招請論文

論文79-16-3-1

Korean Character Processing: Part I. Theoretical Foundation (한글문자의 컴퓨터 처리 : I. 이론)

鄭 元 良*
(Chung, Won Lyang)

要 約

2부에 걸친 한글의 컴퓨터 처리에 관한 논문으로서, 제1부에서는 한글문자의 컴퓨터 처리의 제문제점을 확인하고 이를 위한 해결책으로 이론적 기반의 확립을 제안 하였다. 한글문자의 일차원적 문법과 이차원적 구조를 각각 BNF와 " patternal structure "를 사용하여 정의하고 이로부터 모아쓰기를 위한 lexical analysis 및 parsing algorithm을 이론적으로 토론 하였다. 모아쓰기 algorithm은 한글의 입력과 출력 모두에 응용이 가능하며, device-independence를 위해 "cardinal symbol set"의 개념을 소개하였다. 제2부에서는 한글 컴퓨터 처리의 역사적 개론과 상기 algorithm의 implementation 문제점들을 토론하고자 한다.

Abstract

This is Part I of a two-part article on Korean character processing by a computer. In part I, the problems in Korean character processing are identified and the theoretical foundation is laid out as a viable solution to them. The one-and two-dimensional syntactic structures of Korean characters are formally defined by means of BNF and " Patternal structure " respectively. Formal discussion of lexical and syntactic algorithms is given for character conversion. This character conversion algorithm is applicable to both input and output. For device-independence and implementation-independence, the concept of "cardinal symbol set" is introduced. We will present a historical survey of Korean character processing and discussion of implementation problems for the above algorithm in Part **I**.

1. Introduction

The history of efforts in computerizing the Korean character processing is relatively long considering the short history of computing in Korea. These efforts have been directed to two main areas : (i) discovery of Korean character processing algorithms and underlying theory, and (ii) mass production of input/output devices capable of handling Korean characters.

Naturally, any research and development project would try to be successful in both areas (i) and (ii). In spite of the many attempts, the reason for the long delay of the introduction of commercially available Korean input/output terminals is largely due to the status of computing in Korea during the last decade. However, we have to admit that the scientists and engineers involved in the field are also responsible for the present status of Korean processing by computer.

A number of key problems in Korean character processing can be identified: (i) approaches to the algorithms for Korean I/O are more or less adhoc, (ii) these algorithms are too much device-dependent, and (iii) the lack of standa-

* 正會員, 韓國科學技術研究所
(Korea Institute of Science and Technology)
接受日字 : 1979年 4月 23日

rds in keyboard set or coding scheme causes difficulties in communication and interfacing. High cost of terminals currently available is a serious obstacle to meeting the upsurging demands for Korean I/O terminals.

This article is a sincere attempt to provide a meaningful solution to the above problems (i) and (ii), with a high hope that this will lay a ground stone for future endeavors in the area (iii). It is believed to be a theoretical underpinning of Korean character processing, especially in relation to input/output and the linear to two-dimensional conversion of Korean characters. The syntactic structures of both linear and two-dimensional character representations have been analyzed(in Section 2). The theory of "patternal structure" is presented for the latter. The conversion algorithm is formally discussed(in Section 3) and the notion of the "cardinal symbol set" is introduced to preserve the device-independence. The theoretical background for the discussion is provided by the formal language and language translation theory of computer science.

A final remark should be made regarding the discovery of the algorithm and theory of this article. The discovery occurred independently without the knowledge of other works in late October of 1978. Considering the simplicity (and the accompanying beauty) of Korean character syntax, many independent works should have been carried out without being known. Thus, the claim is not a big one but being made in good faith for the full disclosure of a scholarly work.

2. Structure of Korean Characters

A. Formal Syntax of Korean Characters.

As any other language, the Korean language is generated by an alphabet(generating set of symbols). The "Korean alphabet" is divided into two distinct classes;"consonants"and"vowels", according to the phonetic role of a symbol.

There are 30 consonants and 21 vowels being used in modern Korean. "Consonants" are further broken down to three groups: simple (there are 14 of them), double (5), and compound (11) consonants. Similarly, vowels are grouped into two classes: simple (10) and compound (11). The alphabet symbols are classified and listed in table 1.

A proper combination of these 51 symbols produces a Korean "character" which is a unit of pronunciation and meaning, ie., syllabus. Formation of a character is governed by a fixed set of rules, the subset of the syntax of Korean.

Each character must begin with a consonant, called "initial consonant or sound", which is followed by a vowel called "middle sound". Optional, third component of a character is a consonant which is called "terminal consonant or sound". Simple and double consonants can serve as both initial and terminal sounds.

Table 1.

Symbol class		alphabet symbols
Consoant	Simple(14)	ㄱ,ㄴ,ㄷ,ㄹ,ㅁ,ㅂ,ㅅ,ㅇ, ㅈ,ㅊ,ㅋ,ㅌ,ㅍ,ㅎ
	Double(5)	ㄲ, ㄸ, ㅃ, ㅆ, ㅉ
	Compound (11)	ㄴㅈ,ㄴㅎ,ㄹㄱ,ㄹㅁ,ㄹㅂ, ㄹㅎ,ㅂㅅ,ㄱㅅ,ㄹㅅ,ㄹㅌ,ㅁㄱ
Vowel	Simple(10)	ㅏ, ㅓ, ㅗ, ㅜ, ㅡ, ㅣ, ㅑ, ㅕ, ㅛ, ㅠ
	Compound (11)	ㅐ, ㅔ, ㅒ, ㅖ, ㅚ, ㅟ, ㅢ, ㅘ, ㅙ, ㅝ, ㅞ

Compound consonants can be a terminal sound only. The formal syntax of Korean characters is described by means of BNF notations in Figure 1.

There are two ways that Korean(sentence) can be written down. One way is to spell one symbol after another except for the necessary blanks or sentential marks (commas, colons, for example). That is, each position of a

sentence (or word) is filled by a symbol, and the sentence is a linear representation. The other way is to arrange two or three symbols making up a character in such a way that a character becomes a two-dimensional pattern and occupies a single sentential position. Of course, this two-dimensional configuration follows fixed syntactic rules. For historical reasons (perhaps, the Chinese influence), the second convention is adopted in normal use. However, the abstract syntax and semantics remain the same without regard to the way characters are recorded.

"BNF Specification of Korean Character Syntax"

⟨Korean character⟩ ::= ⟨initial sound⟩ ⟨middle sound⟩ [⟨terminal sound⟩]

⟨initial sound⟩ ::= ⟨initial consonant⟩
::= ⟨simple consonant⟩|
⟨double consonant⟩

⟨terminal sound⟩ ::= ⟨terminal consonant⟩
::= ⟨simple consonant⟩|
⟨double consonant⟩|
⟨compound consonant⟩

⟨middle sound⟩ ::= ⟨vowel⟩

⟨vowel⟩ ::= ⟨simple vowel⟩
⟨compound vowel⟩

⟨simple consonant⟩ ::= ㄱ|ㄴ|ㄷ|ㄹ|ㅁ|ㅂ|ㅅ|ㅇ|ㅈ|ㅊ|
ㅋ|ㅌ|ㅍ|ㅎ|

⟨double consonant⟩ ::= ㄲ|ㄸ|ㅃ|ㅆ|ㅉ

⟨compound consonant⟩ ::= ㄳ|ㄵ|ㄶ|ㄺ|ㄻ|ㄼ|ㄽ|
ㅄ|ㄺ|ㄻ|ㄾ|ㄿ|

⟨simple vowel⟩ ::= ㅏ|ㅓ|ㅗ|ㅜ|ㅡ|ㅣ|ㅑ|ㅕ|ㅛ|ㅠ

⟨compound vowel⟩ ::= ㅐ|ㅒ|ㅔ|ㅖ|ㅘ|ㅙ|ㅚ|ㅝ|ㅞ|ㅟ|ㅢ|

Fig 1.

Due to the idiosyncracies of two-dimensional form of Korean characters, more than one shapes of symbols are required to write proper-looking characters. What we mean by "proper-looking" is not determined by scientific laws but by the conventions and aesthetic considerations of written Korean. Nevertheless, we may find a set of rules of formation for two-dimensionally written Korean characters. We shall give it a name, "patternal structure" of Korean characters.

B. Patternal Structure of Korean Characters.

For systematic discussion of the "Patternal structure", a couple of terms will be introduced. The two-dimensional space occupied by a sentential position is defined as a "slot" or "window". A slot or window is simply an upright rectangular (or square) area whose size is fixed by a writing style of a person or printing method (i.e., font). Within each window, initial and middle, or terminal sounds occupy more or less predetermined positions. These positions are also determined by the kind of a symbol and the presence / absence of a terminal consonant. What is interesting is the fact that these positions, i.e., patternal structure, lead to two-dimensional syntactic rules - as described in the following.

The vowels are subject to the following positional (or patternal) classification:

(i) vertical vowels (V_1): ㅏ, ㅓ, ㅣ, ㅑ, ㅕ, ㅐ, ㅒ, ㅔ, ㅖ

(ii) horizontal vowels (V_2): ㅗ, ㅜ, ㅛ, ㅠ, ㅡ

(iii) mixed shape vowels (V_3): ㅘ, ㅙ, ㅚ, ㅝ, ㅞ, ㅟ, ㅢ

To each of the above classes V_1, V_2 and V_3 corresponds two shapes - "short" and "long". This is based on whether there exists a terminal consonant or not. The positions taken up by these six types of vowels are shown as shaded patterns in windows by Fig 2 (a) and (b) show the combination of short and long types.

The consonants are classified by their shapes and relative positions within a window.

(i) type-1consonants (C_1): initial consonants when followed by long vertical vowels.

(ii) type-2consonants (C_2): initial consonants when followed by long horizontal vowels.

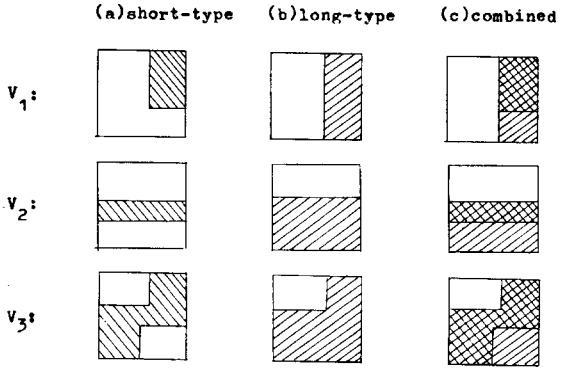

Fig. 2. Elementary patterns for vowels.

(iii) type-3 consonants(C_3): initial consonants followed by mixed-shape vowels.

(iv) type-4 consonants(C_4): terminal consonants preceded by short vertical or mixed-shape vowels.

(v) type-5 consonants(C_5): terminal consonants preceded by short vertical or mixed-shape vowels.

Fig. 3 shows the window patterns for these consonant type. Shaded portions represent the areas occupied by consonants. We will call the above window patterns for vowels and consonants "elementary patterns."

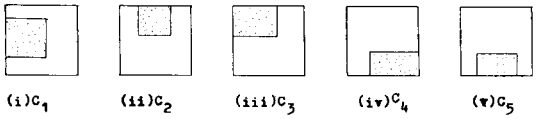

Fig. 3. Elementary patterns for consonants.

We now recall that there are only a finite number of ways that consonants and vowels are combined to comprise a meaningful character. This suggests that patternal structure of Korean characters should be described in terms of finite combination of elementary patterns. And this is exactly the case. The rules and codes for the patternal structure ("two-dimensional syntax" if you like) are succinctly summarized in table 2. As we may see, there are six combined patterns for legal Korean characters. These patterns are shown in fig. 4 together with examples of characters corresponding to each pattern.

Table 2.

Consonant Types \ Vowel Types		Long-type			Short-type		
		V_1	V_2	V_3	V_1	V_2	V_3
Long-type	C_1	C_1V_1	-	-	-	-	-
Short-type	C_2	-	C_2V_2	-	-	$C_2V_2C_5$	-
	C_3	-	-	C_3V_3	$C_3V_1C_4$	-	$C_3V_3C_4$
	C_4	-	-	-	-	-	-
	C_5	-	-	-	-	-	-

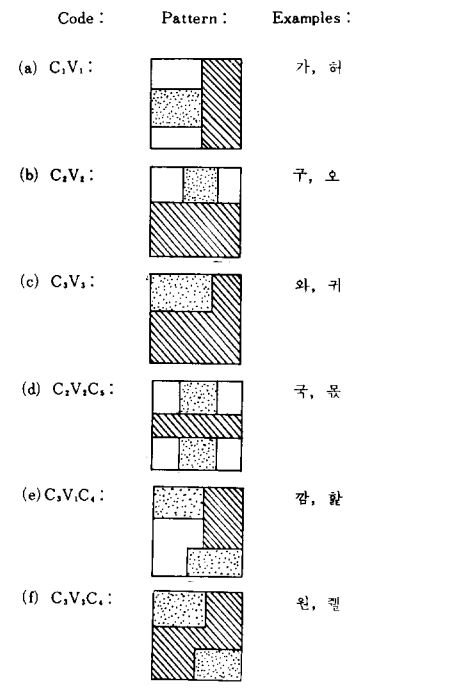

Fig. 4. Patterns for Korean characters.

3. Korean Character Combination/ Conversion Algorithm

We have seen that the differences between the linear and two-dimensional character representations are rather superficial. By this we mean that the basic (abstract) syntax of two representations is identical as described by fig. 1.

This is so even though different concrete syntax based on patternal structure can be derived for two-dimensional character representation.

However, there is an occasion that two character representations make important pragmatic differences. This occurrs for "computer processing of Korean Characters" whether it be in word processing applications or programming system. There arise needs for distinguishing between two representations and for converting from one representation to the other. In particular, the conversion from linear to 2-D representation is a well-known problem. Its fame is due to the significance of the conversion in the aforementioned computer applications. Specifically, both input and output of Korean involve this conversion process.

For the purpose of computer processing, each symbol becomes the unit of internal representation and input while a character serves as a unit of meaning and 2-D output.
For theoretical discussion of the conversion algorithm we need to have a framework which is independent of device-specific features of input/output and internal representations(i.e., coding method). We will start with a few definitions and notations.

DEF. Input to the conversion algorithm is the linear representation of Korean characters which is nothing but a linear string, s, of Korean alphabet symbols and separator symbols.

$s \equiv$ " $a_1 a_2 \cdots\cdots a_n$ "
Where
$a \in A \cup S$ for $i = 1, 2, \cdots\cdots, n$;
$A \equiv$ { ㄱ, ㄴ, …… ㅎ, ㅏ, …… ㅙ } : Korean alphabet set;
$S \equiv$ { . , .. … 1, …, 0, … BLANK, A, … , Z } : Separator set.

DEF. A "Chunk" is defined as a consecutive stream of alphabet symbols without separator symbols.
For example, " ㅎ ㅏ ㄴ ㄱ ㅜ ㄱ " forms a chunk and the the string " ㅈ ㅓ ㄴ ㅅ ㅏ ㄴ ㄱ ㅐ ㅂ ㅏ ㄹ " consists of two chunks.

DEF. Output from the conversion algorithm is the 2-D representation of Korean characters with patternal structure identified.

DEF. The "cardinal symbol set" $\mathcal{C}$ is defined as a minimal subset of A which is necessary for unambiguous linear representation of Korean characters. Explicitly the cardinal set consists of 12 vowels and 19 consonants, i.e.,
$\mathcal{C} \equiv$ { ㄱ, ㄴ, ㄷ, …… ㅋ, ㅌ, ㅍ, ㅎ, ㄲ, ㄸ, ㅃ, ㅆ, ㅉ, ㅏ, ㅓ, ㅣ, ㅑ, ㅕ, ㅡ, ㅜ, ㅛ, ㅠ, ㅢ, ㅐ, ㅔ } .

The remaining symbols, i.e., combined vowels and complex consonants, can be coded as two consecutive symbols from $\mathcal{C}$.
Two practical implications of the cardinal symbol set are noteworthy: first, $\mathcal{C}$ is the minimal symbol set for unambiguous input and output of Korean; second, $\mathcal{C}$ forms the basis of device-independent discussion of the conversion process. Thus, if an input keyboard contains keys for symbols not in $\mathcal{C}$ they are just for convenience. The validity of the above arguments is not difficult to see. We can show the correctness by counterexample: Without inclusion of double consonants, the string " 아 ㄱ ㄱ 다 " becomes ambiguous because it may be either "악기다" or "아끼다". The complex consonants are not cardinal because they are always terminal symbols and thus unambiguously indentified. The vowels " ㅒ " and " ㅖ " should be included because of "ㅙ" and "ㅞ" but, " ㅒ " and " ㅖ " are excluded because they stand by themselves and can be coded as " ㅑ ㅣ " and " ㅕ ㅣ " respectively. If three-position coding of vowels is allowed, then " ㅒ " and " ㅖ " can also be deleted from $\mathcal{C}$. However, the restriction of two-position coding and their inclusion will be beneficial and contribute much to the simplicity of the theory.

With the above preliminary we are now ready to discuss the conversion algorithm. The algorithm consists of two parts:

(1) Lexical analysis : identification of chunks and characters from the linear input string.

(2) Syntactic analysis : determination of symbol combination, i.e., patternal structure, of a 2-D character representation.

For implementation purposes, it is not essential that two parts are organized as separate but connected modules(e.g., coroutines). They may be integrated into a single program module.

A. Lexical Analysis

The most fundamental to lexical analysis is the property of the syntax of Korean that each character must begin with a (simple or double) consonant followed by one or two cardinal vowels.

Lexical analysis algorithm is best described by a finite automaton. It seems most convenient to get two input symbols at a time, which also results in the simplest automaton.

The automaton is described in fig. 5.

The START state is entered for every character and, for all output states, it is necessary to reset all pointers being used with an input string.

All "error states" are suppressed but can easily be added to the figure. Of course, any input symbol $a_i \in \mathcal{C}$.

We note that the basis of the automaton is in fig. 2-1.

Three nonterminal states S_1, S_2, and S_3 admit interesting interpretations:

S_1 means that initial and middle sounds have been identified (for noncardinal vowels, only the first symbol): S_2 means that a compound vowel has been detected: S_3 means hat a cardinal vowel is followed by a terminal consonant.

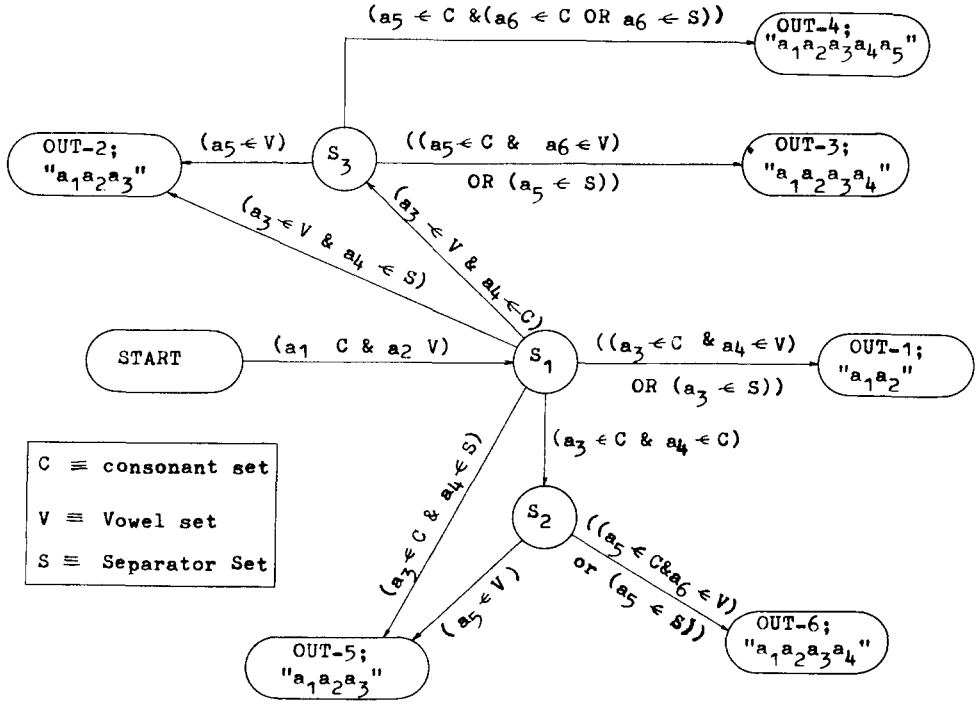

Fig. 5. Automaton for lexical analysis.

B. Syntactic Analysis (Parsing)

Syntactic analysis for Korean characters is extremely simple, to the degree of triviality. Once lexical analysis is done, the vowel (i.e., middle sound) of each character determines the patte rnal structure uniquely. The theoretical basis of this comes from table 2:
each column contains only a single entry. Discrimination between long and short vowel types is trivially done by the absence or presence of terminal consonant. The fact that choice of a column (from the type of a vowel) fixes the types of consonant (s) is useful for syntax error detection. That is, if the input consonant (s) type does not match that of a selected column, it signals the occurrence of a syntax error. From the viewpoint of table 2, syntactic analysis may be thought as the process of determining the patternal structure code, $c_i\ v_j$ or $c_i\ v_j\ c_k$, where $j \in \{1,2,3\}$ and $i, k \in \{1,2,3,4,5\}$

As a supplementary remark we add that proper encoding of alphabet symbols should be seriously considered for efficient lexical and syntactic analysis.

Fig. 6 describes the syntactic analysis algorithm, "SYNA", by means of a pyramid-like symmetric flowchart. Labels in EXIT nodes represent the patternal structure names as in Fig 4.
(We note that the flowchart of fig. 6 is obtained by straight for ward translation of table 2.

4. Conclusion

In this article, we discussed (i) the theory of Korean character syntax and patternal structure (for two-dimensional composition), and (ii) that of lexical and syntactic analysis algorithms for Korean character conversion. We were able to observe

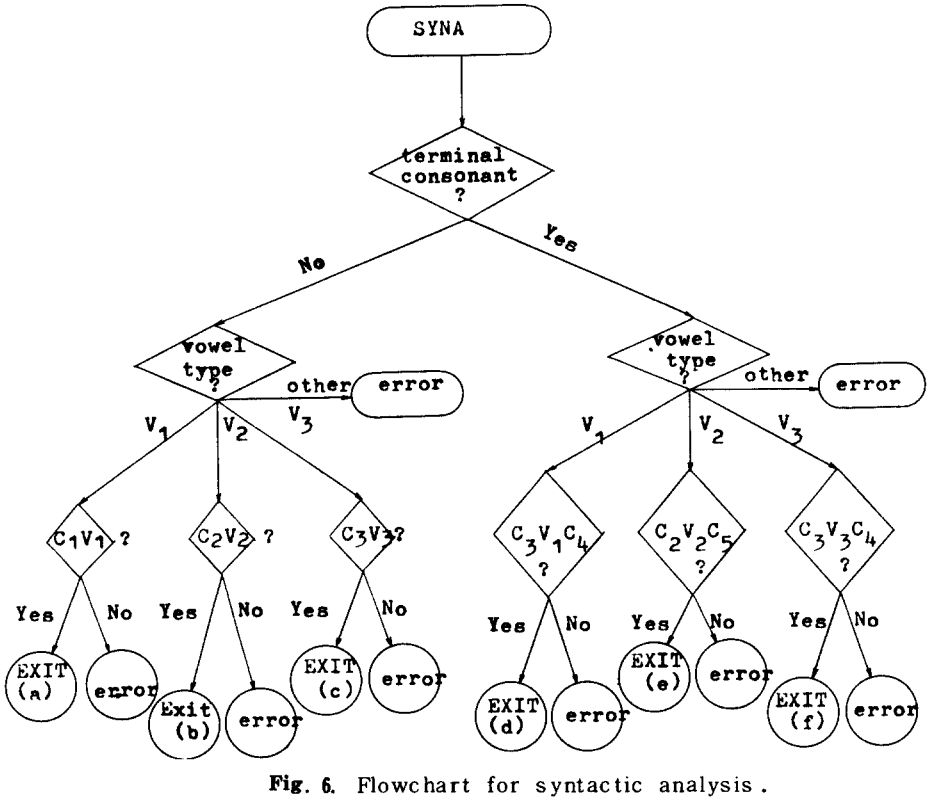

Fig. 6. Flowchart for syntactic analysis.

the utmost simplicity and beauty of Korean characters as the object of formal language and computer processing.

The discourse has been presented in a manner independent of device or implementation specifics. Thus any modification of the discussed algorithms will be straightforward.

We suggest the following applications or extension of the present theory :

(1) application of the patternal structure to on-line character recognition (handwritten or printed).

(2) extension of the conversion algorithm to the lexical and syntactic analyses for a Korean (programming) language processor, and

(3) incoperation of formal and computer processing properties in determining the korean standards for keyboard-set and coding scheme of korean character processing.

Some of these problems will be discussed in the sequel.