[From: *International Computing Symposium*, CINI Foundation, Venice, 12-14 April 1972.]

SLC-II A PROGRAMMING SYSTEM FOR NATURAL-LANGUAGE TEXT PROCESSING. A COMPARISON WITH PREVIOUS SPECIAL-PURPOSE PROGRAMMING LANGUAGES

Sergei PERSCHKE

Commission of the European Communities
Joint Nuclear Research Center
Ispra Establishment

In the late fifties and early sixties, in connection with the large expansion of ventures in machine translation, a series of special-purpose programming languages was designed and implemented, the widest-known of which are COMIT and SNOBOL. However, they did not find a wide application in computational linguistics and machine translation projects as they implied serious restrictions both in efficiency and methodology of the approaches.

SLC-II, which is an advanced development of a software conceived and implemented on IBM 7090 in the framework of the Georgetown-University Machine Translation project, is an attempt of overcoming the limitations of the previous solutions.

The characteristics and capabilities of the system which is being implemented at CETIS/Euratom on IBM 360/65 are discussed.

1. Introduction

There are two principal characteristics in the conception of special-purpose programming language for natural-language data processing in the late fifties and early sixties:

- the belief that the basic task is character-string processing
- on almost total disregard of efficiency as to computer time and storage.

These are to our opinion the reasons why programming languages like COMIT (developed by V. Yngve at MIT) and SNOBOL did not find the wide application the authors had hoped for. On one side, they were inconvenient for practical applications - read machine translation - because of the prohibitive run time and storage occupation. On the other side, they turned out to be inadequate for advanced experimental developments in computational linguistics because of the underlying concept.

In the same period,. there had been another development in connection with the Georgetown University machine translation project, which, actually, represented no progress from the linguistic point of view, since it, too, considered almost exclusively the string processing aspect, but introduced an important progress from the point of view of data processing. The system was called by its author, A.F.R. Brown, somewhat ambitiously Simulated Linguistic Computer (SLC), and the progress consisted, in respect to the other systems, in the fact that the over-all (translation) process was subdivided into two phases

- an information retrieval phase (text analysis and dictionary search)
- a problem phase (source text analysis, transfer, target text synthesis) for which a programming language was defined.

The Russian-English machine translation system developed by the Georgetown University and still in use at various places, indeed, was implemented in SLC, however, as an independent software it remained practically unknown, may-be because it was too closely connected to a project which was not very popular in the environment of computational linguistics, especially in the USA.

In the late sixties, at CETIS (EURATOM) the question of a special-purpose software for application in computational linguistics and automatic documentation was raised again in connection with some projects, especially machine translation, automatic indexing, automatic query formulation and information retrieval.

The basic approach of SLC was considered to be very promitting, and it was decided to design an advanced version (SLC-II) which should cope with the increased requirements.

2. The design of SLC-II

As the acronym SLC (Simulated Linguistic Computer) indicates the design is rather that of a special-purpose computer than of a programming system, and the description of its functions is an attempt of an abstraction of the processes implied in the applications involved. In a very broad sense all of these applications can be interpreted as language translation and question answering problems. The entire process was broken down into a series of cycles, each of which are consist of the following elements:

1. Source data
2. Dictionary
3. Grammar
4. Algorithm
5. Target data(which eventually become the source data of a subsequent cycle)

According to the cycle itself and also to the state of the art of the task involved, the algorithm in certain cases is implemented as a single SLC function (e.g. source text analysis; dictionary search; morphological analysis, etc), while in other cycles no such fixed functions could be defined (e.g. syntactic analysis, semantic analysis etc) as at present no more or less generally accepted models and algorithms are available, and it is uncertain whether such function will be used. In the latter case, micro functions on the level of computer instructions were defined.

2.1. Cycles of the SLC-II System

The analysis of the applications envisaged, permitted to define four major classes of cycles:

1) a data acquisition phase which reads the source text, identifies the "words" and looks them up in the morphological source language search dictionary;

2) a language translation phase which performs the analysis of the source text (syntactic, semantic , statistical etc), the transfer into the target language and the generation of the target text;

3) a text generation and editing phase which is the inverse process of the first cycle and produces a man-readable target text;

4) a question-answering phase which performs the functions of an information retrieval system (e.g. SDI, retrospective searches etc).

Each one of these cycles is subdivided into a series of subcycles which are discussed herebelow.

2.1.1. Data acquisition

The source data are unstructured natural language text in machine-readable form, and the purpose of this first cycle is

1. to identify character strings which according to the input conventions are elements of the source language;

2. to identify them with "words" i.e. lexical units of the source language.

Hence this phase is subdivided into two subcycles, for which the algorithms are invariant, while the dictionaries and grammars can be coded in symbolic form.

2.1.1.1. Source text analysis

For the algorithm, the source text is a continous sequence of unrelated characters, which must be grouped into strings which, according to the dictionary and the grammar, either are elements of the source language ("word items") and are used as arguments in dictionary search, or must be interpreted as mere character strings ("non-word items") which only can be transferred as such into the final translation.

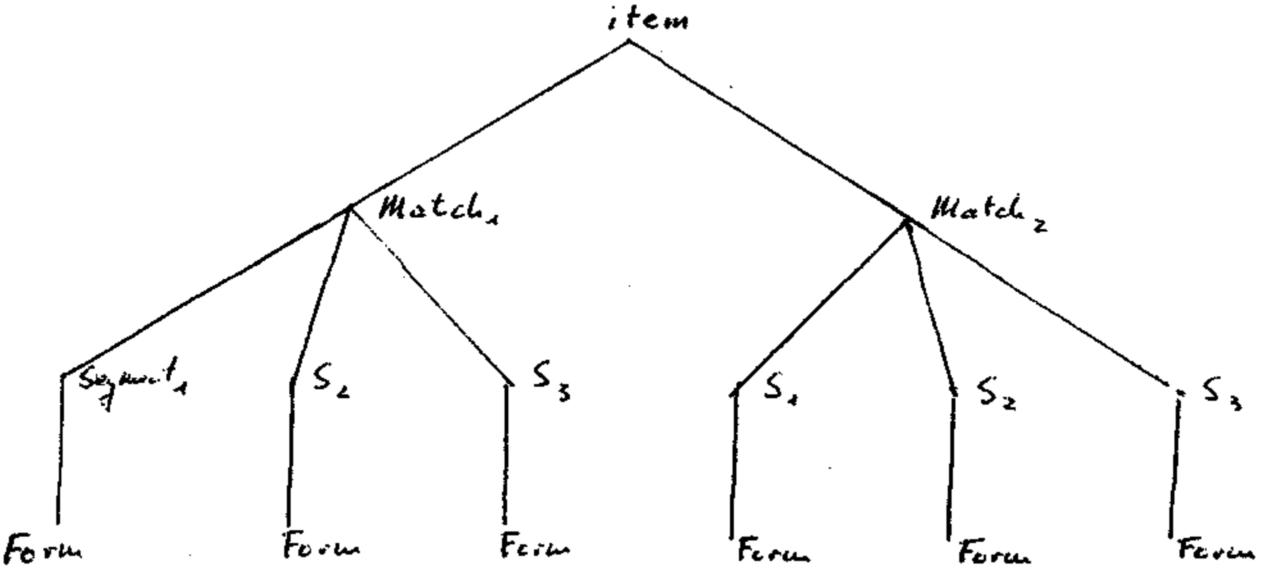
For this cycle compiler techniques with a table-driven syntactic analyzer were used. A dictionary of terminals is established, and the "grammar" is a set of table entries which are either terminals or non-terminals (i.e. push-down subroutines calls). To each table entry an action routine and an error recovery routine are associated. Both the terminals dictionary and the grammar are coded symbolically in SLC and compiled into the form recognized by the analyzer.

2.1.1.2. Dictionary search

The dictionary search, in principle, bases on a character string match between the "word items" and the search arguments in the morphological search dictionary. Matching is performed from left to right, and, as implied functions of the algorithms, the following operations are accomplished:

- morphological analysis
- segmentation of compound words
- detection of lexical homographs (multiple match)
- morphological analysis of words not found in the dictionary.

Both the morphological dictionary and the morphological tables (paradigms) are coded symbolically. The result of dictionary search, for each word item is the identification of the lexical unit of the source language (LXN), and the description of the inflectional form, which can also comprehend word derivation. As multiple match and word derivation are employed, the result may be represented as a three-level tree structure per item:



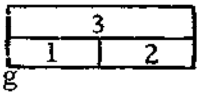
LXN, i.e. the identification code of the source language lexical unit, is used as search argument for loading the source language dictionary entries. At the end of this cycle, the results of dictionary search, for each logical text unit, which may be a sentence, an abstract etc, according to the application, are loaded into core storage as a four-level tree structure containing the items, match variants, segments and word forms. The source language dictionary information is attached to the structure at segment level and may be either a fresh copy in-place or merely a pointer to the general dictionary storage. At the lowest level (form) space for a link to the syntactic structure is provided for.

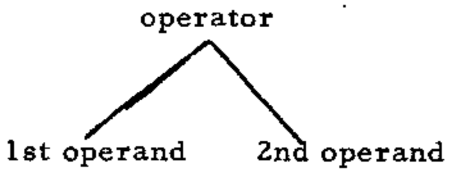
2.1.2. The language translation phase

This phase is subdivided into three cycles - source language analysis, transfer, target language synthesis.

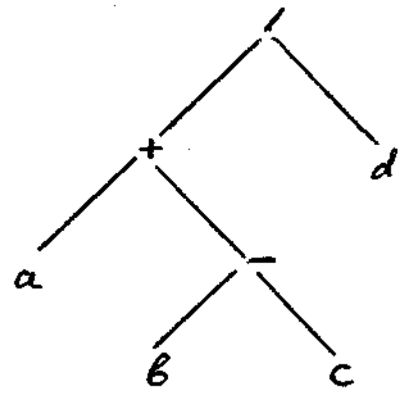
2.1.2.1. Source language analysis

The principal objective of this place is to give a formalized representation of the source text which could be interpreted as a simulation of comprehension. The analysis is, following the latest developments of computational linguistics, syntax and semantics-oriented.

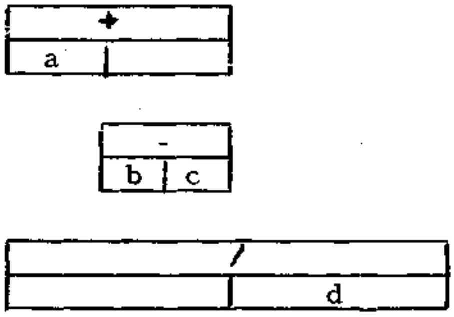
The syntactic model underlying the project design is inspired rather by the operational analysis of thought of the Italian Operational School (Silvio Ceccato) than by the structuralists (Chomsky, Tesnière). The basic syntactic unit is a relation, which always consists of an operator and two operands (analog to algebra) with the typical graphic representation of the Italian operational school  where fields 1 and 2 represent the operands and field 3 the operator. This representation, can be considered to be analog to the form used by compilers in the analysis of arithmetic and logical expressions:



for instance, an expression (a+ (b-c)) /d in this representation can be represented as:



while in the Ceccation representation it would become:



A set of SLC functions permits to construct and to handle such structures.

The analysis provides three levels of depth:

1. the surface structure which starts from a taxative list of all possible syntactic operators in the source language, and according to the grammar rules constructs the syntactic relations without interpreting their meaning;
2. the complement structure which interprets certain syntactic relations as complements of certain elements;
3. the semantic interpretation of the meaning of syntactic relations which attempts a full metalinguistic description of the syntax of a text.

This sort of analysis is in course at CETIS in the framework of the Russian-English machine translation project. As one can see the formalized description of the text is limited rather to the syntax and still considers the meaning of words as elementary units (semantic information on the lexical level has rather an instrumental function for recognizing syntactic structures). Therefore, for translation, a somewhat hybrid cycle - the transfer is necessary, which methodologically, bases on the assumption of a lexical equivalence of words.

2.1.2.2. Transfer

The source data for this cycle is the syntactic structure of the text and a somewhat simplified representation of the original linear structure (the levels match and segment are eliminated, and the items are linked not only to the source language, but also to the transfer dictionary entries.

In most of the cases, the transfer can base on a one-to-one equivalence, however, it must be able to handle the choice between several possible translations, and the transformation of syntactic structures due to a different syntactic value of the target language equivalents.

As it was said above, this cycle is a consequence of an imperfect analysis of the source language. There exist approaches which attempt to describe also the lexical meaning of the words in terms of elementary semantic components and relations between them, and the definition of the target language equivalent through a recomposition of these components. But such attempts are rather rudimentary and not yet mature for practical applications.

2.1.2.3. Target language synthesis

The process is the inversion of the source text analysis. The input data are the syntactic - metalinguistic - representation of the target text, and the output is a sequence of items described in terms of the identification code of the target language word, a binary vector describing the inflectional form and another binary vector defining the lay-out and graphic form.

2.1.3. Text generation and editing

This phase is optionaland is imployed if the results are natural language text. The output of the last translation cycle is a coded text which represents each "word" through the identification code and the binary vector describing the inflectional form. The algorithm locates the "stems" representing the word and attaches the pre- and suffixes which establish the inflectional form. The process, in principle, is the inversion of the morphological analysis. Further, the algorithm uses the display information for the final lay-out of the target text.

2.1.4. Question - answering

This phase is used, when some information retrieval function (e.g. SDI, retrospective search or fact retrieval is associated to the language translation function which, for instance, in this case could be indexing or query formulation.

The source data are automatically formulated queries, while the "dictionary" is the documentation data base. The system provides capabilities of handling both direct and inverted files. However, especially in the case that the IR language is somewhat more sophisticated than mere coordinate indexing, it appears that the use of direct files is advantageous and even can be made more efficient through automatic classification techniques of the document space, which may constitute easily useable master directories for a rapid access to the most pertinent information.

2.2. State of implementation

The first version of the system at present is in an advanced status of completion and is to become fully operational in early 1972. It is a batch processing version and operates in on overlay mode with a static storage management. A subset of the system which comprehends the data acquisition phase and uses the PL/1 programming language for the problem phase has become operational in summer 1971 and is used as the basic software of the automatic indexing project of CETIS.

At present, along with the implementation of the full SLC-II system, a version for a dialog operation is being prepared. The dialog is of eminent importance for applications which inherently involve man-machine interaction, as, for instance, machine-aided translation, or automatic question answering in natural language and information retrieval.

3. Conclusions and prospectives

In a most ambitious prospective, SLC-II should become a standard basic software and programming language in the field of computational linguistics and information science, in the same way, as, for instance, Fortran is in scientific or Cobol in commercial computation. Such a development, among others, could contribute to the solution of one of the most serious problems in these fields - the compatibility of data bases and procedures.

It is evident that SLC-II is already the basic software in all linguistic and information science projects at CETIS, but there already exist reasonable indications for a wider diffusion of the system in a near future.

References

[1] COMIT a) Programmers' Reference Manual
The M.I.T. Press (1961)

b) An Introduction to COMIT Programming
Share Distribution (1961)

c) PERSCHKE, S.: A COMIT Program for a
Provisional Russian-English Machine Translation
Procedure in: Mechanical Translation: The
Correlation Solution pp. 81-128 (1963)

[2] SNOBOL FARBER, GRISWOLD and PLONSKY: "SNOBOL,
A String Manipulation Language", Journal of the
A.C.M., Vol. 11 No. 2 January 1964

[3] BROWN, A.F.R. "Flexibility versus Speed" in Session 10, Proceedings
of the National Symposium on Machine Translation.
pp. 444-450, EDMUNDSON editor (1960)

[4] BROWN, A.F.R. The SLC System and Programming Language for
Machine Translation (2 volumes) Euratom Rep.
EUR 2418.e (1965)

[5] PERSCHKE, S. The Computer Programs of the "SLC" System for
Machine Translation. Euratom Rep. EUR 2583.e (1965)

[6] PERSCHKE, S. The use of the "SLC" System in automatic indexing.
Mechanized Information Storage, Retrieval and
Dissemination, North Holland, Amsterdam (1968)

[7] PERSCHKE, S. Possibilities of further development of automatic
language translation. Linguaggio e Pensiero Vol. 1
N^{o} 1, Milan 1970

[8] PERSCHKE, S. SLC-II one more software for the solution of
linguistic problems. paper presented at the 1971
International Conference on Computational Linguistics
Debrecen, September 4-7, 1971

[9] PERSCHKE, S. SLC-II, eine software für linguistische Datenverarbeitung.
paper presented at "Fachtagung: Information
Retrieval Systeme (IRS) - Management Information
Systeme (MIS)" Stuttgart, December 9-11, 1970

[10] FANGMEYER, H. Stand der Entwicklungsarbeiten für die automatische
Indexierung bei der europäischen Forschungsanstalt
Ispra. paper presented at: Deutscher Dokumentartag,
Bad Herrenalb, October 18-22, 1971