L2/18-248

Rendering of and Prescriptions for Marks in the Multilingual Thai Script

Author: **Richard Wordingham**
Date: **23 July 2018**

Table of Contents

Summary....1
Introduction....1
Remarks....1
References....4

Summary

1. There is a widely distributed rendering system that can already handled the strings considered in L2/18-216.

2. There are consistent encodings for all the combinations considered in that document.

3. <U+0331, U+0E3A> is best interpreted as a a macron below with a vowel killer rendered below it – this is already the intended interpretation in some fonts.

4. The rendering rules can be greatly simplified by normalising strings and using an eclectic choice of canonical form.

5. Inconveniently, both orders <U+0E34, U+0E4A> and <U+0E4A, U+0E34> are already in use.

Introduction

This document examines the rendering rules that are suggested in L2/18-216, “Canonical Ordering of Marks in Thai Script”. It has been prepared using a Thai font (Garuda) provided by the Thai Linux Working Group (TLWG) and generally included in Linux distributions, as interpreted by LibreOffice Version 6.0.5.2 using HarfBuzz Version 1.2.7 and running under Ubuntu 16.04. I conclude that, conceptually at least, reasonable rendering rules are less complex than that document proposes.

Remarks

Point 1:My first point is that, with one exception, the strings shown in L2/18-216 can already be rendered satisfactorily using the Garuda font. The one exception is that because of a simple bug in the TLWG fonts, “กฺ̱” is misrendered as “กฺ̣” (https://github.com/tlwg/fonts-tlwg/issues/7 refers.) I fixed the bug in my own copy to produce this document, and the TLWG has already started corrective work. The separation of the marks is not clear in all the fonts; Garuda has the best separation of the TLWG fonts. The following tables shows how the examples render.

Classes	**Marks in order of classes**	**Eqv?**	**Marks in reverse orders of classes**
Case 1 Non-interacting, non-zero			
9, 107	<U+0E01, U+0E3A, U+0E48> “กฺ่”	≡	<U+0E01, U+0E48, U+0E3A> “กฺ่”
9, 230	<U+0E01, U+0E3A, U+0303> “กฺ̃”	≡	<U+0E01, U+0303, U+0E3A> “กฺ̃”
103, 107	<U+0E01, U+0E38, U+0E48> “กุ่”	≡	<U+0E01, U+0E48, U+0E38> “กุ่”
103, 230	<U+0E01, U+0E38, U+0303> “กุ̃”	≡	<U+0E01, U+0303, U+0E38> “กุ̃”
107, 220	<U+0E01, U+0E48, U+0331> “ก่̱”	≡	<U+0E01, U+0331, U+0E48> “ก่̱”
220, 230	<U+0E01, U+0331, U+0303> “ก̱̃”	≡	<U+0E01, U+0303, U+0331> “ก̱̃”
Case 2: Non-interacting marks, one in Class 0			
Classes 0, 9	<U+0E01, U+0E34, U+0E3A> “กิฺ”	≢	<U+0E01, U+0E3A, U+0E34> “กฺิ”
Classes 0, 103	<U+0E01, U+0E34, U+0E38> “กิุ”	≢	<U+0E01, U+0E38, U+0E34> “กุิ”
Classes 0, 220	<U+0E01, U+0E34, U+0331> “กิ̱”	≢	<U+0E01, U+0331, U+0E34> “ก̱ิ”
Case 3: Two marks below, distinct classes			
Classes 9, 103	<U+0E01, U+0E3A, U+0E38> “กฺุ”	≡	<U+0E01, U+0E38, U+0E3A> “กฺุ”
Classes 9, 220	<U+0E01, U+0E3A, U+0331> “กฺ̱”	≡	<U+0E01, U+0331, U+0E3A> “กฺ̱”
Classes 103, 220	<U+0E01, U+0E38, U+0331> “กุ̱”	≡	<U+0E01, U+0331, U+0E38> “กุ̱”
Case 4: Two marks above, one in class 0:			

Classes	Marks in order of classes	Eqv?	Marks in reverse orders of classes
Classes 0, 107	<U+0E01, U+0E34, U+0E48> “กิ่”	≢	<U+0E01, U+0E48, U+0E34> “ก่ิ”
Classes 0, 230	<U+0E01, U+0E34, U+0303> “กิ̃”	≢	<U+0E01, U+0303, U+0E34> “ก̃ิ”
Case 5: Two marks above, in different non-zero classes:			
Classes 107, 230	<U+0E01, U+0E48, U+0303> “ก่̃”	≡	<U+0E01, U+0303, U+0E48> “ก่̃”

Case 6: Three marks above, all in different classes:			
Canonical eqv. class	**Mark ccc in order**	**First member**	**Second member**
A	0 107 & 230	<0E01, 0E34, 0E48, 0303> “กิ่̃”	<0E01, 0E34, 0303, 0E48> “กิ่̃”
B	107 0 230	<0E01, 0E48, 0E34, 0303> “ก่ิ̃”	
C	107 & 230 0	<0E01, 0E48, 0303, 0E34> “ก่̃ิ”	<0E01, 0303, 0E48, 0E34> “ก่̃ิ”
D	230 0 107	<0E01, 0303, 0E34, 0E48> “ก̃ิ่”	

Point 2: It was remarked that some of the naïve interpretations of the code point sequences cannot be produced because the naïve interpretation of a canonical equivalent is better. This is certainly not true of the rendering system used for this document; the deprecated alternative can be forced by inserting U+200C ZWNJ between the commuting marks. Good rendering in the OpenType system requires complete tables for mark-to-mark positioning.

<U+0E01, U+0E38, U+0E3A> “กฺุ” but <U+0E01, U+0E38, U+200C, U+0E3A> “กุ‌ฺ”

<U+0E01, U+0E38, U+0331> “กุ̱” but <U+0E01, U+0E38, U+200C, U+0331> “กุ‌̱”

<U+0E01, U+0E48, U+0303> “ก่̃” but <U+0E01, U+0E48, U+200C, U+0303> “ก่‌̃”

<U+0E01, U+0E34, U+0E48, U+0303> “กิ่̃” but <U+0E01, U+0E34, U+0E48, U+200C, U+0303> “กิ่‌̃”

<U+0E01, U+0E48, U+0303, U+0E34> “ก่̃ิ” but <U+0E01, U+0E48, U+200C, U+0303, U+0E34> “ก่‌̃ิ”

<U+0E01, U+0E3A, U+0331> “กฺ̱” but <U+0E01, U+0E3A, U+200C, U+0331> “กฺ‌̱”

Point 3: Note that preferred form treats U+0331 as being more likely to be a nukta, its main use in the Thai script, as used to give English pronunciation in some English-Thai dictionaries, and allows for U+0E3A to be used as a vowel killer, its general use when giving pronunciations, and thus potentially applicable to consonants with a nukta.

Point 4: Now, the TrueType-flavoured OpenType TLWG fonts bring about the standardisation by first swapping marks to select a canonical form, and then further swapping interacting marks to select the desired form, as proposed in L2/18-216. However, if the rendering engine takes the approach of first normalising the input string, then the rearrangements can be given effect by selecting bespoke canonically representative strings not with the mark class order <9, 103, 107, 220, 230>, but with the mark class order <220, 9, 103, 230, 107>, and rendering that. Such an approach is already required for Hebrew accents, and is highly desirable (and implemented in HarfBuzz) for Tai Tham tone marks (ccc=230) and invisible stacker (ccc=9).

Thus the seven rendering rules proposed in L2/18-216 can be reduced to the following rules for the bespoke canonical representative:

1) The sequence of above mark followed by below mark is invalid; this can be indicated by the deprecation sign (U+25CC). This is a significant simplification of the current rules.

2) Indic reordering is required for the sequence of tone mark and U+0E33 THAI CHARACTER SARA AM, as at present.

Point 5: There is unfortunately a problem with treating the sequence of <U+0E34 THAI CHARACTER SARA I, U+0E3A THAI CHARACTER PHINTHU> as invalid; Martin Hosken reports its use (in https://lists.freedesktop.org/archives/harfbuzz/2014-January/004060.html) with the explanation (in https://lists.freedesktop.org/archives/harfbuzz/2014-February/004101.html) that U+0E3A is also used as a vowel nukta; this usage is given, without typing sequence, in References 1 and 2.

References

1. Thomas, Dorothy; *Popularizing the Northern Khmer Orthography: Sociolinguistics in Action* Workshop in Northern Khmer Orthography, Surin; Mon-Khmer Studies Journal. 16–17: 255–265

   (available at http://sealang.net/sala/archives/pdf8/thomas1987-1988northern.pdf)

2. Royal Institute of Thailand ราชบัณฑิตยสถาน [Ratchabandit Sathan] *Handbook of the system of writing Northern Khmer in the Thai Script คู่มือระบบเขียนภาษาเขมรถิ่นไทยอักษรไทย ฉบับราชบัณฑิตยสถาน.* [Khumue Rabob Khian Phasa Khamen Thin Thai Akson Thai] (Bangkok, Royal Institute, 2013)