ISO/IEC JTC1/SC2/WG2 N1960
1999-02-01

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

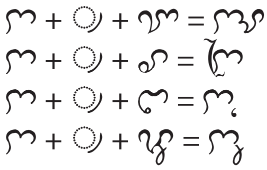
Doc Type: Working Group Document
Title: Response to Ngô Trung Việt on feedback from Cham experts
Source: Michael Everson, EGT (IE)
Status: Expert Contribution
Action: For consideration by JTC1/SC2/WG2 and UTC
Date: 1999-02-01

Ngô Trung Việt was kind enough to forward a fax from Cham expert Bùi Khánh Thế to me, and to translate the text of that fax into English for me. I believe that the coding model which I have proposed for Cham will handle each of the cases which Bùi Khánh Thế has asked us to consider.

The “virama model” proposed for encoding Cham is the model normally used in the UCS (ISO/IEC 10646 and Unicode) for scripts derived from Ashoka’s Brahmi script. (Exceptions to this are Thai and Lao, which are based on the “graphic model” of the Thai industrial standard, and Tibetan, which uses the “subjoin model”, since Tibetan has special stacking features.)

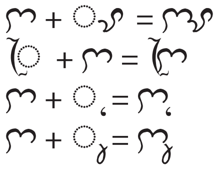
1. Representation of consonant clusters. The virama model uses the virama (Cham character U+xx3F) to “kill” the inherent *-a* vowel of a consonant, and often causes a following consonant to change in some way to join with the first consonant. The font, not the underlying encoding, is responsible for presenting the resulting syllable clusters to the user in the correct shape. The relative order of the signs as *drawn* is therefore irrelevant to the encoding (inputting via the keyboard is a separate issue). Examples in Bùi Khánh Thế’s fax are easy to represent with the virama model:

Consider



ka + VIRAMA + ya = kya
ka + VIRAMA + ra = kra
ka + VIRAMA + la = kla
ka + VIRAMA + va = kwa

In the graphic model used for Thai, the conjunct forms of *ya*, *ra*, *la*, *va* would be coded separately from the base consonants.



ka + Y = kYa (= kya)
R + ka = Rka (= kra)
ka + L = kLa (= kla)
ka + W = kWa (= kwa)

But note that the underlying representation for these is not the same as the phonetic order. This could make certain text operations, such as sorting, somewhat more difficult because *Rka* must sort under *ka*, not under *ra*. Graphic methods for encoding Sinhala and Myanmar were proposed to WG2 but were rejected, because Sinhala and Myanmar experts agreed that the virama model would work for them. It will be easier for implementors who have software that handles Brahmic scripts like Devanagari, Sinhala, or Myanmar, to adapt this software to Cham if Cham uses the virama model.

Compare	Devanagari	क + ◌् + र = क्र	ka + VIRAMA + ra = kra
	Sinhala	ක + ◌් + ර = ක්‍ර	ka + VIRAMA + ra = kra
	Myanmar	က + ◌် + ရ = ကြ	ka + VIRAMA + ra = kra
	Cham	ꨆ + ◌ + ꨣ = ꨆꨴ	ka + VIRAMA + ra = kra

2. Representation of final consonants. The final syllables described in Bùi Khánh Thế's fax can also be correctly rendered by the font, by simply entering the basic characters in sequence:

ꨆ + ◌ꨪ + ◌ꩌ = ꨆꨪꩌ	ka + i + -ṁ = kiṁ
ꨆ + ꨯ◌ꨱ + ◌ꩂ = ꨯꨆꨱꩂ	ka + au + ng = kaung
ꨆ + ꨯ◌ꨱ + ◌ꩌ = ꨯꨆꨱꩌ	ka + au + -ṁ= kauṁ

3. Representation of other syllable-final consonants. The virama model handles other syllable-final consonants quite easily. One encodes them by using the ZERO-WIDTH NON-JOINER (ZWNJ).

ꨆ + ◌ + ZWNJ = ꩀ	ka + VIRAMA + ZWNJ = k.
ꨓ + ◌ + ZWNJ = ꩄ	ta + VIRAMA + ZWNJ = t.

If you write ꨆ + ◌ + ZWNJ + ꨣ, you will get ꩀꨣ (k.ra). Again, inputting software can allow Cham users to type according to their preference regardless of the underlying encoding.

4. Representation of medial and final vowels. The point raised by Bùi Khánh Thế here is a little bit more controversial, but again, the Brahmic model offers a fairly easy solution. The vowels appear to be decomposable in the writing system, but as linguistic entities they are indivisible units. In some other Brahmic scripts (such as Bengali, Malayalam, Oriya, Sinhala, Tamil), similar "multipart" vowels have been encoded in a similar fashion. The font can handle the sequences.

ꨆ + ◌ꨩꨭ = ꨆꨩꨭ	ka + ū = kū
ꨆ + ꨯ◌ꨮ = ꨯꨆꨮ	ka + e = ke
ꨆ + ꨯ◌ꨩ = ꨯꨆꨩ	ka + ai = kai
ꨆ + ꨯ◌ꨱ = ꨯꨆꨱ	ka + au = kau
ꨆ + ◌ꨩꨲ = ꨆꨩꨲ	ka + ư̄ = kư̄

One *could* encode these graphically.

ꨆ + ◌ꨩ + ◌ꨭ = ꨆꨩꨭ	ka + ā + u = kāu = kū
ꨯ◌ + ꨆ + ◌ꨮ = ꨯꨆꨮ	e + ka + ơ = ekơ = ke
ꨯ◌ + ꨆ + ◌ꨩ = ꨯꨆꨩ	e + ka + ā = ekā = kai
ꨯ◌ + ꨆ + ◌ꨱ = ꨯꨆꨱ	e + ka + ơ̄ = ekơ̄ = kau
ꨆ + ◌ꨩ + ◌ꨲ = ꨆꨩꨲ	ka + ā + ư = kāư = kư̄

Or three of these could, in principle, be entered otherwise and still rendered correctly by the font:

ꨆ + ꨯ◌ + ◌ꨮ = ꨯꨆꨮ	ka + e + ơ = keơ = ke
ꨆ + ꨯ◌ + ◌ꨩ = ꨯꨆꨩ	ka + e + ā = keā = kai
ꨆ + ꨯ◌ + ◌ꨱ = ꨯꨆꨱ	ka + e + ơ̄ = keơ̄ = kau

It is better for consistency to have the same coding conventions for each vowel in Cham. This facilitates lexical work, and, as noted above, also makes the software developed for other Brahmic scripts easier to adapt for Cham.

Assuming that the discussion here satisfies Bùi Khánh Thế, there is an important question which I hope he can answer: is the order of the characters in my proposal correct? I have based it on *Từ Điển Chăm Viết*, which I analyzed in order to put the characters into the correct alphabetical order. I am not sure that, algorithmically, the ordering given in *Từ Điển Chăm Viết* is perfect and error-free, but it is the order I endeavoured to follow. In particular I would like confirmation of the position of CHAM SIGN NG (U+xx0B).