CS 82

A DIRECTED GRAPH REPRESENTATION FOR COMPUTER SIMULATION OF BELIEF SYSTEMS

LAWRENCE TESLER

HORACE ENEA

KENNETH MARK COLBY

TECHNICAL REPORT NO. CS 82
DECEMBER 29, 1967

COMPUTER SCIENCE DEPARTMENT
School of Humanities and Sciences
STANFORD UNIVERSITY

A DIRECTED GRAPH REPRESENTATION FOR COMPUTER SIMULATION OF BELIEF SYSTEMS*

Lawrence Tesler
Horace Enea
Kenneth Mark Colby

June (1967)

(To Appear in Mathematical Biosciences, 1967)

Department of Computer Science
Stanford University
Stanford, California

*This research is supported by Grant PHS MH 06645 from the National Institute of Mental Health.

ERRORS

p. 5, 3rd line from bottom should read:

priori, observed, or reasoned) are examples.

p. 8, line 12 should read:

Concepts are represented in this model by nodes of a directed

p. 10, line 7 should read:

(i.e., the set of all persons), SALLY, WOMAN, ANIMAL, MOVE, SPEECH

p. 10, line 8 should read:

BROWN-HAIRED, COLOR, and BRUNETTE.

p. 10, line 10 should read:

(5) PERSONS speak, (6) Brown-haired is a subset of colored,
(7) Sally is brown-haired

p. 10, line 15 should read:

concept person by the node PERSON

p. 10, line 22 should read:

are SALLY ε PERSON,

p. 10, line 25 should read:

person".

p. 12, line 15 should read:

Rules differ from conventional programs on present computers in
that

p. 14, bottom line should read:

$(F)\lambda(e32;e21)=(D)$, and $(A,B,C)\lambda(s12;s23;e12;e23;p32;p21)=(L,M,C,J)$.

p. 19, line 16 should read:

P does/has Q T R

p. 19, bottom line should read:

nodes FEAR and TOWARD.

Abstract

A method of representing data-structures in the form of a directed graph is described. Such a graph is suitable for the data-base of belief systems in particular and of large memory structures in general. ( )

Introduction

There exists a class of problems in the behavioral sciences which have been difficult to manage satisfactorily with information processing methods because of a lack of a good computer representation for very large memory structures. One example is the abstract representation linguists term a "deep structure" into which natural language is translated and to which a transformational grammar is applied in generating natural language sentences. Another example consists of the large data-bases required in computer simulation of human belief systems. It is the purpose of this paper to describe a directed graph we are using for the representation of the data-base of a computer model which simulates the formation and processing of an actual person's or an artificial system's beliefs about interpersonal relations. Although we have written a computer program in a special-purpose language (GRAPPLE) we designed to process a directed graph of this kind, the present paper will be concerned primarily with describing the graph and its relations to the model. A future paper will discuss the details of our particular implementation.

Concepts

A useful approach in building a model is to stipulate a unit or basic component of the data-base. The unit in our model will be the abstract entity, concept. Examples of concepts are parents, fear of women, old men, hating authority, John, hatred.

KINDS OF CONCEPTS

1) Sets
2) Individuals
3) Propositions

A belief in this model is an attitude towards a proposition about concepts. Examples of propositions are men like women, John obeys his father, fear of men leads to avoidance of men, etc. It is convenient to regard a proposition as a special case of a concept. A proposition has one of two functions -- to represent fact, or impart a rule. Thus, all men are mortal is a fact, while if it is believed that x is a man, it shall be believed that x is mortal is a rule. Note that these two statements are logically equivalent, but one is in a form amenable to classification (a fact), while the other is in a form amenable to reasoning (a rule). Since facts and rules form pairs in the above manner, we will recognize the duality of rules and facts.

Concepts are formed in various ways. In our model, we restrict concept formation to three methods:

ORIGIN OF CONCEPTS

1) Starting knowledge (a priori)

2) Observation (perception, hearsay, etc.)

3) Reasoning (identification or induction)

A priori concepts might be self, and desire to observe. Observed concepts would include I see my brother, and my mother tells me sex is bad. Some factual concepts that could be induced from the above concepts and previously formed rules are sex is bad, my mother is crazy, I am curious about sex. A factual concept obtained by identification is, this person is a man. Other methods, such as 'deduction' and 'analogy', are all considered degenerate forms of identification and induction in this model. One justification for this will be given. An analogy such as: 'Cows, four legged animals, give milk, so horses, also four legged animals, probably give milk', really consists of the induction, 'Cows, four legged animals give milk, so: all four legged animals give milk', and the identification, 'horses are four legged animals, all four legged animals give milk, so: horses give milk'. All deductions are applications of formal rules of manipulation; therefore, they can all be performed by analogy to a learned set of deductive patterns (for example, modus ponendo ponens). Since ordinary people can not look at

two complex propositions and determine if they are equivalent without involved and skillful analysis, there is no reason why our model should be a super-logician.

Not all propositions need be beliefs in the model. The degree to which the model is willing to accept a proposition will be called the credibility of the proposition, while the degree to which evidence substantiates a proposition will be called its foundation. Thus, 'There is a man behind that wall' might have little foundation yet be credible. Indeed, 'there is not a man behind that wall' might have the same lack of foundation and also be credible. On the other hand, 'we all use ESP' may be well founded yet incredible. Credibility and foundation are important in a model of belief systems because they are useful criteria for rejecting and accepting observed and induced propositions during periods both of pathological and of normal thought. We will arbitrarily assign values of 0 to 100 for these attributes; e.g., a credibility of 50 means as credible as not (50-50). 'We all use ESP' could be said to have a credibility of 30, but a foundation of 80. 'I am alive' might have a credibility of 100 and a foundation of 100. 'There is a man behind that wall' could have a credibility of 85, but a foundation of 10, as could 'there is not a man behind that wall'. If the credibility and foundation of a proposition p are designated by cred (p) and found (p), and the negation of a proposition p is $\bar{p}$, we have, in normal situations:

$$0 \leq \text{cred } (p) \leq 100$$
$$0 \leq \text{found } (p) \leq 100$$
$$\text{cred } (p) + \text{cred } (\bar{p}) \geq 100$$
$$\text{found } (p) + \text{found } (\bar{p}) \leq 100$$

The credibility inequality arises from the peculiarity that both a proposition and its negation could be quite credible (cf. man behind the wall), although they could not both be incredible, for if p is incredible, there is little choice but to accept $\bar{p}$. Similarly, both a proposition and its negation could be quite unfounded, (cf. man behind

the wall), but if p becomes well-founded, it tends to detract from the foundation of $\bar{p}$*.

The rules of probability, such as prob (p and q) = prob (p) x prob (q), are not realistic predictors of human beliefs. One of the problems encountered in the simulation of belief systems is how to assign credibility and foundation values to propositions, whether entering the system or being produced as a result of reasoning.

Without regard for their status as beliefs, two propositions can still differ only in intensity, e.g., John is firm and John is obstinate, or John strongly believes x and John weakly believes x. We will agree by convention:

$$0 \leq \text{intens}(p) \leq 100$$
$$\text{intens}(p) = \text{intens}(\bar{p})$$

There are thus three values attributable to a proposition; two relate to its status as a belief, the other does not.

MEASURES OF PROPOSITIONS

1) Credibility (degree of acceptance)
2) Foundation (degree of substantiation)
3) Intensity (e.g., relative degree of assertion)

Different concepts (propositions, ideas, and tokens) in a model can vary in their importance to the train of thought, to decision making, and to reasoning processes of a simulation, and a single concept can vary in its importance from time to time. The attribute measuring these differences will be called charge. Thus, sex might be a permanently charged concept to a model, while washing the dishes might be temporarily charged. If c is any concept,

$$0 \leq \text{charge}(c) \leq 100$$

---

*When found (p) + found $(\bar{p}) > 100$ we call p a paradox; and when cred (p) + cred $(\bar{p}) < 100$ we call p a quandary. The above are abnormal situations and the model would try to resolve them if noticed.

If c happens to be a proposition p, there is no special restriction on the relation between charge (p) and charge ($\overline{p}$). One could care less whether or not 'there is a man behind that wall' (charges both low), but become extremely excited whenever someone expresses either agreement or disagreement with 'capital punishment is necessary' (both charges high). Furthermore 'there were no tornadoes in Oregon today' might have low charge while its negation could be quite highly charged. Charge can be resolved into two components, fixed charge and current charge. A temporarily charged concept has a high current charge but not necessarily a high fixed charge. When the current charge of a concept approaches its fixed charge, the concept is said to be active. If it is very much below the fixed charge, it is said to be dormant.

Other quantitative distinctions could be made between concepts, such as longevity (how long ago formed) and inhibition (tendency to be avoided in communication and reasoning). For example, a concept formed long ago having few known properties is easily "forgotten". An inhibited concept which is highly charged is of interest because it may provide a clue to some previous trauma or to a neurotic conflict. These attributes are not central to this paper, but will be included in the following table for completeness:

MEASURES OF CONCEPTS

1) Charge (importance to the belief system)
2) Longevity (time since concept was formed)
3) Inhibition (tendency to avoid being considered)

A belief such as 'I was beaten by my mother' could be charged and long lived, but also be very inhibited. Thus, the model would be expected to avoid reference to it unless forced by extenuating circumstances.

In addition to quantitative measures, concepts have qualitative aspects. The kind (set, individual, or proposition) and the origin (a priori observed, or reasoned) are examples. However, most qualitative information about a concept is supplied by its relationship to other concepts.

Autosimulation

A belief system is itself a model of the universe. Thus, a model of a belief system is a model of a model. The ability of people to model a model is both unusual and important, especially when the target model is the human mind. In fact, reasoning processes are aided by simulation of one's own mind, i.e., by autosimulation. For example, 'If you show me a cat with two tails, I will believe two tailed cats exist' is a prediction of a change of one's own mind in a certain situation. All beliefs which are rules are models of the model itself (cf., 'if it is believed that x is a man, then it will be believed that x is mortal').

Context

During a dialogue, a person observes words and sentences which upon examination out of context are not subject to reliable interpretation. Thus, 'It is blue outside' could refer to the color of the exterior of a car, the loveliness of the weather (blue as in blue sky), or even the dismalness of the weather (blue as in Blue Monday). Out of context, 'it' can not be interpreted. It may be a pronoun whose antecedent could be 'the car', or it may be a part of a figure of speech: 'it is x' specifying a general outlook of condition, especially of the weather.

A name is something uniquely associated with something, while an appellation is what we happen to be using to refer to something. Even a seemingly accurate word like John is really not a name, but an appellation -- it can be used to refer to any of a number of people, depending on the context. We will call this type of multivocality the 'context problem'. Its solution is crucial to those applications in which a model must communicate by means of natural language. In addition, solution of some non-linguistic problems, such as the determination of relevance, may be expedited by regarding them as context problems.

Representation of Beliefs

It has already been indicated that beliefs are attitudes towards propositions, and that propositions are special kinds of concepts. This reduces the problem mainly to one of representing concepts, their values, and the relations between them.

All our examples of concepts so far have been expressed as English phrases. Natural language has room for rich variability, but is uncomfortably vague, ambiguous, multivocal, unspecific and difficult to process usefully on a computer. Some problems of natural language can be solved by adopting the notation of logic and set theory, but the rich variability disappears. In addition, there is no provision in conventional logic for meaningful processing of inconsistent propositions, nor for representation of contextual dependencies. Computer processing of logical and set notation can often be straightforward; however, when we try to simulate human cognition and affect, the inability to cope with conflicting propositions makes set theoretic notation inadequate. Hence we require a different representation.

Several schemes have been proposed for such a representation. We call them 'associative languages', for they allow arbitrary, even inconsistent associations to be specified between concepts in easily processible form. Feldman's APL allows relationships to be stated in terms of triples (6, 12).

LEFT (CIRCLE) = SQUARE
LEFT (CIRCLE) = TRIANGLE
INSIDE (SQUARE) = TRIANGLE
OBJECT (D) = FATHER, where,
D = FEAR (SELF) (I fear my father)

Basically A(B)=C is read, 'One thing which is A of B is C'. APL allows answering questions like:

LEFT (CIRCLE) = ?
INSIDE (?) = TRIANGLE

In which one or two elements of the triple are not specified. All

triples in the system which are of that form are located. Since many 'answers' are possible to a question (e.g., both SQUARE and TRIANGLE satisfy LEFT (CIRCLE) = ?), the answer to such a question is generally a set of things.

One APL triple can be thought of as a subgraph of a directed graph whose arcs associate together three nodes. The chief advantage of this outlook is that all nodes near a given node are readily accessible, while in APL several association questions may have to be asked to determine 'nearness' to identify things related to another thing in a certain way. Other advantages of this representation will emerge later.

A Directed Graph Model

Concepts are represented in the model M by nodes of a directed graph. Simple relationships are represented by directed arcs between pairs of nodes. Each arc is labelled e,s, or p, depending on the type of relationship existing between the connected concepts. If an arc labeled r (where r = e, s, or p) is directed from node A to node B, we say ArB. The types of arcs are distinguished by their formal properties, but notions of their approximate meanings can be outlined:

AeB	Individual A is a member of the set B.
AsB	Set A is a subset of the set B, or Proposition A is a consequence of the proposition B.
BpA	A has B, or A has property B, or B belongs to A, or B is part of A, or the idea of A suggests the idea of B, or A does B

All three kinds of concepts are mentioned in the above outline: individuals, sets, and propositions. By convention, individuals and sets are collectively called tokens. The relationship e is between an individual and a set, s is between two sets or between two propositions, and p is between two tokens. The same node can be a token and a proposition in different contexts, and the same token can be a set and an individual in different contexts. When we say AeB and BeC, we are implying, in the logical sense, that C is a family of sets like B. Although the

terminology of families of sets can be used occasionally in discourse for clarification, families and sets are not explicitly distinguished in the graph.

The formal properties distinguishing the three types of arcs are given by seven axioms for valid graph-enlargement.

Axioms:

Axiom 1.	AsA	s is reflexive.
Axiom 2.	AsB . BsC → AsC	s is transitive.
Axiom 3.	AeB . BsC → AeC	A member of a subset is a member of the set.
Axiom 4.	AsB . CpB → CpA	A property of a set is had by its subsets.
Axiom 5.	AeB . CpB → CpA	A property of a set is had by its members.
Axiom 6.	BpA . BsC → CpA	Having a specific property implies having the more general property.
Axiom 7.	BpA . BeC → CpA	Having a specific property implies having the more general property.

For instance, axiom 2 states that whenever an s arc is directed from A to B, and another s arc from B to C, it is valid (after substitution and detachment) to direct in addition an s arc from A to C. These axioms and their contrapositives are the sole means of valid inference available to the model. The axioms do not vary according to information in the graph. They are considered to be automatically utilized in processes of interpretation and reasoning by the model and no other laws of formal logic are available as axioms to the model. For example, DeMorgan's Laws (-(AB) → -A V -B; -(A V B) → -A . -B) can not be applied as laws of inference by the model to representations in the graph.

Examples:

All persons are persons. (Axiom 1)

If women are persons, and persons are animals, then women are animals. (Axiom 2)

If p is a consequence of q, and q is a consequence of r, then p is a consequence of r. (Axiom 2)

If Sally is a person, and persons are animals, then Sally is an animal. (Axiom 3)

If persons are animals, and animals move, then persons move. (Axiom 4)

If Sally is a woman and women speak, then Sally speaks. (Axiom 5)

If brunette is brown-haired and Sally is brunette then Sally is brown-haired. (Axiom 7)

The graph in Figure 1 represents by circles the tokens **PERSON** (i.e., the set of all persons), SALLY, **WOMEN**, **ANIMAL**, **MOVE**, **SPEECH**, **BROWN**, **COLOR**, and **BRUNETTE**. The relationships (1) Sally is a person, (2) Women are persons, (3) Persons are animals, (4) Animals move, (5) Women speak, (6) Browns are colors, (7) Sally is brown-haired, (8) Brunette is brown-haired, and (9) Sally is brunette are represented by the arcs pointed to by the respectively numbered triangles.

Insert Figure 1 Here

Notice that the concept Sally is represented by the node **SALLY**, and the concept Woman by the node **WOMEN**. Since the information that Sally is a woman is contained in the graph, it is desirable to have a node representing that entire concept. This is accomplished by the convention that every arc has associated with it a single node called the circumstance of that arc, which is drawn as a triangle touching that arc, and whose meaning is, roughly, "the idea that" that arc exists. For example, the triangle 1 in figure 1 is really the circumstance node of the arc SALLY e **WOMEN**, and it is interpreted, "The idea that Sally is a woman". Triangular nodes are propositions; generally, they are propositions concerning the existence of certain relationships between certain concepts. We reserve circular nodes for tokens. As was noted before, occasionally a node will be used as both a proposition and as a token; in such a case it is written as a triangle and its kind is determined by context.

Rules in the Model

To be able to reason, the model obviously needs more than the seven axioms stated earlier. Because human beings generally reason not

by formal logical deduction, but by "common sense" reasoning, analogy, induction, and plausible inference, the consequence relationship denoted by the s arc between propositions is not deductive implication; it is called instead a "rule" to suggest heuristic reasoning, as in "rule of thumb". Rules are propositions in the graph like any other, and are subject to hypothetical formation and verification. They are the "backbone" of the graph, for they provide the principal ability to reason about the universe in a flexible manner. Important characteristics of rules are that each must be well-defined, but several rules may mutually conflict. This allows the model to be self-inconsistent, unlike a conventional logical system, but much like a human being. While the seven axioms and their corollaries are independent of the information in the graph, rules are themselves part of the graph, and can be used and changed during processes of hypothesis formation and verification. The model can represent rules in its graph, and utilize them to enlarge the same graph. Not every application of a rule is expected to draw a valid conclusion, rather, rules should be useful heuristics for generating and testing hypotheses.

Two new kinds of nodes must be introduced to assist in substitution and detachment. A formal node is one which is a variable in the antecedent of a rule; it must be substituted for by a normal node. A creative node is one which specifies graph-enlarging in the consequence of a rule; it results in the creation of a new node during detachment. In the graph, formal nodes are denoted by drawing a diameter inside the circle or triangle, and creative nodes are denoted by writing a C inside.

Insert Figure 2 Here

In Figure 2, the circumstance 1 represents the fact that AeB is a consequence of BpA for any A and B (heuristically).

Rules as Data and as Programs

It was stated in an earlier section that rules enable a model to ponder about what it might think in a hypothetical situation by sim-

ulating itself. When doing this, the model utilizes rules as data, that is, examination of the rules directs the examination and processing of other data. But rules can also become activated and autonomously create new concepts or modify old ones, i.e., they can be programs. Thus, the same subgraph can be both program and data. This observation suggests a duality of program and data. A program can be regarded as a processor of data and of other programs, and a data structure as a retriever of programs and of other data. The duality of program and data is reflected in a related duality of processing and retrieval.*

Some advantages of using the same representation for programs and data are that processing and retrieval can be carried out in the same information base, programs can be created, examined, and changed by the model, and tradeoffs in effort between retrieval and computation can be accomplished by transformations within a single domain.

Rules differ from conventional programs on today's computers in that they are not sequential. The natural way to use rules would be to have each rule continually scan the graph for conditions satisfying its antecedents, and then create new concepts according to its consequences; however, presently this is neither practical nor realistic. Although rules should be executed in parallel, it seems likely that any one rule should only operate in special situations. Such situations are: some rule has just been formulated, and is to be tried out; an observation is made, and the model must evaluate it as a proposition; a fact is highly charged and enters the present context; the model enters a phase of 'contemplation'.

Rules allow the model to make judgments both of observed events and of its own current state. Just as rules such as "When Sam tells me a story about his accomplishments, they are untrue" can be invoked to screen potential beliefs, appropriate rules could be used to notice and resolve conflicts, to make hypotheses about interpretations of natural language input, and to choose among presented hypotheses on the basis of evidence derived from relevant rules and facts.

---

*The duality of facts and rules mentioned on page 2 is a linguistic one, while that of program/processing and data/retrieval is a functional one.

The consequence of a rule can be a new rule. Thus, the model can expand the repertoire of its rules as well as that of its beliefs.

Computer Representation

The directed graph model can be evaluated best either by investigating human behavior to see if it displays the characteristics of the model, or by using the model in a simulation of human cognitive activity to see if it displays the characteristics of a human. The present paper will discuss the latter course, because it can be performed by the reliable technique of interviewing a computer, while the former course requires dependence upon interviewers and subjects, with the associated interference caused by interpersonal relationships and unreliable responses.

One way to represent a directed graph structure such as that used in the model will be presented here. Each node in the graph, including all circumstances of arcs, will correspond to a record (table) in a random-access memory. The record for the node N will have the following fields (attributes):

MODE (normal, formal, or creative).
TYPE (t if a token, e, s, or p if a proposition).
CHARGE (0-100).
LONGEVITY (0-...)
INHIBITION (0-100)
OPERATORS (a list of directed arcs touching N).
NEIGHBORS (a list of the nodes attainable via OPERATORS).

For propositions only...

SIGN (affirmation or negation).
CREDIBILITY (0-100).
FOUNDATION (0-100).
INTENSITY (0-100).

The arcs need not be represented since their circumstance nodes are represented.

There are two ways to look at a graph: locally, and globally. Local examination implies that examination begins at some node, and

proceeds only by following the arcs (in either direction) that touch that node. Global examination requires "stepping back" from the graph and looking for patterns. Both kinds of examinations are useful, but most functions can be performed satisfactorily by local examination, so this method is adopted in the present research.

Beginning at a node N, there are several possible connections that can exist to arcs that touch it.

Insert Figure 3 Here

An arc can be thought of as having three positions at which nodes can touch it: the tail, the center, and the head. Thus the arc in figure 3 which means NeA has N at its tail, K at its center, and A at its head. If the three positions are numbered 1, 2, and 3, we can designate the process of traversing the type e half-arc from N to K by an operator:

e12

In general, if r is one of the relations e, s, or p, the possible operators to traverse half-arcs of type r are:

r12 (tail to center) r21 (center to tail)
r23 (center to head) r32 (head to center)

Operators can be strung out and applied successively to a node to reach any other node. For example, one operator string to reach V from R in figure 3 is

(p21;p32;s12)

It is useful to define a multi-valued function & ("via") whose arguments are a list of nodes and an operator string, and whose value is a list of all nodes attainable from the argument nodes by application to each of the successive operators. In figure 4,

Insert Figure 4 Here

(F)&(e32;e21)=(D), and (A,B,C)&(s12;s23;e12;e23;p32;p21 = (L, M, C, J).

Operator strings provide a limited means of looking for patterns in a graph without resorting to global examination. We can conceive of "factoring" a graph into its structure and its content. Thus, the subgraph of Figure 4 which consists of the path appearing horizontal from B to M has content BPGQM and structure (s32;s21;p32;p21).

The nodes attainable by single operators from N will be called the neighbors of N. The list of neighbors of N is included as a field in the computer's record of N. Corresponding to this list is a list of the respective operators via which the neighbors are attained. For example, the operators and neighbors lists of G in figure 4 are:

(s12,p32,e12,e12,s32)
(P , Q , R , U , V)

If G is in fact a normal token with average charge, age 20 units, and inhibition 10, the complete record of G will appear:

MODE: normal
TYPE: t
CHARGE: 50
LONGEVITY: 20
INHIBITION: 10
OPERATORS: (s12,p32, e12,e12,s32)
NEIGHBORS: (P , Q , R , U , V)

Graph Searching

Now, suppose it is required to find a node x which bears a certain relationship R to N. If R can be expressed as a set of possible operator strings $B_1,\ldots,B_m$, then the solution is theoretically obtainable by computing $(N)\&B_1,\ldots,(N)\&B_m$ in whose union U any node will suffice as X. U could be empty, in which case there is no solution X, or it could have one element, in which case the solution is unique, or it could provide multiple solutions, of which we could choose any one.

The disadvantage of this theoretical solution is that, on a computer, when the operator strings become relatively long (say, ten or twenty operators) the & function begins to become slow because many paths

match the first few operators but fall out of contention later; although they will not contribute to a solution, their paths must be followed out until they "die". If the computer were capable of parallel processing (as future computers will increasingly be), all paths radiating from N could be traced simultaneously, and no time would be wasted. But with sequential processing, methods must be found to minimize the search time if an efficient simulation is to be possible.

Three basic methods are available for graph searching on a sequential machine. The standard method of recursive tree-searching can be adapted to graphs; this will be called "depth-before-breadth".

Consider a person lost in a forest stumbling upon an intersection of two deserted roads. He thinks there is a town a few miles away, and that there may not be another for hundreds of miles. Unfortunately, he has no way to tell which road to follow. So he sets off on one road, leaving a mark in the ground at the intersection so he will remember that he has already tried this route in case he is later forced to backtrack. If he encounters other intersections, he again chooses a route and leaves a mark. If his first choice gets him to the town by sundown, all is well and good. However, if he walks several miles, he might decide that he has chosen the wrong route, and will backtrack to the nearest marked intersection. If there are any roads from this intersection he has not yet tried, he will mark them and try them out. Otherwise, he will again be forced to backtrack, etc. Eventually, either he will have reached the town or will have tried every route out to a certain radius from his starting point. In the example in Figure 5,

Insert Figure 5 Here

the walker started out the wrong way, and exploited many useless paths before discovering the town. We call this "depth-before-breadth" because he always searched as deeply down a path as he ever would before backtracking and broadening the search.

Now in a computer tracing through a graph, it takes one unit of processor time (say, several milliseconds) to traverse a half-arc from

one node to another in the forward searching direction, and a consistent smaller unit of time to backtrack from a dead end to the preceding node. To simplify matters, suppose one forward step takes 10 ms and one backwards step takes 5 ms. Then the search analogous to figure 5 would take 155 ms.

The second method of searching we call "breadth-before depth". This method involves examination of all nodes one step away, then all nodes two steps away, etc., until a solution is found, or a certain number of steps have been tried. This method is not available to our lost traveller, which might explain why it is used less frequently in graph and tree-searching than depth-before breadth. Figure 6 shows how a strong leapfrog might search for the town given the same situation as in Figure 5.

Insert Figure 6 Here

This method requires no backtracking and no recursion. In this particular case, ten steps are necessary; if they take 10 ms each, the search will conclude in 100 ms, as compared to 155 for depth-before-breadth.

The examples in figures 5 and 6 are not proofs that breadth-before-depth is a superior method to depth-before-breadth. It is often the case, however, that breadth-before-depth is better when the node being sought is relatively close to the starting point and is a unique solution, but worse when there are multiple solutions, all distant from the starting point. In richly connected graphs, the situation becomes more complicated, because loops can exist and the same paths searched repeatedly, which may or may not be desirable.

The third method of searching might be called the Monte Carlo approach. Short segments of paths in the graph are chosen at random, and an attempt is made to fit the segments into the operator string. If several segments that are adjacent in the graph become adjacent when compared to the operator string, they are preserved as highly likely candidates; others are eventually discarded. Variations on this method seem more reasonable: one good one would be to search randomly for a

segment matching a substring near the middle of the operator string, and then to search from the middle to both ends. The value of such techniques has not been studied.

Graph Matching

It is often desirable in processing a graph to locate a subgraph R which has the same structure as a given subgraph Q. For example, an input proposition may be evaluated and comprehended by finding relevant beliefs in the graph of the model. This is the kind of problem that seems best solved by global examination; however, the local examination techniques presented here are sufficient.

One necessary step to take before tackling a matching problem is to delineate the subgraph which is to be matched. One way to do this on paper is to draw a "circle" enclosing just those nodes and arcs which belong to the subgraph. In local examination, this is not a very good approach, because there is no natural orientation that can be given to the subgraph so that it can be easily compared with other subgraphs. Also, computer representation of such boundaries is not conveniently accomplished.

The present method takes advantage of the fact that nodes in the graph all represent concepts for which it is desirable to maintain a method of expressing their meaning in a natural linguistic manner. The graph is constructed so that such an interpretation is always attached to each node (see figs. 1 and 7). Furthermore, it takes advantage of the fact that, in general, subgraphs that are to be matched express concepts in a natural linguistic manner, and so the structure guarantees that there is a node in the subgraph that uniquely defines its boundaries; this node will be called the hook of the subgraph. The graph-searching problem then reduces to one of hook-matching, which has a simple local examination solution because the subgraph becomes a tree with its hook the root, and arcs directed away from the hook the branches.

In most cases, the graph-searching problem has restraints that make it even easier. For example, in order to use a rule which includes at least one normal node in the antecedent, a proposition must be under

consideration that contains the normal node with the same role in the structure of the proposition that it has in the structure of the antecedent. From the preceding paragraph, it is determined that the hook of the proposition must likewise have the same role in the structure of the proposition as the hook of the antecedent has in its structure. Thus, we have two subgraphs with at least one node in common, and the problem is, starting at the hook of one, to find the hook of the other (if any).

An example of this problem is shown in Figure 7

Insert Figure 7 Here.

The subgraph on the right has hook B. It means, literally, JOHN is a member of the set: (a member of the family of TOWARDS, belonging to (a member of the set of FEARS, belonging to SELF)); i.e., John is one of what self's fear is towards, or, Self fears John. When a sentence can be put in the form

P does/has Q R I

where P is the subject, Q the noun form of the verb, I a preposition, and R its object, it usually can be represented as shown in Figure 8.

Insert Figure 8 Here.

Since English does not precede direct objects of verbs by a preposition, we can invent one called D.O. so that sentences like JOHN HITS SAM can be encoded from JOHN DOES HIT D.O. SAM.

Returning to figure 7, the left side of the figure has a main subgraph with hook at F which is a rule. The antecedent of the rule has its hook at A, and it is read, V FEARS Y. The consequence has its hook at D, and it is read, V AVOIDS Y. Thus, F is read, IF V FEARS Y THEN V AVOIDS Y. Since the hook B on the right side says SELF FEARS JOHN, we should be able to conclude by autosimulation that SELF AVOIDS JOHN according to this rule. SELF FEARS JOHN can easily be matched with V FEARS Y because of the symmetry of operator strings about the common nodes FEARS and D.O.

Summary

The representation described in this paper has many features in common with recently developed question answering programs (Refs. 1-12). We have attempted to generalize these approaches for use in belief systems in the following ways. First, the logical inference system has been expanded to allow non-deductive inference, including analogy, plausible inference, inductive inference, and modal logic. Second, the data base representation has been generalized from property lists, trees, and triples to a more general directed graph with a compact representation, extensive richness, and a parsimony of forms. Third, the ability to add new relations in a simple and consistent manner has been achieved; that is, the graph can build new relations between concepts using only the three basic arcs, other previously formed concepts, and simple tokens. Question answering programs have shown the value and feasibility of inference programs, consolidation and generalization of these ideas provide an effective and general representation for beliefs and makes possible the modeling of human mental processes.

We have described in detail a directed graph suitable for computer representation of data characteristic of belief systems. This graph constitutes a formal structure capable of abstractly representing the great variety of semantic relationships found in human concept and belief systems. We believe it is sufficiently general to provide a way of representing data-bases of very large memory structures.

References

1. F. S. Black. A deductive question answering system. Ph.D. Thesis in Applied Mathematics, Division of Engineering and Applied Physics, Harvard University, Cambridge, Mass. (1964).

2. D. G. Bobrow. Natural language input for a computer problem solving system. Ph.D. Thesis, Department of Mathematics, M.I.T., Cambridge, Mass. (1964).

3. K. M. Colby and H. Enea. Heuristic methods for computer understanding of natural language in context-restricted on-line dialogues. Mathematical Biosciences, 1 (1967), 1-25.

4. W. S. Cooper. Fact retrieval and deductive question-answering information retrieval systems. JACM 11 (1964), 117-137.

5. J. L. Darlington. A COMIT program for the Davis-Putnam algorithm. Research Laboratory of Electronics, Mechanical Translation Group, M.I.T., Cambridge, Mass., (1962).

6. J. A. Feldman. Aspects of associative processing. Technical Note 1965-13, Lincoln Laboratory, M.I.T., (1965).

7. A. W. Holt, S. O. Chagnon, R. M. Shapiro and S. Warshall. Mem-theory: A mathematical method for the description and analysis of discrete, finite information systems. Applied Data Research Publication 1965.

8. R. K. Lindsay. Inferential memory as the basis of machines which understand natural language. In Computers and Thought, E. A. Feigenbaum and J. Feldman (Eds.), McGraw-Hill, New York, 217-233 (1963).

9. J. McCarthy. Programs with common sense. Proceedings of the Symposium on Mechanization of Thought Processes, I, HM Stationary Office, London, England, 75-91 (1959).

10. R. Quillian. Semantic theory. Ph.D. Thesis, Carnegie Institute of Technology, Pittsburgh, Pa. (1966).

11. B. Raphael. SIR: A computer program for semantic information retrieval. Ph.D. Thesis, Department of Mathematics, M.I.T., Cambridge, Mass. (1964).

12. P. D. Rovner and J. A. Feldman. An associative processing system for conventional digital computers. Technical Note 1967-19, Lincoln Laboratory, M.I.T., Cambridge, Mass., (1967).

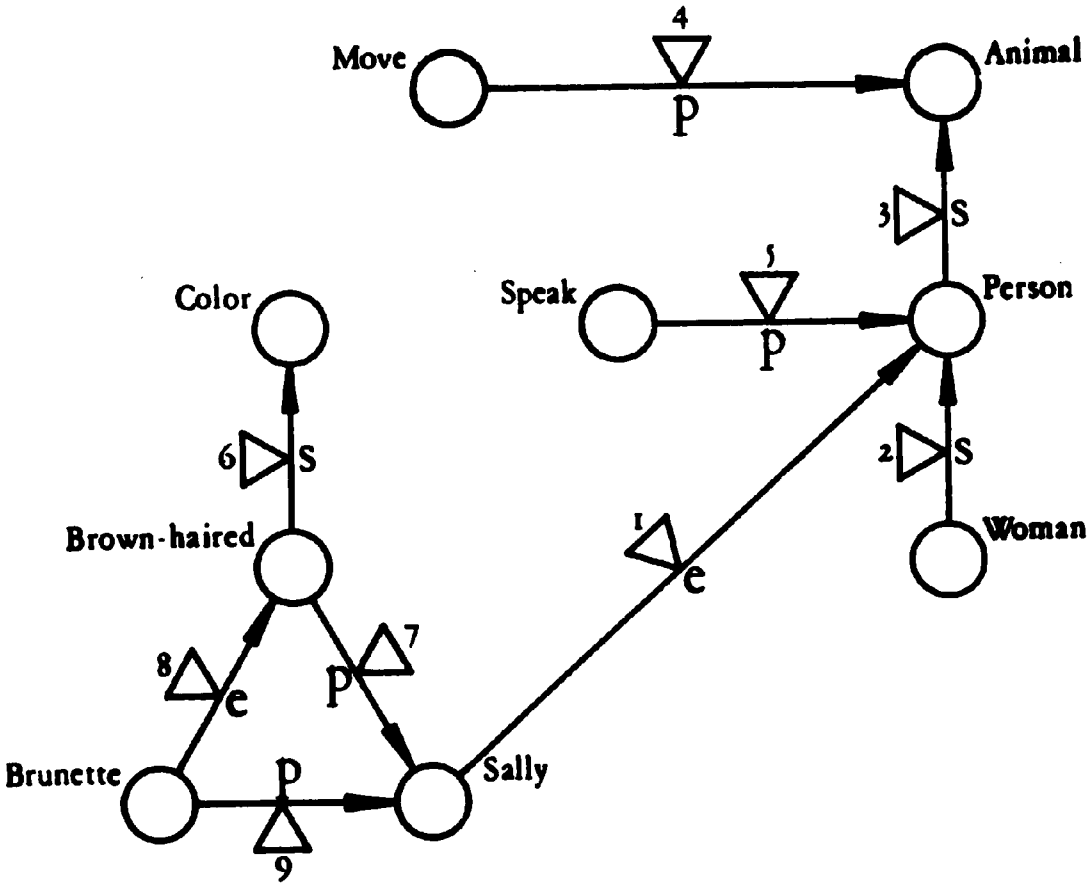
Move
4
p
Animal
3
s
Color
Speak
5
p
Person
6
s
2
s
Brown-haired
1
e
Woman
8
e
7
p
Brunette
p
Sally
9

FIGURE 1

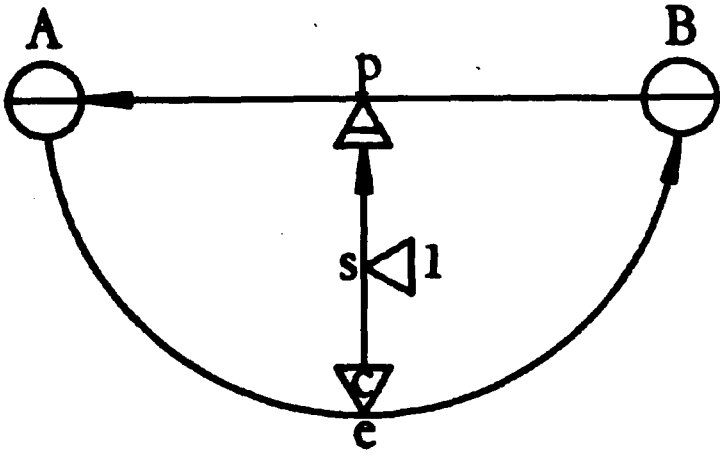

FIGURE 2

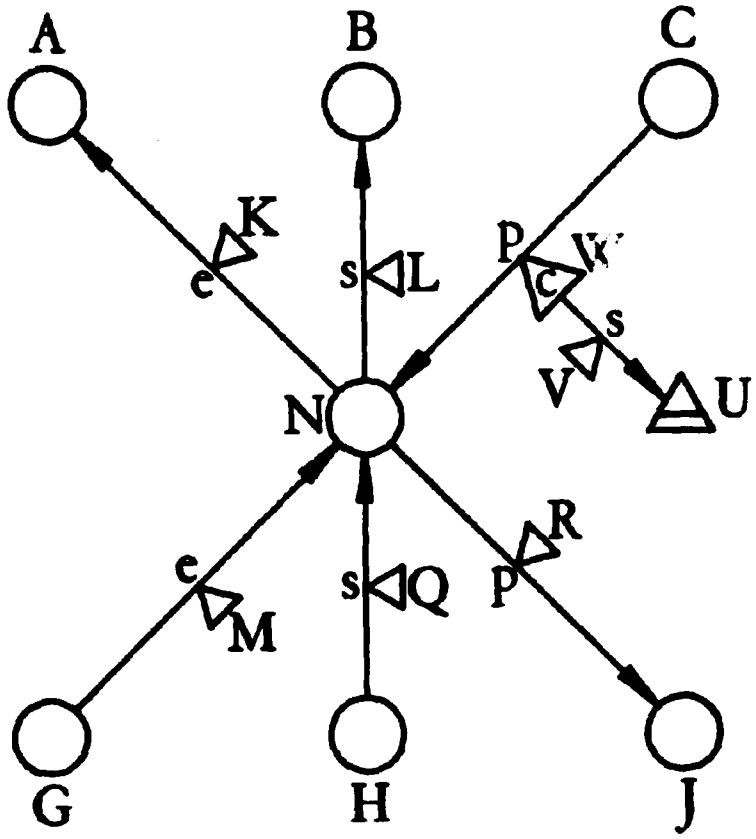

FIGURE 3

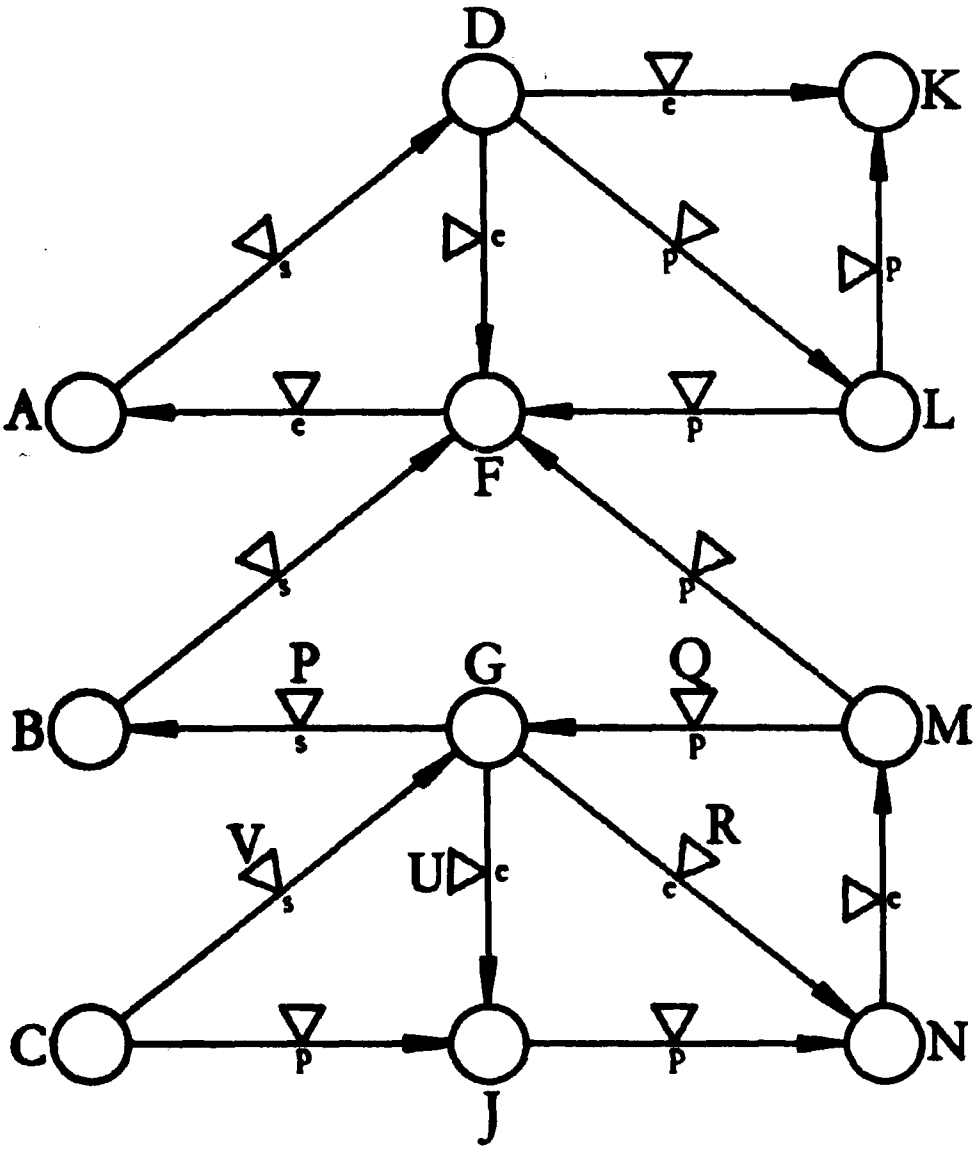

FIGURE 4

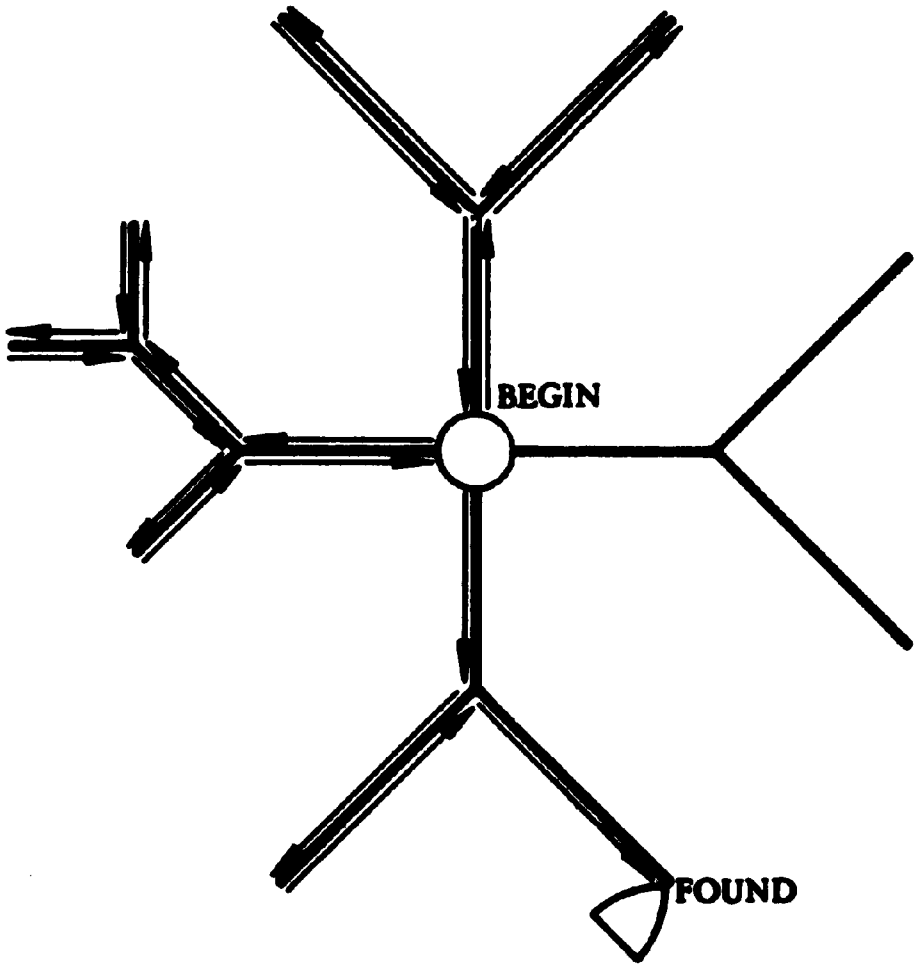

FIGURE 5

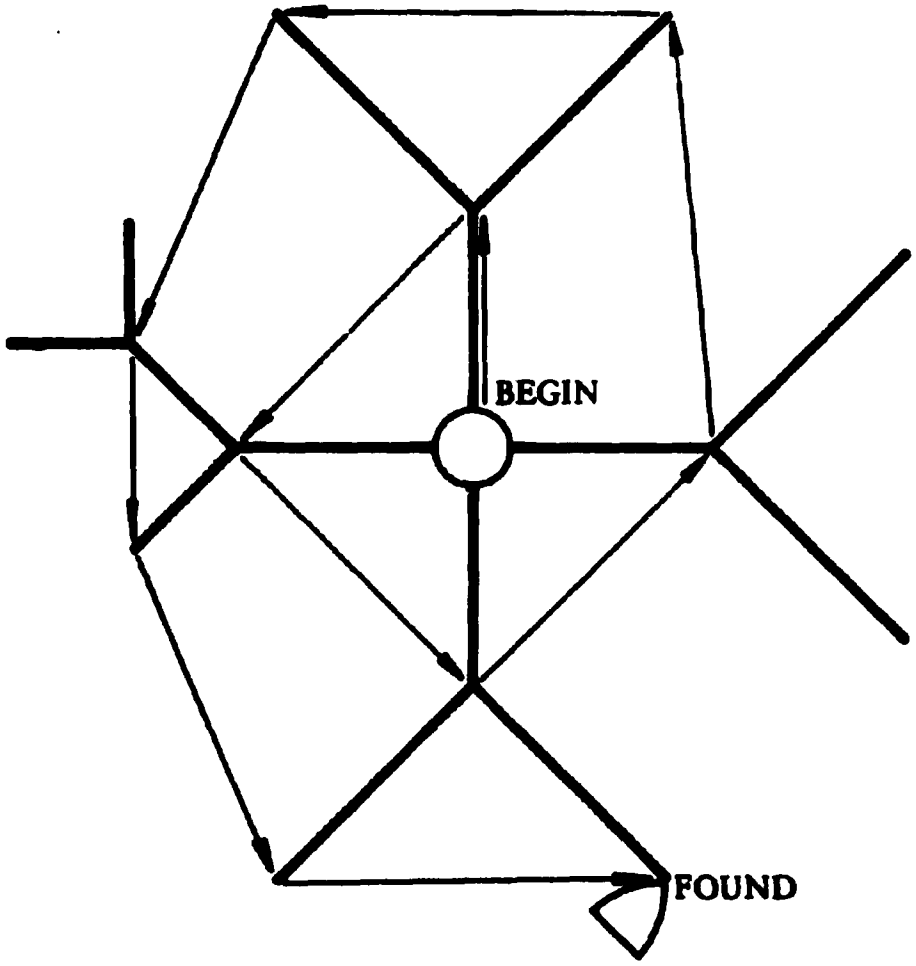

FIGURE 6

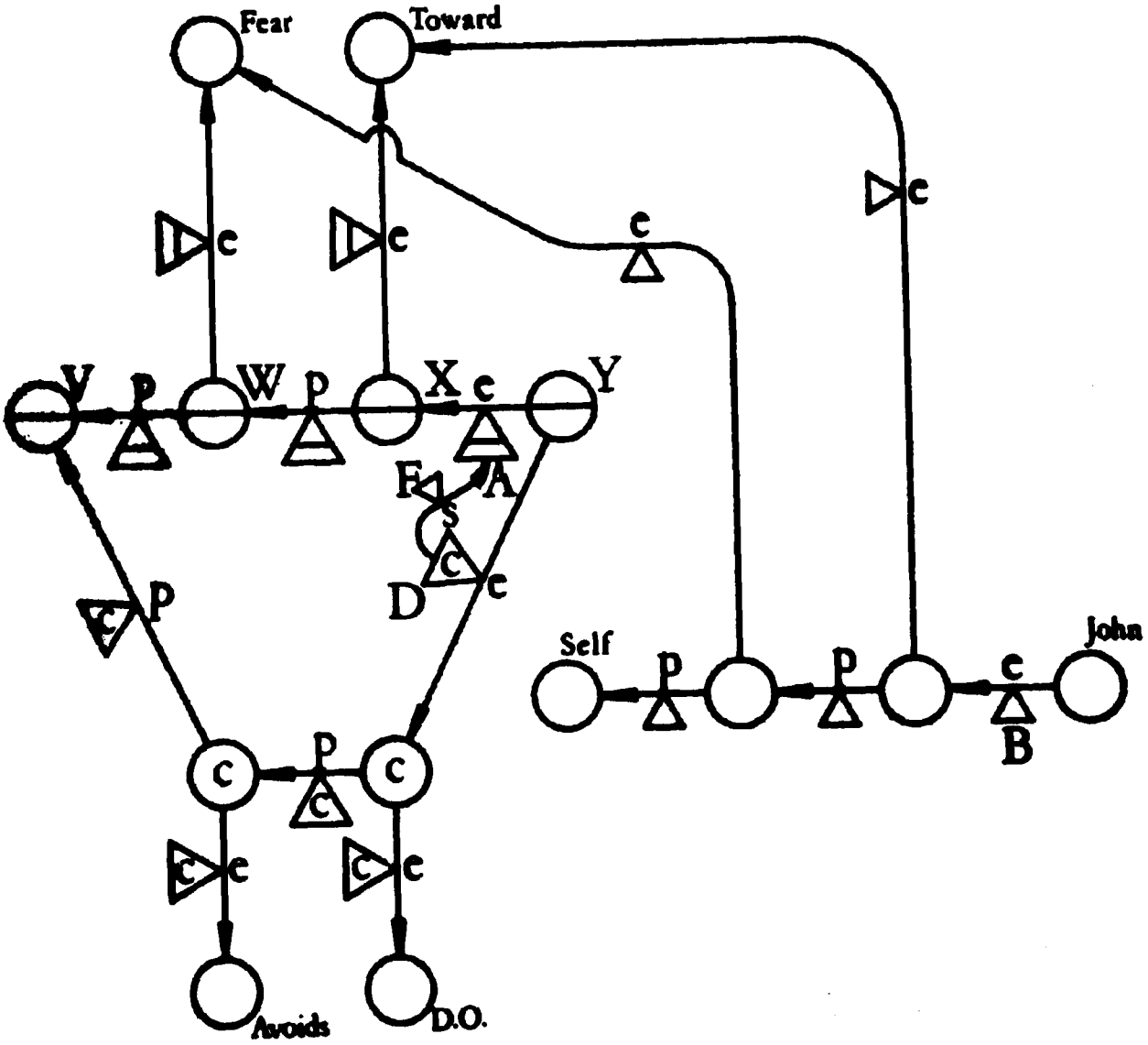
Fear
Toward
e
e
e
e
V
P
W
P
X
e
Y
F
A
S
C
D
e
P
Self
P
P
e
John
B
C
P
C
C
e
C
e
Avoids
D.O.

FIGURE 7

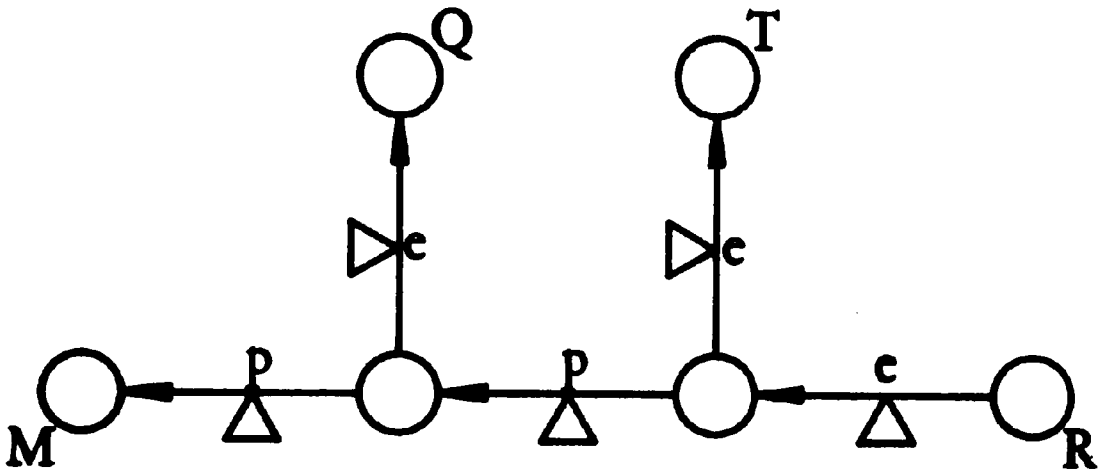

FIGURE 8