A Mechanical Deductive Inference Rule Based on Knowledge Structure

Setsuo OHSUGA*

Abstract

A deductive algorithm that will support the natural language question answering system in a realistic environment involving a large data file is discussed together with the basic internal form of representing knowledge and a knowledge structure. The requirements which the deductive algorithm should meet are presented, and a new algorithm is proposed.

1. Introduction

In this paper we discuss a deductive algorithm that will support the natural language question answering system in a realistic environment involving a large data file. The deductive processor is not only essential for a QA system to understand natural language but very usefull because it allows the system to use general information as well as specific information in the data file resulting in a considerable saving of memory. There are, however, a number of requirements which the deductive processor should meet.

First of all, since the deductive processor has relation to the internal form of knowledge representation; and the internal form must be flexible enough to represent the extensive meanings of words or sentences, the deductive algorithm must adapt to the extensive internal forms.

On the other hand, the deductive processor is required to be theoretically complete. The completeness may be assured without difficulty when the internal form is restricted to the first order predicate; but that's not the case when the more flexible form is used because, then, the ambiguity of the language might come into the system and obscure the definition of completeness. What we can do now is develop the deductive algprithm that is applicable to the extended internal form and shows its completeness when the internal form is restricted to the first order predicate.

In a realistic environment involving a large data file, the deductive operation must

This paper first appeared in Japanese in Joho－Shori (Journal of the Information Processing Society of Japan), Vol. 17, No. 2 (1976), pp. 100～109.

* Institute of Space and Aeronautical Science, University of Tokyo

be preceded by retrieval of the relevant information for deduction. The deductive processor, therefore, must be very efficient for retrieval operation.

So far, a number of deductive processors have been proposed in the area of question answering and theorem proving. None of them, however, meets these requirements. We need a new algorithm.

2. Basic form of representation

To begin with, we define the basic internal form of knowledge representation of which some extention is possible to conform to the language expression.

Let's think of an example sentence: "Man is mortal". Instead of representing it as $(\forall x)[MAN(x) \rightarrow MORTAL(x)]$ as usual, we adopt a way of representing it as $(\forall x/MAN)$ $MORTAL(x)$. Here, MAN denotes a set of entities which is a domain of the variable x, over which the predicate MORTAL is true. Thus, ordinary $(\forall x)[X(x) \rightarrow F(x)]$ and $(\exists x)$ $[X(x) \cap F(x)]$ are modified to $(\forall x/X)F(x)$ and $(\exists x/X)F(x)$, respectively, when X contains the concept of a set of entities. We call these modified forms the predicate too and define from it literals, formulas and so on the same as is done in the theorem proving. This new form of representation has many interesting characteristics, but we do not mention them here.

3. The knowledge structure

The information is organized based on the set-inclusion relationship. Many different sets are brought into the system by words(nouns). Every set is represented in the system by a block of the memory words containing some data peculiar to it. Any of the sets, other than the universe, are definable from the other set or sets either as the subset with special property, or as the result of set operation.

Let Y be a proper subset of a set X and be characterized by a particular property denoted here by an attribute and value pair A(V). Let it be written as Y = X*A(V). If there is another set Y' such as $Y' \subset X$; and if it is represented as Y' = X*A(V') with the same attribute as that of Y but with different value, then $Y \wedge Y' = \phi$ provided V and V' have no common element. $\wedge$ represents the intersection of the sets. For instance, MAN = PERSON*SEX(M) and WOMAN = PERSON*SEX(F). MAN and WOMAN partition PERSON. By using the other attributes, the other partitions of PERSON are defined.

On the other hand, some sets are defined as the intersections of the other sets. For instance, BOY = MAN$\wedge$CHILD. The skeleton structure of information is organized based on these set theoretical relations as shown in Fig.1. Then the assertions are incorporated into the structure in such a way that the assertion containing the domain X is connected to the set X in the skeleton structure. Fig.2 shows, as an example, the case $(\forall x/X)(\exists y/Y)(\forall z/Z)[\sim F(x\ y) \cup G(y\ z)]$ is connected to the sets X, Y and Z.

4. Deductive algorithm

The deductive algorithm shown in this section is substantially the same as the resolution principle but is modified so as to meet the requirements mentioned before.

4.1. Lines of approach

Suppose, given a question C, that

$$P_1 \cap P_2 \cap P_3 \cap \cdots \cap P_n \Rightarrow C \qquad \text{-----(1)}$$

where P_i (i=1,2,---,n) is the assertion in the data file. For ease of explanation, let

$$\tilde{P}_c^0 = P_1 \cap P_2 \cap \cdots \cap P_n = \bigcap_i P_i$$

$$\tilde{P}_c^j = P_1 \cap P_2 \cdots \cap P_{j-1} \cap P_{j+1} \cap \cdots \cap P_n = \bigcap_{i \neq j} P_i$$

$$\tilde{P}_c^{jk\cdots l} = \bigcap_{i \neq j,k,\cdots,l} P_i .$$

Then Eq(1) is written as $\tilde{P}_c^0 = \tilde{P}_c^j \cap P_j \Rightarrow C$.

Note here that the set $\{P_i\}$ is decided depending on C and is not known in advance. Let

$$\mathbb{P}_c = \{ P_i \ / i=1,2,\text{-----},n \} .$$

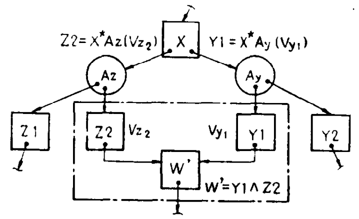


Fig.1 Skeleton structure element.

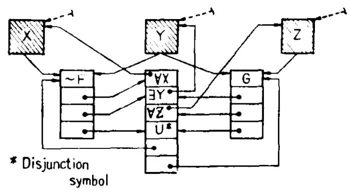


Fig.2 Incorporation of assertion to the structure.

We introduce a new concept of replacement. When a formula P_j' is given, a new formula R_j^* as follows is called the replacement of C by P_j'. Let R be a set of formulae such as $\mathbb{R} = \{ R_j / R_j \cap P_j' \Rightarrow C \}$, then $R^* \subset \mathbb{R}$ and for any formula S, if S implies some R_j in $\mathbb{R}$ then S implies R_j^* too.

Now, suppose $P_j' = P_j \in \mathbb{P}_c$. Then, if R_j^* is true, C is also proved true. If we regard $\tilde{P}_c^j$ as S in the definition of replacement, then $\tilde{P}_c^j \Rightarrow R_j^*$. Thus, $\tilde{P}_c^0 \Rightarrow C$ is equivalent to $\tilde{P}_c^j \Rightarrow R_j^*$. This R_j^* is then regarded as the new question, and this procedure is repeated. Then $\tilde{P}_c^{jk} \Rightarrow R_{jk}^*$, $\tilde{P}_c^{jkl} \Rightarrow R_{jkl}^*$ and so on where R_{jk}^* is the replacement of R_j^* by P_k, and so on. If , at last, a replacement is implied by a disjunction-free formula; then all replacements, as well as C, are proved true.

If, to the contrary, $P_j' \notin \mathbb{P}_c$, then this process will fail. As $\mathbb{P}_c$ is the latent set and is normally a very small fraction of all assertions in the data file, some efficient and systematic search method must be included in this process. This is the first problem of this approach. To obtain the replacement in detail from P and C is the second problem.

Both of the problems are solved by analyzing the condition of $P \cap R \Rightarrow C$, or equivalently, from the condition of unsatisfiability of $P \cap R \cap \sim C$.

Suppose that a formula C is given as the question and that a literal G is chosen arbitrarily in C. Then we represent C as

$$C;\ (Q_{c1}\, x_{c1}/X_{c1})(Q_{c2}\, x_{c2}/X_{c2})---(Q_{ct}\, x_{ct}/X_{ct})[H(x_{k_1}, x_{k_2}, ---, x_{k_s}) * G(x_{j_1}, x_{j_2}, ---, x_{j_p})]$$

where * represents either the conjunction symbol or the disjunction symbol by which the literal G is connected to the remaining part H of C. G is used as the key literal in succeeding one cycle in the deductive process. There are some selection algorithms for G, but we do not mention them here. Q represents either $\exists$ or $\forall$.

On the other hand, every assertion is represented, without loss of generality, as

$$P;\ (Q_{p1}\, x_{p1}/X_{p1})(Q_{p2}\, x_{p2}/X_{p2})---(Q_{pn}\, x_{pn}/X_{pn})[F(x_{i_1}, x_{i_2}, ---, x_{i_m}) \cup K(x_{j_1}, x_{j_2}, ---, x_{j_p})]$$

where F is the set of literals logically connected to each other including the case of the empty set. Fig.3 shows the condition of $P \cap R \cap \sim C$ being unsatisfiable. From this figure, it is clear that in order that P is chosen to generate the replacement, K must be the same with G; and the replacement R should be of the form

$$R;\ (Q_{r1}\, x_{r1}/X_{r1})(Q_{r2}\, x_{r2}/X_{r2})---(Q_{rm}\, x_{rm}/X_{rm})[\sim F(x_{i_1}, x_{i_2}, ---, x_{i_n}) * H(x_{k_1}, x_{k_2}, ---, x_{k_s})]$$

where * is the same logical connective as that of C. In the following sections, further details are discussed.

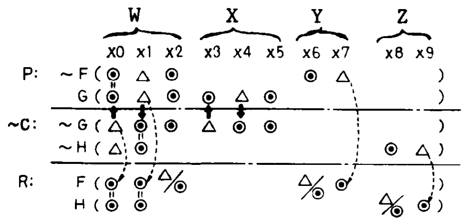
4.2. Retrieval of relevant information

Refering to Fig.1, in particular, considering the conditions of unifiability of G, the following conclusions are derived. (1) As the quantifiers of the corresponding variables in P and C, only those pairs listed in Table 1 are allowed. (2) For every pair of variables, say (x_k, x_l), such a case as $\forall x_k$ precedes $\exists x_l$ in the formula P while $\exists x_l$ precedes $\forall x_k$ in the formula C is not allowed. We call such a case being in circuit condition. Test of this condition is not difficult as shown later. (3) The relations of domains of corresponding variables must be such as those shown in Table 1. This condition is proved later.

These results show the conditions of P to be chosen as the candidate for generating the replacement and also suggest the search algorithm of P ; because such P , if any, is linked to the upper node in the knowledge structure to the node X_{ci} which is the domain of the variable x_{ci} in C if x_{ci} is universally quantified(ref. Table 1).

4. 3. Replacement generation

Same as the case of the literal G, the conditions of unifiabilities of F and H are analyzed so that the literals of the replacement are decided. (4) The quantifier of each variable of R should be decided so that at least one of the corresponding variable in R and P (or $\sim C$) is a universally quantified variable (we say $\forall$ variable from now on). If either $\forall$ or $\exists$ is possible, $\exists$ is exclusively chosen according to the definition of replacement. (5) The order of the variables in the prefix of R should be decided so that each $\exists$-variable comes to the rear as possible as far as the circuit is not formed.



⊙ ; $\forall$-variable, △; $\exists$-variable
➔, --►; substitution

Fig.3 Unsatisfiability condition

In fact, linear ordering is too restrictive to define the replacement. Instead, ordering information is represented, for each $\forall$-variable x_k, as the set S_k^R of $\exists$-variables preceding it. Let S_k^C be defined for C the same as R and let S_k^P be the set of all $\forall$-variables preceding the $\exists$-variable x_k in P, then S_k^R is obtained from S_k^C and S_k^P as either $S_k^R = S_k^P$ or $S_k^R = S_k^C$ depending on either x_k having been $\exists$-variable or $\forall$-variable in P respectively. Then, $S^R = \{ S_k^R / x_k$ is $\forall$-variable$\}$ constitutes the complete order information of R. Moreover, using this notation, the conclusion (2) of the last section is represented as inhibition of $x_k \in S_l^R$ and $x_l \in S_k^R$ to concur.

	Q_{pi}	Q_{ci}	Q_{ri}	X_{ri}	CONDITION
$x_i \in W$	$\forall$	$\forall$	$\forall$	X_{ci}	$X_{pi} \supset X_{ci}$
	$\forall$	$\exists$	$\exists$	$X_{pi} \wedge X_{ci}$	$X_{pi} \wedge X_{ci} \neq \phi$
	$\exists$	$\exists$	$\forall$	X_{pi}	$X_{pi} \subset X_{ci}$
$x_i \in X$	$\forall$	$\forall$	–	(X_{ci}	$X_{pi} \wedge X_{ci}$
	$\forall$	$\exists$	–	$X_{pi} \supset X_{ci}$	$X_{pi} \wedge X_{ci} \neq \phi$
	$\exists$	$\exists$	–	X_{pi})	$X_{pi} \subset X_{ci}$
$x_i \in Y$	$\forall$	–	$\exists$	X_{pi}	
	$\exists$	–	$\forall$	X_{pi}	
$x_i \in Z$	$\forall$	–	$\forall$	X_{ci}	
	–	$\exists$	$\exists$	X_{ci}	

X_{*i} ; The domain of x_i in P/C/R
Q_{*i} ; The quantifier of x_i in P/C/R
$*$; P/C/R

Table 1. Domains and quantifieres of the variables in R

(6) Finally, the domain of each variable is obtained by rewriting $P \cap R \cap \sim C$, by an ordinary set of clauses and by deciding the relation of the domains so as to meet the requirement for unsatisfiability as well as the definition of the replacement. The final results are shown together in Table 1.

5. Deductive process

The deductive process is to: (i) evaluate C, (ii) select a literal in C, (iii) look for and retrieve P which satisfies the conditions (1) through (3) and (iv) generate R and replace C by R, then back to (i).

6. Conclusion

In this paper, we have discussed only the principle of the deductive processor. We must show that with this deductive algorithm we can expand the way of representing knowledge preserving the deductive ability. We will discuss it in the next paper.

In order to assure the completeness, a slight modification of the algorithm from what has been described is necessary. Because we do not have enough space, we will discuss it too in the next paper.

References

1) Ohsuga, S. :"A knowledge structure and a deductive inference rule based on it", Second USA-Japan Computer Conference, pp.195-199, 1975.

2) Robinson, J.A. :"A machine oriented logic based on the resolution principle", J. ACM Vol.12, pp.23-41.

3) Schwarcz, R.M., Burger, J F. and Simmons, R.F. :"A deductive logic for answering English question", Comm. ACM Vol.13, pp.167-183, 1970.

4) Schank, R.C. :"Finding the conceptual content and intention in an utterance in natural language conversation", Second Int'l Joint Conf. on Artificial Intelligence , pp.444-454, 1971.

5) Shapiro, S.C. :"A net structure for semantic information storage, deduction and retrieval", Second Int'l Joint Conf. on Artificial Intelligence, pp.512-523, 1971.

6) Simmons, R.F. and Bruce, B.C. :"Some relation between predicate calculus and semantic net representation of discourse", Second Int'l Joint Conf. on Artificial Intelligence , pp.524-530, 1971.