[*Machine Translation Summit*, September 17-19, 1987, Hakone Prince Hotel, Japan]

The FGCS Project and Machine Translation Systems

Kazuhiro Fuchi

Institute for New Generation Computer Technology (ICOT)

1. INTRODUCTION

This paper covers the following points.

(1) The Fifth Generation Computer Project proposes basic technologies for future advanced information processing systems, including machine translation systems.
(2) The concept of logic programming, which is the basis of the Fifth Generation Computer Project, is extremely well suited to natural language processing.
(3) High-level machine translation systems are typical knowledge information processing systems, and will influence the form of future computer systems.

2. FIFTH GENERATION COMPUTER PROJECT

The Fifth Generation Computer Project, which began in 1982, is a 10-year project operating under the auspices of the Ministry of International Trade and Industry (MITI). It is divided into three stages, an initial stage of three years, an intermediate stage of four years, and a final stage of three years. ICOT has just finished five years of operation.

Briefly, the objective of the project is to build a new framework of computer technology, with the realization of an inference machine as its nucleus. It covers both hardware and software, reviewing the role of both, and attempts to modify the boundary between them.

When one thinks of a machine that can perform logical inference efficiently at high speed, one probably thinks of a parallel machine. One of the ICOT research objectives is the realization of a parallel inference machine. Inference is the way of combining knowledge represented in the form of rules. Consequently, the realization of a knowledge processing machine is another way of expressing this objective. A machine that can efficiently store, manage and retrieve large amounts of knowledge is the focus of research for a knowledge base machine.

These items form the search for new computer architecture. In the fifth generation, computers will appear in many forms, such as supercomputers for parallel numerical computation. Researchers at ICOT see the inference machine as the machine that will take the place of today's general-purpose computers and that will be the focal point of computers. The progress of VLSI and other device technology will support the realization of this new machine group.

Another major focus of our research is creating new software that is based on basic logical inference. Considering the applications of future high-level computers, it can be predicted that AI and related applications will become more important. The basis of these applications is knowledge processing and natural language processing. The focus of research here is methodology for creating high-level software, including knowledge and natural language processing, with high productivity. Logic programming is one of the concepts that supports both the AI viewpoint and the software engineering viewpoint, because the basic operation is logical inference.

The FGCS Project does not directly include machine translation. However, one of the focal research themes of the project is natural language processing. High-level machine translation systems are very likely to be one of the future applications of this new technology. This project may be viewed as the contributor of the element technology for machine translation.

3. LOGIC PROGRAMMING AND NATURAL LANGUAGE PROCESSING

Logic programming and natural language processing have been intrinsically closely related. Prolog, which is a logic programming language, was born from research on machine translation [Colmerauer 78]. It is used not only for natural language processing, but is recognized as having the potential to become a new paradigm for programming in general [Kowalski 74]. To give an example, researchers at ICOT have created SIMPOS, a total OS, using ESP, a form of extended Prolog.

The most frequently used grammar description in natural language processing is one that augments the context-free grammar rules. An example is:

S → NP VP {augmented terms }

This rule denotes that the sentence (S) is formed from a noun phrase (NP) and a verb phrase (VP).

A program that analyses a given sentence using the context-free grammar rules is called a parser. The parser is shown in Figure 1.

$$\begin{tikzcd}
\text{(Augmented) context-free grammar rules} \arrow[rd] & & \\
\text{Dictionary} \arrow[r] & \text{Parser} \arrow[d] & \text{Sentence} \arrow[l] \\
& \text{Sentence analysis result} &
\end{tikzcd}$$

Figure 1. Parser and sentence analysis

The parser is an interpreter which uses linguistic knowledge (dictionaries and grammar rules) to analyze a sentence. The rules (or dictionaries) can be seen as the program (or data) of the interpreter. This kind of grammar description is an extension of a logic programming language such as Prolog.

Pereira et al. [Pereira 80] developed the Definite Clause Grammar (DCG), a simplified form of the Metamorphosis Grammar developed by Colmerauer [Colmerauer 78]. Grammar rules and dictionaries written in DCG. which is one form of augmented context-free grammar, are converted to a Prolog program, which has a one-to-one correspondence with them. For example, the grammar rule given above is converted to a Prolog program, as follows.

```
s(s(NP,VP),S0,S2):-np(NP,S0,S1),vp(VP,S1.S2),{...}
```

When a natural language sentence is processed by the converted Prolog program, top-down, depth-first sentence analysis is performed. This makes the conventional practice of building a parser required for natural language processing unnecessary. Prolog's built-in function is used instead of a parser.

However, this method poses one major problem, that of an endless loop in analysis when left-recursive grammar rules are used. This problem can be avoided by using bottom-up analysis. In the method developed by Matsumoto [Matsumoto 83], when grammar rules written in DCG format are converted to a Prolog program with a correspondence that is close to one-to-one, the Prolog program performs bottom-up, depth-first analysis. This is called the BUP system [Matsumoto 83] [Tanaka 86].

Figure 2 shows the basic concept of sentence analysis using Prolog.

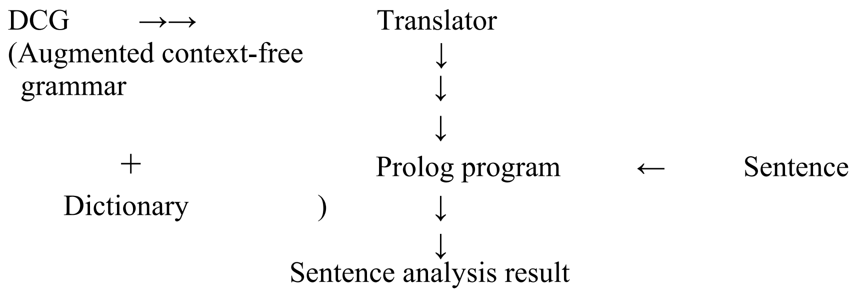

Figure 2. Sentence analysis using Prolog

It should be noted that, in the method shown in Figure 1, a special program must be built into the parser to execute syntactic and semantic analysis at the same time. However, for many parsers, it is difficult to build in this kind of special program satisfactorily. Because of this, parsing that fused syntactic and semantic analysis in the method shown in Figure 1 became a research topic. However, with the method shown in Figure 2. the Prolog program is embedded in the augmented part of the DCG and executed, fusing syntactic and semantic analysis very simply.

Recently, different from BUP that executes sentence analysis depth-first, a method that converts DCG expressions and obtains Prolog programs that execute sentence analysis breadth-first and bottom up has been developed [Matsumoto 86]. This is called the SAX system. This method was intended as an algorithm for parallel execution, but it also provided fast processing for sequential programs. Figure 3 shows part of the most recent experiment results. The experiment environment was a VAX 785, with a QUINTUS Prolog compiler for BUP and a Guarded Horn Clause (GHC) compiler developed at ICOT for SAX. These yielded results for syntactic analysis only.

a) **Sample sentences parsed**

(1) The annotations provide important information for other parts of the system that interpret the expression in the context of a dialogue.

(2) For every expression it analyses, a diagram provides an annotated description of the structural relations that hold among its constituents.

(3) Procedures can also assign scores to an analysis, rating some applications of a rule as probable or as unlikely.

b) **Parse time in msec**

Sentence no.	No. of parse trees	Parse time (msec)	
		BUP	SAX
(l)	5	112,500	1,750
(2)	1	23,490	770
(3)	6	61,490	1,010

Figure 3. Parse time of BUP and SAX

As can be seen from Figure 3, sentence analysis based on logic programming is extremely fast. This is evidence of the practical value of natural language processing based on logic programming.

Not only from the viewpoint of syntactic analysis, but also from that of semantic analysis, logic programming languages like Prolog are natural. Many types of inference (problem solving) are necessary for semantic analysis. Meta-inference and higher-order inference including inference process control, such as default reasoning, are necessary. Prolog is a very suitable base programming language for realizing these types of inference.

It is possible to create languages at a level equivalent to higher-order inference, but these languages can frequently be considered as logic programming languages that are a natural extension of Prolog.

Logic programming languages like Prolog are natural and efficient for realizing language processing functions, including syntactic analysis. This is not accidental; the internal structure is common to both.

There is, or should be, a close internal relation between logic and language. However, linguistics and the study of logic have been separate for a long time. They started moving together again about when Montague theory appeared. At the same time, linguistics started moving away from its focus on syntax and towards a consciousness of semantics. New linguistic theories such as LFG and GPSG are reviewing syntactic systems and are starting to use logic formats for semantic representation.

Needless to say, first-order predicate logic that may be sufficient for expressing formal system of mathematics is not sufficient for expressing natural languages. It is necessary to reorganize logic systems. It is also necessary to review modal logic and intentional logic. Development of situation semantics [Barwise 83] can be viewed as one of the attempts to review logic systems.

These linguistic theories belong to a category that can be called unification grammar formalism [Shieber 86]. Unification in this sense is a natural extension of unification in logic programming languages. The latest focus of research in logic programming is constraint programming. It is an extended unification that incorporates constraints and indeterminates.

At ICOT, researchers are attempting to build a discourse understanding system called DUALS, based on situation semantics [Mukai 85]. The current framework of this system is as follows.

DUALS (discourse understanding system) = information flow model
= situation semantics and situation theory
+ type theory
+ speech act theory
+ constraint logic programming

Mukai has created a language, called CIL, as the semantic representation language for this system. CIL is a logic programming language that is a natural extension of Prolog, with the addition of partially specified terms, constraints and lazy evaluation.

4. KNOWLEDGE INFORMATION PROCESSING SYSTEMS AND MACHINE TRANSLATION SYSTEMS

Figure 4 shows the concept of the global configuration of a machine translation system. As explained in this section, it can be viewed from the standpoint of knowledge information processing systems.

The center of Figure 4 can be seen as the knowledge base. It stores rules of source language and target language dictionaries and grammar supplemented by common-sense rules for translation.

The text in the source language to be translated obtains help from the inference engine, and uses the knowledge about the source language in the knowledge base to analyze syntax, semantics and context. The results are converted to semantic structure.

The results are then sent to the sentence generator. The sentence generator obtains help from the inference engine, refers to grammar rules and dictionaries in the knowledge base for generating the target language, and outputs the text in the target language.

```
                                        (Inference engine)

|  Context   analysis   |      |       |---------------------------- |
|                       |      |       |          ↑    ↓            |       |      |----------- --|
|  Semantic  analysis   | →    |       | -------------------------- |       | →  | Sentence  |
|                       | ←    |       |  (Knowledge base)          |       | ←  |           |
|  Syntax     analysis  |      |       |                            |       |    | generation |
|-----------------------------|      |       |   Dictionaries             |       |    | ------------ |
           ↑                   |  --   |     Grammar rules          |  --   |           ↓
           ↑                           | -------------------------- |                   ↓
           ↑                                                                            ↓
  Source language                                                              Target language
```

Figure 4. Machine translation system

Considered in this way, a translation system can be viewed as a typical knowledge information processing system. (For example, the semantic representation language, CIL, can be viewed purely as a knowledge representation language.) To develop future high-level machine translation systems, the system must hold large amounts of knowledge, including linguistic knowledge. This presents several problems, which can be summarized as follows.

(1) Knowledge representation - how to express knowledge on the computer

(2) The acquisition of large amounts of knowledge

(3) The establishment of technology to process this knowledge

These problems are the base of knowledge information processing technology.

More specifically, if a "question" and "answer" are substituted for the source language and target language respectively shown in Figure 4. Figure 4 can be regarded as a question answering system using natural language. A question answering system is a typical natural language understanding system. Consequently, research into natural language understanding systems and machine translation systems will be mutually beneficial.

Figure 4 shows the necessity for the development of machine translation systems from these points of view to make future machine translation systems more advanced.

5. CONCLUSION

The FGCS Project is working towards establishing new systems of computer technology for integrating knowledge bases and inference mechanisms.

Large-scale, electronic dictionaries will be necessary as knowledge bases for future machine translation. Research in another project at the Japan Electronic Dictionary Research Institute, is expected to yield good results. This project was first conceived and investigated within the FGCS Project, and was established independently [Ishiwata 85]. It is a national project that is a sister to the FGCS Project.

The progress of the CYC Project, being operated by MCC in the U.S.A. also merits attention. The CYC Project is building an electronic encyclopedia knowledge base [Lenat 86], which will form the common-sense rules of translation in the knowledge base described above.

Research in machine translation has a long history, and today, practical systems are being built. It is important to gain practical experience based on ideas and techniques accumulated to date. At the same time, it is also important to promote basic research aiming at future high-level machine translation systems. Important factors in this research are the establishment of linguistic models and the accumulation of linguistic data. For machine translation, it is also essential to review the form of computer systems that will be the foundation for implementing these systems.

REFERENCES

[Barwise 83] Barwise, J. and Perry, J., Situations and Attitudes, MIT Press. 1983

[Colmerauer 78] Colmerauer, A., Metamorphosis Grammar, in Bolc (ed.), Natural Language Communication with Computers, Springer-Verlag, pp.133-190, 1978

[Fuchi 86] Fuchi, K. (ed.), Shizen Gengo no Kiso-ron (Basic Theory of Natural Language), Kyoritsu, 1986 (in Japanese)

[Ishiwata 85] Ishiwata, T. et al., Basic Specifications of the Machine-Readable Dictionary, TR-100, ICOT, 1985

[Kowalski 74] Kowalski, R., Predicate Logic as Programming Language, Information Processing. 1974

[Lenat 86] Lenat, D., Prakash, M. and Shepherd, M., Using Common Sense Knowledge to Overcome Brittleness and Knowledge, Acquisition Bottlenecks, AI Magazine, 6, 4, pp.65-85, 1986

[Matsumoto 83] Matsumoto,Y., et al., BUP - A Bottom-up Parser Embedded in Prolog, New Generation Computing, 1, 2, pp.145-158, 1983

[Matsumoto 86] Matsumoto, Y. and Sugimura, R., Ronri-gata Gengo ni Motozuku Kobun Kaiseki System SAX (SAX - A Structural Analysis System Based on Logic Programming Languages), Computer Software, Japan Software Society, 3, 4, pp.4-11, 1986 (in Japanese)

[Mukai 85] Mukai, K., Unification over Complex Indeterminates in Prolog, ICOT TR-101, 1986

[Pereira 80] Pereira, F. et al.. Definite Clause Grammar for Language Analysis - A Survey of the Formalism and a Comparison with Augmented Transition Networks, Artificial Intelligence. 13, pp.231-278, 1980

[Shieber 86] Shieber, S.M., Pereira, F.C.N., Karttunen, L. and Kay, M. : Compilation of Papers on Unification-base Grammar Formalisms, Parts I and II, CSLI Report, No. SCLI-86-48, April, 1986

[Tanaka 86] Tanaka, H. et al.. Logic Programming to Keisan Gengo-gaku (Logic Programming and Computational Linguistics), Joho Shori (Information Processing). 27, 8, pp.940-946. 19S6 (in Japanese)