Characters with Multiple Accents (e.g. Lithuanian), Recent Keyboard Standards, and Microsoft's MSKLC

Karl Pentzlin – 2012-02-03

1. Introduction

In L2/12-026 = WG2 N4191 "Proposal to add Lithuanian accented letters to the UCS", the Lithuanian NB requests to assign single code points for 35 characters with multiple accents (i.e. encoding them as precomposed characters), although all of these already are represented in Unicode by sequences of existing base characters and existing combining characters. (In fact, all of the requested characters are already listed as "Named Sequences".)

Especially, on p.5 it is stated:

All this means that Lithuanian accented letters should be expressed by sequences. But it is not an option. Text entering, editing and processing is more complicated.

This paper discusses the fact that the Lithuanian NB is correct with this statement, but that the problem does not originates in current standardization, but in missing compliance of current software to current standards.

The presentation issues already are addressed in an earlier paper of the author (L2/12-048 = WG2 N4193).
This paper focuses on the text entering issue.

2. Multiple Accents in Current Keyboard Standards

2.1 Diacritical Marks in ISO/IEC 9995-3:2010

Keyboard layouts are standardized by the international standard series ISO/IEC 9995 "Information Technology — Keyboard layouts for text and office systems".
Herein, ISO/IEC 9995-3:2010 "Complementary layouts of the alphanumeric zone of the alphanumeric sections" introduces diacritical marks (as such are contained in the complementary layouts standardized here).
They are described in the standard as follows in section 5.2 "Operations of keys with diacritical marks":

> *Diacritical marks appear above or below certain letters or overstrike some letters, and all of them are nonspacing characters.*
>
> *Actuating a key with a diacritical mark, or a sequence of such, followed by actuating a key with a letter, a symbol, or another diacritical, shall indicate that the graphical symbols of the implied characters are intended to be combined.*
>
> *The order in which two diacritical marks that apply to the same letter are entered does not matter.*

Moreover, the results of specific combinations of a diacritical mark entered before another character are standardized as "peculiar characters". Examples are:

First character	Second character	Result character
◌́ U+0301	␣ U+0020	´ U+00B4
◌̵ U+0335	␣ U+0020	⏥ U+2212
◌̀ U+0300	◌̀ U+0300	◌̏ U+030F
◌̵ U+0335	T U+0054	Ŧ U+0166
◌̲ U+0331	< U+003C	≤ U+2264

(The symbols used here are standardized in IEC 60417 and are defined in the current drafts of ISO/IEC 9995-7 Amd1 and 9995-10. E.g., the flat rectangle (IEC 60417-6140) denotes that the diacritical mark is to be entered before the base letter, while the dotted circle known from the Unicode code tables (IEC 60417-6137) would denote that the diacritical mark is to be entered after the base character.)

2.2 Diacritical Marks in the German keyboard standard DIN 2137:2012

The reworked German keyboard standard references the ISO/IEC 9995 series. In Part 1, all specific German details are standardized within this framework, while Part 2 addresses all issues which, while not being specific for German, were to be standardized but are not found in international standards at this time.

Especially, in Part 2 Section 4 "Tottasten" ("Dead keys"), the entering of diacritical marks is standardized by the following refining of ISO/IEC 9995-3:

A “dead key” corresponds to entering a “combining character” (according to ISO/IEC 10646).

Step 1. If a combining character is entered, it is buffered by the keyboard functional unit (KFU; i.e. the responsible combination of hardware/firmware/driver software/etc.).

Step 2. If a character is entered while the KFU is buffering characters, proceed as follows:

Step 2.1 If a "backspace" is entered while the KFU is buffering characters and if this "backspace" does not cancel the selection of a group or a level, the character sequence is deleted.

Step 2.2 Otherwise, if the combination of the last character previously buffered and the newly entered character is stated in the list of special combinations of dead keys (see [the relevant tables in ISO/IEC 9995-3:2010]), the last buffered character is replaced by the result character from this list.

Step 2.3 Special case: If this character is [U+200C zero width non-joiner], the saved sequence is output unchanged (without the zero-width non joiner itself). In this case, the character processing accepting the output (text processing software, etc.) is responsible for the processing (e.g. the characters in a Unicode setting can be applied to the previous character)

Otherwise, the newly entered character is appended to the sequence of buffered characters.

Step 2.4 If the last character in the buffered sequence is no longer a combining character, proceed as follows:

Step 2.4.1 a) in a Unicode setting, the last character in the buffered character sequence is re-sorted to the beginning of this sequence; following this, the Unicode normalization form C (canonical composition) is applied to the resulting character sequence.

b) in other settings, analogously combined characters or character sequences are generated in the way appropriate for the setting.

Step 2.4.2 The character sequence thus generated is output by the KFU to the character processing.

(Note that the German standard layouts "T2" and "T3" in fact *have* a ZWNJ key. The special use of ZWNJ in step 2.3 reflects the fact that the semantics of ZWNJ gives no sense to its use in character sequences following a combining character, thus the key can be assigned to another function when entered after dead keys.)

(Note also, the step 4.2.1 expresses the ISO/IEC 9995-3 statement "The order in which two diacritical marks that apply to the same letter are entered does not matter" in a specific Unicode-related way.)

2.3 Conclusion: The input of multiple diacritical marks imposes no problems in standard-conformant systems

For instance, a Lithuanian ę (e with acute and ogonek) is entered simply by the key sequence:

[acute] then [ogonek] then [e],

or, yielding not only the same character visually but also the same character code sequence, by:

[ogonek] then [acute] then [e] — or even by:

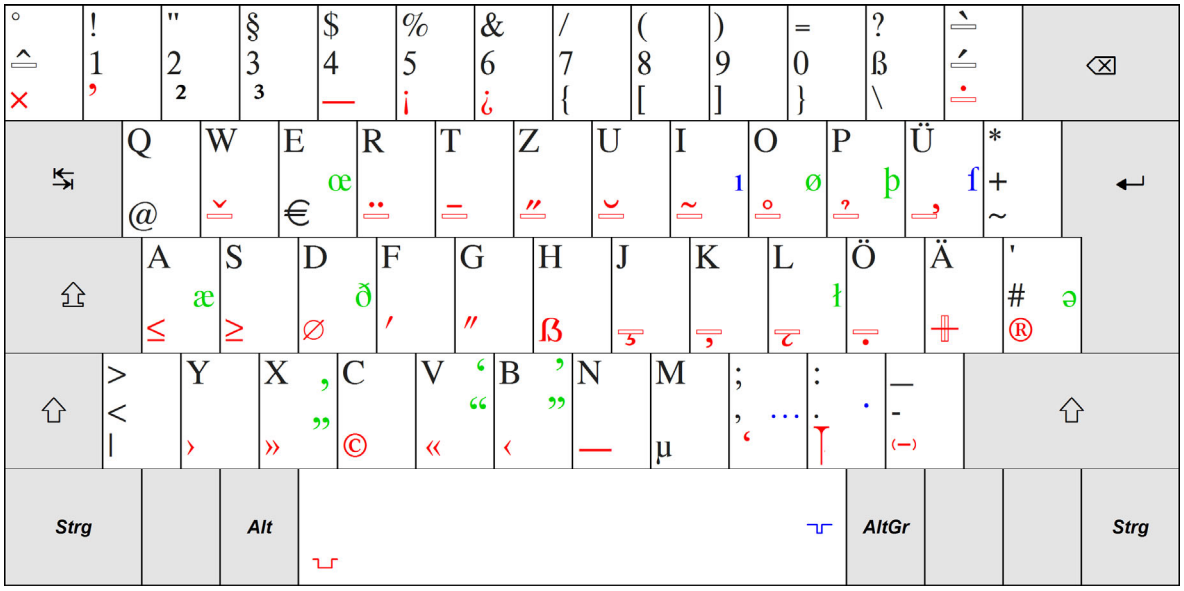
[acute] then [e with ogonek] — (on keyboards which have the latter key, like Lithuanian ones).

Thus, if current software were compliant to international standards, there would be no issue to enter characters with multiple diacritical marks for Lithuanian:

a. The "Lithuanian New" keyboard (as supplied with the German version of Microsoft's Windows Vista, at least) contains an "acute" key as well as an "e with ogonek"key.
   If this layout is based on an Lithuanian standard which in turn is based on the international standard series ISO/IEC 9995 by giving the due references, the input of an e with acute and ogonek has to work.

b. The "common secondary group" defined in ISO/IEC 9995-3:2010 contains all Lithuanian accents.
   Thus, the input of an e with acute and ogonek, while being a little more clumsy (as each diacritical mark input has to be preceded by a Group 2 selection), has to work with all keyboard layouts according to this standard.
   (It is admitted, however, that until now there seems to be no national standard besides the new German DIN 2137:2012 which includes the full "common secondary group".)

c. The new German standard DIN 2137:2012 contains all diacritical marks needed for Lithuanian (in fact, all ones needed for any Latin-written official E.U. language + Turkish and Vietnamese) in the primary group, as the following picture of the new German T2 layout shows:



Thus, in conformant systems, the e with acute and ogonek is simply entered as follows:

[acute] then [AltGr + ł] then [e] — or, with the same effect: [AltGr + ł] then [acute] then [e].

Some notes on the depicted German T2 layout:

Characters to be entered with AltGr are shown in the lower left corner of each keycap.
Red color indicates an additional assignment, compared with the previous standard edition.

The symbol at AltGr+Ä is IEC 60417-6079-1 "Horizontal Stroke Applicator", used e.g. to enter Đ/đ/Ħ/ħ.
The "nail" symbol at AltGr+"." is IEC 60417-6077-1 "Zero-Width Non-Joiner".

Characters to be entered after actuating the Group 2 selector are shown in the right part of the keycap.
Green color on letters indicates that the capital form of that letter can be entered then using the Shift key in the usual way.

The Group 2 is, in compliance with ISO/IEC 9995-3:2010, a subset of the "Common Secondary Group" standardized there.

The Group 2 selector is either a dedicated key right of AltGr, or the key combination Shift + AltGr.

3. A Case Study: MSKLC (Microsoft Keyboard Layout Creator V1.4)

Microsoft generously offers its MSKLC for download and use without charge. This software allows it to create keyboard layouts which then can be installed on all recent versions of Microsoft Windows.
In fact, the tool generates a ".DLL" file which obviously contains tables which are used by the Windows keyboard driver itself, to translate sequences of key actuations into sequences of characters.
Such ".DLL" files are associated also to the keyboard layouts delivered with any version of Microsoft Windows, and the ".DLL" files generated by MSKLC are administrated in the same way as the delivered ones.

However, this tool has some severe constraints:

- Only single diacritical marks can be combined a base character,
- and can only yield a single code point.

All available alternatives ("shareware" software), while being sometime more comfortable in some specific issues, some the same constraint.

This raises the suspicion that the issue is not in the tool, but in the underlying keyboard model used by the operating system:

- The keyboard driver of even the most recent Microsoft Windows versions sold at this time (Windows 7, as of Feb. 2012) is based on a model that combinations of "dead keys" + base character have to denote a character which is represented by a single code point. (presumably a simple table-driven model).

If this is true, it is in fact far behind current software technology, especially ignoring the fact that accented letters are not necessarily given a single code point in Unicode.

This exactly is a situation which makes the Lithuanian request for single code points for their multiply accented letters sound and understandable.

On the other hand, to be compliant to the new German standard, this will not suffice. This requires:

- To give dead keys representing combining characters the function of applying a diacritical mark to the base character entered afterwards, independent of the fact whether the result is encoded in Unicode as precomposed character by a single code point, and independent whether the result is included in any special list.
- To allow sequences of more than one dead keys to enter diacritical marks on the same base character entered after the base characters, employing the mechanism described in section 2.2.
- To allow a secondary group, to be selected as specified in ISO/IEC 9995-2:2009 (i.e., by a dedicated key right of the AltGr key, or the combination Shift + AltGr as a substitute.
- To consider a given list of "peculiar characters" as outlined in ISO/IEC 9995-3:2010, possibly augmented with additional combinations by national standards.
- To support characters outside of the BMP without any restriction whether as combining characters or as base characters (unlike MSKLC, but admittedly not requested by the German standard).

The new German standard will probably go into the requirements for office systems to be used in administrations within Germany.

- ✓ *Any manufacturer is hereby alerted that they probably will lose open tenderings related to such office systems if they are not fully compliant with the German standard, while the competition is.*

Accidentally, as a side effect, compliance to these German requirements includes the proper handling of Lithuanian letters with multiple diacritical marks, without having to change any international standard (neither on keyboards nor on character encoding).