such environments, a constant, language-independent, default case folding is required". Take Microsoft Windows for example, because file name lookups are done with caseless matching, if language-specific case mappings were used, files with names containing lower-case Latin letters would only be retrievable in an English locale (where, e.g., 'A' and 'a' would match) but not in a Lisu locale (where 'A' would map to itself).

(5) Precedence: In Cherokee (U+13A0..U+13FF) over 20 characters look like Latin and yet they are not unified. Why should Lisu?

(6) Imaginary creation: According to the case-folding stability policy, if an upper-case letter is added to the Standard without a corresponding lower case, no corresponding lower-case letter can be added later. This restriction has led some, when unifying with Latin, to create an imaginary lower-case counterpart for encoding with an upper-case letter just in case the former may be needed in the future. This is apparently why the added characters for Sencoten have non-existent lower-case forms (see U+2C65 and U+2C66) which seem to have been added purely for case-folding purposes. Another example is Richard Cook's proposal (N3326) to encode a Latin small letter 'turned j' as the lower-case counterpart to capital letter 'turned J' even though there is no lower-case 'turned j' in Lisu. These examples provide yet another vivid argument against unification: Creating some non-Lisu (or non-Sencoten) characters in order to make the script work with Latin clearly proves that it is **not** Latin!

(7) Visual confusion: The reason that encoding imaginary turned lower-case letters for Lisu is so problematic is the intolerable confusion that would arise with certain upright letters, e.g., d vs. turned p, l vs. turned l, n vs. turned u, p vs. turned d, and q vs. turned b.

As seen from the above, unification would actually create more problems than it could solve and hence would be infeasible.

A better approach is to encode Lisu separately as a distinct script. The major advantage of a separate encoding lies in the fact that behaviour difference can be accounted for at source. To reflect their lack of case, all Lisu letters can be assigned the general category of Lo with no case mappings. It will then be impossible to produce lower-case Latin even if some processes decide to case-fold. This means Lisu letters can be safely used in file systems and IDNs and correctly processed by text applications without the need for any case tailoring. Now the main concern with this approach is the potential confusion between certain Lisu letters and their Latin look-alikes. Questions on legacy implementations have also been raised. These issues are addressed as follows:

(1) Legacy implementations: Some have argued against a separate encoding based on the presupposition that existing implementations use the ASCII Latin encoding (plus a small extension) to represent Lisu letters. The fact is, according to document L2/07-423, all available legacy fonts hack the ASCII code space, discarding ASCII semantics, to encode the Lisu alphabet

as a distinct set separate from Latin. Any counter-argument based on legacy encodings is therefore not valid.

(2) Input methods: Some believe that encoding Lisu separately would make input methods complicated because they would have to distinguish Latin capital letters from Lisu letters. However, this scenario will only occur when you create a two-in-one keyboard that allows you to type both Lisu and Latin letters. This is unnecessary, as it is highly doubtful that such a keyboard will be needed. In practice, separate keyboards are used for typing Lisu and, say, English. To switch from one language to another, the user just toggles the keyboard. There is no need to mix them together.

(3) Data corruption: A separate encoding will allow the co-existence of Lisu and Latin letters in the same text. Because of resemblance between certain members of the two sets, some fear the user might accidentally input a letter from the wrong script resulting in corrupt data. While this is a valid concern, it is of no different a nature than the potential for confusion among Latin, Greek and Cyrillic upper-case letters. In practice, confusion is very unlikely because:

a) A separate keyboard is used to input Lisu (or Greek or Cyrillic) letters. To type Latin letters, the user has to use a different keyboard.

b) Latin text most often contains both upper- and lower-case letters—as a rough estimate, 90% of all printed matter is lower case, which carries no potential for confusion with Lisu letters at all. Even in the case where a single Latin word is embedded in a paragraph of Lisu text (Figure 15), given the large proportion of lower-case letters in the word, which cannot be produced by a Lisu keyboard, the chance of confusing the sole capital letter is remote.

c) Lisu letters are traditionally rendered in a sans serif font in electronic documents. For Latin letters, a serif font is used. Figures 8 and 15 demonstrate clear distinction of the two sets by way of different font faces. While it is true that on occasion serif fonts have been used even for Lisu letters, such usage is confined to specialised domains like decorations, headings, and book prefaces in monolingual or non-Latin bilingual contexts such as books published in China (Figures 7, 12, and 21).

(4) IDN spoofing: Some are concerned that the similarities of certain Lisu letters with Latin characters may allow spoofing of IDNs. They believe if the two are not unified, then Lisu will have to be excluded from internet protocols. This concern is addressed as follows: In theory, IDNA allows IDNs with labels consisting entirely of ASCII capital letters to be input, resolved and displayed to the user. This indeed allows confusion in that IDNs drawn from different scripts can look the same and the user is unlikely to tell the difference. E.g., SPACE.BC.CA will look the same in Latin, Cyrillic, Cherokee, and Halfwidth and Fullwidth Forms (though Cyrillic capitals and Halfwidth and Fullwidth Forms are not allowed to be output according to `idnchars.txt` in UTS #39). However, this is already an existing condition and encoding Lisu separately is not going to create a new problem. If it is believed that Lisu should be banned from IDNs on the basis of visual similarity with Latin, then Cherokee and other similar-looking scripts should be banned as well. This is clearly undesirable. One approach would instead be to remove all upper-case Latin characters from `idnchars.txt` as being allowed to be output, then there would be no problem of confusability. Unfortunately, this is unlikely to happen. Another approach would be to implement rules on the domain authority side as well as on the client side.

As part of their anti-spoofing policies, domain authorities (whether over TLDs or sub-domains) can require that all code points in any IDN label belong to a single script so that it is not possible to create mixed-script confusables. In addition, certain characters such as Lisu tone letters and punctuation can be prohibited in IDNs to avoid confusion with Latin punctuation and symbols

commonly used in IDNs. One can also enforce restrictions to remove the possibility of whole-script confusables by simply disallowing any string that is **entirely** confusable with ASCII, but allowing strings that contain at least one non-confusable character (one of those Lisu letters that look like turned Latin capital letters). As long as one character in the string is unambiguous, and as long as mixed scripts are not allowed, then that string is not going to be visually or functionally confusable with anything from Latin. For example, if someone were to try to register www.MICROSOFT.com using Lisu letters, it would not be allowed because all of the letters in MICROSOFT are confusable with upper-case Latin (even though the IDN clearly stands out against the usual case-folded format displayed in browsers). But if the string MICROSOFT were changed to MꓤCROSOFT, containing one non-confusable character ꓤ LISU LETTER AE, then such a name could be allowed since the string itself is not confusable: it consists of characters from one script block and the whole string is not whole-script confusable with Latin due to the one non-confusable character in it.

If the client wants to make the same check, it can, since it is merely a test to see whether a particular string contains any of a set of characters or not. And since it is upper-case ASCII that is in concern, the probability of a single-syllable string being whole-script confusable would be 20/30 * 5/10 = 1/3 given the general Lisu syllable structure is CV. That gives us a 2/3 chance of a single-syllable label being acceptable. In labels with multiple syllables or exceptional CVC and CVV syllables the probability of acceptance is even higher. This would indeed be very much better than losing all of Lisu in IDNs. And even not yet implemented with this simple check, today's browsers (and certain plug-ins to older browsers) already have other built-in measures that greatly reduce confusion. Under the IDNA model, as long as there is one non-ASCII character in a label, the whole string is case-folded and normalised. In today's browsers (e.g., Firefox 2.0 and Internet Explorer 7.0), however, even all-ASCII IDNs are case-folded before being presented to the user. Since there is no case in Lisu, case-folding will yield the same string whereas Latin characters will be converted to lower case. This easily distinguishes a Lisu letter from a Latin one. Another method, which the IDN-enabling plug-in Quero Toolbar 2.1.0 for older Internet Explorers reportedly adopted, is to display a label with mixed scripts in different colours to warn the user. This can serve as another safeguard on top of the recommendation that domain authorities disallow mixed-script labels altogether. Alternatively, browsers (e.g., Safari) can be configured to display punycode URLs for non-ASCII IDNs. A more advanced approach, which both Mozilla and Opera are using, is to turn on IDN display only for domains run by registries who are taking appropriate anti-spoofing precautions. With all these registry and client measures, the probability of spoofing with Lisu and Latin is basically reduced to zero.

The above analysis suggests that encoding Lisu letters separately is a far better approach than unification, which fails to account for normative differences between Lisu and Latin while having its own implementation problems and usage limitations.

11.2. Issues around Tone Letters. It can be observed that, with the exception of **.,** TONE MYA BO, all Lisu simple tone letters resemble certain Latin punctuation characters. To avoid confusion, some have suggested that they be unified. However, this would lead to undesirable effects in several areas:

(1) Text segmentation: As mentioned in Section 4.1, tone letters are word-forming. If we unify them with their Latin look-alikes, which are non-word-forming punctuation, it would create problems in processes that rely on word-boundary information. On the internet, e.g., search engines would return wrong matches. In word-processing, the user would not be able to select a word by double-clicking, for tone letters would be left out. Yet cursor selection would still be a work-around. A more problematic case would be whole-word searching, whether based on lexical or collation comparison, especially given it is common practice to omit certain tones in writing. E.g., a whole-word search for a toneless W would incorrectly match all toned versions except those starting with MYA BO. (Note that this is a search where the user has explicitly set the whole-word flag and is

different from the general search problem mentioned in Section 4.2. Here the user would rightfully expect a whole-word match; returning sub-string matches would be unacceptable.) In the editing of, say, a 1000-page book, it would really be a pain to manually examine each match and discard wrong ones.

To combat this problem, it has been suggested that separate code points be assigned to all combination tones. While this could prevent a toneless `W` from matching combination-toned versions in the above example, it would still allow matches with those having simple tones and therefore would not work. Furthermore, for reasons covered in Section 4.2, encoding combination tones as units is to be avoided.

Another attempt to account for the word-forming nature of tone letters is to tweak the word- and sentence-breaking rules in UAX#29. The problem with this is that these rules are context-dependent in nature whereas the choice to interpret, say, a unified Latin period as a punctuation (warranting a break) or tone letter (prohibiting a break) is not always so. E.g., the trailing dot in the string `B.` is normally considered a tone letter in prose but must be treated as a period in list numbering (Figure 4, red circle), where `B` is the list number and the dot a separator from the list item. In such environments, no difference in context exists and it is not possible to set computational rules to honour both sets of breaking behaviours. Even application tailoring would be out of the question in this case.

(2) Glyphic distinction: Although the two sets of characters look alike, they are not the same. In general, Lisu tone letters are heavier than Latin punctuation. Take the first tone letter for example, according to Morse, it must have a diameter of at least 175% of the base stem width so that people can see it well. A typical Latin period, on the contrary, is only 110-115% wide and it is not uncommon for fonts (e.g. Arial) to represent it with a square rather than a circle. Unification, therefore, would destroy glyphic differences. In spite of this, there is actually a legacy implementation that unifies four tone letters with Latin punctuation (L2/07-423 Section 4). In this case, whether a dot represents a tone or a punctuation cannot be distinguished by its shape.

(3) Tone spacing: User feedback indicates that tone letters have unique spacing specially designed for combinatorial use. At the same time, Latin punctuation marks are also fixed with specific spacing. Using punctuation to represent tones would result in poor spacing not acceptable in publishing quality materials. This problem, however, could be solved by simple kerning.

As the discussion above shows, unification does not distinguish the difference in normative properties between punctuation and tone letters. In particular, it is not able to account for tone characters being word-forming and thus leads to erroneous results in a number of processes.

A better approach is to encode tone letters separately from Latin punctuation. This approach adequately addresses the word-forming nature of tone letters so that correct word boundaries can be established in all processes to yield meaningful results. In addition, shape differences between tone letters and punctuation are preserved, and so are spacing differences. The main concern here is the potential for confusion between tone letters and punctuation due to their resemblance. However, as the following paragraphs explain, this is not as problematic as it would seem:

(1) Smart implementation: Restricting the keyboard to produce only the tone letters could solve the problem, but as Section 6 shows, the use of Latin punctuation as separators in Lisu number representations necessitate the ability to output punctuation marks in addition to the tone letters. (Even so, there is no need to output the two-dot leader, the Latin look-alike of MYA CYA, for a double-dot is used only for tone-marking and nothing else).

A possible solution is to design a smart keyboard to output the correct characters by context using the same set of keys. E.g., a dot after a letter in the alphabet would be a tone letter whereas one after a digit would be a punctuation. This will work in most cases except when list numbering is involved, as mentioned above, where the dot after a letter must be interpreted as a period rather than a tone letter. In this case, the keyboard can be augmented with a dictionary of valid single-letter first-tone words. E.g., a look-up will reveal that B. is not a valid word and so a dot after LETTER BA must be a period. In those cases where a valid word exists in the dictionary, the keyboard can output the default tone character and allow a user override. Given that:

a) list numbering is relatively less frequent compared to running text (where a tone letter is output by default after a letter in the alphabet),

b) the use of letters in the alphabet to mark list items only applies when multi-levels lists are involved (cf. Figure 4), which further reduces the frequency, and

c) only the first tone letter is affected, whereas the other three can still be determined by context,

a smart keyboard with dictionary look-up plus user override should be sufficient for all perceivable purposes.

(2) Limited damage: Even when a contextual keyboard is not used, one can always map the tone letters to the punctuation keys on a standard Latin keyboard. So long as the tone letters are obvious and the punctuation marks less so (e.g., accessible only via a control key), according to user feedback, people can learn which *dot* to type, for example. And the most likely error, if any, would be to type a tone instead of a Latin punctuation when the latter is needed. E.g., typing a tone letter into a number would just make calculations not work properly in a spreadsheet program; the user would simply need to retype with the correct punctuation. This would not be a serious mistake and would be acceptable to the user community.[1] Moreover, damage would be limited to the single user typing the bad data or that community that use that data alone. It would certainly not cause any unexpected troubles to software implementors.

(3) Small community: The only people going to have any problems with tone-punctuation confusion, if at all, would be Lisu speakers, who constitute only a small minority. The vast majority of computer users are not affected.

In conclusion, a unification approach can avoid confusion but will create unsolvable text segmentation problems, whereas under a separate encoding scheme correct text segmentation is ensured and the concern about confusion can be addressed by a smart keyboard implementation. It is therefore proposed that simple tone letters be encoded separately as laid out in Section 4.1.

12. Unicode Character Properties. All letters in the alphabet have a general category of Lo.

```
A4D0;LISU LETTER BA;Lo;0;L;;;;;N;;;;;
A4D1;LISU LETTER PA;Lo;0;L;;;;;N;;;;;
A4D2;LISU LETTER PHA;Lo;0;L;;;;;N;;;;;
A4D3;LISU LETTER DA;Lo;0;L;;;;;N;;;;;
A4D4;LISU LETTER TA;Lo;0;L;;;;;N;;;;;
A4D5;LISU LETTER THA;Lo;0;L;;;;;N;;;;;
A4D6;LISU LETTER GA;Lo;0;L;;;;;N;;;;;
A4D7;LISU LETTER KA;Lo;0;L;;;;;N;;;;;
```

[1] On the contrary, if Latin punctuation keys were to retain their default mappings, then the error would be in reverse direction. Feedback from the user community reveals that, when using a standard Latin keyboard and a legacy font with separate tone letter encodings (L2/07-423 Section 2), a particularly common error is typing a Latin punctuation instead of a Lisu tone letter after a syllable. Such an error is more serious and is unacceptable. Care must therefore be taken to modify the punctuation keys' mappings to produce tone letters instead if a non-contextual keyboard is used.

```
A4D8;LISU LETTER KHA;Lo;0;L;;;;;N;;;;;
A4D9;LISU LETTER JA;Lo;0;L;;;;;N;;;;;
A4DA;LISU LETTER CA;Lo;0;L;;;;;N;;;;;
A4DB;LISU LETTER CHA;Lo;0;L;;;;;N;;;;;
A4DC;LISU LETTER DZA;Lo;0;L;;;;;N;;;;;
A4DD;LISU LETTER TSA;Lo;0;L;;;;;N;;;;;
A4DE;LISU LETTER TSHA;Lo;0;L;;;;;N;;;;;
A4DF;LISU LETTER MA;Lo;0;L;;;;;N;;;;;
A4E0;LISU LETTER NA;Lo;0;L;;;;;N;;;;;
A4E1;LISU LETTER LA;Lo;0;L;;;;;N;;;;;
A4E2;LISU LETTER SA;Lo;0;L;;;;;N;;;;;
A4E3;LISU LETTER ZHA;Lo;0;L;;;;;N;;;;;
A4E4;LISU LETTER ZA;Lo;0;L;;;;;N;;;;;
A4E5;LISU LETTER NGA;Lo;0;L;;;;;N;;;;;
A4E6;LISU LETTER HA;Lo;0;L;;;;;N;;;;;
A4E7;LISU LETTER XA;Lo;0;L;;;;;N;;;;;
A4E8;LISU LETTER HHA;Lo;0;L;;;;;N;;;;;
A4E9;LISU LETTER FA;Lo;0;L;;;;;N;;;;;
A4EA;LISU LETTER WA;Lo;0;L;;;;;N;;;;;
A4EB;LISU LETTER SHA;Lo;0;L;;;;;N;;;;;
A4EC;LISU LETTER YA;Lo;0;L;;;;;N;;;;;
A4ED;LISU LETTER GHA;Lo;0;L;;;;;N;;;;;
A4EE;LISU LETTER A;Lo;0;L;;;;;N;;;;;
A4EF;LISU LETTER AE;Lo;0;L;;;;;N;;;;;
A4F0;LISU LETTER E;Lo;0;L;;;;;N;;;;;
A4F1;LISU LETTER EU;Lo;0;L;;;;;N;;;;;
A4F2;LISU LETTER I;Lo;0;L;;;;;N;;;;;
A4F3;LISU LETTER O;Lo;0;L;;;;;N;;;;;
A4F4;LISU LETTER U;Lo;0;L;;;;;N;;;;;
A4F5;LISU LETTER UE;Lo;0;L;;;;;N;;;;;
A4F6;LISU LETTER UH;Lo;0;L;;;;;N;;;;;
A4F7;LISU LETTER OE;Lo;0;L;;;;;N;;;;;
A4F8;LISU LETTER TONE MYA TI;Lm;0;L;;;;;N;;;;;
A4F9;LISU LETTER TONE NA PO;Lm;0;L;;;;;N;;;;;
A4FA;LISU LETTER TONE MYA CYA;Lm;0;L;;;;;N;;;;;
A4FB;LISU LETTER TONE MYA BO;Lm;0;L;;;;;N;;;;;
A4FC;LISU LETTER TONE MYA NA;Lm;0;L;;;;;N;;;;;
A4FD;LISU LETTER TONE MYA JEU;Lm;0;L;;;;;N;;;;;
A4FE;LISU PUNCTUATION COMMA;Po;0;L;;;;;N;;;;;
A4FF;LISU PUNCTUATION FULL STOP;Po;0;L;;;;;N;;;;;
```

13. Code Chart. A code chart is given on page 18. The encoding order is adapted from Everson (2006a) with the following changes:

(1) Positions U+A4EA..U+A4ED are mapped differently to reflect the collating order used in China (Section 10).[2]

(2) The original modifier letters at U+A4FC..U+A4FD are removed (Section 5).

(3) Tone letters at original positions U+A4FA..U+A4FB are moved down two positions and switched according to the collating order used in China (Section 10).

(4) Two tone letters are added at positions U+A4FA..U+A4FB (Section 4.1).

Everson (2006a) also maps position U+A4FE to PUNCTUATION COMMA but with a missing dot, which he believes is a quite possible error. The representative glyph in the code chart of this proposal includes the dot. For the most part, character names are taken from Everson (2006a) and adjusted to follow the

[2] It should be pointed out that collating order and encoding order do not dictate one another (see TUS 5.0 Section 2.1, p.12 and Section 5.16, p.179), but for convenience it is common practice to encode characters after a consistent collating order.

guidelines set forth in Annex L of ISO/IEC 10646:2003. Certain vowels are named differently to better reflect their phonetic values. Tone letters are given their Lisu names instead of numbers (Section 4).

14. Bibliography

Anonymous. n.d. Lisu hymn book. n.p.

Anonymous. ca. 1917. Lisu primer (catechism). Kunming, China: China Inland Mission.

Anonymous. 1999. Comic excerpt. In *Rays of Sunlight Magazine.* Yangon: Eastern Bible Institute.

Answers Corporation. 2007. Internationalized domain name. http://www.answers.com/topic/internationalized-domain-name. Accessed 2 October 2007.

Bradley, David. 1994. *A dictionary of the northern dialect of Lisu (China and Southeast Asia). Pacific linguistics* C-126. Canberra, Australia: Australian National University.

_____. 2003. Lisu. In *The Sino-Tibetan languages*, ed. Graham Thurgood and Randy J. LaPolla, 222-235. London, UK: Routledge.

_____. 2005a. Issues in orthography development and reform. In *Heritage maintenance for endangered languages in Yunnan, China*, ed. David Bradley, 1-10. Melbourne, Australia: La Trobe University.

_____. 2005b. *Southern Lisu dictionary*. ms. Published in James A. Matisoff, ed., *STEDT monograph series,* vol. 4 (Berkeley, CA: University of California, 2006).

_____. 2006. Personal interview by Adrian Cheuk. 10 April 2006.

Bradley, David, ed. 2000. *Lisu bride price song.* Transcribed by David Fish. Melbourne, Australia: La Trobe University.

BY YU-LI-Y= 2000. LI-SU ⊥O ⅂ B P ɗ FO, SO.. DU= YI WU. ⊥I PƆN= Chiang Mai: Christian Literature Fellowship.

曹大荣编．2001．《娶亲调》．德宏：德宏民族出版社．

China. 2007. L2/07-423: Documentation on legacy encodings of the Old Lisu script. ms.

戴庆厦、许寿椿、高喜奎主编．1991．《中国各民族文字与电脑信息处理》．北京：中央民族学院出版社．

Davis, Mark. 2008. Proposed update to Unicode standard annex #29: Unicode text segmentation. Rev. 12. http://www.unicode.org/reports/tr29/tr29-12.html. Accessed 22 January 2008.

Davis, Mark and Michel Suignard. 2006a. Unicode technical report #36: Unicode security considerations. Rev. 5. http://www.unicode.org/reports/tr36/. Accessed 2 October 2007.

_____. 2006b. Unicode technical standard #39: Unicode security mechanisms. Rev. 2. http://www.unicode.org/reports/tr39/. Accessed 5 October 2007.

Davis, Mark and Ken Whistler. 2006. Unicode technical standard #10: Unicode collation algorithm. Rev. 16. http://www.unicode.org/reports/tr10/. Accessed 24 January 2008.

Everson, Michael. 2006a. Fraser draft. ms. http://www.evertype.com/standards/iso10646/pdf/fraser-draft.pdf. Accessed 8 March 2006. Later submitted as WG2 document N3323, available at http://www.dkuug.dk/jtc1/sc2/wg2/docs/n3323.pdf.

_____. 2006b. Fraser sample. ms. Also available at http://www.evertype.com/standards/iso10646/pdf/fraser-sample.pdf.

Faltstrom, P., P. Hoffman, and A. Costello. 2003. RFC 3490: Internationalizing Domain Names in Applications (IDNA). http://www.rfc-editor.org/rfc/rfc3490.txt. Accessed 5 October 2007.

Fraser, James O. 1922. *Handbook of the Lisu (Yawyin) language.* Rangoon: Government Printer. Reproduced in Lisu Phonology, http://www.rosettaproject.org/archive/lis/phon-2?page_view=image_view, and Lisu Grammar, http://www.rosettaproject.org/archive/lis/morsyn-1?page_view=image_view, The Rosetta Project. Accessed 4 June 2007.

Freytag, Asmus. 2007. Proposed update to Unicode standard annex #14: Line breaking properties. Rev. 20. http://www.unicode.org/reports/tr14/tr14-20.html. Accessed 21 May 2007.

Gordon, Raymond G., Jr., ed. 2005. Ethnologue: Languages of the world. 15th edn. Dallas, TX: SIL International. http://www.ethnologue.com/. Accessed 8 March 2006.

Handel, Zev. 2003. Proto-Lolo-Burmese velar clusters and the origin of Lisu palatal sibilants. http://faculty.washington.edu/zhandel/Handel_Lisu.pdf. Accessed 8 March 2006.

Harvey, Christopher. 2005. SENĆOŦEN (Saanich, Northern Straits Salish). http://www.languagegeek.com/salishan/sencoten.html. Accessed 2 October 2007.

胡玉来．2006．浅谈傈僳族语言文字的创立和使用价值．ms.

ICANN (The Internet Corporation for Assigned Names and Numbers). 2005. Guidelines for the Implementation of Internationalized Domain Names. Draft Version 2.0. http://icann.org/general/idn-guidelines-20sep05.htm. Accessed 2 October 2007.

Language Museum. 2007. Lisu. http://www.language-museum.com/l/lisu.php. Accessed 4 June 2007.

黎爱蓉、吕晴编．2005．《傈僳族三弦调》．德宏：德宏民族出版社．

丽江纳西族自治县民族事务委员会、教育局编译室编．1985．傈僳文小学课本《语文》第一册．昆明：云南民族出版社．

Lyons, Melinda, et al, ed. 2001. Glossary. http://scripts.sil.org/Glossary. Accessed 2 October 2007.

Morse, David. 2007a. Personal interviews by Adrian Cheuk. 19 February and 20 September 2007.

________. 2007b. Lisu alphabet and tones. ms.

________. 2007c. Lisu vowels. MP3.

Morse, David L. and Thomas M. Tehan. 2000. How do you write Lisu? In *Endangered languages and literacy: Proceedings of the fourth FEL conference (Charlotte, North Carolina, September 21-24, 2000)*, ed. Nicholas Ostler and Blair Rudes, 53-62. Bath, UK: Foundation for Endangered Languages.

mozilla.org. 2007. IDN-enabled TLDs. http://www.mozilla.org/projects/security/tld-idn-policy-list.html. Accessed 2 October 2007.

Saanich Indian School Board. 2004. **ȽÁU,WELṈEW̱** Tribal School. http://www.sisb.bc.ca/intro.html. Accessed 3 October 2007.

Thaprom, Jerry. 1989. ꓶ DO Xꓱ, DO LI PꓷN SO’ M NYI PꓷN= Trans. Joel Khopang. Chiang Mai: Christian Literature Fellowship.

The Rosetta Project. 2006. Lisu Genesis translation. http://www.rosettaproject.org/archive/lis/gen-1?page_view=image_view. Accessed 4 June 2007.

The Unicode Consortium. 2007. The Unicode Standard, Version 5.0.0, defined by: *The Unicode Standard, Version 5.0* (Boston, MA: Addison-Wesley, 2007. ISBN 0-321-48091-0).

徐琳、木玉璋、盖兴之编著．1986．《中国少数民族语言简志丛书・傈僳语简志》．北京：民族出版社．

徐琳、木玉璋、施履谦等编著．1985．《傈汉词典》．昆明：云南民族出版社．

云南省少数民族语文指导工作委员会编．1998．《云南省志・少数民族语言文字志》．昆明：云南人民出版社．

云南省少数民族语文指导工作委员会、怒江州民语委编．1994．《傈僳文识字课本》．昆明：云南民族出版社．

云南省少数民族语文指导工作委员会、维西县文教局编．1981．《傈僳文识字课本》．昆明：云南民族出版社．

中国科学院少数民族语言研究所主编．1959．《傈僳语语法纲要》．北京：科学出版社．

“中国少数民族文字字符总集”课题组主编．2003．《中国少数民族文字字符总集》．北京：中国社会科学院民族学与人类学研究所．CD-Rom.

祝发清．1984．《傈汉小词典》．德宏：德宏民族出版社．

祝发清编．1995．《牧歌》．昆明：云南民族出版社．

15. Acknowledgements

This project was made possible by the help of the following individuals:

- Dr. Deborah Anderson, Researcher, Dept. of Linguistics, UC Berkeley: helped liaise with UTC at various stages of the proposal process.
- Prof. David Bradley, Associate Professor and Reader, La Trobe University, Australia, and renowned scholar in Lisu and related languages: provided valuable advice, samples, and reference materials.
- Adrian Cheuk, Script Technologist, East Asia Group, SIL International: conducted the main research and drafted the proposal.
- Martin Hosken, Non-Roman Script Initiative, SIL International, and Payap University, Thailand: provided technical guidance and liaison with Lisu experts in Thailand.
- David Morse, mother-tongue Lisu speaker and expert in literature production: contributed precious information, fonts, and samples of the Lisu script.

A group of over 15 experts in Lisu literature production in Thailand also gave important user feedback concerning various encoding issues.

TABLE XX - Row A4: LISU

	A4D	A4E	A4F
0	ꓐ	ꓠ	ꓬ
1	ꓑ	ꓡ	ꓱ
2	ꓓ	ꓢ	ꓲ
3	ꓔ	ꓣ	ꓳ
4	ꓕ	ꓤ	ꓴ
5	ꓖ	ꓥ	ꓵ
6	ꓗ	ꓦ	ꓶ
7	ꓘ	ꓧ	ꓷ
8	ꓙ	ꓨ	ꓸ
9	ꓚ	ꓩ	ꓹ
A	ꓛ	ꓪ	ꓺ
B	ꓜ	ꓫ	ꓻ
C	ꓝ	ꓬ	ꓼ
D	ꓞ	ꓭ	ꓽ
E	ꓟ	ꓮ	꓾
F	ꓠ	ꓯ	꓿

TABLE XX - Row A4: LISU

hex	Name
D0	LISU LETTER BA
D1	LISU LETTER PA
D2	LISU LETTER PHA
D3	LISU LETTER DA
D4	LISU LETTER TA
D5	LISU LETTER THA
D6	LISU LETTER GA
D7	LISU LETTER KA
D8	LISU LETTER KHA
D9	LISU LETTER JA
DA	LISU LETTER CA
DB	LISU LETTER CHA
DC	LISU LETTER DZA
DD	LISU LETTER TSA
DE	LISU LETTER TSHA
DF	LISU LETTER MA
E0	LISU LETTER NA
E1	LISU LETTER LA
E2	LISU LETTER SA
E3	LISU LETTER ZHA
E4	LISU LETTER ZA
E5	LISU LETTER NGA
E6	LISU LETTER HA
E7	LISU LETTER XA
E8	LISU LETTER HHA
E9	LISU LETTER FA
EA	LISU LETTER WA
EB	LISU LETTER SHA
EC	LISU LETTER YA
ED	LISU LETTER GHA
EE	LISU LETTER A
EF	LISU LETTER AE
F0	LISU LETTER E
F1	LISU LETTER EU
F2	LISU LETTER I
F3	LISU LETTER O
F4	LISU LETTER U
F5	LISU LETTER UE
F6	LISU LETTER UH
F7	LISU LETTER OE
F8	LISU LETTER TONE MYA TI
F9	LISU LETTER TONE NA PO
FA	LISU LETTER TONE MYA CYA
FB	LISU LETTER TONE MYA BO
FC	LISU LETTER TONE MYA NA
FD	LISU LETTER TONE MYA JEU
FE	LISU PUNCTUATION COMMA
FF	LISU PUNCTUATION FULL STOP

hex	Name

Figures

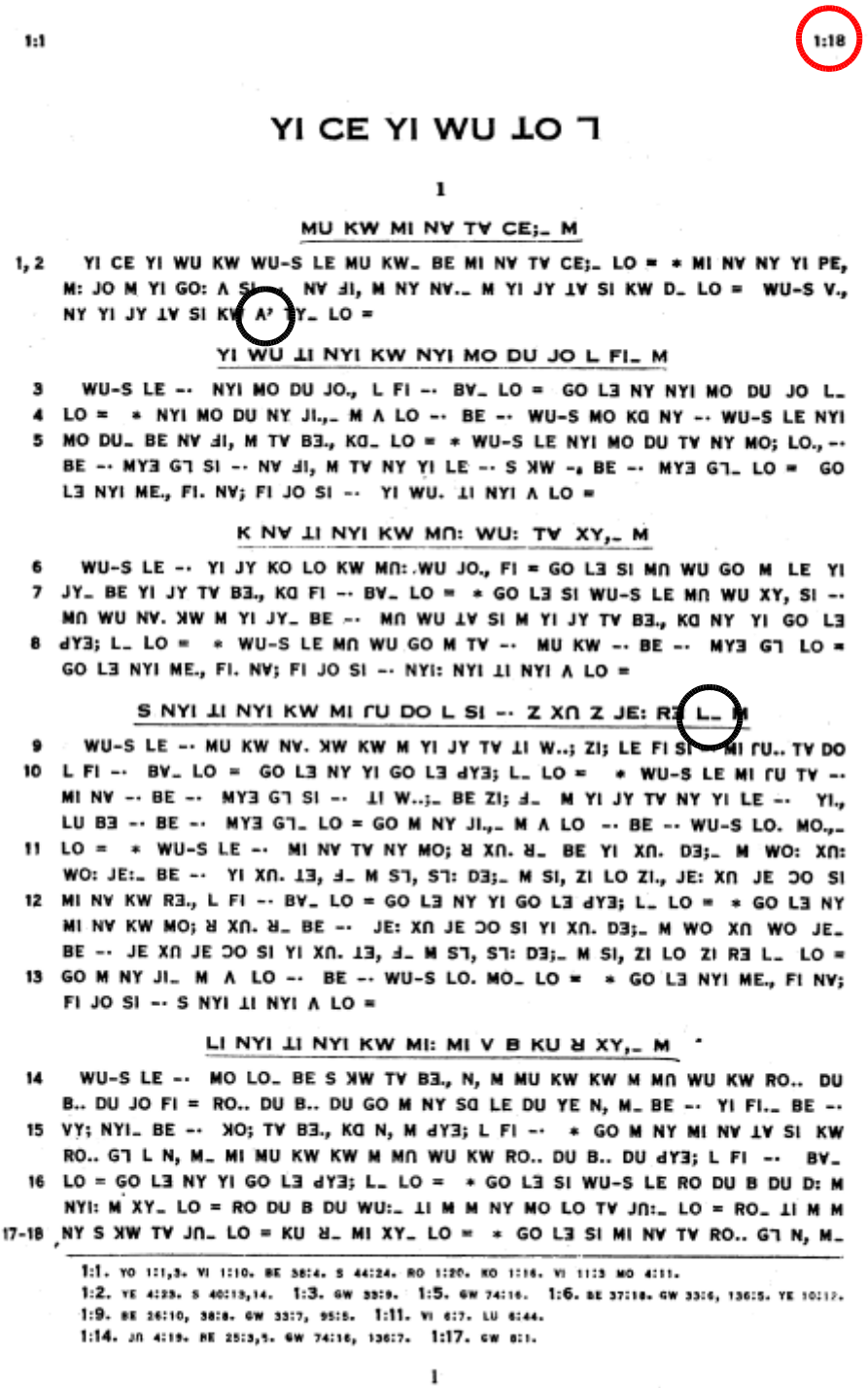


Figure 1: Sample from a 1968 Lisu Bible (Genesis 1:1-17), showing examples of the *nasalisation mark* and the *A glide* (black circles). The vertical position of the latter is contrasted with that of the underlining. Circled in red is an example where the Latin colon is used to separate chapter and verse numbers.

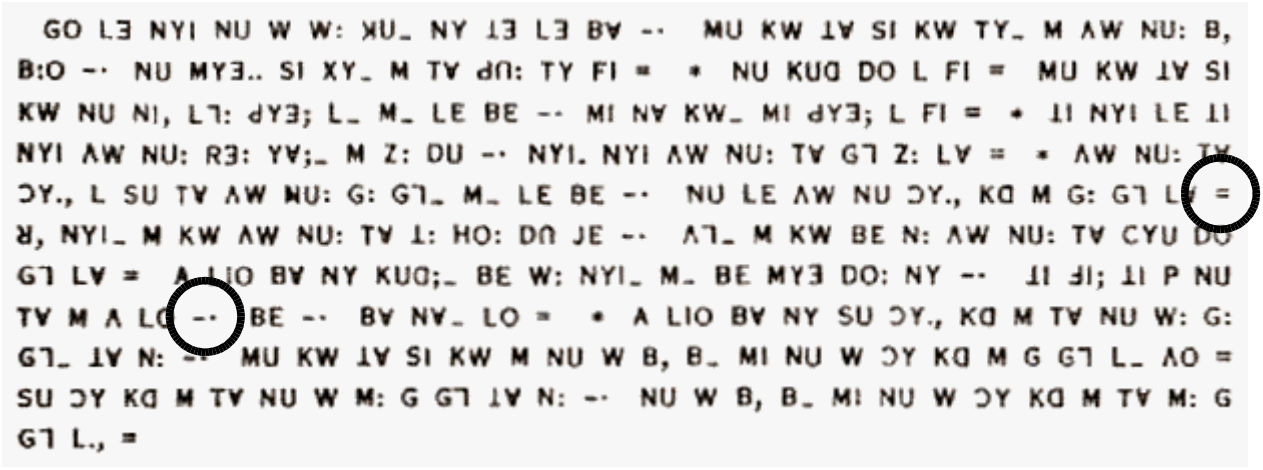


Figure 2: Sample from a Lisu Bible (Matthew 6:9-12), showing -. PUNCTUATION COMMA **and =** PUNCTUATION FULL STOP**.**

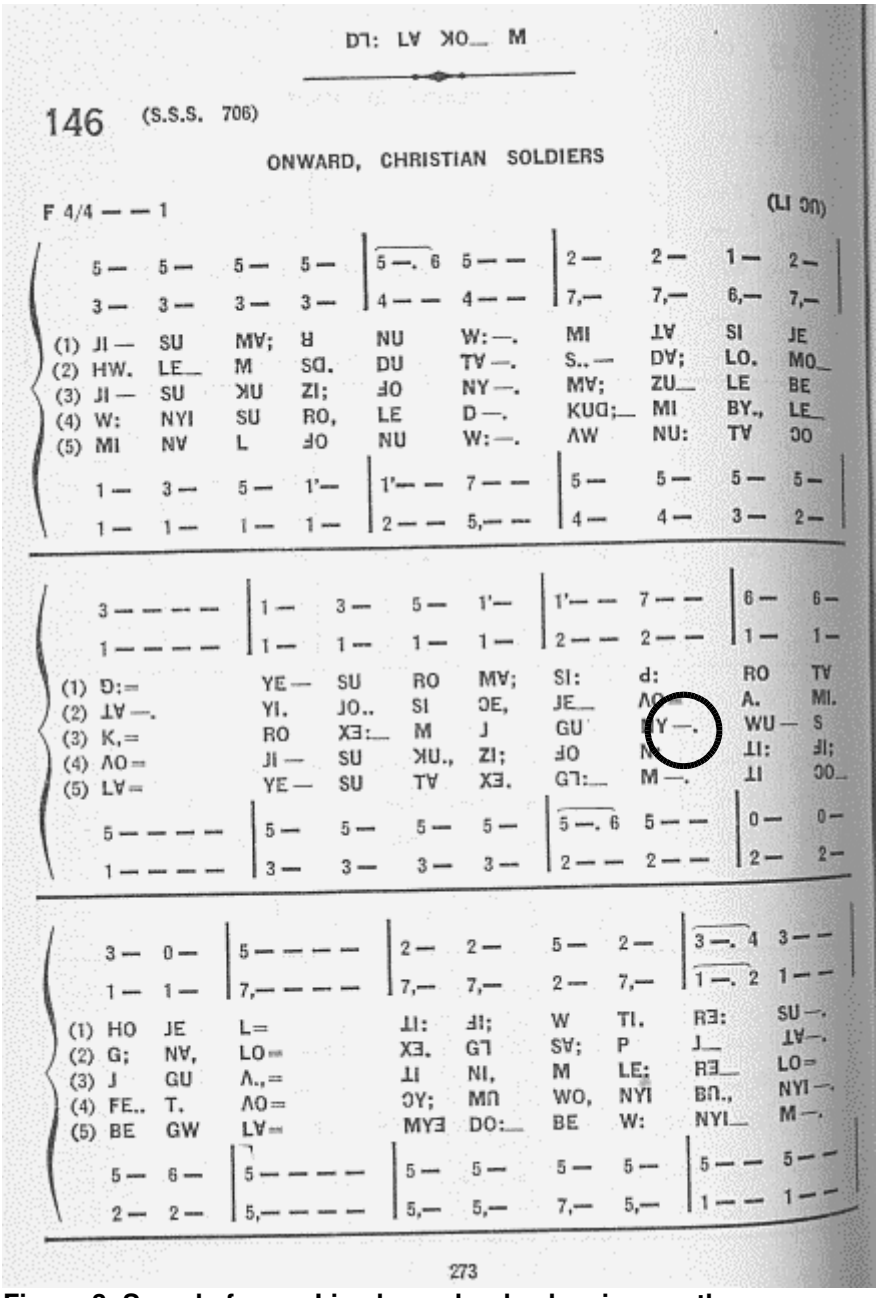


Figure 3: Sample from a Lisu hymn book, showing another rendering of -. PUNCTUATION COMMA**.**

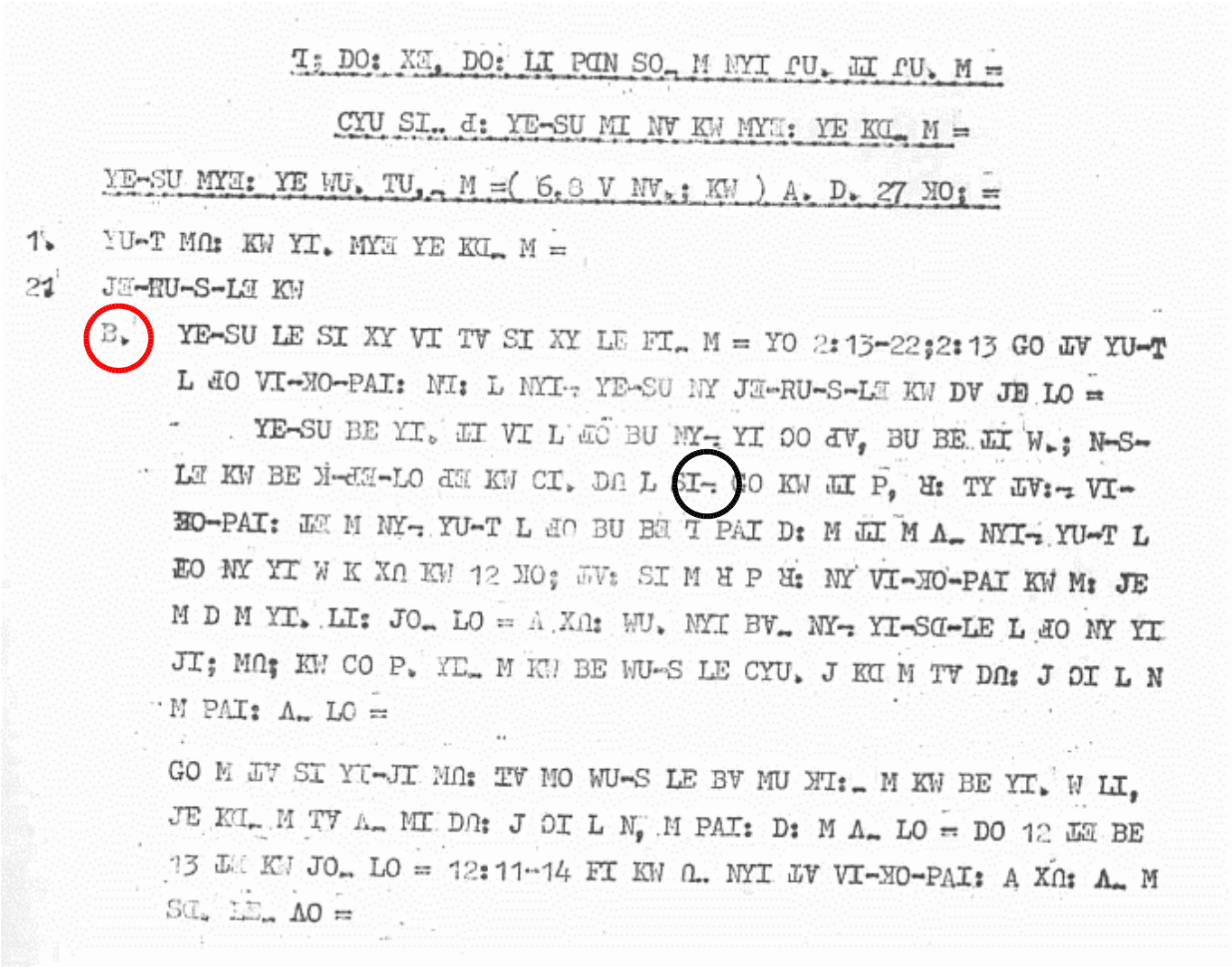


Figure 4: Sample from a Lisu Bible study resource. Circled in black is a third rendering of -. PUNCTUATION COMMA**. The red circle shows** LETTER BA **being used as the first number in the second level of a list.**

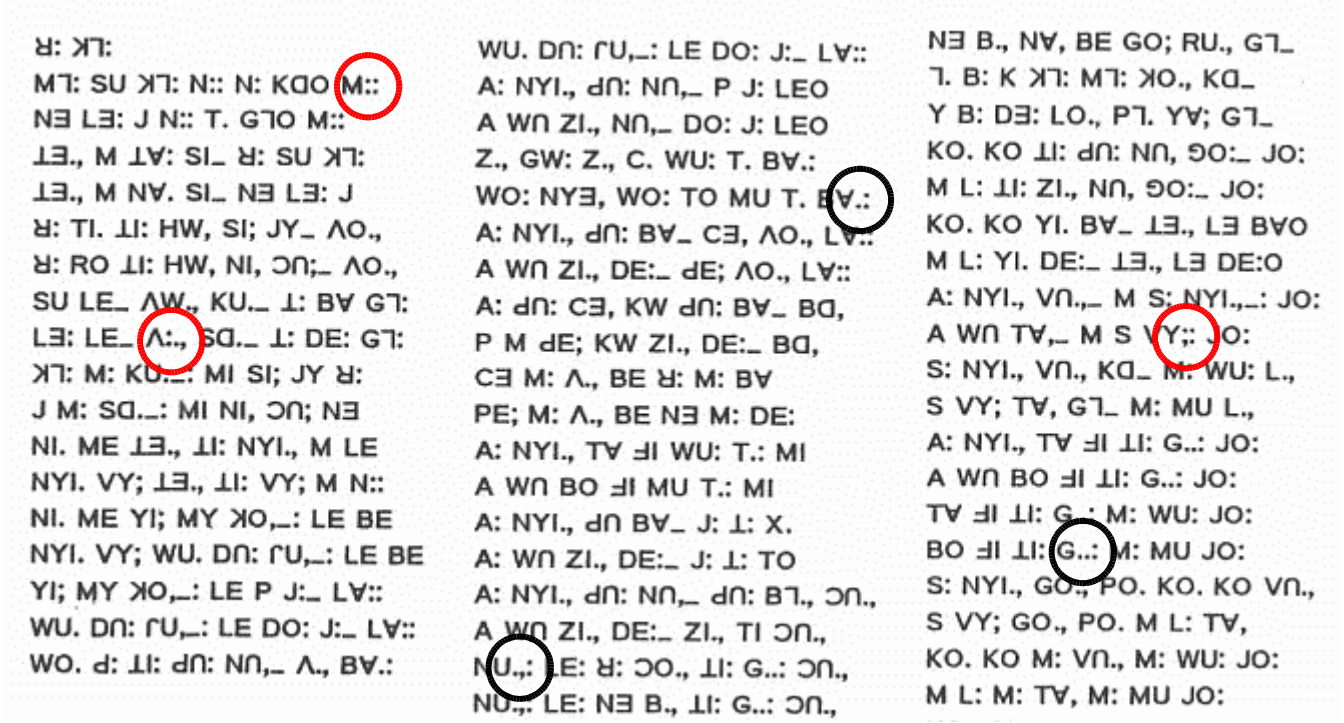


Figure 5: Samples from a Lisu song book, showing various combination tones. Those circled in red are exceptional permutations used to transcribe special intonations and vowel lengths as the song is sung.

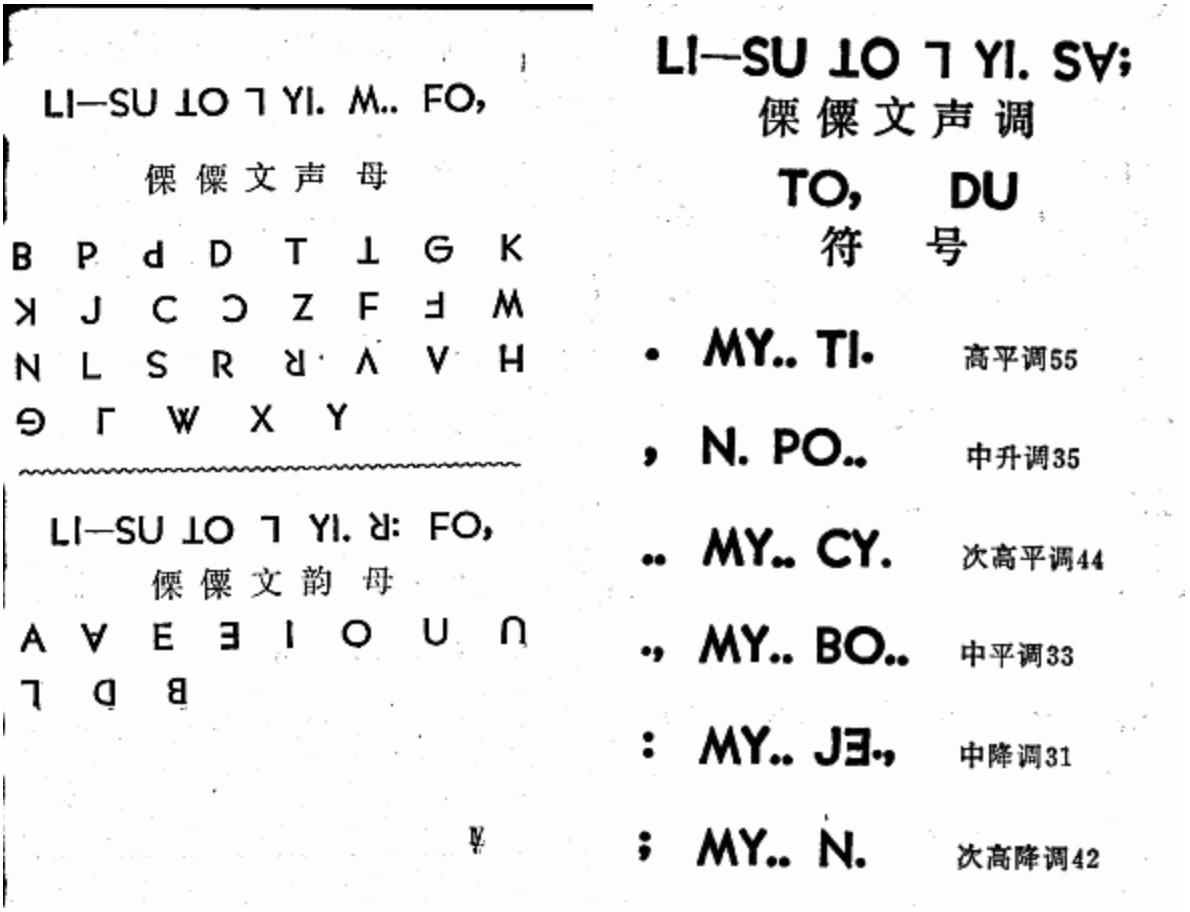


Figure 6: Samples from a Lisu-Chinese dictionary, showing the traditional alphabetical order (left) and tone order with tone names (right).

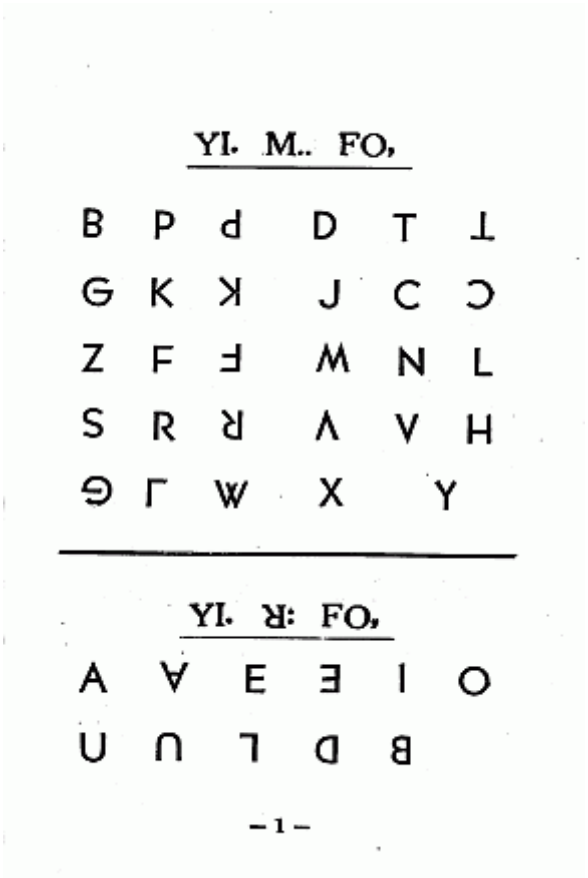
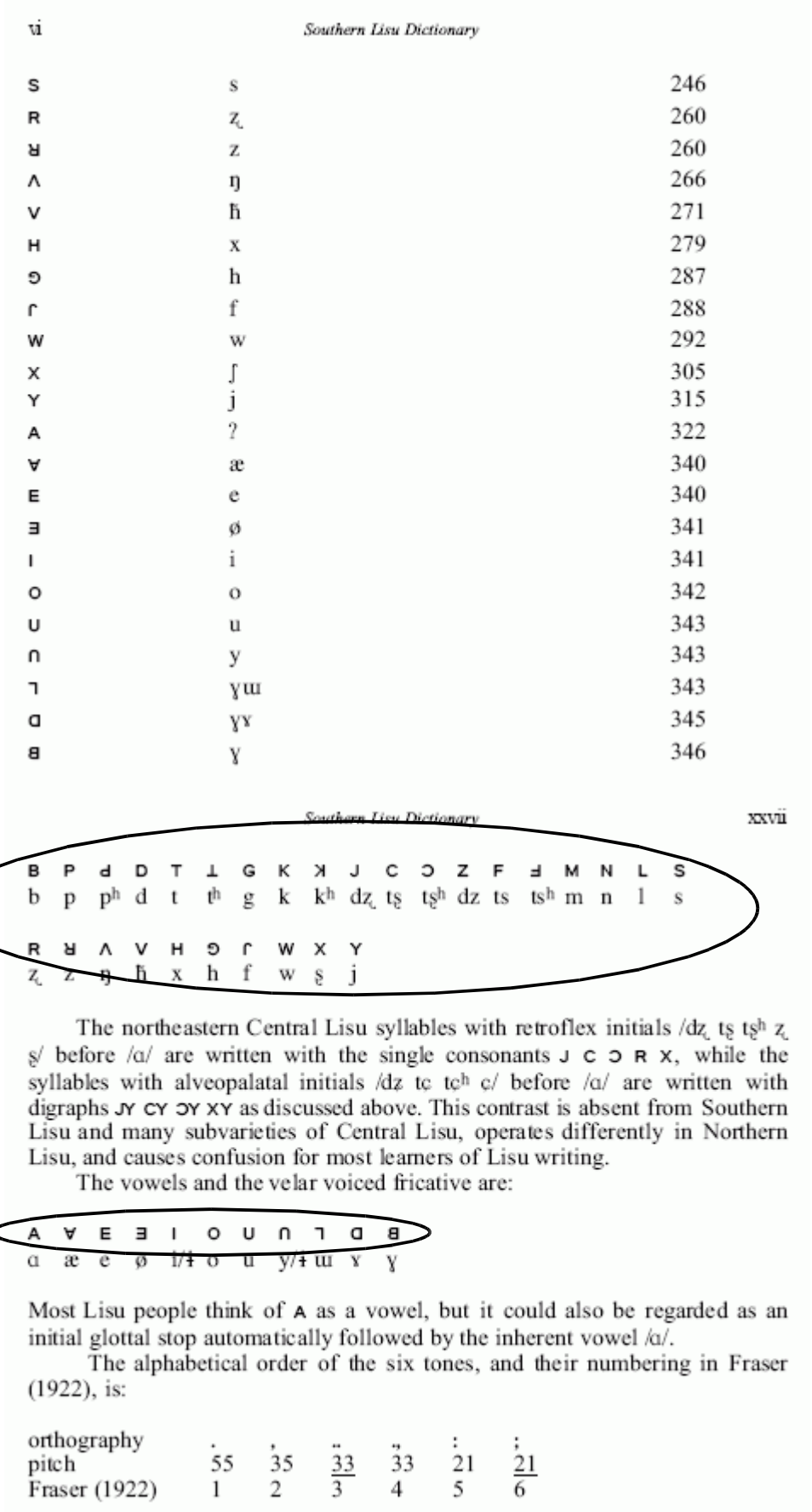


Figure 7: Sample from a Lisu primer, showing the same alphabetical order.



vi *Southern Lisu Dictionary*

S	s	246
R	ʐ	260
ʁ	z	260
Λ	ŋ	266
V	ɦ	271
H	x	279
Ɔ	h	287
ꟻ	f	288
W	w	292
X	ʃ	305
Y	j	315
A	ʔ	322
Ɐ	æ	340
E	e	340
Ǝ	ø	341
I	i	341
O	o	342
U	u	343
Ո	y	343
ꟻ	ɣɯ	343
ꓷ	ɣʏ	345
ꓭ	ɣ	346

Southern Lisu Dictionary xxvii

B	P	ԁ	D	T	⊥	G	K	Ж	J	C	Ↄ	Z	F	ԍ	M	N	L	S
b	p	pʰ	d	t	tʰ	g	k	kʰ	dʑ	tʂ	tʂʰ	dz	ts	tsʰ	m	n	l	s

R	ʁ	Λ	V	H	Ɔ	ꟻ	W	X	Y
ʐ	z	ŋ	ɦ	x	h	f	w	ʂ	j

The northeastern Central Lisu syllables with retroflex initials /dʐ tʂ tʂʰ ʐ ʂ/ before /ɑ/ are written with the single consonants J C Ↄ R X, while the syllables with alveopalatal initials /dʑ tɕ tɕʰ ɕ/ before /ɑ/ are written with digraphs JY CY ↃY XY as discussed above. This contrast is absent from Southern Lisu and many subvarieties of Central Lisu, operates differently in Northern Lisu, and causes confusion for most learners of Lisu writing.

The vowels and the velar voiced fricative are:

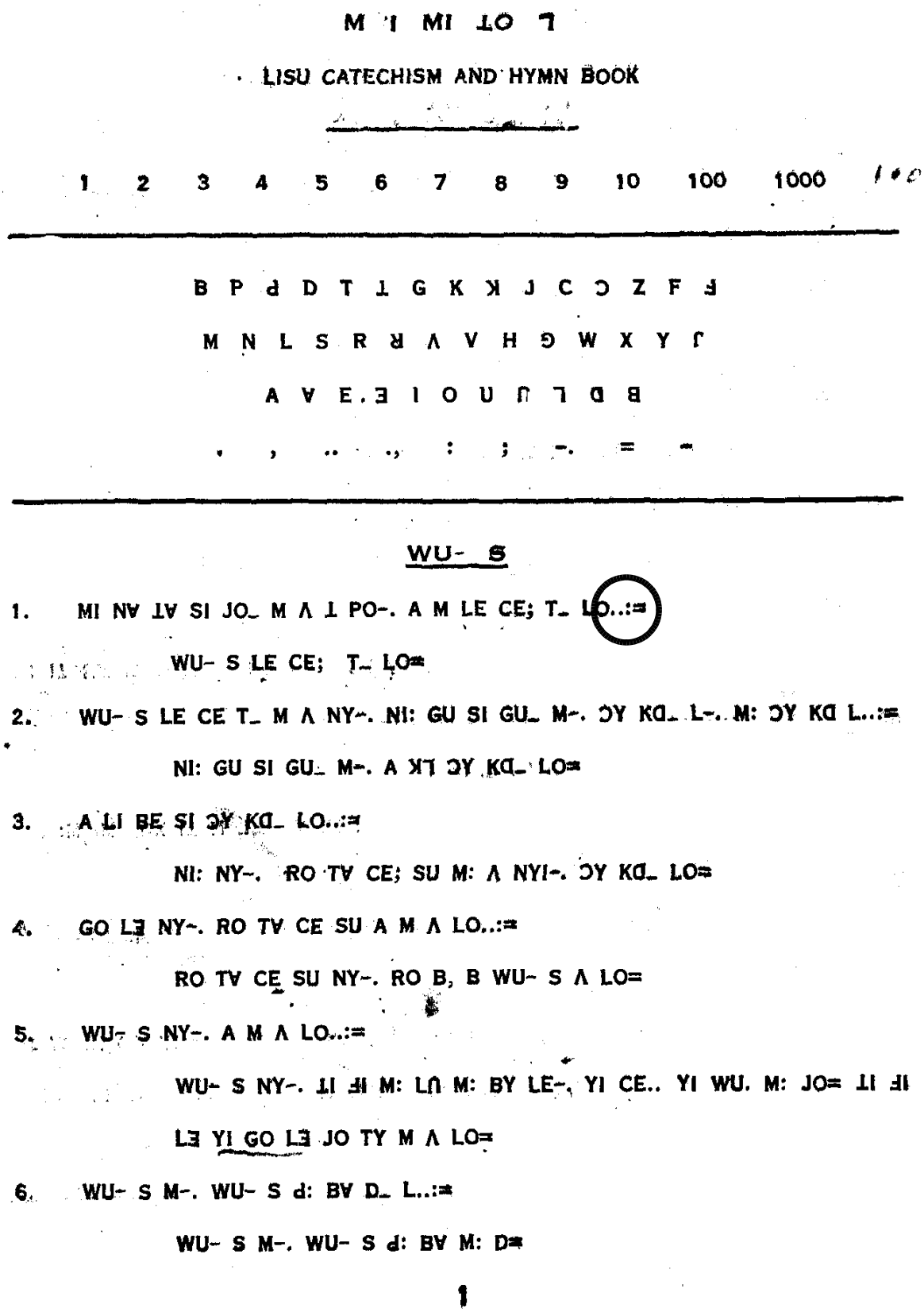
A	Ɐ	E	Ǝ	I	O	U	Ո	ꟻ	ꓷ	ꓭ
ɑ	æ	e	ø	i/ɨ	o	u	y/ʉ	ɯ	ʏ	ɣ

Most Lisu people think of A as a vowel, but it could also be regarded as an initial glottal stop automatically followed by the inherent vowel /ɑ/.

The alphabetical order of the six tones, and their numbering in Fraser (1922), is:

orthography	.	,	¨	ˌ	:	;
pitch	55	35	33	33	21	21
Fraser (1922)	1	2	3	4	5	6

Figure 8: Samples from a Lisu-English dictionary, showing the same alphabetical order (circled) and a corresponding look-up order (top; only second part shown). The traditional tone order is also listed (bottom).



LISU CATECHISM AND HYMN BOOK

1 2 3 4 5 6 7 8 9 10 100 1000

B P d D T ꓕ G K ꓘ J C ꓛ Z F ꓝ

M N L S R ꓤ Λ V H ꓖ W X Y ʃ

A V E ꓱ I O U ꓵ ꓶ ꓷ ꓸ

WU- S

1. MI NV ꓕV SI JO_ M Λ ꓕ PO-. A M LE CE; T_ LO..:=

WU- S LE CE; T_ LO=

2. WU- S LE CE T_ M Λ NY-. NI: GU SI GU_ M-. ꓛY Kꓷ_ L-. M: ꓛY Kꓷ L..:=

NI: GU SI GU_ M-. A ꓘꓶ ꓛY Kꓷ_ LO=

3. A LI BE SI ꓛY Kꓷ_ LO..:=

NI: NY-. RO TV CE; SU M: Λ NYI-. ꓛY Kꓷ_ LO=

4. GO Lꓱ NY-. RO TV CE SU A M Λ LO..:=

RO TV CE SU NY-. RO B, B WU- S Λ LO=

5. WU- S NY-. A M Λ LO..:=

WU- S NY-. ꓕI ꓝI M: Lꓵ M: BY LE-, YI CE.. YI WU. M: JO= ꓕI ꓝI

Lꓱ YI GO Lꓱ JO TY M Λ LO=

6. WU- S M-. WU- S d: BV Dꓷ_ L..:=

WU- S M-. WU- S d: BV M: D=

1

Figure 9: Sample from a Lisu catechism, showing an alphabetical order with a different placement of ʃ LETTER FA (top). Note the use of a tone sequence to signal a question (circled).

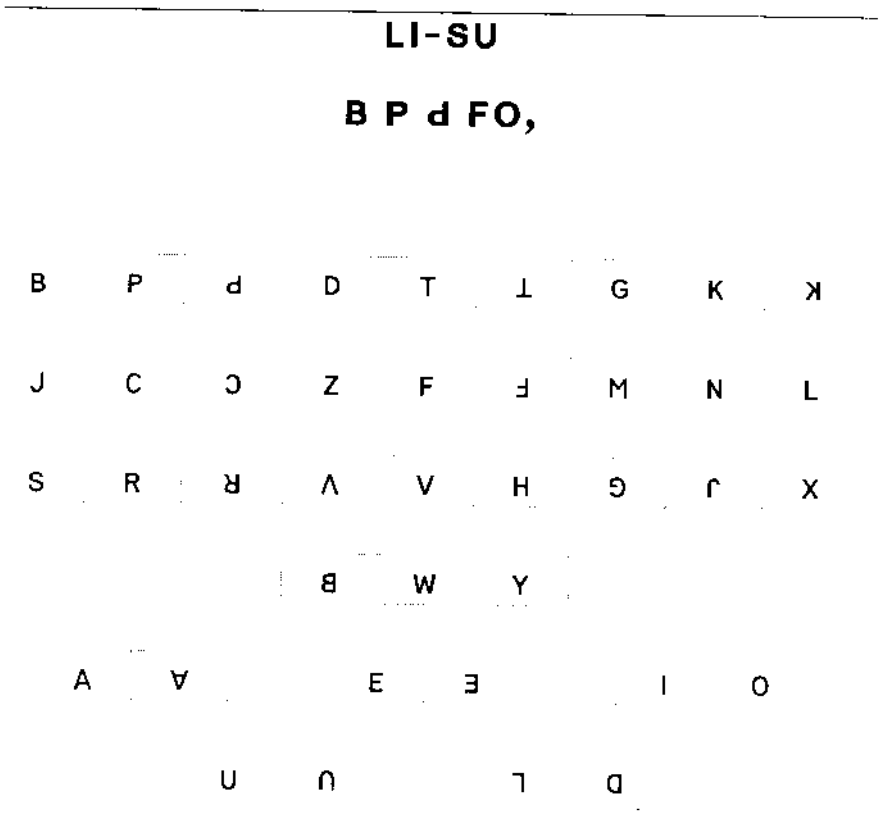
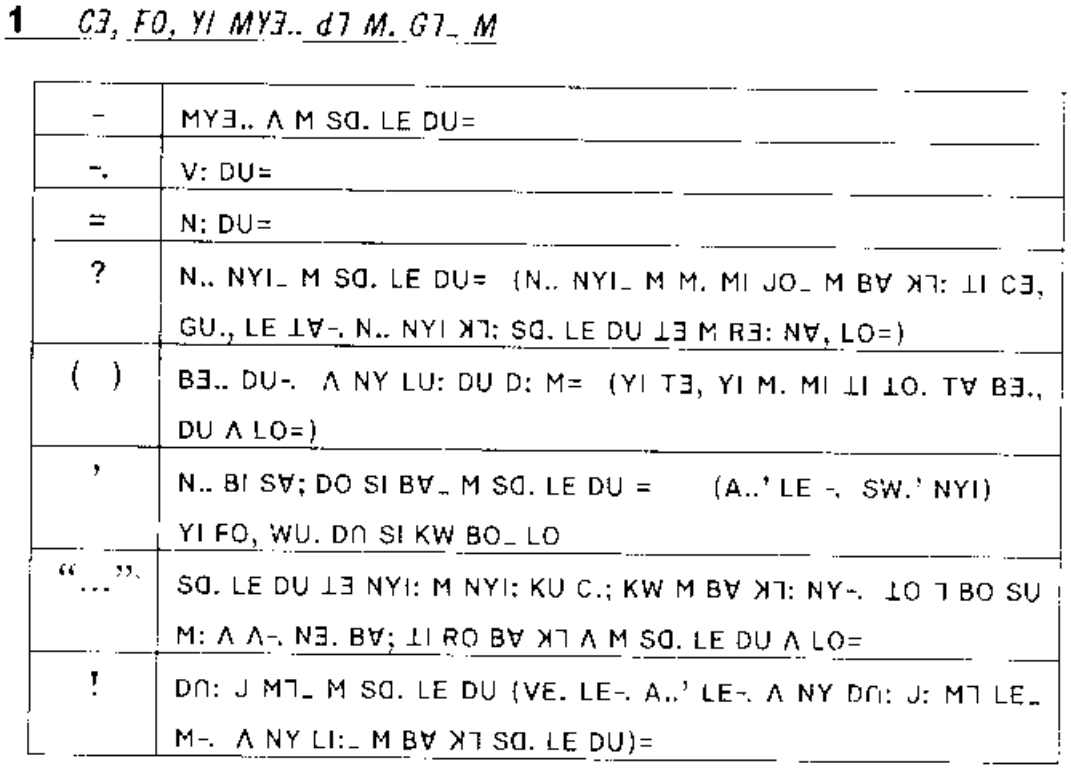
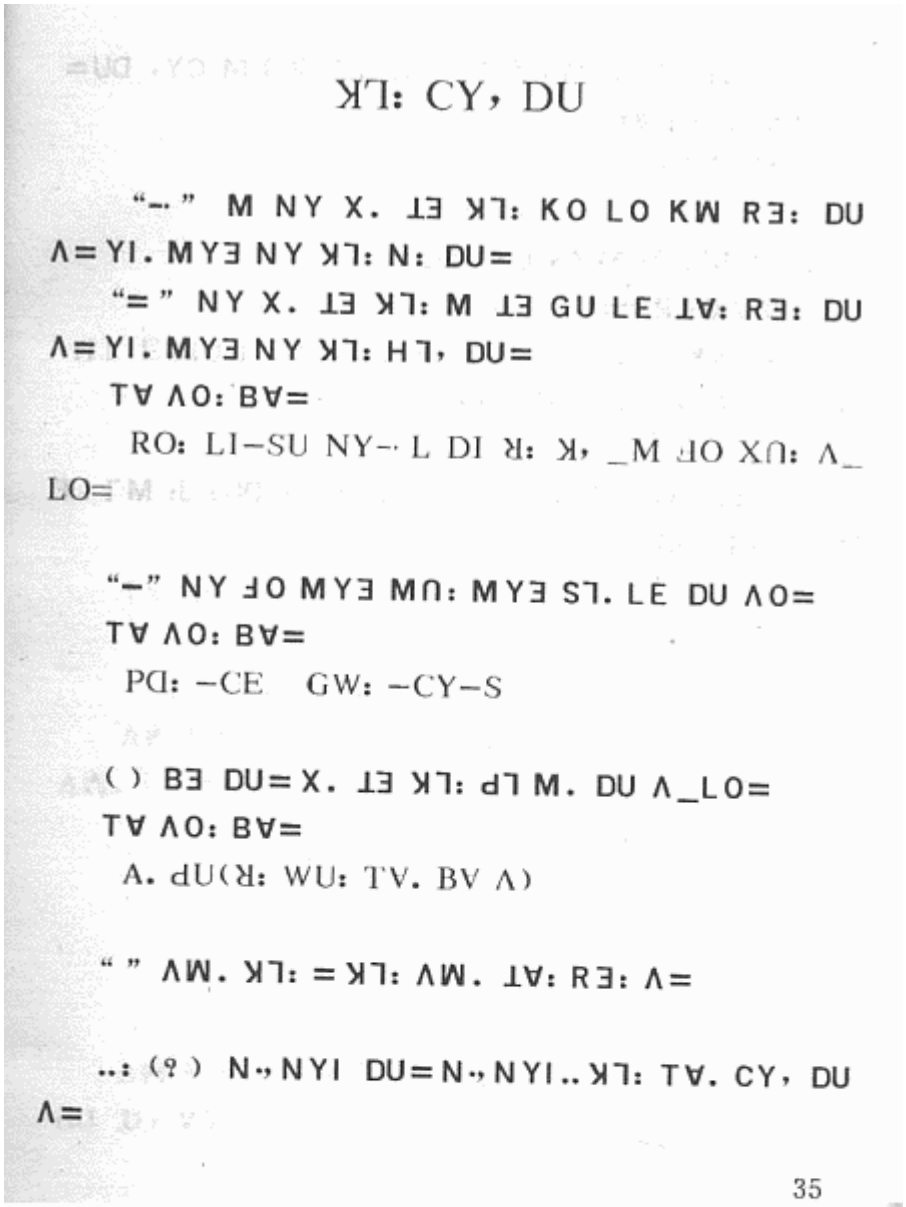


Figure 10: Sample from a Lisu primer used outside China, showing an alternate alphabetical order. Note the letter positions after ϽHHA**.**



-	MYƎ.. Λ M SD. LE DU=
-.	V: DU=
=	N: DU=
?	N.. NYI_ M SD. LE DU= (N.. NYI_ M M. MI JO_ M BA ꓘ: IT CƎ, GU., LE TA-. N.. NYI ꓘ: SD. LE DU TƎ M RƎ: NA, LO=)
( )	BƎ.. DU-. Λ NY LU: DU D: M= (YI TƎ, YI M. MI IT TO. TA BƎ., DU Λ LO=)
’	N.. BI SA; DO SI BA_ M SD. LE DU = (A..’ LE -. SW.’ NYI) YI FO, WU. DՈ SI KW BO_ LO
“...”.	SD. LE DU TƎ NYI: M NYI: KU C.; KW M BA ꓘ: NY-. TO ꓶ BO SU M: Λ A-. NƎ. BA; IT RO BA ꓘ Λ M SD. LE DU Λ LO=
!	DՈ: J Mꓶ_ M SD. LE DU (VE. LE-. A..’ LE-. Λ NY DՈ: J: Mꓶ LE_ M-. Λ NY LI:_ M BA ꓘ SD. LE DU)=

Figure 11: Sample from a Lisu primer used outside China, describing how punctuation marks are used.



35

Figure 12: Samples from a Chinese Lisu primer, describing how punctuation marks are used.

声调表

文字符号	调值	例词		
		文字	国际音标	汉意
.	˥55	LO.	lo˥	(狗)叫
,	˧35	LO,	lo˧	过(去)
. .	˦44	LO..	lo˦	扔
. ,	˧33	LO.,	lo˧	轻
;	˦42	LO;	lo˦	够
:	˩31	LO:	lo˩	(斤)两

Figure 13: Sample from a Chinese minority script journal describing the Lisu tones. Note the switched order of the last two tones.

SA; ZƎ.,

YI SA; (SA; TI.) NYI: M ⊥I W.,; JO L M NY-. (8) M JO_ LO=
(⊥Ǝ M TA SA; ZƎ.,-. BE BA_ LO=)

SA; TI.=	.	,	..	.,	:	;	-
SA; ZƎ.,=					.,:	.,;	
					..:	..;	
					,:	,;	
					.:	.;	

Figure 14: Sample from a Lisu primer used outside China, listing six simple tones and eight combination tones.

ƎO ϟ, NA.; SI XY M J GU KW MYƎ: HO: SU NY-. ƎO ϟ, KW M ƎO WU: ƎO M BU TA A LI YE NA, M-. MYƎ: TA HO: WU. MU SU YI ZU., Я (Committee) ⊥I ZU., SI.. GㄱNA, LO= SI: XY MYƎ: KW HO: WU. MU SU NY ƎO ϟ, L ƎO BU Λ MI NI, ԀՈ: NI, LƎ LƎ SA; ⊥U; L N, M BƎ ㄱ SA; GㄱNA, LO= MYƎ: HO: SU_ MI MYƎ: YE M. GㄱNA, LO=

Figure 15: Sample from a survey document, showing the Latin characters (Committee) **in a serif font distinguished from the surrounding sans serif Lisu characters.**

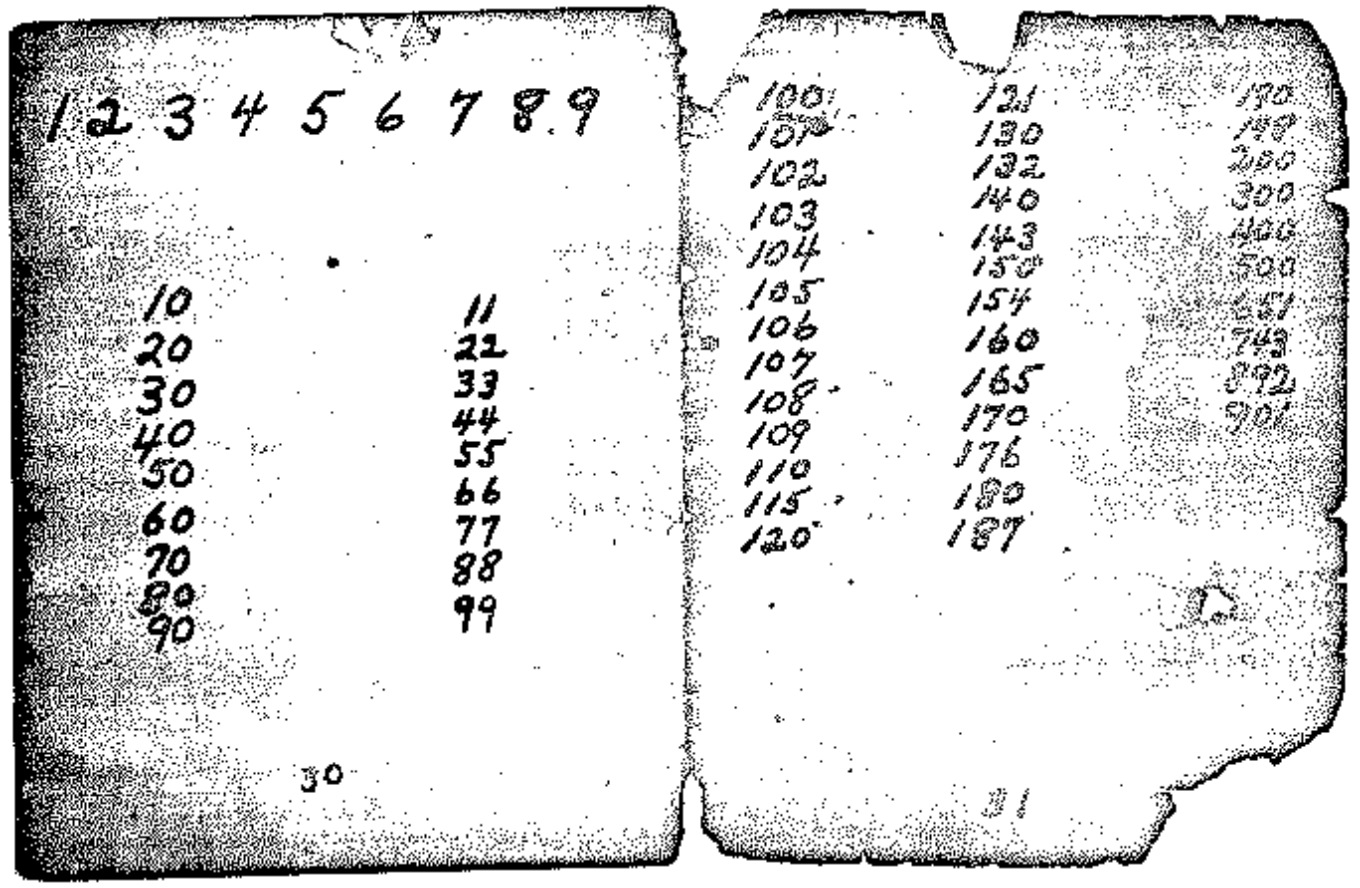


Figure 16: Sample from a handwritten Lisu primer, showing numbers represented with Arabic numerals.

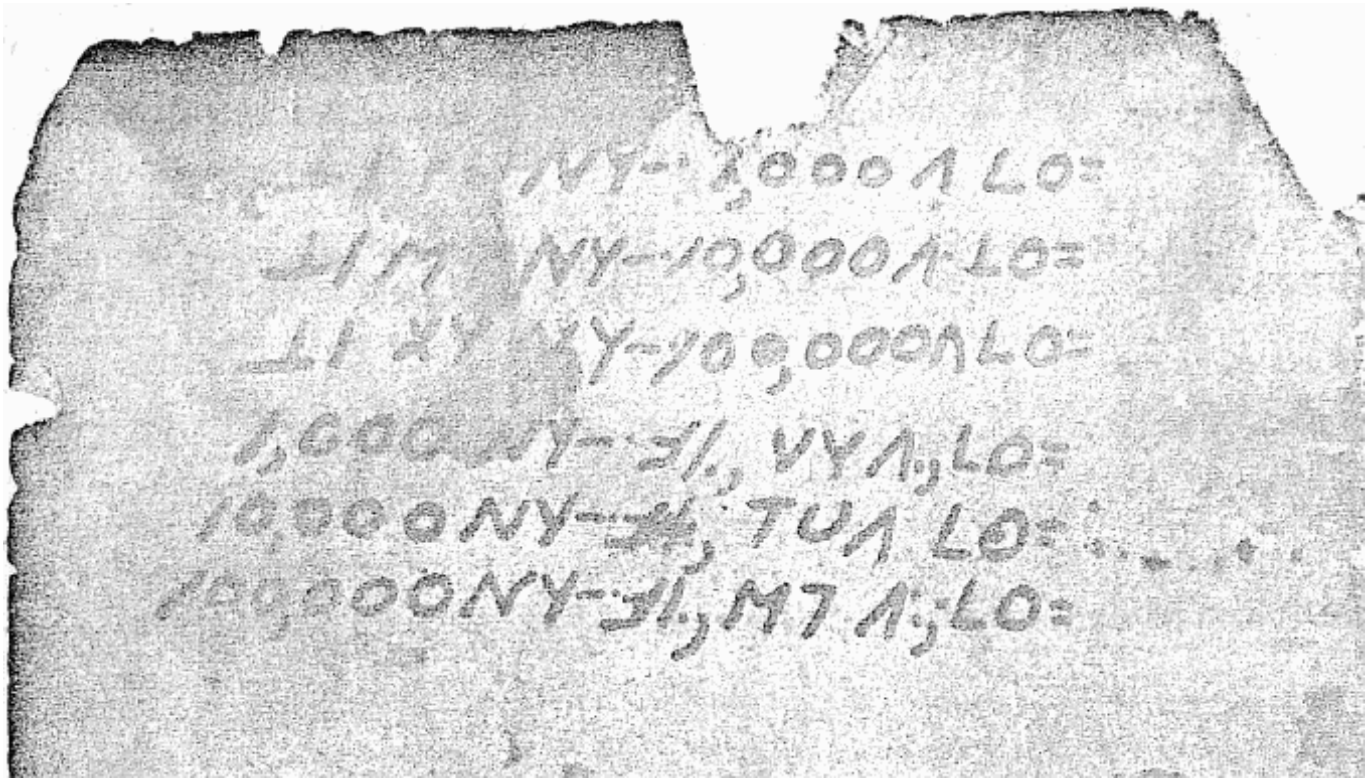


Figure 17: Sample from a handwritten Lisu primer, showing the thousand separator represented with the Latin comma.

Figure 18: Banner in front of a shopping mall in Yunnan, China.

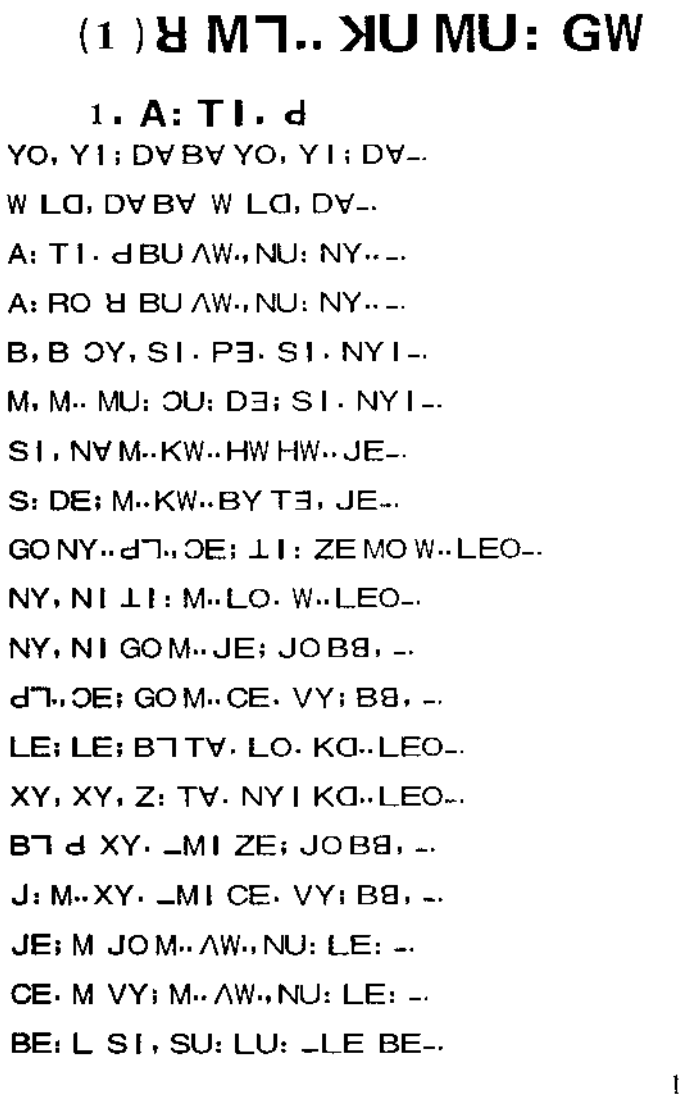


Figure 19: Sample from a Lisu song book.

Figure 20: Sample from a Lisu magazine in Yangon, Myanmar.

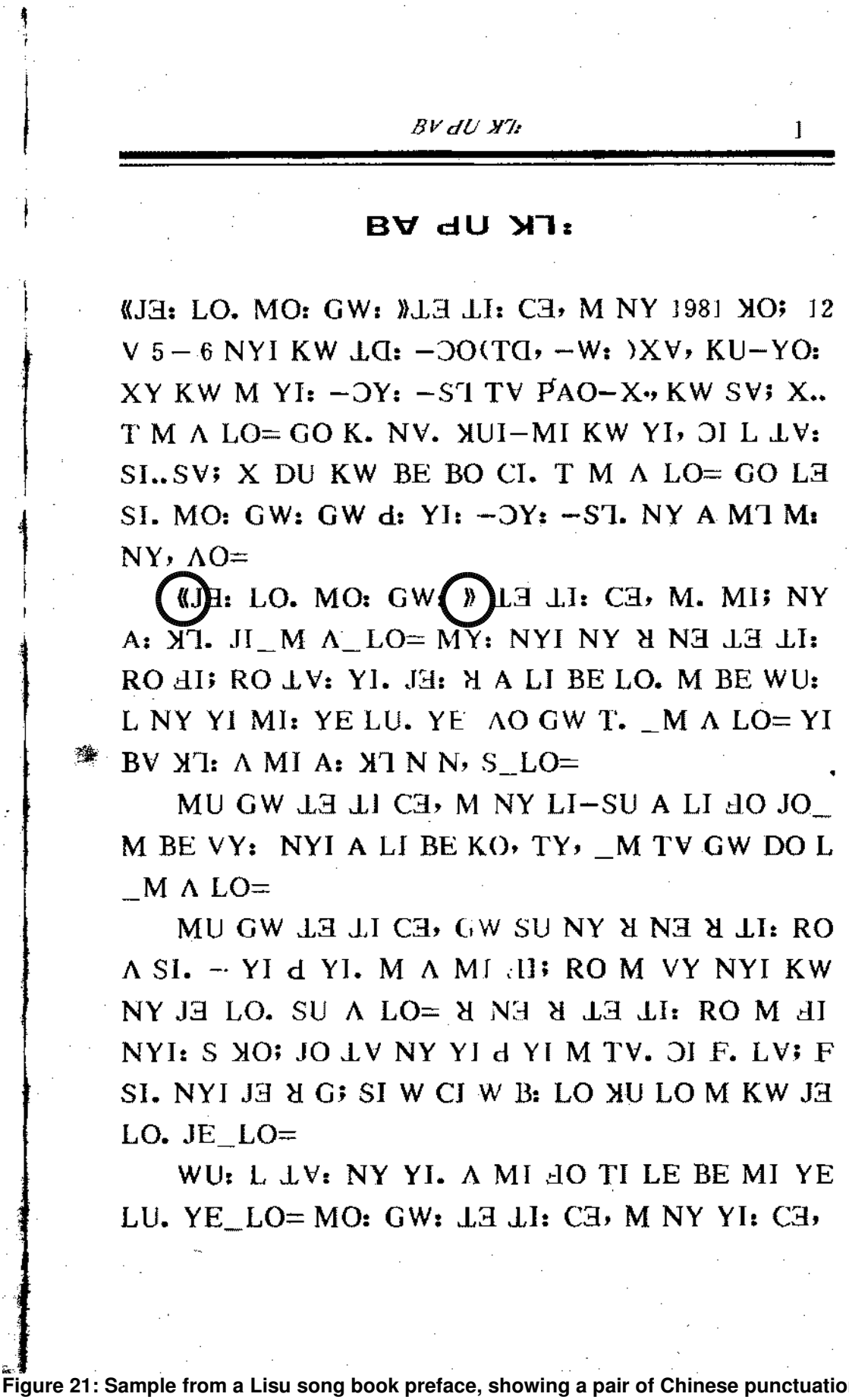


Figure 21: Sample from a Lisu song book preface, showing a pair of Chinese punctuation used to mark book titles (circled).

ISO/IEC JTC 1/SC 2/WG 2
PROPOSAL SUMMARY FORM TO ACCOMPANY SUBMISSIONS
FOR ADDITIONS TO THE REPERTOIRE OF ISO/IEC 10646[3]
Please fill all the sections A, B and C below.
Please read Principles and Procedures Document (P & P) from http://www.dkuug.dk/JTC1/SC2/WG2/docs/principles.html **for guidelines and details before filling this form.**
Please ensure you are using the latest Form from http://www.dkuug.dk/JTC1/SC2/WG2/docs/summaryform.html**.**
See also http://www.dkuug.dk/JTC1/SC2/WG2/docs/roadmaps.html **for latest *Roadmaps*.**

A. Administrative

1. **Title:** *Proposal for encoding the Lisu script in the BMP of the UCS*
2. Requester's name: *China*
3. Requester type (Member body/Liaison/Individual contribution): *Member body*
4. Submission date: *2008-04-22*
5. Requester's reference (if applicable): *CN/08-001R*
6. Choose one of the following:
   This is a complete proposal: *Yes*
   (or) More information will be provided later:

B. Technical – General

1. Choose one of the following:
   a. This proposal is for a new script (set of characters): *Yes*
      Proposed name of script: *Lisu*
   b. The proposal is for addition of character(s) to an existing block:
      Name of the existing block:
2. Number of characters in proposal: *48*
3. Proposed category (select one from below - see section 2.2 of P&P document):
   A-Contemporary X | B.1-Specialized (small collection) | B.2-Specialized (large collection)
   C-Major extinct | D-Attested extinct | E-Minor extinct
   F-Archaic Hieroglyphic or Ideographic | G-Obscure or questionable usage symbols
4. Is a repertoire including character names provided? *Yes*
   a. If YES, are the names in accordance with the "character naming guidelines" in Annex L of P&P document? *Yes*
   b. Are the character shapes attached in a legible form suitable for review? *Yes*
5. Who will provide the appropriate computerized font (ordered preference: True Type, or PostScript format) for publishing the standard? *David Morse*
   If available now, identify source(s) for the font (include address, e-mail, ftp-site, etc.) and indicate the tools used: *ntcm0@yahoo.com*
6. References:
   a. Are references (to other character sets, dictionaries, descriptive texts etc.) provided? *Yes*
   b. Are published examples of use (such as samples from newspapers, magazines, or other sources) of proposed characters attached? *Yes*
7. Special encoding issues:
   Does the proposal address other aspects of character data processing (if applicable) such as input, presentation, sorting, searching, indexing, transliteration etc. (if yes please enclose information)? *Yes*
   Addressed throughout proposal. See esp. Sections 4, 10, and 11.

8. Additional Information:

Submitters are invited to provide any additional information about Properties of the proposed Character(s) or Script that will assist in correct understanding of and correct linguistic processing of the proposed character(s) or script. Examples of such properties are: Casing information, Numeric information, Currency information, Display behaviour information such as line breaks, widths etc., Combining behaviour, Spacing behaviour, Directional behaviour, Default Collation behaviour, relevance in Mark Up contexts, Compatibility equivalence and other Unicode normalization related information. See the Unicode standard at http://www.unicode.org for such information on other scripts. Also see http://www.unicode.org/Public/UNIDATA/UCD.html and associated Unicode Technical Reports for information needed for consideration by the Unicode Technical Committee for inclusion in the Unicode Standard.

[3] Form number: N3102-F (Original 1994-10-14; Revised 1995-01, 1995-04, 1996-04, 1996-08, 1999-03, 2001-05, 2001-09, 2003-11, 2005-01, 2005-09, 2005-10, 2007-03)

C. Technical - Justification

1. Has this proposal for addition of character(s) been submitted before? *Yes*
   If YES explain *N3317, L2/07-344*
2. Has contact been made to members of the user community (for example: National Body, user groups of the script or characters, other experts, etc.)? *Yes*
   If YES, with whom? *Yunnan Minority Language Commission, David Morse, David Bradley, over 15 Lisu experts in literature production in Thailand*
   If YES, available relevant documents:
3. Information on the user community for the proposed characters (for example: size, demographics, information technology use, or publishing use) is included? *Yes*
   Reference: *See Section 1 of this document.*
4. The context of use for the proposed characters (type of use; common or rare) *Common*
   Reference: *See Section 1 of this document.*
5. Are the proposed characters in current use by the user community? *Yes*
   If YES, where? Reference: *China, Myanmar, Thailand, India*
6. After giving due considerations to the principles in the P&P document must the proposed characters be entirely in the BMP? *Yes*
   If YES, is a rationale provided? *Yes*
   If YES, reference: *It is widely used among the Lisu communities, which number 1 million. See Section 1 of this document.*
7. Should the proposed characters be kept together in a contiguous range (rather than being scattered)? *Yes*
8. Can any of the proposed characters be considered a presentation form of an existing character or character sequence? *No*
   If YES, is a rationale for its inclusion provided?
   If YES, reference:
9. Can any of the proposed characters be encoded using a composed character sequence of either existing characters or other proposed characters? *Yes*
   If YES, is a rationale for its inclusion provided? *Yes*
   If YES, reference: *See Section 4.1 of this document.*
10. Can any of the proposed character(s) be considered to be similar (in appearance or function) to an existing character? *Yes*
    If YES, is a rationale for its inclusion provided? *Yes*
    If YES, reference: *Although some appear similar to Latin characters, this is a different script altogether with different behaviours. Hence, it would be best to encode them as a block. See Section 11 of this document.*
11. Does the proposal include use of combining characters and/or use of composite sequences? *No*
    If YES, is a rationale for such use provided?
    If YES, reference:
    Is a list of composite sequences and their corresponding glyph images (graphic symbols) provided?
    If YES, reference:
12. Does the proposal contain characters with any special properties such as control function or similar semantics? *No*
    If YES, describe in detail (include attachment if necessary)
13. Does the proposal contain any Ideographic compatibility character(s)? *No*
    If YES, is the equivalent corresponding unified ideographic character(s) identified?
    If YES, reference: