Syllable Based Dual Weight Algorithm for Line Breaking in Myanmar Unicode

Overview

Myanmar usually does not have spaces typed after every word, but only at sections and other grammatically appropriate break points. However, in tight columns of text it is frequently necessary to break a line more frequently than the locations of spaces.

This document outlines an algorithm to compute potential line breaking points in a section of line breaking text. It is often sufficient to just compare two characters to determine whether a break is possible between them. However, in certain cases ambiguity will arise, in which case a more complicated context analysis is necessary.

The steps to determine whether a break is possible between a pair of characters can therefore be summarized as:

1. Determine character classes for the pair of characters Table 1.
2. Look up the pair in Table 2 to determine the break status.
3. If step 2 is ambiguous perform further Context Analysis around the pair using and and the extra character classes in .

The algorithm presented assumes that data is stored in Myanmar Unicode Canonical Order, it is important to realise that this is **not** the same as the **rendered order**.[1]

Potential break points are considered using a 2 weight approach. Weight 2 Potential Break Points should only be used if there are no Weight 1 Potential Break Points close to the optimal break position. (The meaning of "close" is discussed below).

Word Breaks

This algorithm assumes that accurate information, such as a dictionary lookup for accurate word break determination is unavailable. Many of the issues discussed here would be removed if such a lookup was available.

However, the algorithm could also be used as a crude method of determining word-breaks for such applications as double-click word selection. Both Weight 1 and Weight 2 Potential Break Points would need to be considered as potential selection boundaries. Spaces and punctuation would need further consideration, as is already done by existing software in other languages.

Character Classes

The Character classes shown split the Myanmar Code range into a number of classes.

A number of additional Western punctuation characters have also been included, for the sake of completeness, though traditionally Myanmar would not use these.

The character U+25CC has been included as a consonant, because this is the suggested way of rendering a Myanmar diacritic on its own. This might be necessary in grammatical teaching texts, for example.

The Classes are denoted by a 2 letter acronym (shown in brackets) to simplify the expressions shown later.

1 For more details, see: *Representing Myanmar In Unicode: Details and Examples*, Martin Hoskens & Maung Tuntunlwin, http://www.unicode.org/notes/tn11/.

Class	***Unicode Code Points***	***Characters***[2]
Consonants & Independent Vowels (CI)	[U+1000...U+102A]; U+104E; U+25CC; U+002D	ကခဂဃငစဆဇဈဉ ညဋဌဍဎဏတထဒဓနပဖဗဘမယရလဝသဟဠအ ဣဤဥဦဧဩဪ◌-
Virama (VI)	U+1039	
Medial (ME)	[U+103B...U+103E]	ျ ြ ွ ှ
E vowel (EV)	U+1031	ေ
Lower Vowel (LV)	U+102F; U+1030	ု ူ
Upper Vowel (UV)	[U+102D, U+102E, U+1032]	ိ ီ ဲ
A Vowel (AV)	U+102C	ာ
Anusvara (AN)	U+1036	ံ
Killer (KI)	U+103A	်
Lower Dot (LD)	U+1037	့
Visarga (VG)	U+1038	း
Myanmar Digits (MD)	[U+1040...U+1049]	၁၂၃၅၆၇၈၉
Section (SE)[3]	U+104A; U+104B;[4] U+002C; U+002E; U+003A; U+003B;	၊ ။ , . : ;
Various Signs (VS)	[U+104C; U+104D; U+104F][5]	၌ ၍ ၏
Sanskrit Letter (PL)	[U+1050...U+1055]	
Sanskrit Vowel (PV)	[U+1056...U+1059]	
Space (SP)	U+0020; [U+2000...U+200B]	

2 The actual glyph displayed may be context dependent. Only the most common glyph is shown here.

3 The Western section delimiters COMMA, FULL STOP, COLON and SEMICOLON have been added to this for completeness.

4 U+104A and U+104B have the properties Sentence_Terminal and in Terminal_Punctuation according to Unicode Public Review Issue 12, http://www.unicode.org/review/pr-12.html. The position just after these is given Weight 1 below.

5 [U+104C...U+104F] have the properties Punctuation_Other according to Unicode Public Review Issue 12, http://www.unicode.org/review/pr-12.html. The position just after these is given Weight 1 below. However, U+104E can start a phrase, so it is treated as CI.

Class	***Unicode Code Points***	***Characters***
Left Quote (LQ) Parentheses and Quotation marks[6]	U+0028; U+005B; U+007B; (U+00AB;) U+2018; U+201C; (U+2039) The codes in brackets are optional.	([{ « ‘ “ ‹
Right Quote (RQ) Parentheses and Quotation marks[7]	U+0029; U+005D; U+007D; (U+00BB;) U+2019; U+201D; (U+203A;) The codes in brackets are optional.	)] } » ’ ” ›
ZWNJ (NJ)	U+200C	
Word Joiner (WJ) and ZWJ	U+2060; U+200D	
Other (OT)	Any Unicode character not mentioned above.[8]	

Table 1: Main Character Classes

Pair comparison

Table 2 Shows how to compute whether a break is allowed between a pair of characters. The horizontal rows represents the first character (B), the vertical column represents the second character (A). Each cell specifies whether a break is prohibited between B and A in the sequence BA.[9]

6 These would not be used in Traditional Myanmar.
7 These would not be used in Traditional Myanmar.
8 Sequences of characters in class OT should be handled using an appropriate line breaking algorithm for the appropriate script / language concerned.
9 The sequence BA is used to be consistent with .

Classes		***Class of 2nd Character (A)***																			
		CI	ME	VI	EV	UV	LV	AV	AN	KI	LD	VG	MD	SE	VS	PL	PV	SP	LQ	RQ	WJ
Class of 1st Character (B)	CI	?	×	×	×	×	×	×	×	×	×	×	_1	×	×	_2		_1	_2	×	×
	ME	?	×		×	×	×	×	×		×	×	_1	×	×	_2		_1	_2		
	VI	×				×							×					_1	_2	×	×
	EV	?				×	×	×	×	×	×	×	_1	×	×	_2		_1	_2	×	×
	UV	?					×	×	×	×	×	×	_1	×	×	_2		_1	_2	×	×
	LV	?						×	×	×	×	×	_1	×	×	_2		_1	_2	×	×
	AV	?		×					×	×	×	×	_1	×	×	_2		_1	_2	×	×
	AN	_2								×	×	×	_1	×	×	_2		_1	_2	×	×
	KI	_2	×	×	×	×	×	×	×		×	×	_1	×	×	_2		_1	_2	×	×
	LD	_2										×	_1	×	×	_2		_1	_2	×	×
	VG	_2											_1	×	×	_2		_1	_2	×	×
	MD	_1		×	×	×	×	×	×	×	×	×	×	×	_1	_2	_1	_1	_2	×	×
	SE	_1											_1	×	_1	_1		_1	_2	×	×
	VS	_1									×		_1	×	×	_1		_1	_2	×	×
	PL	_2	×	×	×	×	×	×	×	×	×	×	_1	×	×	_2	×	_1	_2	×	×
	PV	_2			×	×	×	×	×	×	×	×	_1	×	×	_2		_1	_2	×	×
	SP	_1	_1										_1	_1	_1	_1	_1	_1	_1	×	×
	LQ	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	_1	×	×	×
	RQ	_2	_2	_2	_2	_2	_2	_2	_2	_2	_2	_2	_2	_2	_2	_2	_2	_1	_2	×	×
	WJ	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	_1	×	×	×
	OT	_1	_1	_1	_1	_1	_1	_1	_1	×	_1	_1	_1	_1	_1	_1	_1	_1	_1	×	×

Table 2: Pair Table

Notation used in Table 2:
× indicates that breaking is prohibited.
$_{}^{1}$ indicates a potential break point – weight 1 (high).
$_{}^{2}$ indicates a potential break point – weight 2 (low).
? indicates further context analysis is necessary – see below.
Where the box is empty, it means that the sequence is a breach of Myanmar Canonical Ordering.
The default in these cases should be to allow a break – weight 2 (low).

It should be noted that the table can be simplified by combining EV, LV, UV, AV, AN, LD, VG into one diacritic class DC. In this case the sequence DC followed by DC should be taken as DC×DC. The sequence DC followed by CI should always receive context analysis DC?CI. In this case if the first character is AN, LD or VG, then there is always a weight 2 break point.

Context Analysis

In the case where context analysis is necessary, the break status can be determined from the Context Analysis Algorithm shown in . The analysis should be performed in every case where the Pair Table shows a '?' character. The figure uses both the Character Classes in Table 1 and Unicode code points for specific characters which require special handling.

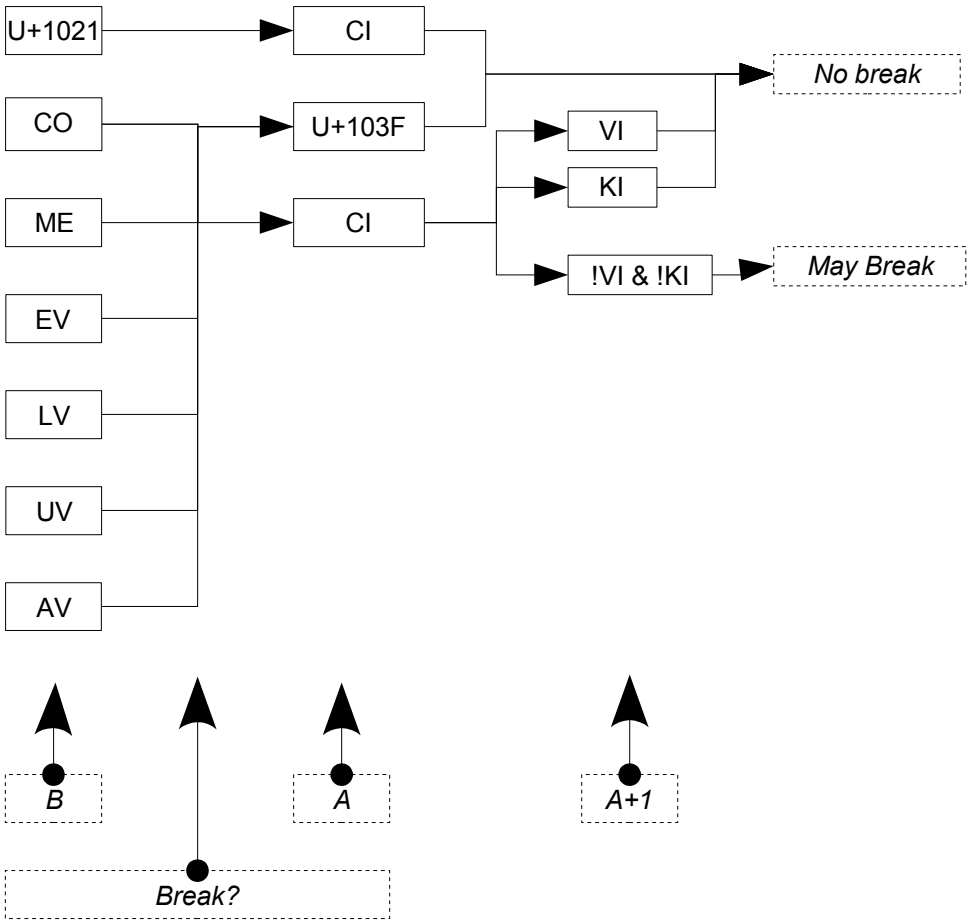

Figure 1: Context Analysis Algorithm

The algorithm assumes that the text is using Myanmar Canonical order[10] and requires the algorithm to access at most 2 characters ahead of A.

If '!' prefixes the class name, then the meaning is any character except those in the class.

Implementation Details

If possible the application should support a 2 level weighting system for potential line break points. This will give the best performance and minimize the chance of a break occurring in the middle of a word.

Two Pass Line Breaking Algorithm

A possible Line Breaking Algorithm is shown in Figure 2. It could be extended to have many different break weights, but as presented here, the maximum weight would be 2, so only 2 passes would be performed.

The assessment of whether the "Break Point is Close to the Line End" is obviously subjective. "Close" could be defined in several ways, for example:

10 See *Representing Myanmar In Unicode: Details and Examples*, Martin Hoskens & Maung Tuntunlwin, http://www.unicode.org/notes/tn11/.

- As a percentage of the remaining space over the total available being less than a certain value. e.g. <20% of total column width.
- As the remaining space being less than an absolute distance in pixels or some other device units e.g. 75 for a 75 DPI screen.

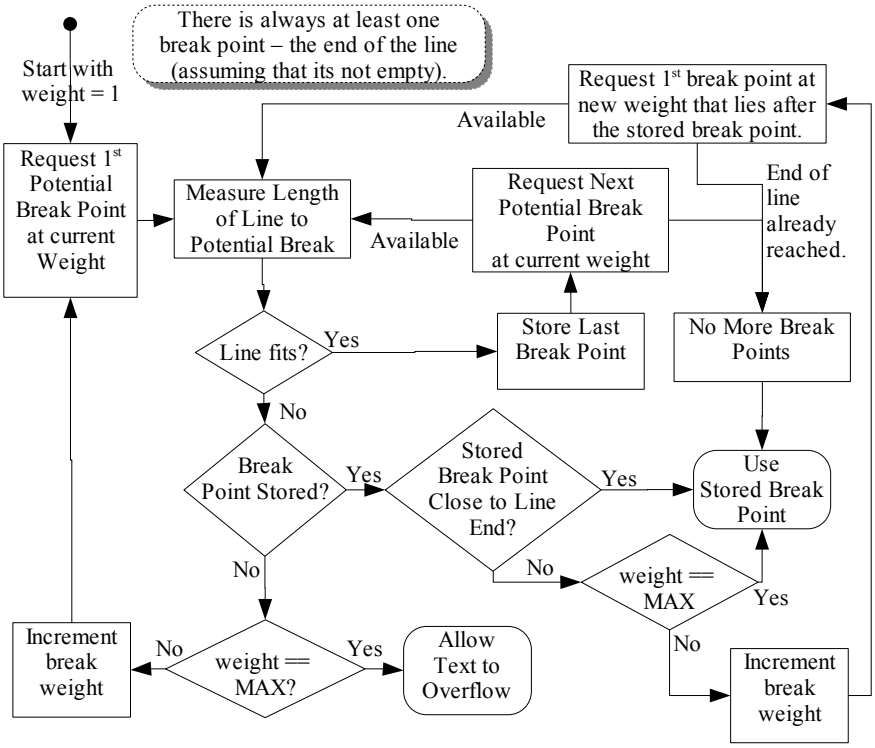

Figure 2: Multi-Pass Line Breaking Algorithm

Single Pass / Single Weight Line Breaking Algorithm

The simplest Single Pass Line Breaking Algorithm would just ignore weight 2 line breaks. However, this is likely to give poor results in narrow column text unless the text was typed by a professional using ZWSP characters to mark every word break.

The other alternative would be to treat weight 1 and weight 2 breaks as the same. This will give good performance in narrow columns, but will allow cause some multiple syllable words to be split. In this case WJ could be used to prevent breaks in multi-syllable words.

Appendix

Example using Pair Algorithm to determine potential Break Points

Consider the Myanmar sentence fragment:

ကောင်လေးတွေကျောင်းသို့သွားကြသည်။[11]

The following lines show the Unicode on the 1st line and the consonant classes and break status on the 2nd line.

```
U+1000 U+1031 U+102C U+1004 U+103A|U+101C U+1031
```

11 Example taken from *Representing Manmar In Unicode: Details and Examples*, Martin Hoskens & Maung Tuntunlwin, http://www.unicode.org/notes/tn11/.

```
  CO  ×  EV  ×  AV  ?× KI  ×  KI  _2  CO ×  EV  ×
U+1038|U+1010 U+103D U+1031|U+1000 U+103B
  VG  _2  CO  ×  ME ×  EV ?_2  CO ×   ME ×
U+1031 U+102C U+1004 U+103A U+1038|U+101E U+102D
  EV  ×  AV  ?× KI  ×  KI  ×  VG  _2  CO ×  UV  ×
U+102F U+1037|U+101E U+103D U+102C U+1038|U+1000
  LV  ×  LD  _2 CO  ×  ME  ×  AV  ×  VG  _2  CO ×
U+103C|U+101E U+100A U+103A U+104B
  ME ?_2  CO ?×  CO ×  KI  ×  SE  _1
```

References

Burmese/Myanmar: a dictionary of grammatical forms, John Okell and Anna Allott, Curzon Press, Richmond, England, December 2000.

Burmese: an introduction to the script, John Okell. Center for Southeast Asian Studies, Northern Illinois University, DeKalb, July 1994.

Canonical Ordering in Unicode Technical Note 11, *Representing Myanmar In Unicode: Details and Examples*, 2003, Martin Hoskens & Maung Tuntunlwin, http://www.unicode.org/notes/tn11/ (also available at http://www.myanmars.net/unicode/doc/20040101_myanmar_uni.pdf).

Unicode Public Review Issue 12, http://www.unicode.org/review/pr-12.html.

Unicode Standard Annex 14, *Line Breaking Properties*, Asmus Freytag, http://www.unicode.org/unicode/reports/tr14/

Acknowledgements

The author is grateful for additional feedback from Frank Tang (AOL), William W.L.K. and Maung Tuntunlwin (Myanmar Linux Users Group) and especially John Okell for carefully reviewing the document.

Keith Stribley
http://www.thanlwinsoft.org/
Issue 1.1
18 September 2006