The Worldwide Multilingual Computing (4):

4Q－4 Essentials for the Multilingual Text Manipulation

Tomoko Inagawa Kataoka*, Kazutomo Uezono†, Tadao Tanaka‡, Toshio Oya†, Hidejiro Daikokuya†, Kenji Maruyama‡, Shoichiro Yamanishi†, Yutaka Kataoka* and Hiroyoshi Ohara†

* Centre for Informatics, Waseda University † School of Science and Engineering, Waseda University
‡ Research and Development, Japan Computer Corporation

1. Introduction

Researches on the scripts all over the world have revealed the finite mapping patterns among a *codepoint*, a *glyph*, and a *character* [1]. Information contained under the traditional name of *Orthography* is found to be classified into sub-categories, one of which is *Writing Convention*, information to determine a final glyph.

Text Processing has been generalized to be the *Basic Text Manipulation* as far as truly language/codeset/fontset independent manipulations are concerned, as deletion, insertion, search, line separation and *displaying* and *pointing* characters. The process unit has been normalized as one *character*, our WC codepoint. Each manipulation is realized with the language/codeset/fontset independent operating functions.

More advanced language/codeset/fontset dependent text processing like natural language processing or parsing, is analyzed to utilize more flexible unit defined as *TMC* (Text Manipulation Code) [2]. Such specific information is stored in TMC bitfield, which has been designed as customizable according to its users' requirements and needs.

2. Character Codeset Designs

A codeset must be able to generate the sufficient numbers of codepoints computable for *character* and *glyph*. Every character can be defined as *position dependent* and *direction dependent*, and it is proved in [3] that one character is a set of, like a label for, 16 final glyph candidates [Also refer to Talk 1]. A codeset can be classified into two categories: 1) *Character Name defined*, or 2) *Glyph defined*, but as a matter of fact, the Glyph definition (Case (7) in Figure 1) is out of the question, for a character cannot be determined from glyph(s).

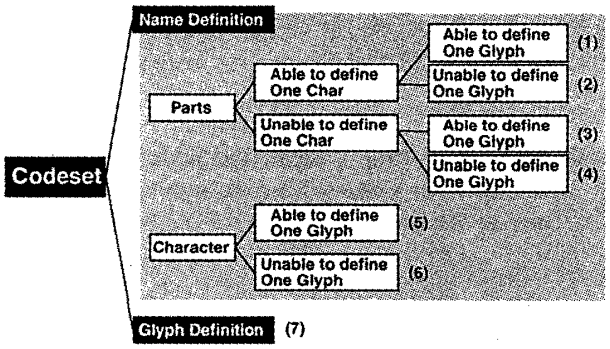

Figure 1: Codeset Designs

Name defined codesets are divided into: a) *Parts defined*, or b) *Character defined* ones. Only viable combinations of the potential character and glyph definitions are cases (1) and (5).

For a Multilingual I/O and TM/C System to operate, all codesets with extended codepoints are specifiable by ISO 2022 [4] and ISO 6429 [5]. And final glyph sets of non-fixed length codepoints of Conjunct Syllabic Scripts have been supported beyond ISO specifications [6].

TIS 620-2533:1990 [7] is a typical example of Parts definition. It has no internal rules to derive a character from the parts, so is supplemented with such rules. In certain cases it has to be given a delimiter to decide the extent of one character, process unit (Case (3)). On the other hand, IS 13194:1991 [8] is able to provide the computable codepoints for one character.

GB 8045:87 for Mongolian scripts defines both glyphs/glyph parts and character/glyph names, but lacks amounts of ligatures and script variants. More than one character which happen to have the same glyph are given the same codepoint. Thus, even if it is made to display all final glyphs with those missing glyph sets supplied, the codeset is still uncomputable for one character (Case (4)); it cannot be utilized for the text manipulation. ISO 10646 causes all troubles.

3. Basic Text Manipulation with WC

Multilingual means *generalized*. Text processing is an operation with specifying its target, a process unit or a scope of processing on the memory image. Usually the target process unit is a character, so once the definition and specification of the extent of "process unit = one character" is established, text manipulation can be generalized whatever codeset/language/script is involved.

As to be clarified in Talk 5, there will be mismatches between the memory images and the sequences to be displayed, so our WC has been redefined to normalize itself as one character, with its bitfield storing the essential information for text manipulations including displaying functions [Talk 2]. Our Multilingual Text Widget to absorb the codeset dependencies provides the supplement windows and extended cursor information. It also meets the requirement of the principled display of more than one line; it is able to specify where and to which direction the line feeds for any multilingual text.

It is fairly easy to specify one character for Phonemic, Ideogrammic, Pure Syllabic or Phonemic Conjunct Syllabic Script; a glyph is. (Notice that by a 'glyph' we

do not mean a final glyph for Position Dependent Perso-Arabic or Mongolian.) In Syllabic Conjunct Syllabic Scripts like Devanagari or Thai, non-fixed, m-by-n mappings should be supported between glyph and character. In fact, the concept of one character could be varied according to each native user, and for such users hoping to manipulate units other than our WC, TMCs and their associated functions are ready to serve.

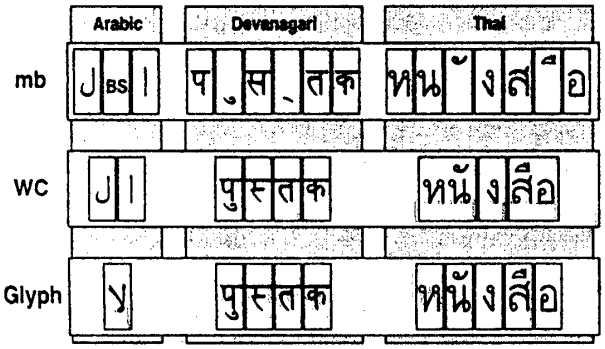

Figure 2: Process Units

4. Advanced Text Processing with Language/ Codeset Dependency

We propose for the other category in text processing beyond the scope of the language/codeset independent basic text manipulation. The distinctions between the two categories had been obscure. This advanced language/codeset dependent text processing treats extended comparisons/searches for Perso-Arabic or Mongolian cases by manipulating the position dependency information stored (and changed) in *TMC Attribute field.*

More specific operations in terms of glyph or even glyph size, or of *script variants* will also be possible. A mixed text with Arabic Nasxi and Nastaliq styles with potential different writing conventions, or the one with more than one language, Sanskrit and Hindi, for example, described by the same Devanagari scripts, will also be supported. Highly user-specific text processing with canonical orderings will be realized.

With these flexibilities to cope with the language/ codeset/fontset/style dependencies, reinforced by the researches on the specification of where to dissect the sequence into words, especially for those Syllabic Conjunct Syllabic as Thai or Lao, the further step of *Multilingual Natural Language Parsing* is on the way. Multilingual programming languages based on Multilingual FORTH [Talk 7] will play a major role here.

5. Summary and Further Remarks

Researches described so far made the essential information and functions clearer. Those compared with ISO specifications [9, 10, 11] in terms of character and its manipulations, it is possible now to redefine those specifications to cover multilingual processings.

Layered, finer structures of text processing, namely, the basic text manipulation and the advanced text processing with WC and TMC codepoints so designed, can lead to various applications in the fields of language education, linguistics or logic. Apparent mysteries and complexities around the scripts over the world have been swept, so new viewpoints and technologies will emerge for the library and database utilities. Those advanced researches have already started and the results been used to prove the whole system.

The multilingual researches based on characters, orthographies and generalized essential processings brought a completely new horizon to computability and data exchangeability.

References

[1] Kataoka, Y., et al., 1994. Multilingual I/O and Text Manipulation System(1): The Total Design of the Generalized System based on the World's Writing Scripts and Code Sets, Proceedings of the 49th General Meeting of IPSJ, Vol. 3, September 1994, pp. 299-300.

[2] Kataoka, T. et al., 1994. Multilingual I/O and Text Manipulation System (3): Extracting the Essential Informations from World's Writing Scripts for Designing TMC and for the Generalizing Text Manipulation, Proceedings of the 49th General Meeting of IPSJ, Vol. 3, September 1994, pp. 303-304.

[3] Kataoka, Y. et al., 1995. Codeset Independent Full Multilingual Operating System: Principles, Model and Optimal Architecture, IPSJ SIG System Software & Operating System, 68-4, pp. 25-32.

[4] ISO/IEC 2022: 1986, Information processing – 7-bit and 8-bit coded character sets – Code extension techniques.

[5] ISO/IEC 6429: 1992, Information processing – Control functions for coded character sets.

[6] Uezono, K. et al., 1994. Multilingual I/O and Text Manipulation System (2): The Structure of the Output Method Drawing the World's Writing Scripts beyond ISO 2022, Proceedings of the 49th General Meeting of IPSJ, Vol. 3, September 1994, pp. 301-302.

[7] TIS 620-2533 (1990), Thai Character Codes for Computers, Thai Industrial Standards Institute, Ministry of Industry, Thailand.

[8] IS 13194:1991, Indian Script Code for Information Interchange--ISCII, Bureau of Indian Standards, India.

[9] ISO/IEC 9945-1: 1990, Information technology – Portable Operating System Interface (POSIX) Part 1: System Application Program Interface (API) [C Language].

[10] ISO/IEC 9899: 1990. Programming language C.

[11] ISO/IEC 9899: 1990/DAM 3, *Draft* Amendment 1:1994 (E), Programming languages – C AMENDMENT 1: C Integrity.