L2/18-081
2018-03-26

Preliminary proposal to encode Rejang Numbers in Unicode

Anshuman Pandey
pandey@umich.edu

March 26, 2018

1 Introduction

This is a preliminary proposal to encode characters used in a numerical notation system associated with the Rejang script. The characters were described by Christopher Miller in Unicode Technical Note #35 "Indonesian and Philippine Scripts" and recommended for encoding in Unicode (p. 31–32). Describing various extensions for Rejang, Miller writes that the "most important extension to the block consists of the Angka Bejagung numerals" (p. 33).

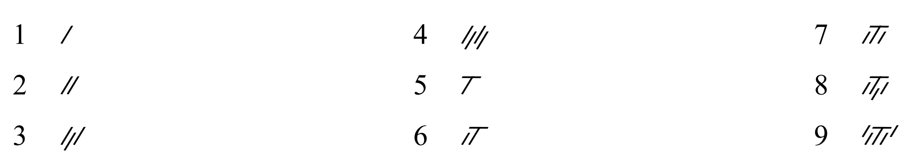
There are two possible models for encoding these Rejang numbers, which are described here. Feedback is requested from experts. Research is ongoing and a formal proposal with additional details and specimens of usage is forthcoming.

As the 'Rejang' block in the BMP is nearly full, these Rejang numbers should be placed in a new 'Rejang Extended' block in the SMP, which may also be used for other Rejang additions.

2 Notation system

The notation system is known as *angka bejagung* in Rejang. It is a quinary system based upon the additive principle. Unique signs exist for 1, 5, 10, 50, 100, and 1000, 10000, 100000, and 1000000. Three different patterns of numerical expression are discernable: 1) primary units and tens; 2) hundreds; 3) thousands, ten thousands, hundred, thousands, and millions. These are described below:

- Primary units Expressed using combinations of / 'one' and ⊤ 'five'. 2–4 are repetitions of 'one'; 5–9 are the forms for 1–4 grouped around 'five'.



- Tens Expressed using combinations of 'ten' and ⊕ 'fifty'. 20–40 are repetitions of 'ten'; 50–90 are the forms for 10–40 grouped around 'ten'.

10	40	70
20	50 ⊕	80
30	60	90

- Place values of the tens are written using the signs for the tens along with subjoined, reduced, or rearranged forms of signs for the primary units:

11	14	17
12	15	18
13	16	19

- Hundreds One hundred is produced using the base sign to which the sign or is attached. The sign for 'two hundred' is produced by adding a diagonal stroke to the top extender of 'one hundred'. That for 'three hundred' is derived from 'two hundred' through the addition of a stroke through the circle. The rest of the hundreds are written using combinations of one hundred, two hundred, three hundred. The unit is tabulated in groups of three: 100–300 using the base signs; 400–600 using 'three hundred' + the base signs; 700–900 using 'six hundred' + the base signs. The sequences are written close together to indicate that they belong to a group, eg. instead of .

100	400	700
200	500	800
300	600	900

- Thousands This is produced using the sign beneath which a dot . is placed for marking 1000. The numbers of this unit are written using repetitions of one thousand. The sequences are written closely together to indicate that they belong to a group, eg. instead of .

1,000	4,000	7,000
2,000	5,000	8,000
3,000	6,000	9,000

- Ten thousands 10,000 is represented using the sign 'ten thousand'. This sign is based upon 'one thousand' with the addition of a diagonal check mark at the top right. This check mark serves as a unit mark for orders of the thousands. Multiples other than 10,000 are not attested, but the pattern would follow that of the thousands. Also, it is likely that the sequences are written closely together to indicate that they belong to a group.

10,000	40,000	70,000
20,000	50,000	80,000
30,000	60,000	90,000

- Hundred thousands 100,000 is represented using the sign  ‘hundred thousand’. This sign is based upon  ‘one thousand’ with the addition of two check marks at the top right (or alternately, by the addition of one check mark to ‘ten thousand’). Multiples other than 100,000 are not attested, but the pattern would follow that of the thousands. Following the pattern for other orders of thousand, sequences of this unit are written closely together to indicate grouping.

100,000		400,000		700,000	
200,000		500,000		...	
300,000		600,000			

- Millions 1,000,000 is represented using the sign  ‘one million’. This sign is based upon ‘one thousand’ with the addition of three check marks at the top right (or alternately, by the addition of one check mark to ‘hundred thousand’). Multiples other than 1,000,000 are not attested, but the pattern would follow that of the thousands. Sequences are written closely together to indicate that they belong to a group.

1,000,000		4,000,000		...
2,000,000		5,000,000		
3,000,000		6,000,000		

These are encoded as atomic characters. It could be possible to use multiple instances of in plain text, but this approach does not provide for the space adjustments between the characters. It also disrupts the sequence at end of line.

3 Encoding Model

Miller noted that the signs for *angka bejagung* can be “decomposed into eight base symbols”; however, “the complex manners in which they are combined may justify treating at least some, if not all of them, as single compound glyphs with their individual code points” (p. 33). Although Miller does not specify the eight, the following base symbols are identifiable: ‘one’, ‘five’, ‘ten’, ⊕ ‘fifty’, ‘one hundred’, ‘one thousand’. Encoding these is not sufficient. Based upon the patterns described above, there are two possible models for encoding the signs:

3.1 Model #1

This approach takes into consideration the positioning and grouping of signs.

- Primary units Although written using ‘one’ and ‘five’, the positioning of in various combinations does not permit a model that uses only ‘one’ and ‘five’. For example, 9 is not simply the linear (‘five’ + ‘four’), but , in which the instances of are clustered around , but with adjustments to size and position. Therefore, it is practical to encode the numbers 1–9 as atomic characters:

Glyph	Character name	Value
𑤁	REJANG NUMBER ONE	1
𑤂	REJANG NUMBER TWO	2
𑤃	REJANG NUMBER THREE	3
𑤄	REJANG NUMBER FOUR	4
𑤅	REJANG NUMBER FIVE	5
𑤆	REJANG NUMBER SIX	6
𑤇	REJANG NUMBER SEVEN	7
𑤈	REJANG NUMBER EIGHT	8
𑤉	REJANG NUMBER NINE	9

- Tens Following the pattern of the primary units, the tens are written using 𑤊 'ten' and ⊕ 'fifty'. For this unit also, the positions of 𑤊 do not permit a model that uses only 'ten' and 'fifty'. Such a model would work for 70, eg. 𑤊⊕𑤊 could be composed as <𑤊, ⊕, 𑤊>, but such an approach does not work for 80 𑤊⊕𑤊. If 20 were written as *𑤋 instead of 𑤊𑤊, then an opportunity to produce 30 as *<𑤋, 𑤊> or 90 as *<𑤋, ⊕, 𑤋> might be possible. It is impractical to represent some tens as composites of primitives and some as distinctive signs. Therefore, it is sensible to treat the signs for 10–90 as atomic characters.

Glyph	Character name	Value
𑤊	REJANG NUMBER TEN	10
𑤊𑤊	REJANG NUMBER TWENTY	20
𑤋𑤊	REJANG NUMBER THIRTY	30
𑤋𑤋	REJANG NUMBER FORTY	40
⊕	REJANG NUMBER FIFTY	50
⊕𑤊	REJANG NUMBER SIXTY	60
𑤊⊕𑤊	REJANG NUMBER SEVENTY	70
𑤊⊕𑤋	REJANG NUMBER EIGHTY	80
𑤋⊕𑤋	REJANG NUMBER NINETY	90

- Composites of the primary units and tens These values are expressed using the tens and subjoined forms of the primary units. For example, 16 is represented as 𑤋𑤍, not as 𑤊𑤆. Given the positioning of the primary units with regard to the tens, the signs for the former may be considered as modifiers. It is practical to encode them as combining signs that attach to a base number. For example, 𑤋𑤍 would be encoded using the sequence <𑤊 'ten', ◌𑤍 combining 'six'>.

Glyph	Character name	Value
	REJANG NUMBER COMBINING ONE	1
	REJANG NUMBER COMBINING TWO	2
	REJANG NUMBER COMBINING THREE	3
	REJANG NUMBER COMBINING FOUR	4
	REJANG NUMBER COMBINING FIVE	5
	REJANG NUMBER COMBINING SIX	6
	REJANG NUMBER COMBINING SEVEN	7
	REJANG NUMBER COMBINING EIGHT	8
	REJANG NUMBER COMBINING NINE	9

- Hundreds Although one hundred could be decomposed into × and the sign or , the unit should be considered an atomic sign for 100. Similarly, two hundred and three hundred should be treated atomically. The linear positioning of these signs permits a model that uses the above three characters. For example, 500 would be composed as < three hundred, two hundred>. This approach reduces the need to encode 100–900 as atomic characters, and only requires characters for 100–300.

Glyph	Character name	Value
	REJANG NUMBER ONE HUNDRED	100
	REJANG NUMBER TWO HUNDRED	200
	REJANG NUMBER THREE HUNDRED	300

- Thousands using / 'one thousand'. For example, 6000 is //////. This reduces the need to encode 1000-9000 atomically, and requires only a character for the sign 1000 'one thousand'.

Glyph	Character name	Value
/	REJANG NUMBER ONE THOUSAND	1,000

- Ten thousands The sign is derived from / with the addition of a check mark. Despite the potential to decompose the sign into two primitives, it is practical to consider it an atomic character. The model for the ten thousands is the same for the thousands. This reduces the need for atomically encoding signs for 10000–90000, and requires only the character 'ten thousand'.

Glyph	Character name	Value
	REJANG NUMBER TEN THOUSAND	10,000

- Hundred thousands The sign [glyph] is derived from [glyph] with the addition of two [glyph] check marks. Despite the potential to decompose the sign into two primitives, it is practical to consider it an atomic character. The model for the hundred thousands is the same for the thousands. This reduces the need for atomically encoding signs for 100000–900000, and requires only the character 'hundred thousand'.

Glyph	Character name	Value
[glyph]	REJANG NUMBER HUNDRED THOUSAND	100,000

- Millions The sign [glyph] is derived from [glyph] with the addition of three [glyph] check marks. Despite the potential to decompose the sign into two primitives, it is practical to consider it an atomic character. The model for the millions is the same for the thousands. This reduces the need for atomically encoding signs for 1000000–9000000, and requires only the character 'one million'.

Glyph	Character name	Value
[glyph]	REJANG NUMBER ONE MILLION	1,000,000

- Number joiner The sequences are written close together to indicate that they belong to a group, rather than being separated by a normal space. Therefore, sequences of characters of the same unit of the hundreds and higher orders may be indicated by the use of a script-specific number joiner:

Glyph	Character name	Value
[RNJ]	REJANG NUMBER JOINER	—

  This number joiner is placed between two characters of the same unit:

  – 500 [glyph]: <[glyph], joiner, [glyph]>

  – 8000 [glyph]: <[glyph], joiner, [glyph], joiner, [glyph], joiner, [glyph], joiner, [glyph], joiner, [glyph], joiner, [glyph], joiner, [glyph]>

  – 60,000 [glyph]: <[glyph], joiner, [glyph], joiner, [glyph], joiner, [glyph], joiner, [glyph], joiner, [glyph]>

3.2 Model #2

This approach provides for a linear representation of numbers that does not account for the positioning and grouping features. A minimal set of characters are required and the positioning features would be handled as ligatures of a sequence of signs. The difference between this and the previous model is the representation of the primary units, tens, and place values of the tens.

- Primary units The linear model requires only [glyph] 'one' and [glyph] 'five'. For example, 9 would be produced using the sequence <[glyph] 'five' + [glyph] 'one' + [glyph] 'one' + [glyph] 'one' + [glyph] 'one'>, where the linear form [glyph] would be rendered as the ligature [glyph].

Glyph	Character name	Value
𑤁	REJANG NUMBER ONE	1
𑤅	REJANG NUMBER FIVE	5

- Tens The tens would be produced using two characters: 𑤃 'ten' and ⊕ 'fifty'. The number 70 would be expressed as <⊕ 'fifty' + 𑤃 'ten' + 𑤃 'ten'>, and rendered as 𑤃⊕𑤃.

Glyph	Character name	Value
𑤃	REJANG NUMBER TEN	10
⊕	REJANG NUMBER FIFTY	50

- Composites of the primary units and tens These values would be expressed using the characters for the tens followed by those for the primary units: 16 would be <𑤃 'ten' + 𑤅 'five' + 𑤁 'one'>, which is rendered as 𑤃𑤁.

- The hundreds, thousands, ten thousands, hundred thousands, and millions would be represented using the characters specified for Model #1.

- The number joiner may be used for controling ligatures.

4 Acknowledgments

This project was made possible in part by a grant from the U.S. National Endowment for the Humanities, which funded the Universal Scripts Project (part of the Script Encoding Initiative at UC Berkeley). Any views, findings, conclusions or recommendations expressed in this publication do not necessarily reflect those of the National Endowment of the Humanities.

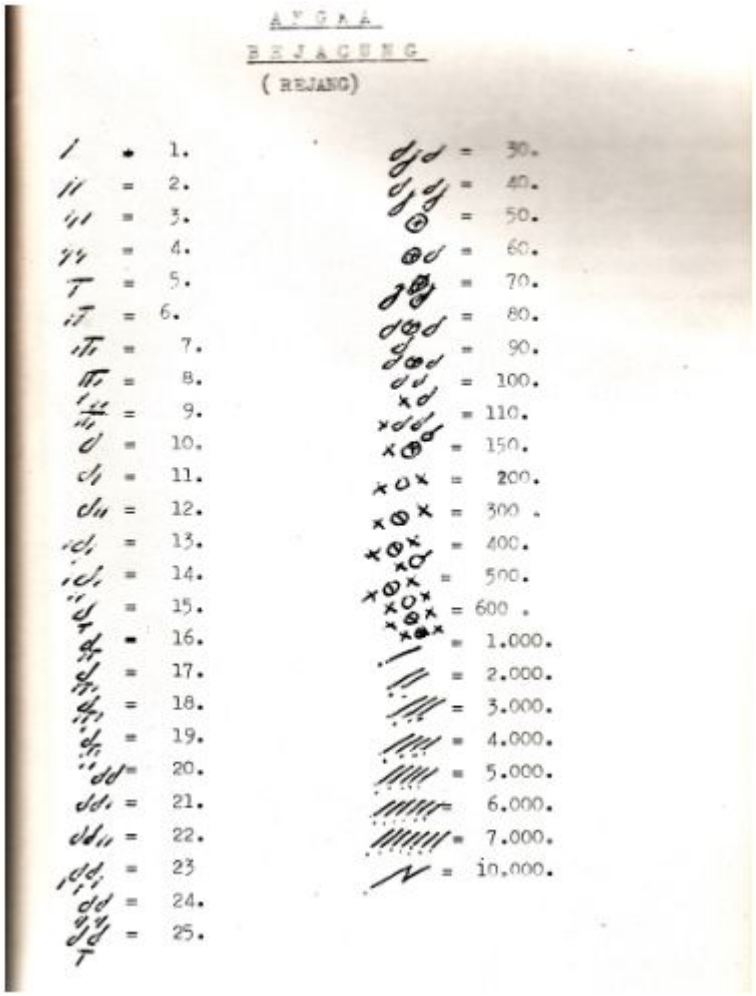


Figure 1: Examples of *angka bejagung*.

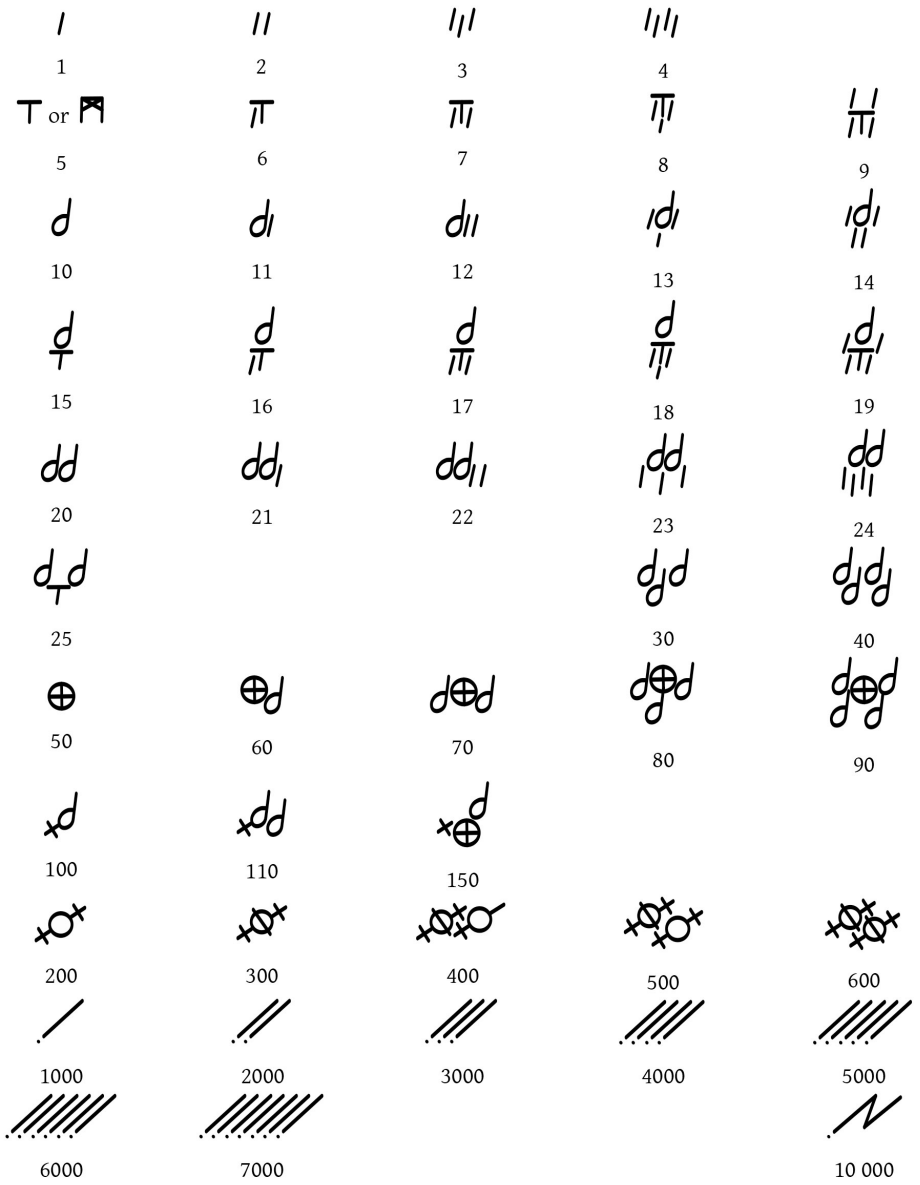


Figure 2: Examples of *angka bejagung* (UTN #35, p. 32).