"QUESTION-ANSWER" - A WULTIPURPOSE INFORMATION SYSTEM

Tu.A.Buchstab and S.S.Kamynin
Institute of Applied Mathematics
of the USSR Academy of Sciences,
Moscow, USSR.

ABSTRACT

We describe a "QUESTION-ANSWER" information system implemented on computer BESM-6 in the time-shared system. The "QUESTION-ANSWER" system is capable of deducing facts that have not been explicitly given to it by using a large data base and interpreting some of data as rules of inference. The system employs a special procedure which allows not to use contradictory information even if it is contained in the data base.

1. INTRODUCTION

The "<3JESTI0N-ANSWER" system developed at the Institute of Applied Mathematics of the USSR Academy of Sciences is capable of accepting various information and, by using such information, it can answer questions posed to it. New information received by the system broadens the stock of its "knowledge" and, hence, widens the scope of questions the system can answer. The "QUESTION-ANSWER" system is multipurpose, since the field of its application is not fixed beforehand, being determined by the contents of the information introduced into the system. When answering questions, the system carries out logical analysis of the information available to it; in the course of such analysis the system deduces new facts that are not explicitly given to it. Thus, the system not only searches the asked-for data among the particular facts known to it, but also carries out deductive inference interpreting some of the received information as rules of inference.

When developing the present system, we have taken into account the experience gained in the creation of other deductive information systems such as "Advice Taker" of J.McCarthy (1): "SIR" of B.Raphael (2); "SQA" of F.Black (3); "DEDUCOM" of J.Slagle (4); "QA3" of C.Green (5); "ROF" of R.Levien and .Maron (6); etc. Evidently, the main factors which determine the possibility of practical application of a deductive information system are the volume of the data base the system can effectively use and the high-speed of the programs that realize logical inference. In this connection, when developing the "QUESTION-ANSWER" system, we tried that it should combine the possibility of accepting a large bulk of factual data with an ability of an effective carrying out of the logical analysis of these data, so as to minimize the time needed for answering questions that require deduction at the "common sense" level.

It should be noted that in case of a large data base there arises a problem associated with that information introduced into the system may prove to be contradictory. The "QUESTIOBf-ANSWER" system employe a special procedure which, on the one hand, makes it possible to remove during the input a considerable portion of the information which comes in contradiction with the "knowledgeof the system and, on the other hand, allows the system, when answering questions, not to use contradictory information even if it is contained in the data base.

The creation of a multipurpose information system employing deductive inference involves the following main problems:

1. Choice of a language for presenting information and posing questions.
2. Realization of logical inference, i.e. the choice of methods which would allow carrying out effective logical analysis of the information known to the system with a view to obtaining requested data.
3. Organization of the data base. 1. e. the choice of methods of presentation and arrangement of information in machine memory, that would ensure rapid information retrieval.

2. LANGUAGE FOB COMMUNICATION WITH THE

In our choice of a communication language we were guided by two principles, the first being mari mum simplicity of the language adopted for communication with the system and the second, its sufficient flexibility for the system to demonstrate good "understanding" in most diverse applications. Usually attempts are made that the language for external presentation of information should be close to a natural language, with language of predicate calculus or property lists serving as the basis. Evidently, the use of property lists for presenting information is sufficiently convenient for man—system communication, since such a presentation can be regarded as a simplest model of a natural language. On the other hand, presentation of input information with the aid of property lists allows easy creation of a model that will provide faster question answering than with the use of the theorem-proving procedure.

The basic language structure of the "QUESTION-ANSWER" system for forming statements and questions is a triple <.attribute> object> <value> (<a> <o> <•> for short). Such a notation means that an attribute <a> of an object with the name <o> has the value <v> • For instance, the fact that the colour of a car is blue can be expressed by the following triple: COLOUR CAR BLUE. An essential feature of such presentation of information is that in the description of properties of an object by means of attribute-value pairs we use "words" whose meaning is defined, generally speaking, within the framework of a natural language. Properties are ascribed to an object which is defined by an individual name. An individual name can also be used as a value of a certain attribute, and therefore descriptive structures of any degree of complexity can be created. Though words of a natural language can be used as individual names of objects, these words must be maximum unique in the sense that passing over from one object to another should always cause a change of the name. In this connection the use of special standard names of objects is allowed, these standard names being words of the form C1,C2,C3.... • On request of the user the system delivers a current free standard name and also inhibits the use of the name whose number exceeds the current value.

Besides triples proper, the language also permits the use of negations, i.e. structures of the form FALSE <triple>. Thus, in the description of particular information, employed as statements are triples, negations, and also composite statements having the form P1 AND P2 (where P1 is a triple or a negation and P2 is a triple or a negation or a composite statement)•

For the system to be able to execute deductive inference proceeding from particular facts known to it, it is necessary to specify rules of inference. In the present system the rules of inferexice have the form of conditional statements! IF <condition> THEN <consequence> • Triples describing the condition and the consequence besides words contain variables of the form X1,X2,X3,.»* • For example: IF SMALLER X1 12 AND EQUAL X2 X3 *TBM* SMALLER X1 X3.

Hence, the "knowledge" of the system can be defined as a set of facts provable within the framework of the following formal theory.

1. Atoms: words, i.e. strings of letters and/or numerals.

2. Elementary propositions: <axo> <v> or FALSE <axo><v> (where <a> , <o> , <v> are atoms).

3. Axioms; a set of particular elementary propositions known to the system.

4. Rules of inference: methods of combining elementary propositions, that generate new elementary propositions. The rules of inference have the form: IF A1 AND A2 AND...AND Am THEN B1 AND B2 AND... AND Bn, where Ai. and Bj(1<i<m, 1<j<n) are elementary propositions extended by attaching variables X1,X2,X3... as additional atoms.

Elementary questions that can be posed for the system to answer have the form <a> <o> <v> ? and FAIfiB <a><o>or>? The system can also be asked a composite question consisting of several elementary questions, e.g. COLOUR CAR BLUE AND SIZE CAR LARGE? . Possible answers of the system to these questions are "Tea", "No" and "UNKNOWN". Moreover, the system can be asked questions which require enumeration of answers. Triples in such questions contain variables of the form K1,K2,K3,«- • When answering questions that require enumeration of answers, the system selects appropriate values of the variables and thus forms a set of answers. Naturally, a question requiring enumeration of answers may be composite, and in this case the values selected for the same variables entering in different triples are matched As the answer to a question requiring enumeration of answers the system delivers a sequence of records in the question format with the values substituted for the variables. If no appropriate values can be found for the variables entering into the question, the answer to such a question will be "Unknown".

It should be pointed out that when a certain particular statement is communicated to the system, this statement is not entered at once into the data base, but the system itself preliminarily substitutes it by a question. If the answer to such question is "Tes" or "No", the system does not accept this fact and answers, respectively, either;"Known" or "Statement contradicts to fact known to system". The system also checks the incoming new rules of Inference for novelty and consistency with the "knowledge" it has at the moment. To this end, the system first of all checks the deducibility of the consequent of a given rule from the conditions determined by its antecedent, using the already known rules of inference. The system also checks whether any particular facts contradictory to the rule of inference being checked are deductible. Thus, the deductive possibilities of the system are used for data input check

Besides statements and questions, means of communication with the system are furnished by prescriptions. The use of prescriptions makes it possible to obtain additional information, change the state of the data base, etc. Particularly, the user can require that the system should present the proof of the answer it gave to the preceding question. He can also require that the system should "forget" a certain particular statement or rule of Inference.

3.ORGANIZATION OF THE SYSTEM

All words entering into every incoming message are encoded by using a special dictionary. Therefore the internal presentation of triples has a simple fixed structure, so that such structures can be effectively searched for and compared.

When answering questions, the system carries out deductive inference by comparing questions with the consequents of the rules of inference and processes subquestions obtained from corresponding antecedents in the same manner as the questions. First the very tree of inference is constructed, with due accovint of cycles that may originate, and measures are taken that elimination of such cycles should not cause loss of answers. Then the constructed tree is used by the search procedure of "depth-first" type to obtain answers. Preliminary constructing of the inference tree offers a possibility to ignore non-resultant branches in proof presentation.

As has been noted above, when deducing answers, the system does not employ contradictory Information. This is ensured by storing the current situation after the obtaining of answers to each sub question and by making an attempt to obtain answers to the negation of a given sub question. If answers to the negation of the subquestion are obtained, contradictory answers are excluded from the list of answers to the subquestion proper. The revealed contradictions are stored and can be delivered on request of the user who, together with the answer to the question, then receives a message about the presence of contradictions in the data base.

One of the main factors determining the system effeciency is the organization of information files. The "CJJESTION-ANSWER" system employs various information structures, namely, lists, trees, mixed tables, ordered files. Similarly to the case of the "RDF" system, the data base in our system consists of three files, each of them being ordered with respect to one of the triple positions. This allows rapid readout of the necessary data. To minimize data recording time, the memory is made two-staged. The incoming information is first accumulated in a special (relatively small) buffer and, after the buffer is filled, this information is merged with each of the three files.

4-,EXAMPLE

The example given below is a record of a dialogue actually conducted with the "QUESTION-ANSWER" system with the aid of a time-sharing terminal on the computer BESM-6. Naturally, this simple example which is purely illustrative in character demonstrates only some potentialities of the system that is capable of solving far more complicated, problems.

In the operation of the "SJJESTIOH-ANSWER" system the answers of the machine are printed with indentions.

```
        15.10.74          17.52.
           AT YOUR SERVICE OS IPM
  YOUR NAME:      KB
FOR WORK WITH PROBLEM ENTER WORD "GIVE"
AND NAME OF PROBLEM
GIVE ANSWER
WAIT
   START WORK
SPECIALTY ALEXANDR MATHEMATICIAN.
   UNDEFINED INFORMATION FILE
NEW:
   CLEAR
SPECIALTY ALEXANDR MATHEMATICIAN.
   CLEAR
IF SPECIALTY X1 MATHEMATICIAN THEN KNOWS
X1 FORTRAN.
   CLEAR
KNOWS ALEXANDR FORTRAN?
   YES
IF KNOWS X1 X2 AND IS X2 ALGORITHMIC-LAN
GUAGE THEN CAN X1 PROGRAM.
   CLEAR
CAN ALEXANDR PROGRAM?
   UNKNOWN
IS FORTRAN ALGORITHMIC-LANGUAGE.
   CLEAR
CAN ALEXANDR PROGRAM?
   YES
PRESENT PROOF:
   (1) AXIOMS:
   SPECIALTY ALEXANDR MATHEMATICIAN
   (2) ACCORDING TO STATEMENT:
   IF SPECIALTY X1 MATHEMATICIAN THEN
   KNOWS X1 FORTRAN
   FROM (1)
   KNOWS ALEXANDR FORTRAN
   (3) AXIOMS:
   IS FORTRAN ALGORITHMIC-LANGUAGE
   4 ACCORDING TO STATEMENT:
   IF KNOWS X1 X2 AND
   IS X2 ALGORITHMIC-LANGUAGE THEN
   CAN X1 PROGRAM
   FROM (2),(3)
   CAN ALEXANDR PROGRAM
   ALL
IF SPECIALTY X1 MATHEMATICIAN THEN CAN
X1 PROGRAM.
   KNOWN
PRESENT PROOF:
   LET:
   SPECIALTY C1 MATHEMATICIAN
   (1) AXIOMS:
   SPECIALTY C1 MATHEMATICIAN
   (2) ACCORDING TO STATEMENT:
   IF SPECIALTY X1 MATHEMATICIAN THEN
   KNOWS X1 FORTRAN
   FROM (1)
   KNOWS C1 FORTRAN
   (3) AXIOMS:
   IS FORTRAN ALGORITHMIC-LANGUAGE
   4 ACCORDING TO STATEMENT:
   IF KNOWS X1 X2 AND
   IS X2 ALGORITHMIC-LANGUAGE THEN
   CAN X1 PROGRAM
   FROM (2),(3)
```

```
   CAN C1 PROGRAM
   ALL
FALSE KNOWS ALEXANDR FORTRAN.
   STATEMENT CONTRADICTS FACTS KNOWN
   TO SYSTEM
 K1 ALEXANDR K2?.
   CAN ALEXANDR PROGRAM
   KNOWS ALEXANDR FORTRAN
   SPECIALTY ALEXANDR MATHEMATICIAN
CO-WORKER ALEXANDR C100.
   STANDARD NAME C100 NOT DEMANDED
PRESENT NAME:
   C63
CO-WORKER ALEXANDR C63.
   CLEAR
NAME C63 K1?
   UNFAMILIAR WORD:   NAME
NAME C63 VLADIMIR AND FALSE KNOWS C63
FORTRAN.
   CLEAR
IF CO-WORKER ALEXANDR X1 THEN KNOWS
X1 FORTRAN.
   STATEMENT CONTRADICTS FACTS KNOWN
   TO SYSTEM
PRESENT PROOF:
   1- AXIOMS:
   CO-WORKER ALEXANDR C63
   ON THE OTHER HAND:
   1 AXIOMS:
   FALSE KNOWS C63 FORTRAN
   ALL
IF CO-WORKER ALEXANDR X1 AND X2 ALEXANDR
X3 THEN X2 X1 X3.
   CLEAR
K1 C63 K2?
   CAN C63 PROGRAM
   CO-WORKER C63 C63
   NAME C63 VLADIMIR
   SPECIALTY C63 MATHEMATICIAN
   (CONTRADICTIONS DETECTED WHEN DEDUCIN
   ANSWER)
PRESENT CONTRADICTIONS:
   CONTRADICTIONS:
   KNOWS C63 FORTRAN
PROVE INCONSISTENCY KNOWS C63 FORTRAN:
   (1) AXIOMS:
   CO-WORKER ALEXANDR C63
   (2) AXIOMS:
   SPECIALTY ALEXANDR MATHEMATICIAN
   (3) ACCORDING TO STATEMENT:
   IF SPECIALTY X1 MATHEMATICIAN THEN
   KNOWS X1 FORTRAN
   FROM (2)
   KNOWS ALEXANDR FORTRAN
   4 ACCORDING TO STATEMENT:
   IF CO-WORKER ALEXANDR X1 AND
   X2 ALEXANDR X3 THEN X2 X1 X3
   FROM (1),(3)
   KNOWS C63 FORTRAN
   ON THE OTHER HAND:
   1 AXIOMS:
   FALSE KNOWS C63 FORTRAN
   ALL
FORGET FALSE KNOWS C63 FORTRAN:
   CLEAR
SPECIALTY K1 K2 AND KNOWS K1 K3?
   SPECIALTY ALEXANDR MATHEMATICIAN AND
   KNOWS ALEXANDR FORTRAN
   SPECIALTY C63 MATHEMATICIAN AND
   KNOWS C63 FORTRAN
END
YOU FINISHED WORK AT 18.04.GOODBYE.
```

5 CONCLUSIONS

The "9JESTI0N-ANSWKR" system is written In the autocode of the computer HESM-6. The program contains about 9,000 instructions. By now the system has been tested In solving a number of problems (particularly such simple but In a certain sense "reference" problems as "Monkey and Bananas", "Mikado" etc.). The system answers questions quite rapidly. Thus, for example, the solution of the "Monkey and Bananas" problem takes about 0.01 second of the processor time. During the tests the data base of the system could contain more than 4,000 statements. Principally the volume of the data base Is determined by the capacity of the disc memory and can be substantially increased. Our nearest plans for the development of the system include a certain extension of the language to communicate with it, as well as the uee of special heuristics with a view to further improving its work.

We believe that systems of the "QUESTION-ANSWER" type may prove useful not only for solving information retrieval problems, but may also constitute an important component in a number of other systems being developed within the framework of the "artificial intelligence" project, such as systems for understanding natural language, analysis of scenes, planning of actions, etc.

RERENCES

1. J.McCarthy, Programs with common sense, Symp. Mechanisation of Thought Processes, Teddington, Nat.Phye. Lab., 1958.
2. B.Raphael, A computer program which "understands", Proc. AFIPS, Fall Joint Computer Conference, vol.26. 1964.
3. F.Black, A deductive question-answering system, Semantic Information Processing, MIT Press, Cambridge. Mass., 1968.
4. J.R.Slagle, Experiments with a deductive question-answering program, Communications of the ACM, vol. 8, No. 12, 1965.
5. C.Green, Theorem-proving by resolution as a basis for question-answering systems, Machine Intelligence, vol. 4, Edinburg Univ. Press, 1969.
6. R.E.Levien, M.E.Maron, A computer system for inference execution and data retrieval, Communications of the ACM, vol. 10, No 11, 1967.