L2/22-041

꧁ Javanese orthographic syllables ꧂

Norbert Lindenberg
2022-02-11

Proposal

This document proposes to replace the description of the "Ordering of Syllable Components" in section 17.4, Javanese, of The Unicode Standard, with the following:

> ***Encoding Order of Orthographic Syllable.*** The structure of an orthographic syllable is, using the notation and character classes described in Appendix A – Extended BNF:

```
Syllable := (Base | Generic_Base) Nukta*
            (Virama Base Nukta*)*
            [Consonant_Medial && [Bottom_And_Left || Bottom]]?
            [Consonant_Medial && Bottom_And_Right]?
            [Vowel_Dependent && Left]*
            [Vowel_Dependent && Top]*
            [Vowel_Dependent && Bottom]*
            [Vowel_Dependent && Right]*
            Bindu*
            Visarga*
            Consonant_Final*
         := (Base | Generic_Base) Nukta*
            (Virama Base Nukta*)*
            Virama
Base := Consonant | Vowel_Independent | Number
```

The "notation and character classes described in Appendix A" rely on additions to that appendix of The Unicode Standard that are currently under review by the Editorial Committee in the context of similar documentation for the Kawi script. These additions are also shown in Appendix A below.

Error report

This proposal takes up an error report that David Corbett sent to the Unicode Consortium on 2019-12-08:

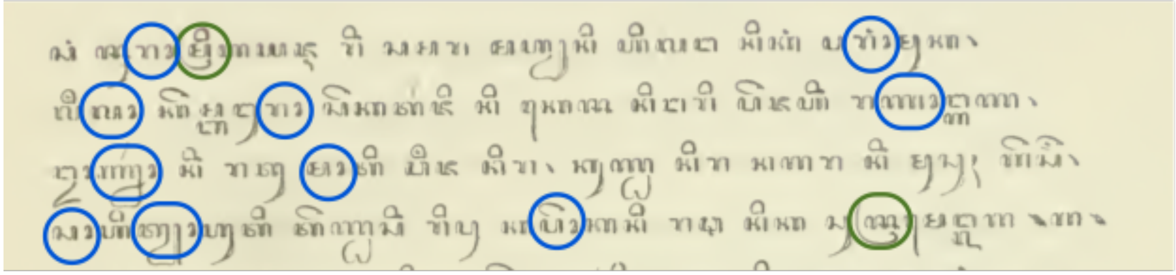
> **Subject:** Bad Javanese BNF
>
> The Javanese syllable BNF is {C F} C {{R}Y} {V{A}} {Z}. That means that -ra may only occur if -ya occurs, and -aa may only occur if another vowel sign occurs. Both restrictions are wrong.

Evaluation

The Unicode Standard up to version 14.0 describes the ordering of syllable components in Javanese as follows:

> ***Ordering of Syllable Components.*** The order of components in an orthographic syllable as expressed in BNF is:
>
> ```
> {C F} C {{R}Y} {V{A}} {Z}
> ```
>
> where
>
> `C` is a letter (consonant or independent vowel), or a consonant followed by the diacritic U+A9B3 JAVANESE SIGN CECAK TELU
> `F` is the virama, U+A9C0 JAVANESE PANGKON
> `R` is the medial -ra, U+A9BF JAVANESE CONSONANT SIGN CAKRA
> `Y` is the medial -ya, U+A9BE JAVANESE CONSONANT SIGN PENGKAL
> `V` is a dependent vowel sign
> `A` is the dependent vowel sign -aa, U+A9B4 JAVANESE VOWEL SIGN TARUNG
> `Z` is a consonant sign: U+A980, U+A981, U+A982, or U+A983

David Corbett is correct in that medial *-ra* can occur without medial *-ya*, and the long vowel *-aa* can occur without another vowel in Javanese orthographic syllables. The use of tarung may be limited to the combination with taling to write the vowel *-o* in the modern Javanese language, but it is used by itself in writing Old Javanese and Sanskrit. Examples are highlighted in the Old Javanese text sample from Bharatayuddha: Oudjavaansch Heldendicht (1903) below.



The BNF for Javanese orthographic syllables has other flaws:

- It describes the initial consonant group as `{C F} C`, which would be appropriate for Devanagari half-forms, but is not appropriate for Javanese, where the virama always combines with a following consonant to a sub- or post-joined form.
- It does not allow for a cluster that ends with a virama. Such clusters commonly occur before punctuation or at the end of the line, and occasionally in other situations.

- It provides no space for the third medial consonant, A9BD JAVANESE CONSONANT SIGN KERET, anywhere in the syllable.
- It does not allow the use of consonant placeholders such as U+25CC DOTTED CIRCLE.

The proposed new description of the syllable structure doesn't try to fix the existing description in a minimal way, but adopts the de-facto standard set by the OpenType Universal Shaping Engine. The USE provides a general cluster model, which is tailored to each supported script by using several properties of the Unicode character data: General category, Indic syllabic category, Indic positional category, and Arabic joining type. Several Javanese fonts and at least one Javanese keyboard have been implemented based on this model and are serving their users well.

Appendix A

The following additions to the Notational Conventions in the Unicode Standard, Appendix A, are currently being considered.

The operators || and `&&` are allowed in character class definitions to support class union and intersection, respectively.

The following section is added after the existing subsection "Character Classes":

> ***Predefined Character Classes for Indic properties.*** In the descriptions of certain Brahmic scripts, regular expressions specifying the encoding order of orthographic syllables can use predefined character classes based on the values of the properties Indic_Syllabic_Category and Indic_Positional_Category. For each property value, there is a predefined character class that contains the characters that have the script being discussed in their Script_Extensions property and the given property value in the Indic_Syllabic_Category or Indic_Positional_Category property. For example, within the context of the description of the Kawi script, the character classes `Consonant` and `Right` are defined as:
>
> ```
> Consonant := [\p{scx=Kawi} && \p{InSC=Consonant}]
> Right := [\p{scx=Kawi} && \p{InPC=Right}]
> ```
>
> In addition, a character class `Generic_Base` is predefined to enable the inclusion of placeholders such as U+25CC DOTTED CIRCLE and U+00A0 NO-BREAK SPACE:
>
> ```
> Generic_Base := [\p{sc=Common} && \p{InSC=Consonant_Placeholder}]
> ```