A cost effective question-asking strategy for Horn clause systems

Jinchang Wang[a,*] and Evangelos Triantaphyllou[b,**]

[a]*Department of Business and Economics, Missouri Western State College, St. Joseph, MO 64507-2294, USA*
E-mail: wang@acad.mwsc.edu

[b]*Department of Industrial and Manufacturing Systems Engineering, Louisiana State University, 3134C CEBA Building, Baton Rouge, LA 70803-6409, USA*
E-mail: ietrian@lsuvm.sncc.lsu.edu

In many applications of knowledge based systems, the initial data are insufficient to fulfill inference. In that case, knowledge based systems ask users questions in order to acquire more information. When and what to ask is determined by a question-asking strategy. This paper deals with question-asking strategies for a Horn system in which the response costs of the questions and the probablistic estimates of the answers are given. We introduce a question sequencing rule and enhance an efficient question-asking strategy. Our computational experiments show that the proposed question-asking strategy is very effective.

1. Introduction

With the development of intelligent information systems, knowledge bases have found numerous applications in decision support, data manipulation, process control and expert systems. A knowledge base in which knowledge is represented in terms of "*if... then...*" statements is called a *rule base*. Given a set of initial data about a particular situation, a rule-based system makes inference by using its stored rules. When the initial data are insufficient, the rule based system may ask the user for additional information. A typical example is a medical diagnostic expert system in which the system, like a human physician, needs to know the patient's symptoms and/or results of various medical tests. A strategy of identifying and sequencing the questions to be asked when initial data are not sufficient to reach the final conclusion is called a *question-asking (or Q-A) strategy.*

A good Q-A strategy should quickly generate a couple of key questions, which lead to a final conclusion by spending a small total cost. On the other hand, a poor Q-A strategy selects irrelevant and expensive questions, which would retard the

* Supported in part by the Office of Naval Research grant N00014-94-1-0355.
** Supported in part by the Office of Naval Research grant N00014-95-0639.

inference process and build up unnecessary costs to the end user. Q-A strategies play an important role in many applications of knowledge based systems, such as expert systems for consulting, diagnostics, or training.

The issue of developing an effective and efficient Q-A strategy has been considered by researchers in building expert systems. The system EXPERT [7] uses pre-ordered lists of rules and questions. The system PROSPECT [6] uses a scoring function for selecting questions. The "*Alpha-Beta*" pruning strategy was introduced by Mellish [11] for acyclic inference nets. Wang and Vande Vate [13] proved that an effective strategy, which optimally selects the next question in order to minimize the *total number of questions*, is never efficient, even in Horn clause systems. They also developed an efficient strategy, called the minimum usage set (or MUS) strategy which approximates an effective Q-A strategy.

This paper first introduces an optimal question sequencing rule, for selecting the next question from a set of candidate questions towards a conclusion. This rule is then used to enhance the MUS strategy for Horn clause knowledge bases so that the final conclusion could be reached at a low cost. Computational experiments on randomly generated problems show that the new strategy provides an effective guide to inference when the given data are not adequate to reach the final conclusion.

Horn systems are a special class of a logical system which is used widely in practice. PROLOG, a popular logic programming language, relies on Horn clauses. The most distinguishing characteristic of Horn systems is that inference, which is computationally hard in general logical systems, can be performed in linear time in a propositional Horn system [5].

De Kleer [4] and Williams [3] studied the problem of diagnosing faults in electrical circuits. The behavior discrepancy of an electrical circuit between the observed and predicted performances indicates malfunctioning of the circuit. When behavior discrepancies occur, one wants to identify what causes these discrepancies (malfunctions). De Kleer et al. developed an entropy technique for minimizing the number of measurements needed to detect the faulty components. Although the entropy technique can be used to test the integrity (validity) of a Horn clause knowledge base, it does not work for "question-selection" which is discussed in this paper. In the problem of "question-selection", all the rules in a knowledge base are *assumed* to be functioning perfectly well, i.e., no behavior discrepancy exists, and the objective is to reach a final conclusion at a low cost.

Section 2 introduces the fundamental definitions and concepts for Horn systems and Q-A strategies. Section 3 states and proves two propositions related to logic circuits, which are used in determining an optimal question sequencing. Section 4 describes the proposed question-asking strategy for a Horn system in which we know the *response costs* and the *probability* of having a positive response for each unconfirmed question. The proposed strategy is compared in our computational experiments with a set of other strategies. The experiment results are presented in section 5.

2. Horn systems and Q-A strategies

A *literal* is either a propositional assertion A (a positive literal) or its negation $\neg A$ (a negative literal). A *Horn clause* is a disjunction of literals of which at most one is positive. For instance, $Q \vee \neg A_1 \vee \neg A_2 \vee \cdots \vee \neg A_s$ is a Horn clause. A *Horn system* is a conjunction of Horn clauses.

A Horn clause $Q \vee \neg A_1 \vee \neg A_2 \vee \cdots \vee \neg A_s$ can also be written as an implication $Q \leftarrow A_1 \wedge A_2 \wedge \cdots \wedge A_s$, where Q is called the *head* (or *conclusion*) and $A_1 \wedge A_2 \wedge \cdots \wedge A_s$ is called the *body* (or *premises*). Therefore, a Horn clause represents an "*if... then...*" rule in which a set of positive assertions implies at most one positive assertion.

For the convenience of presentation we define the following notation for representing a Horn system. The notation was used for the first time in [8]. Suppose that there are n assertions in a Horn system. Let $N = \{1, 2, \ldots, n\}$ be the assertion index set. Let $D(i)$ be the set of natural numbers indexing the clauses which have as head the assertion A_i. A *fact* can be viewed as a special clause which has one positive conclusion without any premises. We use a pair of indices (i, d) to identify a clause (i.e., an "*if... then...*" rule), where (i, d) refers to the dth clause which has assertion A_i as its conclusion. The clause $(i, 0)$ is reserved for the fact about A_i, $i \in N$. Let $I(i, d)$ be the set of natural numbers indexing the premises of the clause (i, d). By using the previous notation, a Horn system can be written as follows:

$$A_i \leftarrow \bigwedge_{j \in I(i,d)} A_j, \quad \text{for } i \in N,\ d \in D(i),\ \text{where } D(i) \neq \emptyset. \tag{1}$$

We may construct a *directed hypergraph for a Horn system*, as follows. For each assertion there is an assertion-node. For each clause there is a clause-node. There is an arc from an assertion-node to a clause-node exactly if the assertion is a premise of the clause. There is an arc from a clause-node to an assertion-node exactly if the assertion is the conclusion of the clause.

An assertion A_i in a Horn system H is *proved* (*forced*) *to be true* if H is consistent but $H \wedge \neg A_i$ is inconsistent. An assertion A_i can be proved to be true in either of the two cases: (1) "$A_i =$ *true*" is given as a fact; (2) "$A_i =$ *True*" is deduced, i.e., when all the premises in a clause (i, d) which has A_i as its conclusion are proved to be true.

The *selection of an assertion* A_i, denoted as $\delta(i)$ (where $\delta(i) \in D(i)$), indicates that the rule used to prove A_i is $(i, \delta(i))$. For any $k \in N$, we recursively define a Horn subsystem $H^\delta(k)$ corresponding to the selection δ, as follows:

$$A_k \in H^\delta(k);$$

$$\text{If } A_j \in H^\delta(k), \text{ then rule}(j, \delta(j)) \in H^\delta(k);$$

$$\text{If rule}(i, \delta(i)) \in H^\delta(k), \text{ then } A_j \in H^\delta(k) \text{ for each } A_j \in I(i, \delta(i)).$$

$H^\delta(k)$ is also called a *potential proof* of assertion A_k.

We say that the clause (i, d) is an immediate cause of its conclusion A_i and that the assertion A_i is an immediate result of each clause (i, d) concluding it. Furthermore, the clause (i, d) is an immediate result of each assertion A_j in its body, and the assertion A_j is an immediate cause of each clause (i, d) including A_j in its body. Transitive applications of the relations "*is an immediate cause of*" and "*is an immediate result of*" lead to the relations "*is a cause of*" and "*is a result of*" in the natural way. If assertion A_s is a cause of A_t with k times transitive applications of the relation "is an immediate cause of", then we say that there are *k levels between A_s and A_t*. A Horn clause system is *acyclic* if no assertion is a result of itself, otherwise it is *cyclic*.

If an assertion occurs in a Horn system only as the conclusion of some clauses and never as a premise in any clause, it is called a *top level conclusion*. In most of cases, top level conclusions reflect the final conclusions the system is aimed at. For example, possible diseases are top level conclusions in a knowledge base for a disease diagnostic system. An assertion can be viewed as a *binary variable* with value either "*true*" or "*false*". If the value of an assertion is known, the assertion is called *confirmed*, otherwise it is called *unconfirmed.*

An assertion is a *first cause* if it is not a result of any other assertion. If the value of an assertion can be provided by the user, then the assertion is called an *observable assertion*. If an observable assertion is not yet confirmed, it is called an *unconfirmed observable assertion* (*UOA*). The cost of confirmation of a UOA is called the *response cost* of that UOA.

Let A_j be a potential top level conclusion. A set of unconfirmed observable assertions is an *unconfirmed observable set* (*or UOA set*) *of A_j* exactly if the assertion A_j can be proved true when all the assertions in that set have values "*true*", and if any one of those assertions were false then A_j could not be concluded from the other assertions of that set. In other words, a UOA set of A_j is a minimal set of observable assertions which may prove the truth of A_j. It is common for an assertion A_j to have many UOA sets. Among all the UOA sets of A_j, the UOA set with a lowest total response cost is called the *economical inquiry set of A_j*. Among all the UOA sets of all top level conclusions, the UOA set with a lowest total response cost is called the *global economical inquiry set*.

Forward chaining and backward chaining are two frequently used inference methods. *Forward chaining* begins with the facts already provided by the user. It proceeds by iteratively applying rules to the confirmed facts to obtain new facts. These new facts are in turn used to generate more new facts. This process continues until one of the top level conclusions is reached. On the other hand, *backward chaining* begins with taking a pre-specified top level conclusion as the goal to prove. It checks which rules could be used to prove this conclusion. The premises of those rules become the new goals which are called subgoals. This process continues, working backwards through successive subgoals, until the user confirmed assertions (i.e., initial facts) are reached. If the initial facts cannot lead to the pre-specified top level conclusion, then another top level conclusion is chosen as the goal and the proof of the new goal is pursued again. Although forward and backward chaining are not in general complete (i.e., being able to identify every implication of a given set of clauses), they are complete for Horn clause systems.

In general, a Q-A strategy is composed of two steps [7]: goal selection and question selection. In the *goal selection step*, the strategy chooses a potential top level conclusion to pursue. In the *question selection step*, the strategy selects a question which would help to reach the potential top level conclusion being pursued.

A criterion for goal selection is the *expected* total response costs of UOA sets. The top level conclusion associated with the global economical inquiry set is a good choice of a potential conclusion to pursue. As a matter of fact, the global economical inquiry set gives a lower bound of the total cost required to reach a top level conclusion. If all the responses for the assertions in that set happen to be positive, then the associated top level conclusion is proved at the lowest possible cost. However, if the response to one of these questions is negative, then the global economical inquiry set is no longer useful and one has to choose another UOA set. Applying the concept of UOA sets in goal selection has an analogue in human inference: When a human being feels that the given information is not enough, he/she would first choose a possible conclusion which might be confirmed easily.

Once a potential conclusion and associated UOA set is selected, one has to decide which question should be asked first. In the next section, we prove a question selection rule for the situation when a UOA set has already been chosen.

3. Sequencing rules for AND and OR gates

AND and OR gates are concepts in the design of electronic logic circuits [9]. This section describes two theorems regarding the order of checking inputs when determining the output of an AND or an OR gate. These theorems can be directly applied into question selection problems.

A "*gate*" in logic circuits is an electronic component with inputs and outputs. AND and OR gates have many inputs but only one output. The status of the output depends on the status of the inputs. The gates are supposed to operate at two distinct electrical levels which are represented by binary 1 and binary 0. For example, the status of voltage lower than 5 V and status of voltage higher than 5 V are two electrical levels, which may be represented as binary 0 and 1, respectively.

An *AND gate* corresponds to the logic operation AND. That is, if all the inputs are binary 1, then the output is binary 1, otherwise the output is binary 0. Let AND$[(I_1, I_2, \dots, I_u), X]$ denote an AND gate which has u inputs: $I_1, I_2, \dots, I_u$, and one output X.

An *OR gate* corresponds to the logic operation OR. That is, if any of the inputs is binary 1, then the output is binary 1, otherwise the output is binary 0. Let OR$[(I_1, I_2, \dots, I_u), X]$ denote an OR gate which has u inputs: $I_1, I_2, \dots I_u$, and one output X.

Let c_i denote the *cost* of checking the electrical status of input I_i, and p_i the likelihood (or probability) that the status of the assertion A_i is binary 1. Let the *ordered* set $\{s_1, s_2, \ldots, s_u\}$ denote a *checking sequence* of the input values of an AND gate AND$[(I_1, I_2, \ldots, I_u), X]$ or an OR gate OR$[(I_1, I_2, \ldots, I_u), X]$, where $s_1, s_2, \ldots, s_u$ are the subscripts of the inputs. For instance, the ordered set $\{1, 2, \ldots, u\}$ represents a sequence in which I_1 is checked first, then $I_2, I_3, \ldots, I_u$.

We define the *CP ratio* (standing for: Cost-Probability ratio) of an input I_k as the value of the ratio: c_k/p_k. Similarly, the *CNP ratio* (standing for: Cost-Null-Probability ratio) of I_k is the value of the ratio: $c_k/(1-p_k)$. The CP ratio of I_k may be interpreted as the average cost of obtaining a value 1 on I_k. The CNP ratio of I_k may be interpreted as the average cost of obtaining a value 0 on I_k.

The testing decision rules in [1, 2] can be directly used as rules of examining a checking sequence for an AND gate and an OR gate so that the expected checking costs are minimized. These checking sequence rules are next described in Result 3.1 and Result 3.2.

Result 3.1. Let c_i be the cost of determining the value of input I_i of an AND gate AND$[(I_1, I_2, \ldots, I_u), X]$, and p_i be the probability for I_i to have value 1. When determining the value of output X, a checking sequence $S = \{s_1, s_2, \ldots, s_u\}$ yields the minimum expected cost if and only if the following condition is satisfied:

$$\frac{c_{s_k}}{1-p_{s_k}} \leqslant \frac{c_{s_{k+1}}}{1-p_{s_{k+1}}}, \quad \text{for } k = 1, 2, \ldots, u-1. \tag{2}$$

Result 3.1 indicates that, in order to minimize the total expected cost of determining the output value of an AND gate, one should check the inputs of the AND gate according to their CNP ratios. That is, the element with a smallest CNP ratio should be checked first. Since the CNP ratio of input I_k can be interpreted as the average cost of obtaining a value 0 at input I_k, Result 3.1 indicates that we should start the input checking with the input which has the smallest average cost of obtaining a value "*false*". If all the inputs have an identical checking cost, then one should first check the input which is most likely to be false (see also [12]).

Result 3.2. Let c_i be the cost of determining the binary value of input I_i of an OR gate OR$[(I_1, I_2, \ldots, I_u), X)]$, p_i be the probability for I_i to have a binary value 1. Then, when determining the value of output X, a checking sequence $S = \{s_1, s_2, \ldots, s_u\}$ yields the minimum expected cost if and only if the following condition is true:

$$\frac{c_{s_k}}{p_{s_k}} \leqslant \frac{c_{s_{k+1}}}{p_{s_{k+1}}}, \quad \text{for any } k = 1, 2, \ldots, u-1. \tag{3}$$

According to Result 3.2, the input checking should start with the input which has the smallest CP-ratio in an OR gate. Since the CP-ratio of input I_k can be interpreted as the average cost of obtaining a value 1 at input I_k, Result 3.2 indicates that we should start input checking with the input with the smallest average cost of obtaining a value "*true*".

In order to establish the correlation between a Horn system and AND/OR gates, let us consider two status of an assertion A_i in a Horn system, as follows:

Status One: A_i is forced to be true. That is, A_i must be true, otherwise logical contradiction would occur. For instance, A_i is given to be true; or A_i is proved to be true if all the premises of a rule (i, d) are forced to be true.
Status Two: A_i is not forced to be true. That is, A_i can be either true or false.

If one takes Status One as binary 1 and Status Two as binary 0, then a Horn rule (i, d) can be viewed as an AND gate in which an input corresponds to a premise and the output corresponds to conclusion A_i. The conclusion A_i must be true if and only if each premise is true. An assertion which can be proved by several rules may be taken as an OR gate in which the rules form inputs and the assertion is the output. The assertion can be proved to be true by any of those rules.

Similarly, a UOA set and the corresponding top level conclusion form an AND gate, in which the top level conclusion is proved to be true if the responses for all the unconfirmed observable assertions in the UOA set are positive. In this way Result 3.1 can be applied directly as a question selection rule *after* a UOA set is selected. This is formally stated in the following corollary.

Corollary 3.1. Let $U = \{A_1, A_2, \ldots, A_u\}$ be a UOA set which corresponds to the top level conclusion A_x. Let c_i denote the cost of determining the value (true or false) of assertion A_i, and p_i the probability that the value is true (for $i = 1, 2, \ldots, u$). Then, when determining whether A_x can be proved by U, a questioning sequence $S = \{s_1, s_2, \ldots, s_u\}$ (where s_i, $i = 1, 2, \ldots, u$, are subscripts of the assertions in the UOA set) yields the minimum expected cost if and only if the following condition is true:

$$\frac{c_{s_k}}{1 - p_{s_k}} \leqslant \frac{c_{s_{k+1}}}{1 - p_{s_{k+1}}}, \quad \text{for any } k = 1, 2, \ldots, u - 1. \tag{4}$$

Note that if we associate logic value "true" with binary 1, "false" with binary 0, a Horn rule is no longer equivalent to an AND gate. That is because the conclusion of a Horn rule *can* be true even though some premises are false, but it never occurs that the output of an AND gate is 1 while some inputs are 0.

4. The Cost-Effective Question-Asking (CEQA) strategy

The Q-A strategy described in this section is designed for a Horn clause knowledge base in which each UOA (unconfirmed observable assertion) is associated with a response cost and a probability (or likelihood) of having a positive response. Uncertainty *is not assumed* in either responses or the rules.

The proposed Q-A strategy is composed of two steps. The first step chooses a possible top level conclusion and a corresponding UOA set by using the *LABELING* procedure (to be described next). The second step selects questions to ask from the chosen UOA set according to the sequencing rule described in Corollary 3.1. When a negative response is reached, the procedure *UPDATING* (to be described next) is used to prune off the rules which are no longer useful and the assertions which are un-provable.

In the LABELING procedure of the first step, we consider both the response cost and probability of proving an assertion to be true. We maintain two labels for each rule and each assertion: the *cost label C* and the *probability label P*. With the cost labels and probability labels, we may define the CP ratio and CNP ratio of an assertion or a rule in a manner similar to those for inputs of AND or OR gates, as described in section 3. That is, the *CP ratio of an assertion* (*or a rule*) is the ratio of the assertion's (or the rule's) cost label to its probability label; the *CNP ratio of an assertion* (*or a rule*) is the ratio of the assertion's (or the rule's) cost label to the difference between 1.00 and its probability label.

Assertions and rules (i.e., Horn clauses) are labeled as follows. For a rule, its cost label is equal to the sum of the cost labels of its premises, and its probability label is equal to the product of the probability labels of its premises. For an assertion A_i which can be proved by a couple of rules, we choose the rule which has a *smallest* CP ratio as the selection $\delta(i)$, and pass the cost and probability labels of the selected rule to the assertion A_i. In this way the labels are propagated until all rules and assertions are labeled or a cycle is encountered. To break a cycle, we calculate the CP ratios of the rules which are labeled but their heads are not labeled; select the rule (k, d^*) which has the minimum CP ratio, and label assertion A_k by using the label of the rule (i, d^*). The label of A_k obtained in this way will not be changed subsequently although A_k is in a cycle [13].

Next, a top level conclusion which has a minimum CP ratio is identified. By tracing back from the selected top level conclusion using the selections set up in the LABELING procedure, we find the UOA set which leads to the conclusion with the minimum CP ratio.

Once a UOA set is determined, questions are asked in the second step in the order of the optimal sequence stated in Corollary 3.1, until either all assertions in the UOA set are confirmed true or a negative response is received. In the former case, we are done; the top level conclusion associated with the UOA set is proved to be true. In the latter case, we update the Horn system by using the procedure UPDATING, and start another iteration (by choosing another UOA set and asking questions as above).

The purpose of using the procedure UPDATING is to simplify the Horn system by using the newly obtained responses from the user. Starting from the newly positively (or negatively) confirmed UOA's, we find other assertions which are confirmed positively (or negatively) now. When all the premises of a rule are true, then this rule is called *fired* and the conclusion of this rule is proved true.

When an assertion is known to be false or unprovable, it is called *defunct*, and it may cause other rules or assertions to become defunct as well. Formally, *defunct assertions and rules* are defined recursively as follows:

An observable assertion is defunct if it is given a value false;

A rule is defunct if at least one of its premises is defunct;

An non-observable assertion is defunct if all the rules which have the assertion as their conclusion, are defunct.

Defunct assertions and rules are no longer considered in the subsequent iterations. Moreover, a rule is called *active* if it is neither fired nor defunct. *An assertion is active* if its truth value is not yet known and it is not defunct.

The CEQA Strategy works for *regular* Horn systems which are defined as follows. *An assertion is regular* if it has a potential proof in which all the first causes (see its definition in section 2) are observable. *A Horn system is regular* if each assertion in the system is regular. In other words, a system is regular if each assertion may be inferred to be true from some truth assignments of the observable assertions. If a Horn system is not regular, then some assertions can never be proved to be true, regardless of the truth values of the observable assertions. A situation like this indicates that either the rules of the system contain a logical error, or we fail to correctly identify the observable assertions. The CEQA strategy, along with its related procedures, are formally described next.

Algorithm CEQA;

```
BEGIN {Algorithm CEQA};
  Step 1:
      Label the unconfirmed assertions and rules by using the LABELING procedure;
      Select a potential top level conclusion $A_c$ with a smallest CP ratio;
      Trace and find the UOA set, say $U$, which corresponds to $A_c$ and the smallest CP ratio;
  Step 2:
      Select a question $A_k$ to ask from the UOA set $U$, with $A_k$ having a smallest CNP ratio in $U$. That is:

          $\frac{C_k}{1-P_k} = \underset{i \in U}{\mathrm{MIN}} \left\{ \frac{C_i}{1-P_i} \right\}.$    (5)

      Get a response from the user about the value of $A_k$;
  Step 3:
      If the response is positive (i.e., the value of $A_k$ is true), then go to Step 2.
      Otherwise, update the rule system by using the procedure UPDATING and go to Step 1.
  Stopping Rules:
      Stop if a top level conclusion is confirmed true;
      Stop if all unconfirmed assertions in a selected UOA set $U$ are confirmed
           true. In this case the associated potential conclusion is proved;
      Stop if all top level conclusions are defunct. In this case no conclusion can
           be reached.
END {Algorithm CEQA};
```

```
Procedure LABELING;

BEGIN {LABELING};
  Step 1:
      Each unconfirmed observable assertion $A_i$, cost label $C_i$, and probability
      label $P_i$ are initialized as follows: $C_i = c_i$, and $P_i = p_i$.
  Step 2:
      Alternately use PROPAGATE and ASSIGN to label the assertions and
      clauses until no new labels are applied in either subroutine.
END {LABELING};

Procedure PROPAGATE;

BEGIN {PROPAGATE};
    While there is an unlabeled element $t$, in which all immediate causes have
    been labeled, label $t$ according to the following rule:
    If the element $t$ is a rule $(i,d)$ whose unconfirmed premises are all labeled,
    then use:
        $$C_{id} = \sum_{j \in I(i,d) C_j} C_j, \quad \text{and} \quad P_{id} = \prod_{j \in I(i,d)} P_j. \qquad (6)$$
    If the element $t$ is an unconfirmed assertion $A_i$, such that the rules with con-
    clusion $A_i$ are all labeled, then use:
        $$C_i = C_{i,\delta(i)}, \quad P_i = P_{i,\delta(i)},$$
    where: $\delta(i) \in D(i)$ and:
        $$\frac{C_{i,\delta(i)}}{P_{i,\delta(i)}} = \underset{d \in D(i)}{\mathrm{MIN}} \left\{ \frac{C_{id}}{P_{id}} \right\}. \qquad (7)$$
END {PROPAGATE};

Procedure ASSIGN;

BEGIN {ASSIGN};
    Among all labeled rules whose conclusions are not labeled, choose the one,
    say $(k, d^*)$, with a smallest CP ratio. That is:
        $$\frac{C_{kd^*}}{P_{kd^*}} = \underset{(i,d) \text{ labeled}, A_i \text{ unlabeled}}{\mathrm{MIN}} \left\{ \frac{C_{id}}{P_{id}} \right\}.$$
    Let: $C_k = C_{kd^*}$, $P_k = P_{kd^*}$, and $\delta(k) = d^*$.
END {ASSIGN};
```

```
Procedure UPDATING;

BEGIN {UPDATING};
        Find as many as possible assertions which are confirmed true, starting with the
        most recently confirmed observable assertions and applying forward chaining
        to proceed;
        Find as many as possible defunct assertions, starting with the observable as-
        sertion which was just confirmed false.
        NOTE: the updated rule base is composed by only active assertions and active
        rules.

END {UPDATING};
```

If the Horn system is acyclic, then the procedure LABELING runs in time linear to the number of occurrences of assertion, otherwise, LABELING takes log-linear time [13].

The cost label of a rule or an assertion indicates the so-called "*minimum pseudo-cost*" of using the rule or proving the assertion, rather than a real minimum cost. To prove an assertion, there exists a proof with a minimum cost. However, finding such a proof is NP-hard even in Horn clause systems [13]. In the LABELING procedure, the costs are propagated through a *myopic* calculation in which the cost of using a rule is the summation of the costs of its premises. The myopic calculation does not guarantee a minimum cost. For instance, in a two-clause Horn system: $\{A_1 \leftarrow A_2 \wedge A_3,\ A_2 \leftarrow A_3 \wedge A_4\}$, A_3 and A_4 are unconfirmed observable assertions and suppose that their confirming costs are equal to 1. Although the minimum cost to prove A_1 is 2, the procedure LABELING yields a cost 3, since it *myopically* labels A_1 as the summation of labels of A_2 and A_3, and misses the information that the cost of A_3 is already included in the label of A_2. Therefore, the cost labels are *approximations of minimum proving costs*, and thus they are called *pseudo-minimum-costs*.

The cost label of a rule (i, d), as calculated in (6), is the cost of proving the rule's conclusion A_i by using rule (i, d). Since a rule is an AND gate as discussed in section 3, A_i can be proved by using rule (i, d) only when all the premises of rule (i, d) are forced to be true. Therefore, if a rule (i, d) could be used to prove its conclusion A_i, then all its premises must have been checked and so the total cost would be the sum of its premises' costs. The cost label of an assertion A_i, as calculated in (7), is the cost of a relatively inexpensive potential proof of A_i. When the cost labels propagate, by using (6) and (7), to reach a top level conclusion A_t, then the cost label of A_t can be taken as a lower bound of the cost for proving A_t. In fact, cost labeling in procedure LABELING gives, for each top level conclusion, a lower bound of the proving cost.

Note that the cost label of rule (i, d), as defined in (6), is *not* the expected cost of determining the "provableness" of A_i by rule (i, d). A_i is provable by using rule (i, d) only when all the premises of rule (i, d) *must* be true. A_i is not provable (A_i can be either true or false) by using rule (i, d) if any of the premises is not necessarily true.

If one checks the premises in the sequences $S = \{s_1, s_2, \ldots, s_u\}$, then the expected cost, EC_{id}, of determining whether A_i' is provable by using rule (i, d) is

$$EC_{id} = C_{s_1} + \sum_{k=2}^{u} \left(C_{s_k} \prod_{j=1}^{k-1} p_{s_j} \right). \tag{8}$$

The above concept of expected cost of a rule was used to determine the checking sequence S, as in Result 3.1, Result 3.2, and Corollary 3.1 when a UOA set is given, but it is not appropriate to identify a UOA set towards *proving* a top level conclusion. In fact, (8) is the expected cost of *determining whether rule* (i, d) *can be used to prove* A_i, and it is not the cost of any proof of A_i. On the other hand, the cost label C_{id}, defined in (6), is the cost of *proving* A_i *by using rule* (i, d), and it is associated with a particular potential proof of A_i.

The example in Appendix I demonstrates an application of the CEQA strategy.

5. Some computational experiments

In this section we describe the computational experiments in which we compared the CEQA strategy with three other strategies and an "*ideal*" strategy which was used as a lower bound. The experiments were carried out on randomly generated Horn clause knowledge bases.

Although CEQA works for general Horn systems, the randomly generated Horn clause bases in our experiments were *all acyclic*. That was because the main focus of these experiments was on the effectiveness of the heuristic on the "pseudo-costs". When dealing with a cyclic system, the CEQA strategy uses the procedure ASSIGN to break cycles during the labeling process. Note that ASSIGN is *not a heuristic procedure*.

The three Q-A strategies, other than the CEQA strategy, in the experiments are more or less based on random question-selection. The "ideal" strategy, SUBIDEAL, provides a lower bound for the best result a Q-A strategy may achieve. However, the "ideal" strategy cannot be used in practical problems since it is computationally hard. These strategies are described in more detail in the following paragraphs.

Strategy *UOARND* selects an unconfirmed observable assertion *randomly*. This strategy does not have a goal selection step. It selects a question directly from the pool of the unconfirmed observable assertions. Forward chaining is used for deduction to see whether a top level conclusion is reached. If a top conclusion is not reached, then a question is selected and asked and forward chaining is applied again.

Strategy *SETRND* selects a UOA set randomly, then selects questions from that UOA set randomly. If the response for the question is positive, then the next question is still from the same UOA set. This continues until all the UOAs in the set are confirmed truth, which means that the top level conclusion is proved. If the response for a question is negative, then another UOA set is selected randomly.

Strategy *DEPTH1ST* applies backward chaining for deduction. In DEPTH1ST, a possible top level conclusion is randomly selected, from which tracing is going through the rules that may prove it following the "*depth first*" rule. When an unconfirmed observable assertion (UOA) is met during the tracing, its value is asked. The response to the question determines the branching to be performed next [10].

Strategy *SUBIDEAL* is an approximation of the best (or ideal) strategy and describes an *unrealistically fortunate* question sequencing situation. *Suppose* that we *knew* the actual values of the responses of all the questions in advance, but we still needed to ask the questions. In that (unrealistic) case we could simply identify a UOA set in which the UOA's have positive values and which has the *minimum* total response cost. The so-called *global economical inquiry set* (see also the definitions in section 2) is such a UOA set. Therefore, a Q-A strategy which finds the global economical inquiry set is an "*ideal*" strategy. However, since the responses are *not known* in advance in real life problems, the "*ideal*" strategy cannot be applied in real life problems.

The cost of the global economical inquiry set provides the total response cost to reach a top level conclusion under the most fortunate question sequencing situation. We may use the cost of the global economical inquiry set as a *lower bound* of the total response cost in our experiments. A difficulty to use the global economical inquiry set is that finding such a UOA set is an NP-hard problem, even when we unrealistically assume that all responses are known in advance. Therefore, we used the heuristic procedure introduced in [13] to approximate the global economical inquiry set searching.

Each Horn clause base generated for the experiments is called a *scenario*. As it was stated earlier, these rule bases were acyclic. The conclusions of the clauses (top level conclusions or intermediate conclusions) can be viewed as belonging to different levels. The highest such level is called the *level of the system*.

A Horn clause base (i.e., scenario) was generated randomly with the following parameters: number of rules (i.e., clauses), number of assertions, number of premises, number of top level conclusions, number of unconfirmed observable assertions, levels of the system, and distribution of response costs and probabilities of having a "*true*" value. Each observable assertion was assigned a response cost and a probability to be true. The real truth value of an observable assertion was assigned randomly according to its probability to be true. These probabilities were assumed to be *independent* from each other. The truth values of unconfirmed observable assertions were used as the responses to questions asked. These values were also used by the SUBIDEAL strategy in order to find an "*optimal*" UOA set.

For each scenario created as above, the five strategies were used separately. The total cost that each strategy used to reach a top level conclusion was recorded. For each set of parameters, 20 random scenarios were generated. The average total cost was calculated for each strategy after it ran on the 20 scenarios.

Fifteen percent of the total number of assertions were top level conclusions, and 20% of the assertions were unconfirmed observable assertions. The level of the system was two if the number of rules was less than 80, and three otherwise. Costs of unconfirmed observable assertions were initially assumed to be uniformly distributed over the interval [0, 100]. The probabilities for a UOA to be true were uniformly distributed in the interval [0.50, 1.00]. The interval [0.50, 1.00] was used instead of [0.00, 1.00] in order to generate more true values and thus consider scenarios with more questions (which are more challenging than scenarios with fewer questions).

Table 1 presents the average costs of the five strategies; CEQA, UOARND, SETRND, DEPTH1ST, SUBIDEAL, or different sizes of Horn clause bases. In the table, the size of a knowledge base is represented as (*number of assertions* $\times$ *number of rules*). For instance, 150×150 represents the Horn clause bases with 150 assertions and 150 rules.

Table 2 is obtained from Table 1. Table 2 provides the cost ratios of the strategies to the proposed strategy CEQA. UOARND resulted in total costs which were 180% to 700% as much as those of CEQA. SETRND and DEPTH1ST resulted in total costs which were 135% to 300% as much as those of CEQA. CEQA took only 20% more expensive total costs than the *unrealistic* SUBIDEAL strategy.

In practice, response costs are often *not uniformly* distributed. Costs for some of the questions may be significantly higher than the others. For instance, in a medical environment the cost for an X-ray photograph may be much higher than that for taking body temperature. Moreover, some questions may have significantly lower probability to be true. People would try to avoid those high cost or low probability questions unless the cheap questions with higher probabilities are tried first. Tables 3 and 4 present the results of experiments reflecting such a situation. For the scenarios in Table 3, 90% of the UOA's have costs between 0 and 100 distributed uniformly, and the rest 10% of the UOA's have costs distributed uniformly in the interval [1000, 2000]. The computational results illustrate that average total costs of the CEQA strategy remained on three digits. This indicates that the CEQA strategy can reach a conclusion by bypassing the expensive questions. Moreover, the cost ratios to CEQA of UOARND, SETRND and DEPTH1ST are ranging from 2.5 to 14 (see Table 4).

In Tables 1 and 3, one can see that the average total response costs for different scenarios do not increase significantly with the sizes of the rule bases. That happens because in the experiments we maintained three premises in all rules, and the number of levels in the systems was either two or three. The results presented in Tables 1 and 3 are graphically depicted in Figs. 1 and 2, respectively.

The computational results demonstrate that the CEQA strategy is very effective in reaching a final conclusion by asking questions which would yield a small total cost. Among the three random based Q-A strategies, UOARND is significantly worse than SETRND and DEPTH1ST, and the latter two performed similarly. That happened partially because SETRND and DEPTH1ST fix a possible top level conclusion to pursue and then ask questions chosen from a UOA set which may lead to that conclusion. In DEPTH1ST, the UOA set selection is not explicit, but the depth first searching procedure leads to a UOA set. The UOARND strategy, on the other hand, selects questions from the pool of unconfirmed observable assertions, rather than from a UOA set.

Table 1
Average total questioning costs for different Q-A strategies. Case I: With regular answering costs.

Scenario	Question-asking strategy				
	CEQA	UOARND	SETRND	DEPTH1ST	SUBIDEAL
30×30	159.92	388.91	299.53	265.09	146.66
40×40	219.17	403.82	342.44	296.53	180.26
50×50	220.85	532.35	420.52	376.57	193.02
60×60	240.29	591.86	425.00	372.54	193.82
70×70	307.25	729.31	469.29	450.39	213.18
80×80	232.86	758.28	534.59	480.14	174.46
90×90	321.53	710.21	547.29	464.38	218.83
100×100	218.84	779.33	521.15	470.17	172.00
110×110	189.18	804.47	514.24	513.99	145.87
120×120	245.11	1,169.73	572.92	531.21	192.21
130×130	228.46	1,107.25	694.26	688.81	140.30
140×140	202.69	1,223.60	657.03	576.56	145.29
150×150	218.83	1,543.38	648.41	606.30	153.35
Average	231.15	826.35	511.28	468.67	174.56

Table 2
Cost ratios with the results of the CEQA strategy for the results in Table 1.

Scenario	Question-asking strategy				
	CEQA	UOARND	SETRND	DEPTH1ST	SUBIDEAL
30×30	1.00	2.43	1.87	1.66	0.92
40×40	1.00	1.84	1.56	1.35	0.82
50×50	1.00	2.41	1.90	1.71	0.87
60×60	1.00	2.46	1.77	1.55	0.81
70×70	1.00	2.37	1.53	1.47	0.69
80×80	1.00	3.26	2.30	2.06	0.75
90×90	1.00	2.21	1.70	1.44	0.68
100×100	1.00	3.56	2.38	2.15	0.79
110×110	1.00	4.25	2.72	2.72	0.77
120×120	1.00	4.77	2.34	2.17	0.78
130×130	1.00	4.85	3.04	3.02	0.61
140×140	1.00	6.04	3.24	2.84	0.72
150×150	1.00	7.05	2.96	2.77	0.70
Average	1.00	3.57	2.21	2.03	0.76

Table 3
Average total questioning costs for different Q-A strategies. Case II: Some of the answering costs are very high.

Scenario	Question-asking strategy				
	CEQA	UOARND	SETRND	DEPTH1ST	SUBIDEAL
30×30	599.67	1,018.33	882.11	939.44	559.53
40×40	654.10	1,714.76	1,560.17	1,473.31	533.90
50×50	672.02	2,477.40	1,695.77	1,619.29	550.29
60×60	296.66	3,308.18	2,309.98	2,242.11	262.26
70×70	335.84	2,562.59	1,996.46	1,836.12	272.24
80×80	531.96	3,316.59	2,156.38	1,785.18	405.18
90×90	352.50	4,625.54	2,616.10	2,442.72	285.80
100×100	350.60	4,589.58	2,275.59	2,258.48	285.80
110×110	533.84	4,617.13	3,470.18	3,686.32	389.86
120×120	280.92	4,617.02	2,829.39	2,780.89	219.86
130×130	209.02	5,404.04	2,016.72	2,059.13	160.62
140×140	507.50	5,564.33	1,941.54	2,263.31	401.21
150×150	453.39	6,457.50	3,125.93	2,683.48	313.95
Average	444.46	3,867.15	2,221.26	2,159.21	356.93

Table 4
Cost ratios with the results of the CEQA strategy for the results in Table 3.

Scenario	Question-asking strategy				
	CEQA	UOARND	SETRND	DEPTH1ST	SUBIDEAL
30×30	1.00	1.70	1.47	1.57	0.93
40×40	1.00	2.62	2.39	2.25	0.82
50×50	1.00	3.69	2.52	2.41	0.82
60×60	1.00	11.15	7.79	7.56	0.88
70×70	1.00	7.63	5.94	5.47	0.81
80×80	1.00	6.23	4.05	3.36	0.76
90×90	1.00	13.12	7.42	6.93	0.81
100×100	1.00	13.09	6.49	6.44	0.82
110×110	1.00	8.65	6.50	6.91	0.73
120×120	1.00	16.44	10.07	9.90	0.78
130×130	1.00	25.85	9.65	9.85	0.77
140×140	1.00	10.96	3.83	4.46	0.79
150×150	1.00	14.24	6.89	5.92	0.69
Average	1.00	8.70	5.00	4.86	0.80

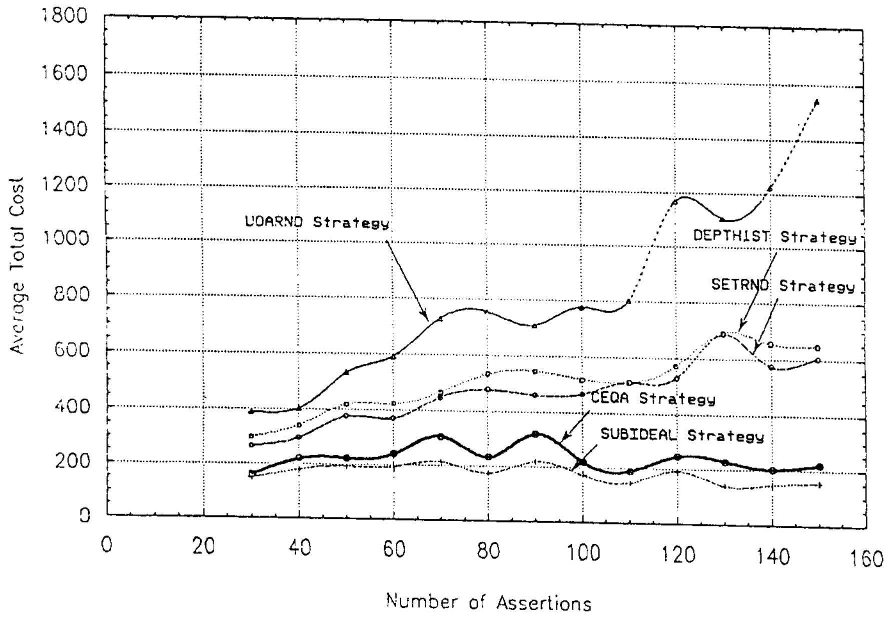

Fig. 1. Average total questioning costs for different Q-A strategies. Case I: With regular answering costs (data are from Table 1).

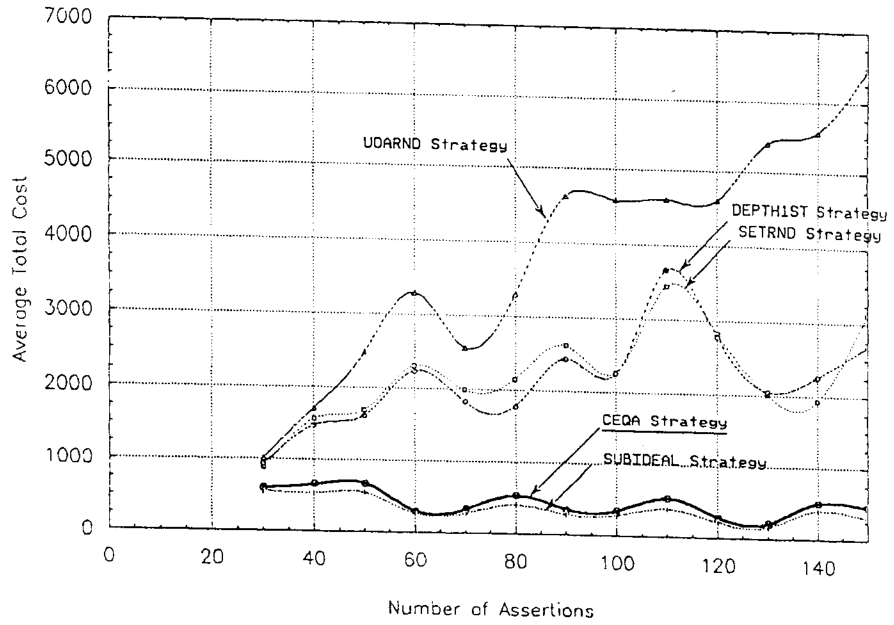

Fig. 2. Average total questioning costs for different Q-A strategies. Case II: Some of the answering costs are very high (data are from Table 3).

6. Concluding remarks

A good inference procedure is required to possess capabilities in two aspects: (1) when the given data are sufficient, the procedure should quickly find an implied top level conclusion; and (2) when the given data are not sufficient, the procedure should intelligently identify the missing facts and guide the inference towards reaching some top level conclusions. The second aspect is concerned about question-asking strategies. The issue discussed in this paper is for propositional Horn systems in which answering different questions involves different costs. The proposed Q-A strategy, CEQA, attempts to reach a final conclusion with a small total response cost. CEQA is an efficient heuristic. The computational results presented in this paper demonstrate that the CEQA strategy is also highly effective.

Propositional Horn clauses can form a knowledge base by their own for some applications, and they are the basis for predicate Horn systems. Further research on question-asking strategies may proceed on predicate Horn systems, systems with uncertainty considerations, and other more general systems.

Acknowledgement

The authors would like to thank the two reviewers for their valuable and constructive comments which improved the presentation of the paper.

Appendix I (An example of applying the CEQA strategy)

Consider the following Horn system:

$A_1 \leftarrow A_3 \wedge A_4$	Rule(1, 1)
$A_1 \leftarrow A_4 \wedge A_5$	Rule(1, 2)
$A_2 \leftarrow A_6$	Rule(2, 1)
$A_4 \leftarrow A_7$	Rule(4, 1)
$A_5 \leftarrow A_8$	Rule(5, 1)
$A_6 \leftarrow A_9 \wedge A_{10}$	Rule(6, 1)
$A_{10} \leftarrow A_8$	Rule(10, 1)
$A_{10} \leftarrow A_6$	Rule(10, 2)

Assume that A_3, A_7, A_8 and A_9 are unconfirmed observable assertions. Suppose that their response costs and probabilities to be true are: $c_3 = 10$, $p_3 = 0.9$, $c_7 = 15$, $p_7 = 0.7$, $c_8 = 12$, $p_8 = 0.8$, $c_9 = 14$, $p_9 = 0.9$. Also note that Rule(6, 1) and Rule(10, 2) form a cycle.

The directed graph associated with this Horn system is shown in Fig. 3.

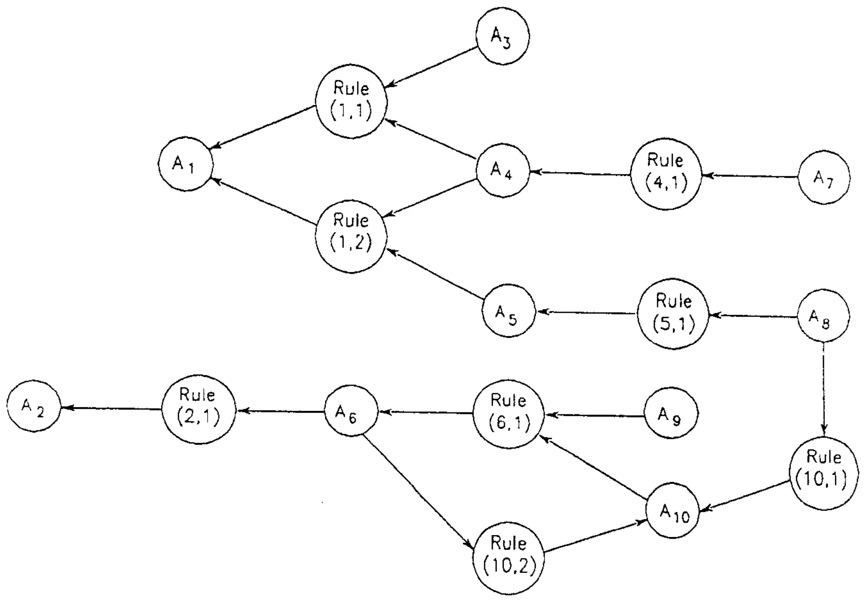

Fig. 3. The graph associated with the example in Appendix I.

The following is the step-by-step results by applying CEQA on this Horn system.

Iteration 1:

Step 1: Apply procedure LABELING:

Apply procedure PROPAGATE as follows:

For A_3:	$C_3 = c_3 = 10,\ P_3 = p_3 = 0.9$;
For A_7:	$C_7 = c_7 = 15,\ P_7 = p_7 = 0.7$;
For A_8:	$C_8 = c_8 = 12,\ P_8 = p_8 = 0.8$;
For A_9:	$C_9 = c_9 = 14,\ P_9 = p_9 = 0.9$;
For Rule$(4,1)$:	$C_{41} = C_7 = 15,\ P_{41} = P_7 = 0.7$;
For Rule$(5,1)$:	$C_{51} = C_8 = 12,\ P_{51} = P_8 = 0.8$;
For Rule$(10,2)$:	$C_{10,2} = C_8 = 12,\ P_{10,2} = P_8 = 0.8$;
For A_4:	$C_4 = C_{41} = 15,\ P_4 = P_{41} = 0.7$;
For A_5:	$C_5 = C_{51} = 12,\ P_5 = P_{51} = 0.8$;
For Rule$(1,1)$:	$C_{11} = C_3 + C_4 = 25,\ P_{11} = P_3 \times P_4 = 0.63$;
For Rule$(1,2)$:	$C_{12} = C_4 + C_5 = 27,\ P_{12} = P_4 \times P_5 = 0.56$;
For A_1:	$\text{MIN}\{C_{11}/P_{11}, C_{12}/P_{12}\}$ $= \text{MIN}\{25/0.63, 27/0.56\} = 25/0.63$;

Therefore, the selections are: $\delta(1) = 1,\ C_1 = C_{11} = 35$, and $P_1 = P_{11} = 0.63$;
Labels cannot be propagated any more because of the cycle.

By applying procedure ASSIGN:

The rule which is currently labeled but its conclusion is not labeled is only (10,2).

Let $C_{10} = C_{10,2} = 12$, $P_{10} = P_{10,2} = 0.8$;

By applying procedure PROPAGATE:

For Rule(6, 1):	$C_{61} = C_9 + C_{10} = 26$, $P_{61} = P_9 \times P_{10} = 0.72$;
For A_6:	$C_6 = C_{61} = 26$, $P_6 = P_{61} = 0.72$;
For Rule(2, 1):	$C_{21} = C_6 = 26$, $P_{21} = P_6 = 0.72$;
For Rule(10, 1):	$C_{10,1} = C_6 = 26$, $P_{10,1} = P_6 = 0.72$;
For A_2:	$C_2 = C_{21} = 26$, $P_2 = P_{21} = 0.72$.

At this point all assertions have been labeled. Next, select a promising top level conclusion as follows:

Because: $C_1/P_1 = 39.7 > C_2/P_2 = 36.1$, we choose A_2 as the potential conclusion A_c to be proved by CEQA. Tracing back from A_2, we find the corresponding UOA set U to be as follows:

$$U = \{A_8, A_9\}.$$

Step 2: Ask questions as follows:

Because $C_8/(1 - P_8) = 60 < C_9/(1 - P_9) = 140$, we first ask about A_8. Suppose that the response for A_8 is positive (i.e., it has a true value).
Next, we ask about the value of A_9. If the response for A_9 were positive, then we would be done; the top level conclusion A_2 is proved to be true. To illustrate the following steps, let us suppose the response for A_9 is *negative*, then we have to go to Step 3.

Step 3: Apply procedure UPDATING as follows:

Since A_9 is confirmed false, Rule(6, 1) is inactivated;
Since Rule(6, 1) is defunct, A_6 is not able to be proved and thus it becomes defunct;
Since A_6 is defunct, Rule(2, 1) and Rule(10, 1) are defunct;
Since Rule(2, 1) is defunct, A_2 is not provable and thus it becomes defunct;
Since A_8 is confirmed true,

A_5 is proved true by firing Rule(5, 1), and Rule(1, 2) is simplified as $A_1 \leftarrow A_4$;

A_{10} is proved to be true by firing Rule(10, 2). (Note that Rule(6, 1) is defunct, therefore it is not necessary to simplify that rule)

The active rules left in the system are thus as follows:

$A_1 \leftarrow A_3 \wedge A_4$	Rule(1, 1)
$A_1 \leftarrow A_4$	Rule(1, 2)
$A_4 \leftarrow A_7$	Rule(4, 1).

Iteration 2:

Step 1: Apply procedure LABELING as follows:
$C_3 = 10,\ P_3 = 0.9;$
$C_4 = C_7 = 15,\ P_4 = P_7 = 0.7;$
$C_{11} = C_3 + C_4 = 25,\ P_{11} = P_3 \times P_4 = 0.63;$
$C_{12} = C_4 = 15,\ P_{12} = P_4 = 0.7;$
For A_1, since $C_{11}/P_{11} = 39.7 > C_{12}/P_{12} = 21.4$, $C_1 = C_{12} = 15$, $P_1 = P_{12} = 0.7$.
Since A_1 is the only top level conclusion now, we choose A_1 as the potential conclusion A_c, and the corresponding UOA set U, is: $U = \{A_7\}$.

Step 2: Ask about the value of A_7. Suppose that the response is "*true*". We are done, the goal A_1 is proved to be true.

References

[1] Y. Ben-dov, Optimal testing procedures for special structures of coherent system, *Management Science* 27(12) (1981) 1410–1420.

[2] R. Butterworth, Some reliability fault-testing models, *Operations Research* 20 (1972) 335–343.

[3] J. De Kleer and B.C. Williams, Diagnosing multiple faults, *Artificial Intelligence* 32 (1987) 97–130.

[4] J. De Kleer, Using crude probability estimates to guide diagnosis, *Artificial Intelligence* 45 (1990) 381–391.

[5] W.F. Dowling and J.H. Gallier, Linear time algorithms for testing the satisfiability of Horn formulae, *J. Logic Programming* 3 (1984) 267–284.

[6] R. Duda, J. Gasching and P. Hart, Model design in the PROSPECT consultant system for mineral exploration, in: *Expert Systems in the Micro-electronic Age* (Edinburgh Univ. Press, Great Britain, 1979).

[7] F. Hates-Roth, D.A. Waterman and D.B. Lenat, *Building Expert Systems* (Addison-Wesley, Reading, MA, 1983).

[8] R.G. Jeroslow and J. Wang, Dynamic programming, integral polyhedra and Horn clause knowledge bases, *ORSA's Journal on Computing* 1 (1989) 7–19.

[9] J.D. Kershaw, *Digital Electronics – Logic and Systems* (Wadsworth, Belmont, CA, 1976).

[10] G.F. Luger and W.A. Stubblefield, *Artificial Intelligence and the Design of Expert Systems* (Benjamin /Cummings, Redwood City, CA, 1989).

[11] C.S. Mellish, Generalized alpha-beta pruning as a guide to expert system question selection, *Expert System '85, Proceedings of the Fifth Technical Conference of the British Computer Society Specialist Group on Expert Systems, Univ. of Warwick* (Cambridge Univ. Press, Cambridge CB2 1RP, 1985) pp. 31–41.

[12] E. Triantaphyllou and J. Wang, The problem of asking the minimum number of questions in Horn clause systems, *Math. Comput. Modeling* 20(9) (1994) 75–87.

[13] J. Wang and J. Vande Vate, Question-asking strategies for Horn clause systems, *Annals of Mathematics and Artificial Intelligence* 1 (1990) 359–370.

VOLUME 17 (1996)
Nos. 3,4

ISSN 1012 2443
Published October 1996

ANNALS of MATHEMATICS and ARTIFICIAL INTELLIGENCE

Editor-in-Chief:
Martin Charles Golumbic

LAST ISSUE OF THIS VOLUME

ARTIFICIAL INTELLIGENCE and MATHEMATICS V

Editors:
Endre Boros – Martin Charles Golumbic

BALTZER SCIENCE PUBLISHERS

ANNALS OF MATHEMATICS AND ARTIFICIAL INTELLIGENCE

VOLUME 17 (1996) Nos. 3,4

Published October 1996

Contents

ANNALS OF MATHEMATICS AND ARTIFICIAL INTELLIGENCE

VOLUME 17 (1996) Nos. 3,4

Published October 1996

Contents