4S－6

Multilingual I/O and Text Manipulation System (4): The Optimal Data Format Converter to/from MB/WC/TMC

Tadao Tanaka†, Yutaka Kataoka*, Kazutomo Uezono†, Tomoko Kataoka* and Hiroyoshi Ohara†
* Center for Informatics, Waseda University † School of Science and Engineering, Waseda University

1. Introduction

In order to ensure computability and interoperability, *the Multilingual I/O and TM/C System* is based on the *Meta Converter System* described in the first paper of this series[1].

Basically, conversions are classified into the following categories, 1) encoding scheme conversion, 2) trans-character-set conversion and 3) trans-unit conversion. The encoding scheme conversion is required to convert/reverse-convert a graphic or a control character set by ISO 2022[2]. The trans-character-set conversion is required to convert/reverse-convert multiple elements of multiple sets to an element of a set with ISO/Non-ISO extension rules[3]. The trans-unit conversion is required to generate/reverse-generate an element of field oriented data set (*TMC*, described in the third paper of this series[4]) from WC. The Meta Converter System was designed and developed to execute such conversion with the fastest speed and with the largest flexibility. The system used compiler technology to realize such abilities.

2. Informations in mb, WC, IWC and Final Glyph Set

Mb (multi-byte code set) consist of *graphic character sets* (GCS) and *control character sets* (CCS) defined by ISO and national standards. Relations between codepoint in GCS and codepoint in final glyph set are classified into the following categories, 1) 1 by 1 mapping, 2) 1 by M(Multiple), 3) M by 1 and 4) M by M. And sequence in mb contains designation, invocation and control sequence (character). Designation and invocation sequences are used to specify a set. On the other hand, control sequence contains functions that select glyph shape by *Current Writing Direction* and *Origin of Line*. Thus, the following informations must be kept in WC to ensure reverse conversion, 1) Font ID, 2) Glyph Index, 3) Writing Direction, 4) Origin of Line, 5) GCS-ID, 6) CCS-ID and 7) Determinative/Control Function ID. By the fields above, each element in WC can be always unique and be definable in our system.

For drawing, information 1 to 4 are used in IWC. In OM, IWC is converted to informations for primitive drawing function(s) (described in the second paper of this series[5]).

3. TMC and Generalized Text Manipulation

A codepoint of a TMC has *Character-ID* field for identifying it and *Attribute* field to retain character categories that make text manipulation functions code set independent[4].

The basic text manipulation operations are *Insertion, Deletion, Comparison, Search, Replacement* and *Line Separation* and the operations are executed with TMC-ID. To realize them, the following informations are essential in TMC-ID 0, 1) Permission of line separation, 2) Permission to put it at the top of line, 3) Permission to put it at the end of line, 4) Control character or not, 5) Visible or not, 6) Variable shaped glyph or not, 7) Writing direction, 8) Line direction, 9) Origin of line and 10) Origin of line direction. Since the Attribute field is definable, it is possible to add more informations into the field.

4. The Meta Converter System

The Meta Converter System consists of *Meta Converter Table Compiler, Trans-Unit Converter, Character Information Functions, Text Manipulation Functions*, and *Inter-Process Communication Assisting Functions*. The encoding scheme converter and the trans-character-set converter are parts of Trans-Unit Converter.

4.1. The Meta Converter Table Compiler

The Meta Converter Table Compiler (MCTC) compiles data tables and generates automaton bodies that are loaded and used by Trans-Unit Converter. The data files has three categories, 1) Relation Tables describing *GCS Tables* and *CCS Tables* to be loaded by the compiler, 2) GCS Tables and CCS Tables describing informations specifying each element of WC, and 3) *TMC Tables* describing conversion information from/to WC.

In Relation Tables, locale, multi-locale and global informations are specified. For one GCS, by different extension rules, multiple GCS Tables can be defined.

The compiler was implemented as independent software. The compiler reads the Relation Tables, then it reads GCS/CCS Tables and TMC Tables. After reading all files, the compiler generates encoding scheme automata, Trans-code-set automata, trans-unit automa and TMC automata. The structure of automaton is pointers to predefined optimal functions with calling parameters for minimum storage size and for portability of source codes. Each automaton can be selected by only changing pointers to determine the automaton. Thus, after compilation, all automata can be invoked without locale and GCS table dependencies. Those automata are loaded when the initializer in our library is executed.

4.2. The Trans-Unit Converter

The Trans-Unit Converter consists of mb/WC Converter, WC/TMC Converter, mb/IWC Converter and WC/IWC Converter. The mb/IWC and the WC/IWC Converters are used in the OM.

The mb/WC Converter processes mb through the following steps, 1) Control Functions and Determinative processing, 2) Direction determination and 3) Position determination. At the third step, informations of wctypes functions are stored in a bit field in WC for compatibility to POSIX – note that those functions do not always work correctly. Reverse-conversion executes through 3 to 1. Since the Control Functions and Determinative processing use a set of functions with rules associating GCS/CCS and WC retains an ID of the function, it is possible to reverse-convert by a pair of functions and its reverse-function with the same function ID. Thus, by the function pairs, fixed length data type can be used for WC.

Conversion from WC to TMC is done by WC/TMC Converter through the following steps, 1) generating TMC Character-ID field from GCS/CCS-ID, Font ID, Glyph Index of the WC codepoints and rules, 2) generating TMC Attribute field from rules in the converter and 3) generating the rest of Character-ID field that retains informations to convert TMC to WC. Reverse conversion to WC can be done by using of the informations to convert TMC to WC.

Since all conversion paths were determined clearly, the best algorithm optimized to speed was implemented with minimum number of condition statements that break pipe-line stage of CPUs. The algorithm, the implementation and the architecture are described in detail in another paper (in preparation).

4.3. The Character Information Functions

Character Information Functions query and change informations in the Attribute field informations in one element of TMC. Each attribute in the field is saved as one bit. In order to generalize the functions, each attribute name in the field is set in a TMC Table and the functions retain the name and its bit location in the field by data generated by the MCTC. Thus, the functions are called with the attribute name(s) and TMC-ID. The essential functions[4] are as follows, 1) QueryTMCAttribute, 2) QueryTMCInformation, 3) ChangeTMCInformation, 4) QueryTMCCtrlCName. Finding line separation can be done by checking an attribute field held for the purpose.

4.4. The Text Manipulation Functions

The TMC Text Manipulation functions are also called with TMC-ID. Since each TMC is independent from another TMC, it is impossible to mix different TMCs by the functions.

The basic TMC string functions[4] are the following two, 1) TMCStringLength and 2) TMCStringCopy. A TMC string concatenation function can be derived from those above.

The essential functions for TMC text manipulation[4] are as follows, 1) InsertTMCString, 2) DeleteTMCString, 3) CompareTMCString, 4) ExtCompareTMCString, 5) SearchTMCString, 6) ExtSearchTMCString, 7) ReplaceTMCString and 8) ExtReplaceTMCString. Those functions above adjust Position Dependency attribution fields.

4.5. The Inter-Process Communication Assisting Functions

The functions are used for the Global IOTMC model to inform GCS/CCS and association between designation sequences and extension rule sets. And to re-associate designation sequences and extension rule sets, function ChangeGCSExt is used. This function is essential for IS 13194:1991, because ISO does not supply such re-association. Since WC codepoints in a rule set is unique when the rule is changed, it is possible to mix all scripts derived from *Brahmi* simultaneously.

5. Summary

Since the Multilingual I/O and TM/C System we developed is based on the Meta Converter System, the system can cover all scripts in all code sets consistently with the best performance and the largest flexibility. And the Meta Converter System loads all compiled tables, thus, the system can change Drawing/Processing rule sets for any purpose without any limitation beyond ISO specifications.

References

[1] Kataoka, Y. et al., 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, September 1994, 4S-3 (In this volume).

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

[3] ISO/IEC 6429: 1988, Information processing – Control functions for 7-bit and 8-bit coded character sets.

[4] Kataoka, T. et al., 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, September 1994, 4S-5 (In this volume).

[5] Uezono, K. et al., 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, September 1994, 4S-4 (In this volume).