L2/08-147

Document Type:	Working Group Document
Title:	Discussion of Batak Vowel Ordering
Source:	Rick McGowan and Ken Whistler
Status:	Individual Contribution
Action:	For consideration by WG2 and UTC
Date:	2008-04-14

We have examined document document WG2 N3320 (L2/08-011) especially Section 4, page 3, the paragraph beginning with:

> The main peculiarity of Batak rendering...

It is our opinion that this case is actually somewhat *different* from the familiar cases of "matra reversal" in, for example, the Devanagari script. In Devanagari one has the short "i" matra which reverses over previous consonant clusters. The sequence "ta + i + virama +..." is rendered as "i (ta virama ...)".

That is, a vowel sign always logically follows, but in some cases may be re-ordered around the consonant (or cluster) that it occurs *with*.

However, in Batak, according to the model presented in N3320, when a CVC syllable is closed with pangolat, the vowel sign is always re-ordered around the *following* consonant. This isn't the same behavior at all.

In this model, what is rendered as "ta pa i virama" is pronounced as "ta i pa (virama)" or "tip".

A perfectly usable alternative model to that might be to simply allow the vowel signs to be written in the normal writing order for Batak. Then, the documentation of the script would say something like this:

> ***Natural Language Processing Implications***
> The main peculiarity of reading and pronouncing Batak syllables as they are written has to do with the consonant to which the vowel sign belongs when the CVC syllable is closed with a virama (pangolat). In that case, the vowel sign is written after the second consonant, but logically belongs with the first. Thus, in the character sequence "TA + PA + I + PANGOLAT" the vowel sign "I" is pronounced with (associated with) the first consonant TA, not the second PA. Thus, in natural language processing for example, intermediate tokens should be constructed with this in mind, by re-arrangment of the phonemes in parsing:
>
> TA+PA+I+PANGOLAT → (TA+I → TI) + (PA+PANGOLAT → P) → "TIP"

That would be just as viable as the model espoused in N3320, and should not be confusing for people who are writing Batak. They are *used* to writing in that order, presumably, and then *parse* it as above when they read.

This model considerably simplifies *font* construction and *input* methods for Batak, while (hopefully) not complicating the reading, spell-checking, etc. Only syllabic parsing for NLP would be affected.

It should not noticeably affect string search, either.

There is an impact on string searching, but primarily for *substrings*, and not for whole words. The proposal, for example, notes that the visual sequence:

{pa ra o \ ka sa i \}

with closed syllables, and read "por-kis", needs to be distinguished
in searching from:

{pa ra o ka sa i}

with open syllables, and read "parokasi".

The proposal opts for encoding these as:

<pa, o, ra, \, ka, i, sa, \> versus <pa, ra, o, ka, sa, i>

so that each syllable gets a distinct string representation, and so that a substring match for "paro" would not match the first 3 characters of "por" by mistake.

However, whole word matching would not be a problem for either approach.

And in any case, it is an issue to be debated and discussed--not just assumed--how substring matching should best proceed for a script like Batak, anyway. If a visual sequence {pa ra o} is typed in as a search term, why should it *not* match the initial substring of the visual sequence {pa ra o \}, even if the latter sequence is read differently?

There is also the potential for complicating sorting, depending on how collation is done for Batak.

Our guess is that, like most Brahmi-based scripts, sorting would be done more or less on a syllabic basis, so the user would expect:

```
ka ki ku kak kik kuk kap kip kup kat kit kut
pa pi pu pak pik puk pap pip pup pat pit put
ta ti tu tak tik tuk tap tip tup tat tit tut
  etc., etc.
```

To that kind of sorting right, one must parse out and weight entire syllables -- and for that it doesn't really make any difference whether one is grabbing and weighting <ta, pa, i, pangolat> or <ta, i, pa, pangolat> for the "tip" syllable.

But it might make things more difficult for *default* sort tables, which don't parse out syllables. To get UCA approximately correct for Batak, if it were done in logical order, all it needs for the above order is:

ka < pa < ta < i < u

Of course that won't be enough to handle multisyllabic string comparisons correctly, but then that is the essential problem with syllabic orderings, anyway.

But if Batak closed syllables are encoded in visual order, then to do the same thing, the UCA will probably need to incorporate contraction entries for:

<ka, i, pangolat>
<ta, i, pangolat>
<pa, i, pangolat>

and so on for each final consonant + vowel combination. That is comparable to what needs to be done for Thai and Lao, although the particulars are different. (Thai unambiguously weights certain <V, C> combinations as if <C, V>; Batak will unambiguously weight certain <C, V, pangolat> combinations as if <V, C>.)

This would make the UCA default table entries for Batak for complex--something that the proposal was hinting at. But it isn't a different kind of complication than we already need to deal with for Thai and Lao. Particularly if Batak will required tailored syllabic sorting to get it *right*, then this becomes close to a no-op in terms of tradeoff, and doesn't weigh heavily compared to the potentially significant gains for input and rendering.