

# THE FRAMEWORK OF KNOWLEDGE REPRESENTATION AND ITS RETRIEVAL IN LGS -- THE LITERATURE GUIDE SYSTEM

Toyo-aki NISHIDA and Shuji DOSHITA  
Department of Information Science  
Kyoto University  
Sakyo-ku, Kyoto 606, Japan

This paper describes the framework of knowledge representation and the techniques for information retrieval in a program called LGS (the Literature Guide System). LGS will answer questions about scientific literatures in natural language. Model theoretic approach is employed to represent the meanings of sentences in the abstracts of literatures. The knowledge base is a set of interpreted logical formulas, and is partitioned into possible worlds. User's own view to the knowledge base can be incorporated into objects called script frames. A script frame is defined in terms of logical formulas. A query to the knowledge base is presented as a pattern. A list containing all items which match the pattern will be returned as the result. A set of rules for pattern matching are defined in terms of logic, and are designed so as to cover the cases of paraphrases and categorical implications. LGS is composed of an accommodator, a retriever, and a natural language interface. It is under development on the personal LISP system.

## 1. INTRODUCTION

LGS [1] is an experimental system which integrates knowledge about scientific literatures and which will answer questions about them. LGS is toward a natural language data base system (NLDB system). In any NLDB system, the problems of knowledge representation and information retrieval are crucial.

We employ the model theoretic approach to a NLDB system. The model theory gives the formal semantics to the knowledge representation in a NLDB system. The meaning of a sentence is described as a logical formula of the intensional logic. Many aspects of natural language expressions such as modalities can be clearly explained by the concept of a possible world. Our approach to a NLDB system can be summarized in the following schema;

NATURAL LANGUAGE EXPRESSION  
(STEP1) syntactic analysis  
LAMBDA EXPRESSION  
(STEP2) lambda calculus  
LOGICAL FORMULA OF INTENSIONAL LOGIC  
(STEP3) interpretation  
DOMAIN DESCRIPTION BY META-LANGUAGE  
(STEP4) accommodation  
THE KNOWLEDGE BASE

The idea about STEP1 and STEP2 is formalized in MG (Montague Grammar) [2] which is implemented in the natural language interface. In this paper, we concentrate on STEP3 and STEP4. In STEP3, meta-language expressions are created from logical formulas, and in STEP4, they are accommodated into the knowledge base with some auxiliary data objects which are called frames.

Information retrieval plays an important part in question answering. Pattern matching among meta-expressions is a basic technique in information retrieval. Rules for pattern matching are defined in terms of logic. The concept hierarchy is defined as logical implications between concept predicates. The rules are presented to cope with the cases of paraphrases and categorical implications.

We are implementing those ideas as LGS. Currently, we have a limited version, LGS-0, where simple dialogs are available using Japanese or query commands.

## 2. KNOWLEDGE REPRESENTATION AND THE KNOWLEDGE BASE

### 2.1 Meta-language Expressions

In general, the interpretation of natural language, that is, the proper generation of domain description from a given natural lan-

guage expression, is accompanied with much difficulty, especially when treating quantifiers. So, by employing some conventional expressions in meta-language to leave ambiguity in translation, we shall put aside those problems temporarily. Here we show some of those representations. (Representations in natural language are enclosed by "...".)

(E1) indef[x; program(x)] for "a program"  
 (E2) def[x; program(x)] for "the program"

Def and indef are a definite operator and an indefinite operator, respectively.

(E3) love("she", "he") in a possible world P, for "he believes that she loves him."

Where the possible world P is the one which is "believed by him."

## 2.2 The Knowledge Base

Conceptually, the knowledge base is a set of meta-expressions which are partitioned into possible worlds. In the actual implementation, the knowledge base is constructed from data structures, called paragraphs, units, and frames. A paragraph is a possible world management block. A unit is a data structure for internal representation of a meta-expression. There are four types of a unit, I-, P-, F-, and O-unit. An I-unit (Individual-unit) is used for an object, which includes a set of references to the object. A P-unit (Predicate-unit) is for a predicate. Not only a set of references but also a list of 'case slot and filler' pairs for the predicate are included. A F-unit (Function-unit) is for a function. An O-unit (Operator-unit) is for an operator.

A frame is a data structure for representing more complicated but stereotyped knowledge. Two levels of a frame, a basic frame and a script frame are defined. A basic frame is a data structure defined for each predicate symbol and function symbol. It includes the definition of the symbol and the index of its occurrences in the knowledge base. The concept hierarchy, which plays a key part in inference on the knowledge base, is embedded in the collection of basic frames as implication relationships among predicate symbols. A script frame is a data structure defined for more complicated situation. It includes case structure, frame definition, and knowledge base search strategies. Users can define script frames with their own case structure and knowledge base search strategies. Here we shall show an example of a script schema about a "natural language understanding system."

```
script-name:=
  NATURAL-LANGUAGE-UNDERSTANDING-SYSTEM;
  1) VARIABLE DECLARATIONS:
    DCL x,y,z,w objects;
    DCL P,Q,R,S,T,U predicates;
  2) INSTANTIATION CONDITIONS:
    P:understand(x,y) & Q:written-in(y,z) &
    R:natural-language(z) & S:system(x);
  3) CASE DEFINITIONS:
    name:=w such that name(x,w),
    theme:=y,
    method:=T such that method(P,T),
    performance:=U such that actor(U)=x;
```

Script frames will be created for all knowledge structures that fill the instantiation conditions.

## 2.3 Interpretation and Accommodation

In this section, we shall overview how the meta-expression is created from a given logical formula, and accommodated into the knowledge base. First, a paragraph is created for the actual world. Then the knowledge structure construction program reads logical formulas one by one. Let a given formula be f, then the construction process proceeds as follows;

- 1) set a pointer (WP) to the actual world.
- 2) seek a possible world which is related to the actual world by a modal operator or an intensional operator in the formula. If it is found, move WP to the possible world, and if not, create a new possible world corresponding to the operator and move WP to the new world. Remove the operator from f.
- 3) remove indef and def operators if possible. If f is a predicate P(...indef[x;q(x)]...), and P is an extensional predicate, create a new object c in the possible world pointed by WP, and generate internal structure corresponding to P(...c...) and q(c). If f is a predicate P(...def[x;q(x)]...), search the object that satisfies q(x), and if it is found, replace def[x;q(x)] by the object.
- 4) construct a network structure for the resulting formula and put it into the world pointed by WP.
- 5) add an index to the related basic frames.

Steps 2-5 may be called recursively when the given formula is nested. Script frames may be instantiated for the generated knowledge structure if the instantiation conditions are filled.

## 3. INFORMATION RETRIEVAL

An important type of an information re-

trieval request may be "for the given predicate P, find all objects (x's) such that P(x) is true in some possible world." Any object y should be retrieved if  $D(y) \rightarrow P(y)$  holds in some possible world, where  $D(y)$  is the conjunction of all descriptions about y. In LGS, the pattern matching method is used to test the implication, since the direct proof of the implication by a resolution-based theorem prover is inefficient and inflexible. Each formula is regarded as a symbolic pattern. The predicate  $\text{match}[P^*, y]$  is tested instead of  $D(y) \rightarrow P(y)$ , where  $P^*$  is  $\text{indef}[x; P(x)]$ .

The retriever consists of an index searcher, a matcher, and a modality generator. The index searcher examines the index structure of basic frames to produce a list of candidates. Then for each element y in the list of candidates, the matcher tests if the pattern matches y. Any candidates that do not match the pattern are removed from the list. Finally, the modality generator may add appropriate modal operators to the remaining candidates so that they may be the predicates in the actual world.

A set of rules are defined for pattern matching. Especially,

(RULE1) categorical implication rule:  
a pattern P matches another pattern D if  
 $D \vdash P$ .

(RULE2) paraphrasing rule:  
a pattern P matches another pattern D if there exists a predicate Q such that  
 $P \vdash Q$  and  $D \vdash Q$ ,  
where natural deductions are used with the set of axioms for the concept hierarchy.

RULE1 covers the cases of categorical implications. For example, the pattern "animal" matches "human-being". RULE2 covers the cases of paraphrases. For example, a predicate  
 $\text{give}(\text{actor}=\text{A}, \text{object}=\text{B}, \text{direction}=\text{C})$   
can match the predicate  
 $\text{get}(\text{actor}=\text{C}, \text{object}=\text{B}, \text{source}=\text{A})$   
where the predicate Q is  
 $\text{transfer}(\text{object}=\text{B}, \text{source}=\text{A}, \text{direction}=\text{C})$ .  
The range of the applications of deductions in RULE2 should be limited for plausibility.

#### 4. EXPERIMENTS

LGS is under development on the personal LISP system (LISP1.7) [3]. We currently have a limited version of LGS (LGS-0). LGS-0 can integrate the knowledge about the literatures and answer questions. The interactions can be done in natural language. LGS-0 consists of three program modules (the natural language interface, the accommodator, and the retriever) and the

knowledge base. The natural language interface is used (1) to translate both sentences of literatures and queries into logical formulas of intensional logic, and (2) to generate natural language outputs as the replies to queries. The details are described in [1].

LGS-0 can perform simple dialogs like below in Japanese.

U: do you know any literature about a question answering system?  
(Information retrieval is done.)  
S: YES, I KNOW 20 SUCH SYSTEMS.  
U: take one of them!  
U: what is the name?  
S: LGS  
U: what kind of questions can it answer?  
S: IT WILL ANSWER QUESTIONS ABOUT SCIENTIFIC LITERATURES.  
(Question answering about the found object.)  
U: which papers describe it?  
S: THE PAPER THAT IS SUBMITTED TO THE 6TH IJCAI.  
(A literature is guided.)  
U: what aspects does it describe?  
S: THE FRAMEWORK OF KNOWLEDGE REPRESENTATION AND INFORMATION RETRIEVAL IS DESCRIBED.  
... (U: a user, S: the system)

#### 5. CONCLUSION

In this paper, we have attempted to show our model theoretic approach to a NLDB system. We think LGS is a good example for such an approach. Of course, there are lots of problems to be solved in future. Especially, the problem of inferences in the possible world domain [4] is important.

#### REFERENCES

- [1] T.Nishida "A Knowledge Based Question Answering system on Scientific Literatures." Master Thesis at Kyoto University, 1979, (Japanese)
- [2] D.Dowty "A Guide to Montague's PTQ." Indiana University Linguistic Club, 1978
- [3] S.Doshita, K.Hiramatsu, K.Kakui "Implementation of LISP System Using Direct Accessible Bulk Memory." Trans. of IECE Japan, vol.J61-D, no.5, 1978, (Japanese)
- [4] R.Moore "Reasoning about Knowledge and Action." in Proc. IJCAI-77, pp.223-227, 1977