영어교육 27호
1984년 2월

The SNOBOL 4 Programming Language for TESOL or TEFL

Myung-Un Lee
(Kongju National Teachers' College)

What is a programming language?

As this terminology is quite new, most students feel confused at the beginning.

> A programming language is a language in which one can also write declarations and commands. We are concerned in this primer with the language SNOBOL, which stands for String-Oriented Symbolic Language, a language in which one can write statements to be read and responded to by a computer. (Allen Forte, 1967 : 1)

The SNOBOL language was designed by Mr. Ralph E. Griswold, Mr. Polonsky, and Mr. Farber, of Bell Telephone Laboratories in 1971. Version 3 of SNOBOL 4 is the reference version for comparison and MACRO SPI TBOL is an implementation of the SNOBOL 4 computer language which was coded during 1974/75. The SNOBOL 4 language has many important features extracted from many languages and was devised for manipulating natural languages. In short, it is a kind of string processing (manipulation) of language. The characters used in the programming languages are generally fewer than those used in natural languages.

In SNOBOL there are exactly 48 admissible characters, 26 alphabetics (upper case only), 10 numerals (0 1 2 3—9), and 12 miscellaneous.

In programming, a distinction is made between data and the program, which is a set of instructions that specifies the processing. A computer program is general and remains constant, but the data may change. The term

string, when used in connection with SNOBOL, means a sequence of characters, or string of characters.

Here is the brief synopsis of SNOBOL 4 provided by Dr. James L. Wyatt at the Florida State University in the course of "computational linguistics" during the spring quarter, 1981.

(a) STATEMENT FORMAT (any element may be optional; all blanks obligatory)

label stringname pattern=replacement goto

col. 1	stringname(s) : (label)
(optional)	and/or : S(label)
	characters : F(label)
	in quotes : S(label)
	F(label)
	: ($ stringname)

(b) INPUT-OUTPUT

Stringname=INPUT causes data card to be read and given stringname.

Stringname=TRIM (INPUT) same as above but trims trailing blanks.

OUTPUT=stringname(s) and/or characters in quotes. Above causes line to be printed.

(c) PATTERN MATCHING

(label) stringname pattern

character strings within quotes

stringname(s), alternate patterns separated by 1(=)

ANY ('list of characters') matches any single character.

NOTANY ('list of characters') matches if none found.

SPAN ('list of characters') matches a run.

ARB stands for any remaining characters

not at end.

REM stands for any remaining characters at end.

BREAK ('list of characters') up to but not including the "break" character.

LEN (integer) measures off indicated number of characters.

SIZE (stringname) counts number of characters.

$ Stringname $ means "which has as its name whatever is contained in:"

Here, ANY, NOT ANY, SPAN, ARB, REM, BREAK, SIZE, are all cover symbols for pattern matching.

(d) **CONDITIONS** (often used to determine program flow of input-output)

. INDENT (character string, character string)
1 2

DIFFER (character string, character string)
1 2

Character strings are identified by stringname or are characters in quotes.

EQ (integer, integer)
1 2

NE (integer, integer)
1 2

GT (integer, integer)
1 2

LT (integer, integer)
1 2

GE (integer, integer)
1 2

LE (integer, integer)
1 2

integers are identified by stringname or actual integer.

The two IDENT, DIFFER conditions are only for strings, and the other EQ (=equal), NE (=not equal), GT (=greater than), LT (=less than), GE (=greater or equal), LE (=less or equal) are the numerical conditions.

(e) **ARITHMETIC OPERATORS** (obligatory blank on each side)

† addition
- subtraction
* multiplication
/ division

This SNOBOL programming language is very hard, confusing for first-timer to understand, but through some examples of problems or statements, everyone can understand it and write a computer program finally— a series of statements in the SNOBOL language which describes a procedure— without involving himself in the complexities of machine operations.

Ex. 1) Write a statement to get your name into the computer and call that string of characters MYNAME.

MYNAME= ' MYUNG LEE '

This is a statement, and MYNAME is the name of the string. My name, MYUNG LEE, is read (input) into the computer. Notice the blanks before and after the equal mark and before the name of the string.

Ex. 2) The following programming statement will cause one punched card (or its "image" stored on magnetic tape, disc or drum) to be read into the computer and for the programmer's convenience to have the name TEXT:

TEXT=INPUT

If a punched card had on it *NOW IS THE TIME.* and had been read by the statement above, the character string named TEXT would contain *NOW IS THE TIME.* plus 64 blanks, since computer cards have 80 columns for alphabetic and/ or numeric characters and any unused columns are treated as blanks by statements of type above.

Were the character string named TEXT to be printed, the blanks would be invisible and would cause no trouble, but if that string were to be fo llowed by another in computer output, would be separated by 64 blanks, unless something were done to suppress the blanks.

A string solution to this inconvenience is to trim off the trailing blanks at the time the card or its image is read into the computer. The following

statement does this:

TEXT=TRIM (INPUT)

EX. 3) Write a statement to determine whether a string named LINE contains *SLIPPER, SLIMY, SLICK,* and if so, name the word found SLADJ. To solve this problem, we had better use the naming period, or immediate value assignment ".".

LINE (' SLIPPER' | ' SLIMY' | ' SLICK')
. SLADJ

(' SLIPPER' | ' SLIMY' | ' SLICK') is a pattern matching of the string named LINE and because the naming period can name any of three words, parentheses should be used in this problem.

EX. 4) Write a statement to change the contents of the string named SENTENCE from *WHERE ARE YOU GOING TO?* to *WHERE ARE YOU GOING?*

SENTENCE ' TO'=NULL

Here, SENTENCE is the name of string, and "=" mark means deletion. 'TO' is a pattern matching, the contents of string named SENTENCE.

EX. 5) Write a statement to determine whether there is a period followed by a blank in the sequence FIRSTNAMES, and if so, make a transfer by means of a GOTO to a program section labelled INITIAL.

FIRSTNAMES '. ':S (INITIAL)

Here, ' ' means toquote (= ↑ ↑), and: is GOTO. S means 'if succeeds,' or 'if so,'.

So, above statement can be explained like this: "Quote one period followed by a blank, the contents of character string named FIRSTNAMES, in to the computer, and if so(succeeds), go to the label named INITIAL."

And then, simple, easy programs can be written with no difficulties, which are nothing but a series of statements already learned.

EX. 1) Write a program to print all data cards in a deck.

DATA CARD=INPUT: F(END)
OUTPUT=CARD: (DATA)
END

Both DATA and END are the names of labels.

Above program can be explained as follows: "Read all the data cards in a deck, and if fails, go to the label named END, but if succeeds, print (OUTPUT) all the cards and go to the label of DATA. If all the cards are printed, then go to the label named END, and the program ends."

EX. 2) Write a program to print only those data cards containing any of the following punctuation: . ; : , ? !

```
PUNCTUATION='. ; : , ? !' (This is outside of loop.)
DATA CARD=INPUT: F(END)
CARD PUNCTUATION: F(DATA)
OUTPUT=CARD: (DATA)
END
```

Using the SNOBOL language features, various analysis, contrast or comparison of learning problems, data and materials are available. Some students can compare the cognate words of English and German or work out the literary specific technique of a writer, and contrast and compare the structural devices of L-1 and L-2. Of course this kind of contrast or analysis is a somewhat complicated program, or project. However, the easiest way of being able to use this kind of technically advanced project lies in understanding the fundamental statements of SNOBOL language because a computer project is nothing but a series of statements of instruction.

> Each SNOBOL statement is punched in a single card, to a maximum length of 72 columns. The deck containing data follows immediately after the program card that carries the statement label END. The program is read in automatically according to instructions on control cards that should be obtained from your installation. (Forte, 1967: 72)

Using the SNOBOL 4 features, we can generate English copulative sentences with 'be verb' and transform them into interrogative and negative forms. First, make the data cards of copulative sentences. For example, suppose one of the data cards of copulative sentences has *I/am/a/student.* or *You/are/a/teacher.*

Each card uses/to delimit subject, verb, and complement, and has only one sentence on it. Second, provide the project (program) mechanics.

```
1. READ SENTENCE=TRIM (INPUT): F(END)
2. SENTENCE BREAK ('/'). SUBJ LEN (1)
        † BREAK ('/')
        †. VERB LEN (1) BREAK ('/'). DET LEN (1)
        † BREAK ('.'). COMP '.'=SUBJ VERB DET COMP '.'
3. PRINT 1 OUTPUT=SENTENCE
4. SENTENCE SUBJ VERB DET COMP.=VERB SUBJ DET COMP '?'
5. PRINT 2 OUTPUT=SENTENCE
6. SENTENCE VERB SUBJ DET COMP?=SUBJ VERB 'not' DET
        †COMP '.'
7. PRINT 3 OUTPUT=SENTENCE: (READ)
8. END
```

Translation:

1. Read (input) the string named SENTENCE, and trim off any trailing blanks if any (TRIM), if if fails (F) goto (:) the label named END.

2. Make the contents of the string named SENTENCE(pattern matching), read up to/and name it SUBJ, measure off one and read up to/, name it VERB and measure off one, read up to /, name it DET, measure off one, and then read up to. and name it COMP, input.,and change the contents of the string named SENTENCE into SUBJ VERB DET and. (equal mark means replacement, and † means concatenation.)

3. Print the sentence (generate copulative sentence).

4. Change the contents of the string named SENTENCE from SUBJ VE RB DET COMP and. to VERB SUBJ DET COMP and quote?

5. Print the sentence (transform into interrogative form).

6. Change the contents of the string named SENTENCE from VERB SUBJ DET COMP and ? into SUBJ VERB, quote (input) not and DET COMP and quote.

7. Print the sentence (from interrogative to negative form) and goto (no blank) the label of READ (1).

8. END

Likewise, we can generate Korean copulative sentence and then transform them into interrogative and negative forms.

By contrasting both project mechanice of English and Korean, Korean students will understand the differences of structural devices between English and Korean.

The following is MACRO SPITBOL implementation of SNOBOL project which generates Korean copulative sentences and replaces them with interrogative and negative forms. In Korean, 'ipnida' is 'be verb' in English, and it becomes 'ipnika?' in interrogative sentences and '*an*ipnida' in negative forms.

A scientific study of the differences/similarities in the system of any two languages, which is known as 'contrastive studies', comes under contrastive linguistics.

```
1 READ SENTENCE=TRIM (INPUP): F(END)
2 SENTENCE BREAK('/'). SUBJ LEN(1) BREAK('/'). COMP LEN(1) +REM
  VERB=SUBJ COMP VERB
3 PRINT OUTPUT=SENTENCE
4 SENTENCE ARB. FIRST 'IPNIDA.'=FIRST 'IPNIKA?'
5 PRINT 2 OUTPUT=SENTENCE
6 SENTENCE ARB. FIRST 'IPNIKA?'=FIRST 'ANIPNIDA.'
7 PRINT 3 OUTPUT=SENTENCE: (READ)
8 END
STORE USED          1246
STORE LEFT          3752
COMP ERRORS            0
REGENERATIONS          0
COMP TIME-MSEC        85
KEYOJANEUN KANHOWON IPNIDA.
KEYOJANEUN KANHOWON IPNIKA?
KEYOJANEUN KANHOWON ANIPNIDA.
KEJOJANEUN SENSAENG IPNIDA.
KEJOJANEUN SENSAENG IPNIKA?
KEJOJANEUN SENSAENG ANIPNIDA.
KEJOJANEUN HAKSAENG IPNIDA.
KEJOJANEUN HAKSAENG IPNIKA?
KEJOJANEUN HAKSAENG ANIPNIDA.
```

KENEUN WUNJENSU IPNIDA.
KENEUN WUNJENSU IPNIKA?
KENEUN WUNJENSU ANIPNIDA.
TANGSINNEUN HANKOOKIN IPNIDA.
TANGSINNEUN HANKOOKIN IPNIKA?
TANGSINNEUN HANKOOKIN ANIPNIDA.
KENEUN KYOSU IPNIDA.
KENEUN KYOSU IPNIKA?
KENEUN KYOSU ANIPNIDA.
TANGSINNEUN KYOWHANSU IPNIDA.
TANGSINNEUN KYOWHANSU IPNIKA?
TANGSINNEUN KYOWHANSU ANIPNIDA.
NANEUN HAKSAENG IPNIDA.
NANEUN HAKSAENG IPNIKA?
NANEUN HAKSAENG ANIPNIDA.
NANEUN EUISA IPNIDA.
NANEUN EUISA IPNIKA?
NANEUN EUISA ANIPNIDA.
KEKOSEUN NAMU IPNIDA.
KEKOSEUN NAMU IPNIKA?
KEKOSEUN NAMU ANIPNIDA.

Such a study can predict the areas of difficulty a learner will face while learning a second language. Once the findings of the contrastive study are available, one could prepare the instructional materials in the second language either in the traditional manner for use in the regular classrooms or for use in intensive courses, through language laboratories and through other A-V aids. Both these approaches, however, can hardly satisfy the impending problems of the population explosion and the explosion in education. So, under the current situation the techniques of 'programmed learning' have come to us providentially.

Despite the present simplicity of these techniniques, their potentiality is enormous. There is hardly any doubt that parts of the learning process can be automated and that the programmed textbooks are here to say— wide-spread use of programmed material would result in a change in the role of teachers and the programmes

would serve the teacher as a springboard for creative thinkng. (Robert C. Lugton, 1970: 143)

Here are some types of problems for writing a program.

1. Two languages have vowel systems represented by the characters A, E, O, U, and they have consonant systems represented by the characters P, T, K, B, D, G, F, V, S, Z, M, N, L, R.

A linguist has a corpus of each language on punched cards and wants to find the percentage of consonants and vowels in each corpus. The words in each corpus are separated on punched cards by one or more blanks.

Write a program to assist your linguist colleague.

2. A committee which chooses textbooks for very young children wants to select the book which has the shortest words and shortest sentences. The committee has three textbooks on punched cards, and each card has one and only one sentence on it. There are no punctuation marks within the sentence, and all sentences end with period.

Write a program to aid the committee in making its decision.

3. A literary critic is interested in the surface structure (that part of language which is heard or read) of the poetry of a movement called "modernist". Modernists used a vocabulary known for a high frequency of exotic words, especially those referring to the Orient. Also colors were prominent, especially blue.

Write a program to help the literary critic decide whether a large corpus of poetry by an unknown writer might be "modernist". Set some arbitrary "threshholds" to make the determination.

4. A government plans to censor all correspondence containing reference to sensitive information in the areas of diplomacy, military operations, and geography. Lists of "sensitive" words are available for each of the areas to suffer censorship. All mail is to be put on punched cards and to be processed by a computer program to delete the sensitive reference and replace each reference with one of these three messages:

*** unauthorized diplomatic reference***

unauthorized military reference

unauthorized geographic reference

5. A manufacturer of plastic pressure-sensitive letters wants to supply kits of letters with a statistically adequate number of each letter in the alphabet.

Write a program to process a large corpus of English to determine the percentages of A's, B's, C's, etc. the manufacturer should place in each kit. Assume the words on the punched cards are all followed by a blank.

Reference

Forte, Allen. *SNOBOL 3 Primer*. Cambridge, Mass: MIT Press, 1967.

Lugton, Robert C. *English as a Second Language: Current Issue*. Philadelphia: The Center for Current Development, 1970.

The Manual For MACRO SPITBOL, The Florida State University Computer Center, 1980.

〈국문 초록〉

TESOL 혹은 TEFL을 위한 SNOBOL 4 프로그래밍 언어의 응용에 대한 고찰

이 명 언
(공주사대)

SNOBOL 4 프로그래밍 언어는 1971년에 미국 Bell전화 회사 연구소의 Griswold씨 등에 의해서 처음으로 고안된 언어인데 String-Oriented Symbolic Language를 의미하는 바, 이는 곧 Statements를 콤퓨터에 input시켜서 respond하게 하는 언어라고 본다.

이 중 MACRO SPITBOL은 1974년과 75년 사이에 coding이 된 SNOBOL의 implementation이라고 할 수 있고, 換言한다면, SNOBOL 언어는 많은 自然言語에서 추출된 많은 중요한 特性(features)을 지니고 있으면서 자연언어를 processing하고 操作할 수 있도록 考案이 된 것이다. 이 SNOBOL 특성을 利用해서 학습문제의 여러가지 分析, 對照와 比較를 할 수 있고 data나 자료의 수집 乃至는 L-1과 L-2의 構文의 相異性과 유사성을 비교해 볼 수도 있다. 곧 이를 이용한 학습과정의 自動化가 이루어지면 교사는 보다 더 학습현장에서 creative thinking을 하는 springboard의 役割을 하게 될 것이다.

영어교육의 문제의식으로 진통을 겪고 있는 한국의 現狀下에서 그리고 인구와 정보(지식)의 폭발에 직면한 현대에서 교육의 문제를 풀 수 있는 key로서 SNOBOL의 기대는 크고 확신해 볼 수 있다고 생각된다.

모쪼록 수다한 한국 영어교육의 문제점을 풀 수 있는데 이 SNOBOL이 큰 一助가 될 수 있기를 빌며 또 믿는다. ——한국도 본격적인 Computer시대에 突入했기에.