# AN ALGORITHM FOR APPROXIMATING VALIDITY FUNCTIONS

by

Phillip F. Smith

Submitted in Partial Fulfillment of the Requirements

for the Degree of

Master of Science

in the

Mathematics and Computer Science

Program

Adviser 12/10/87
Date

Dean of the Graduate School December 10, 1987
Date

YOUNGSTOWN STATE UNIVERSITY

December, 1987

# ABSTRACT

## AN ALGORITHM FOR APPROXIMATING VALIDITY FUNCTIONS

Phillip F. Smith

Master of Science

Youngstown State University, 1987

In this thesis, an algorithm is presented for the approximation of validity functions. It can be applied to approximate logical entailment in uncertain knowledge bases and is practical even if the knowledge base is large. In the process of developing the algorithm, several interesting results concerning validity functions are shown. A directed graph is used to conceptualize both the knowledge base and the behavior of the algorithm. Finally, this graph indicates a method for estimating conditional validities.

# ACKNOWLEDGEMENTS

I wish to thank my research advisor, Dr. Santos, and Dr. Burden and Dr. Schueller for reading this thesis. I also would like to express my thanks to Dr. Faires for his help in many ways.

# TABLE OF CONTENTS

# CHAPTER I

## INTRODUCTION

In the last several years, much interest has developed in trying to incorporate into knowledge based systems the ability to use and manipulate uncertain information. This is expecially true in the field of "expert systems." Very seldom does an expert have complete information or know all data with certainty. Thus, a great need arose for researchers to address this problem and since then the field has flourished.

One of the first AI systems to include methods for dealing with uncertainty was MYCIN,[1] a medical diagnosis system which used certainty factors. Many other techniques have since been developed and are based on a variety of concepts: Bayes rule,[2] Schafer-Dempster theory,[3] logic of likelihood,[4] fuzzy reasoning,[5] "probabilistic logic",[6] and others.

---

[1] E.H. Shortliffe, Computer Based Medical Consultations: MYCIN (New York: Elsevier, 1976).

[2] R.O. Duda, P.E. Hart, and N.J. Nilsson, "Subjective Bayesian Methods for Rule-base Inference Systems," in: Proceedings 1976 National Computer Conference, AFIPS 45 (1976) pp. 1075-1082.

[3] G.A. Schafer, Mathematical Theory of Evidence (Princeton: Princeton University Press, 1979).

[4] J.Y. Halpern and M.O. Rabin, "A Logic to Reason About Likelihood," Artificial Intelligence, 32 (1987) pp. 379-405.

[5] L.A. Zadeh, "Fuzzy Logic and Approximate Reasoning," Synthese, 30(1975) pp. 407-428.

[6] N.J. Nilsson, "Probabilistic Logic," Artificial Intelligence, 28 (1986) pp. 71-87.

The main objective of a knowledge based system is general logical entailment: Given a sentence S and a knowledge base $\mathcal{K}$, what is the truth value of S (probability of S, belief value of S, etc.) based on the information in $\mathcal{K}$? One approach is given by Nilsson[7] who discusses construction of a large matrix M used in the "probabilistic entailment" of S. Given M, linear programming can be used to determine the probability bounds of S. However, M can have dimensions, in general, $n \cdot 2^n$ (where n is the number of sentences in the knowledge base) which makes linear programming applicable only when the the knowledge base is very small. Even the construction of M itself is very difficult and time-consuming. Most of the other approaches mentioned share this same problem of being impractical for a realistic knowledge base. Clearly, an alternate approach is needed.

In this thesis, an algorithm will be presented which provides approximate bounds for the entailment of a sentence S, given a consistent knowledge base $\mathcal{K}$ (even if $\mathcal{K}$ is large). The knowledge base is restricted to contain only sentences which are minterms (conjunctions of atomic propositions or their negations). This is not a severe restriction since many knowledge bases could be expressed in this form. Also, it will be clear that the techniques presented could be modified to apply to more generalized knowledge bases. The algorithm is based on results from validity functions in Santos.[8] A directed graph is used to conceptualize the knowledge base and

---

[7]Nilsson, pp. 75-79.

[8]E.S. Santos and E. Santos Jr., "Reasoning With Uncertainty in a Knowledge Based System," in: Proceedings the Seventeenth International Symposium on Multiple-Valued Logic, (May 1987) pp. 75- 81.

information inferred through the algorithm. A study of the graph then indicates how the algorithm can estimate conditional validities.

# CHAPTER II

## PRELIMINARIES

### Validity Functions

The symbols $\sim$, $\vee$, $\wedge$, $\rightarrow$, $\equiv$, $\forall$, $\exists$, $\subset$, $\supset$, $\cap$, $\cup$, $\in$, **T** and **F** will denote, respectively, negation, disjunction, conjunction, implication, equivalence, for every, there exists, subset, superset, intersection, union, member, true, and false.

Let $\mathcal{P}$ be a collection of atomic propositions. $\mathcal{L}^*(\mathcal{P})$, or $\mathcal{L}^*$, denotes the set of all finite sentences over $\mathcal{P}$. $\mathcal{L} \subset \mathcal{L}^*$ is a knowledge base if $\mathcal{L}$ is finite.

Let v be a function from $\mathcal{L}^*$ into [0,1], the closed unit interval. v is a validity function if and only if v satisfies the following conditions:

(1) $v(S_1) = v(S_2)$ if $S_1 \equiv S_2$;

(2) $v(\mathbf{T}) = 1$, $v(\mathbf{F}) = 0$;

(3) $v(S_1) \leq v(S_2)$ if $S_1 \rightarrow S_2$; and

(4) $v(S_1 \vee S_2) = v(S_1) + v(S_2)$ if $S_1 \wedge S_2 \equiv \mathbf{F}$.

v(S) may be viewed as the probability that S is true, the certainty factor of S, the belief value of S, or any other appropriate interpretation.

**Theorem 1** Let v be a validity function. Then

(a) $v(\sim S) = 1 - v(S)$,

(b) $v(S_1) + v(S_2) = v(S_1 \vee S_2) + v(S_1 \wedge S_2)$, and

(c) $v(S_1 \rightarrow S_2) = 1 - v(S_1 \wedge \sim S_2)$.

Proof. See Santos.[9]

Let $\mathcal{L} \subseteq \mathcal{L}^*$. A function f from $\mathcal{L}$ into [0,1] is extensible if and only if there exists a validity function v such that $v(S)=f(S)$ for all $S \in \mathcal{L}$. v is called an extension of f.

A function g from $\mathcal{L}$ into $2^{[0,1]}$, the power set of [0,1], is extensible if and only if there exists a validity function v such that $v(S) \in g(S)$ for all $S \in \mathcal{L}$. v is called an extension of g. g is a generalization of f.

The functions f and g represent two ways of specifying what is known about the sentences in $\mathcal{L}$. Clearly, f signifies more complete knowledge of $\mathcal{L}$ than does g. If $S \in \mathcal{L}$ and f is given, there is full knowledge of S because $v(S)=f(S)$. If g is given then there is only partial knowledge of S since $v(S) \in g(S)$. Given $S_e$, a sentence to be entailed ($S_e \notin \mathcal{L}$), it is, in general, not possible to determine the validity of $S_e$ uniquely but only up to a certain bound. (This is obvious if g is given but also holds for f as well.)

Define $V(g,S_e) = \{v(S_e) : v \text{ is an extension of } g\}$. Let $v^+(g,S_e) = \text{lub } V(g,S_e)$ and $v^-(g,S_e) = \text{glb } V(g,S_e)$. $v^-(g,S_e)$ and $v^+(g,S_e)$ are called the lower and upper validity, respectively. $[v^-(g,S_e),v^+(g,S_e)]$ is the bound mentioned above and is the interval of consistent values for $v(S_e)$. If $v^-(g,S_e) = v^+(g,S_e)$ then $S_e$ is uniquely entailed by $\mathcal{L}$ with respect to g.

---

[9] Santos and Santos, p. 4.

In the rest of this discussion, $v^+(g,S)$ and $v^-(g,S)$ will be denoted $v^+(S)$ and $v^-(S)$. The function g is implied and assumed to be given with the knowledge base.

Let $\mathcal{K}^*(\mathcal{P})$, or $\mathcal{K}^*$, denote the set of all finite minterms over $\mathcal{P}$. $\mathcal{K} \subset \mathcal{K}^*$ is a restricted knowledge base if $\mathcal{K}$ is finite.

Suppose $S_e \in \mathcal{K}^*$ is the sentence to be entailed ($S_e \notin \mathcal{K}$). $av^-(S_e)$ and $av^+(S_e)$ will be used to denote the approximate lower and upper validity of $S_e$ calculated by the ensuing algorithm. It should be noted that $av^-(S_e) \le v^-(S_e) \le v^+(S_e) \le av^+(S_e)$. (1)

That is, the approximate upper and lower validity will approach the actual upper and lower validity always from the outside. This is extremely important. If an approximate bound [a,b] is found but it is not known whether $[v^-(S_e), v^+(S_e)] \subseteq [a,b]$, then the bound is not helpful. In fact, many approximations just find some single point value $c \in [0,1]$. Clearly, that is not nearly as meaningful as (1).

Finally, $\mathcal{K}$ will be used to refer to a (restricted) knowledge base in general; $\mathcal{K}^O \subset \mathcal{K}^*$, called the original knowledge base, refers to a specific set of sentences along with their associated validities; $\mathcal{K}^W \subset \mathcal{K}^*$, called the working knowledge base, refers to the sentences "created" by the algorithm as it tries to entail $S_e$ (the sentence of interest), given $\mathcal{K}^O$. (Note: $\mathcal{K}^W \cap \mathcal{K}^O = \emptyset$.) $\mathcal{K}^W$ also includes the current approximate upper and lower validity associated with each sentence in $\mathcal{K}^W$.

## Introduction to the Algorithm

The algorithm that will be discussed is called ENTAIL. Its general workings are as follows: Given a knowledge base $\mathcal{K}^{O}$ and $S_e$, the sentence to be entailed, ENTAIL calls four procedures, each with distinct goals in trying to approximate $v^{-}(S_e)$ and $v^{+}(S_e)$. When ENTAIL is finished, it returns $av^{-}(S_e)$ and $av^{+}(S_e)$, the approximate lower and upper validity for $S_e$. Sometimes information can be inferred directly from $\mathcal{K}^{O}$. Usually, however, these procedures recursively call ENTAIL with a new sentence, $S_{ei}$, to be entailed. (This is somewhat similar to backward-chaining theorem provers where given an initial goal, new subgoals are continually derived until one is proven true.) The entailing of this 'new' sentence $S_{ei}$ will cause ENTAIL to be called recursively with even more sentences which in turn will do the same.

ENTAIL can be viewed as constructing and then traversing a directed tree-like graph where each vertex corresponds to a sentence. Let $V^{*}$ be a set of vertices such that there exists a function $t:\mathcal{K}^{*}\longrightarrow V^{*}$ where t is onto and one-to-one. Therefore, $\forall\ S\in\mathcal{K}^{*}\ t(S)\in V^{*}$. Define a graph $G=(V,E)$ where $V\subset V^{*}$ and E is a set of directed edges. $V = \{t(S): S \in \mathcal{K}^{O}$ or $S \in \mathcal{K}^{W}\}$. In other words, V contains vertices which correspond to every sentence in both the working knowledge base and the original knowledge base. $(j_1, j_2) \in E$ implies that the sentence $S_2=t^{-1}(j_2)$ was used in entailing the sentence $S_1=t^{-1}(j_1)$. Specifically, it does not mean that $S_2$ *did* affect the validity of $S_1$, but rather that it *could* have affected it. (This will be important later when discussing the approximation of conditional validities.)

Suppose that the sentence being entailed is $S_e = A \wedge B \wedge C \wedge D$ and that both $S_1 = A \wedge B$ and $S_2 = C \wedge D$ were entailed directly because of $S_e$. In turn, assume that both $S_3 = \sim A \wedge B$ and $S_4 = A$ were entailed while working on $S_1$. This could be viewed as the graph in figure 1.

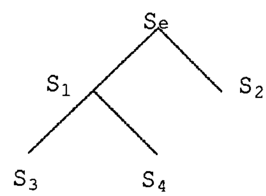


Fig. 1.--Graphical representation of the partial entailment of $S_e$

It is evident that ENTAIL traverses the graph in a depth-first order. Thus, some means must be provided to prevent ENTAIL from diving indefinitely. This is accomplished with a parameter called LEVEL. The value of LEVEL is the maximum depth that ENTAIL is allowed to reach. This is implemented by decrementing the value of LEVEL every time a recursive call to ENTAIL is made. For example, if ENTAIL is called with sentence $S_1$ and LEVEL=4, denoted ENTAIL($S_1$,4), and $S_2$ is now entailed, the LEVEL parameter for $S_2$ will be equal to 3 (ENTAIL($S_2$,3)). Clearly, at some point a sentence $S_i$ will be entailed with LEVEL=0. When this occurs, ENTAIL will not try to entail the sentence at all, but rather will return $av^-(S_i)=0$ and $av^+(S_i)=1$, meaning that nothing is known about $S_i$.

Two more points need to be made before discussing the algorithm in detail. If ENTAIL is called with the sentence S and $S \in \mathcal{K}^0$ (S is part of the original knowledge base) then the algorithm will not try to entail S but will simply return $av^-(S)=v^-(S)$ and $av^+(S)=v^+(S)$. (The values returned are the given validities for S.) This means that $S \in \mathcal{K}^0$

$\longrightarrow \nexists\ j_1 \in V$ such that $(j, j_1) \in E$ where $j=t(S)$. ($t(S)$ will have no children.) Vertices in the set $t(\mathcal{K}^O)=\{t(S): S \in \mathcal{K}^O\}$ are called *anchor* vertices. No edges originate from them. Edges can only terminate there. Also, all information inferred form the traversal of the graph is ultimately dependent on the sentences in $\mathcal{K}^O$ which correspond with the set $t(\mathcal{K}^O)$.

Secondly, what if ENTAIL is called with sentence S but S has already been enatiled? Should the approximate validities calculated previously be returned, or should S be entailed again? This question is handled by making use of the value of the LEVEL parameter. Suppose S had already been entailed with LEVEL=5 and now it is to be entailed with LEVEL=4. Clearly, there is no need to entail S any further for the depth of the subgraph rooted at t(S) is greater than what is needed. (The subgraph rooted at t(S) is the subgraph traversed while entailing S.) For $j \in V$, let L(j) denote the value of the LEVEL parameter used in entailing $S=t^{-1}(j)$. Suppose L(t(S))=2 and ENTAIL is called with sentence S again and LEVEL=4. It would not make much sense to return the current approximate validity of S because LEVEL=4 implies that more effort should be expended in entailing S than has already been done. Therefore, the subgraph rooted at t(S) is extended (to a depth of four) and the approximate upper and lower validity is updated.

The fact that a sentence S can be entailed more than once indicates that t(S) can have many parents and therefore the graph G cannot be a tree.

# CHAPTER III

## ALGORITHM ENTAIL

The algorithm is divided into five procedures: an initial and four main. It is assumed the algorithm was invoked as ENTAIL($S_e$,LEVEL).

### Initial Procedure

Let $P_e$ denote a parent sentence of $S_e$. (More accurately, $P_e$ is a sentence whose entailment resulted in the current entailing of $S_e$.)

```
if Se ∈ 𝒦^O then return av⁻(Se)=v⁻(Se) and av⁺(Se)=v⁺(Se)
if Se ∈ 𝒦^W (Se has already been entailed) then do
     if L(Se)≥LEVEL then return av⁻(Se) and av⁺(Se)
                                  (calculated previously)
                        else do
                              set L(Se):=LEVEL
                              goto PROCEDURE ONE
                        end do
end do
if LEVEL=0 then return av⁻(Se)=0 and av⁺(Se)=1
add t(Se) to V         (Se has never been entailed.)
add Se to 𝒦^W
L(t(Se)):=LEVEL
goto PROCEDURE ONE
```

## Procedure One

This procedure uses disjoint sentences to reduce the value of $av^+(S_e)$. Let $K^{p1} = \{S \in \mathcal{K}^O : S \wedge S_e = \mathbf{F}\}$. $K^{p1}$ contains all sentences in $\mathcal{K}^O$ which are disjoint with $S_e$. Clearly, $v^+(S_e) \le 1 - v^-(S)$ $(S \in \mathcal{K}^O)$. Suppose $S_e = A \wedge B$, $S_1 = {\sim}A \wedge C$, $S_2 = {\sim}B \wedge {\sim}C$, and $S_1, S_2 \in \mathcal{K}^O$. $S_e$ is disjoint with both $S_1$ and $S_2$. However, $S_1$ and $S_2$ are also disjoint. Therefore, $v^+(S_e) = 1 - (v^-(S_1) + v^-(S_2))$. Based on this idea, procedure one searches for a subset $K^{s1}$ of $K^{p1}$ in which every sentence in $K^{s1}$ is disjoint with every other sentence in $K^{s1}$ and the subset is maximum with respect to the sum of the lower validities of each sentence in $K^{s1}$, which will be denoted as $v^-(K^{s1})$. $\left(v^-(K^{s1}) = \sum_{S \in K^{s1}} v^-(S)\right)$. Again, $v^+(S_e) \le 1 - v^-(K^{s1})$. Set $av^+(S_e) \le 1 - v^-(K^{s1})$.

$K^{s1}$ is not necessarily a unique set. This does not matter, though, because what is important is not what $K^{s1}$ contains but rather the value of the maximum sum $(v^-(K^{s1}))$ mentioned above. The search for $K^{s1}$ can be thought of as being over all subsets of $K^{p1}$. In actuality, some simple heuristics are used to shorten this search. However, the worst case complexity of this search is d! where d is the size of $K^{p1}$.

It should be noted that procedure one does not entail any new sentences. It works directly with information in $\mathcal{K}^O$.

## Procedure Two

Let $K^{p2} = \{S \in \mathcal{K}^O : S \to S_e\}$. For any $S \in K^{p2}$ there exists a sentence denoted $(S - S_e)$ which has no atoms in common with $S_e$ and such that $S_e \wedge (S-S_e) \equiv S$. Call ENTAIL($(S-S_e)$,LEVEL-1). It is obvious that $v^-(S_e) \geq v^-(S)$. Set $av^-(S_e) \geq v^-(S)$. For the upper validity, the following theorem can be used:

**Theorem 2** $\quad v^+(S_e) \leq 1 - v^-((S-S_e)) + v^+(S)$ (2)

Proof. $\forall\, S_1, S_2 \quad v(S_1) + v(S_2) - 1 \leq v(S_1 \wedge S_2)$ (Theorem 1)

$\Longleftrightarrow v(S_e) + v((S-S_e)) - 1 \leq v(S_e \wedge (S-S_e)) = v(S)$

$\Longleftrightarrow v(S_e) \leq 1 + v(S) - v((S-S_e))$

Since v is arbitrary, $\Rightarrow \exists\, v_0$ (a specific validity function) such that $v^+(S_e) \leq 1 + v_0(S) - v_0((S-S_e))$

$\leq 1 + v^+(S) - v^-((S-S_e))$ □

Thus, set $av^+(S_e) \leq 1 - av^-((S-S_e)) + v^+(S)$. (3)

For example, suppose $S_e = A \wedge B$, $S_1 = A \wedge B \wedge C \in \mathcal{K}^O$, $v^+(S_1) = .5$, and $S_1 \in \mathcal{K}^O$. $S_1 \rightarrow S_e$ so $S_1 \in K^{p2}$ and $v^-(S_e) \geq v^-(S_1)$. The sentence $(S_1 - S_e) = C$ is entailed and assume it returns $av^-(C) = .6$. Equation (3) then yields $av^+(S_e) \leq 1 - .6 + .5 = .9$.

Procedure two also uses another technique in approximating lower validities. Suppose $S_e = A \wedge B$, $S_1 = A \wedge B \wedge D \wedge E \in \mathcal{K}^O$, $S_2 = A \wedge B \wedge \sim D \in \mathcal{K}^O$, $v^-(S_1) = .3$, and $v^-(S_2) = .4$. $S_1 \rightarrow S_e$ and $S_2 \rightarrow S_e$ imply $S_1, S_2 \in K^{p2}$. Thus, $v^-(S_e) \geq v^-(S_1) = .3$ and $v^-(S_e) \geq v^-(S_2) = .4$. But $S_1$ and $S_2$ are disjoint. Therefore, $v^-(S_e) \geq v^-(S_1) + v^-(S_2) = .3 + .4 = .7$. In a similar manner as in procedure one, procedure two searches for a subset $K^{s2}$ of $K^{p2}$ in which every $S \in K^{s2}$ is disjoint with every other $S \in K^{s2}$ and such that $v^-(K^{s2})$ is a maximum. $v^-(S_e) \geq v^-(K^{s2})$.

## Procedure Three

Let $K^{p3} = \{S \in \mathcal{K}^O : \exists\, S'$ such that $S' \leftarrow S_e$, $S' \supseteq S$, and $S' \neq \mathbf{T}\}$. $K^{p3}$ contains all sentences in $\mathcal{K}^O$ which share at least one common atom with

$S_e$. If $S \in K^{p3}$, let $c(S)$ denote the sentence comprised of all atoms that $S$ and $S_e$ have in common. For example, if $S=A \wedge B \wedge C$ and $S_e=A \wedge C \wedge D \wedge E$ then $c(S)=A \wedge C$. Let $K^c = \{c(S) : S \in K^{p3}\}$. For every $S_c \in K^c$, call ENTAIL($S_c$,LEVEL-1). Clearly, $S_c \in K^c$ implies $v^+(S_e) \le v^+(S_c)$ and thus set $av^+(S_e) \le v^+(S_c)$.

Let $S_1, S_2, \ldots, S_n \in K^c$ such that $S_1 \wedge S_2 \wedge \ldots \wedge S_n \subseteq S_e$. (4)

**Theorem 3** Given that (4) is true,

$$v^-(S_e) \ge v^-(S_1)+v^-(S_2)+\ldots+v^-(S_n) - (n-1). \quad (5)$$

Proof. (Via mathematical induction) For proof when n=2 see Santos.[10] Assume true for n<m.

$v(S_e) \ge v(S_1 \wedge S_2 \wedge \ldots \wedge S_m)$ (Theorem 1)

$\Rightarrow v^-(S_e) \ge v^-(S_1 \wedge S_2 \wedge \ldots \wedge S_m)$

$\ge v^-(S_1) + v^-(S_2 \wedge S_3 \wedge \ldots \wedge S_m) - 1$ (n=2<m)

$\ge v^-(S_1) + \left[v^-(S_2) + v^-(S_3) + \ldots + v^-(S_m) - (m-2)\right] - 1$

(n=m-1<m)

$= v^-(S_1) + v^-(S_2) + v^-(S_3) + \ldots + v^-(S_m) - (m-1)$ □

Therefore, set $av^-(S_e) \ge v^-(S_1) + v^-(S_2) + \ldots + v^-(S_n) - (n-1)$. For example, let $S_e=A \wedge B \wedge C$, $S_1=A \wedge B$, $S_2=B \wedge C$, $v^-(S_1)=.7$, and $v^-(S_2)=.6$. By theorem 3, $v^-(S_e) \ge v^-(S_1) + v^-(S_2) - (2-1) = .7 + .6 - 1 = .3$. Procedure three searches for a subset $K^{s3}$ of $K^c$ that maximizes equation (5) and then sets $av^-(S_e)$ greater than or equal to that maximum.

However, theorem 3 can be generalized even further.

**Theorem 4** Assume $S_1 \wedge S_2 \wedge \ldots \wedge S_n \subseteq S_e$ and let $S'' \in \mathcal{K}^*$.

$$v^-(S_e) \ge v^-(S_1 \wedge S'') + v^-(S_2 \wedge S'') + \ldots + v^-(S_n \wedge S'') - (n-1)v^+(S'') \quad (6)$$

---

[10] Santos and Santos, p. 79.

Proof. (Mathematical induction and theorem 1 (b) will be used.)

Show for n=2. $v(S_e) + v(S'') \geq v(S_1 \wedge S_2) + v(S'')$ $(S_1 \wedge S_2 \subseteq S_e)$

$= v(S_1 \wedge S_2 \wedge S'') + v((S_1 \wedge S_2) \vee S'')$ (Theorem 1 (b))

$= v(S_1 \wedge S'') + v(S_2 \wedge S'') - v((S_1 \wedge S'') \vee (S_2 \wedge S'')) + v((S_1 \wedge S_2) \vee S'')$

$\geq v(S_1 \wedge S'') + v(S_2 \wedge S'')$ $\big[-v((S_1 \wedge S'') \vee (S_2 \wedge S'')) + v((S_1 \wedge S_2) \vee S'') \geq 0$

since $(S_1 \wedge S_2) \vee S'' \leftarrow (S_1 \wedge S'') \vee (S_2 \wedge S'')\big]$

$\Longleftrightarrow v(S_e) \geq v(S_1 \wedge S'') + v(S_1 \wedge S'') - v(S'')$

Since v is arbitrary,

$v^-(S_e) \geq v_0(S_1 \wedge S'') + v_0(S_2 \wedge S'') - v_0(S'')$ (for some specific $v_0$)

$\geq v^-(S_1 \wedge S'') + v^-(S_2 \wedge S'') - v^+(S'')$

Assume (6) is true for n<m.

$v^-(S_e) \geq v^-(S_1 \wedge S'') + v^-(S_2 \wedge S'') + \ldots + v^-(S_{m-2} \wedge S'') + v^-(S_{m-1} \wedge S_m \wedge S'')$

$- (m-2)v^+(S'')$ (n=m-1<m)

$\geq v^-(S_1 \wedge S'') + v^-(S_2 \wedge S'') + \ldots + v^-(S_{m-2} \wedge S'')$

$+ \left[v^-(S_{m-1} \wedge S'') + v^-(S_m \wedge S'') - v^+(S'')\right] - (m-2)v^+(S'')$ (n=2<m)

$= v^-(S_1 \wedge S'') + v^-(S_2 \wedge S'') + \ldots + v^-(S_m \wedge S'') - (m-1)v^+(S'')$ □

Suppose $S_e = A \wedge B$, $v^-(A \wedge C) = .4$, $v^-(B \wedge C) = .5$, and $v^+(C) = .6$. Then, $v^-(S_e) \geq v^-(A \wedge C) + v^-(B \wedge C) - (2-1)v^+(C) = .4 + .5 - .6 = .3$. This is implemented by defining a function $r: K^c \longrightarrow K^{p3}$. $r(S_c)$ $(S_c \in K^c)$ is the sentence in the original knowledge base of which $S_c$ is a "part". That is, define r such that $c(r(S_c)) = S_c$. (Recall that $K^c = \{c(S) : S \in K^{p3}\}$ where c(S) denoted the sentence comprised of all atoms that S and $S_e$ have in common.) $\forall S_c \in K^c$ $r(S_c) \subset S_c$. Given $S_1, S_2, \ldots, S_n \in K^c$ satsifying (4), it is obvious that $r(S_1)$, $r(S_2)$, ..., $r(S_n)$ also satisfy (4). Find $S'' \in \mathcal{K}^*$ such that

$r(S_i) \subset S''$ for $i=1,2,\ldots,n$ and $c(S'') = \mathbf{T}$. (7)

($c(S'') = \mathbf{T}$ means that S" and $S_e$ have no atoms in common.) Call

ENTAIL(S",LEVEL-1).

$v^-(S_e) \geq v^-(r(S_1)) + v^-(r(S_2)) + \ldots + v^-(r(S_n)) - (n-1)v^+(S")$ (8)

Set $av^-(S_e) \geq v^-(r(S_1)) + v^-(r(S_2)) + \ldots + v^-(r(S_n)) - (n-1)av^+(S")$.

What if the only S" satisfying (7) is S"=**T**? Substitute in equation (8) $v^+(\mathbf{T})=1$ for $v^+(S")$, and (8) (almost) reduces to equation (5). In this case, though, $v^-(r(S_i))$ is used as opposed to $v^-(S_i)$. Since $v^-(r(S_i)) \leq v^-(S_i)$, the 'reduced' equation (8) will not be as accurate as (5).

Just as it did with equation (5), procedure three searches for a subset of $K^c$ that maximizes equation (8) and sets $av^-(S_e)$ greater than or equal to that maximum.

Procedure three is called the "heuristic parts" entailer. This is because every sentence $S_c \in K^c$ that is entailed is a "part" of $S_e$. Also, this procedure does not entail blindly but uses some heuristics, entailing only those parts for which it has detected that there is information in the knowledge base.

It should be noted that this procedure also incorporates sentences in the original knowledge base that are supersets of $S_e$. ($S \in \mathcal{K}^O$ and $S_e \subset S$ imply that $S \in K^{p3}$.) Sentences of this type are particularly valuable. (In earlier versions of ENTAIL, a separate procedure was used for such sentences. It soon became apparent, though, that it was more efficient to handle them in the "parts" entailer, procedure three.)

## Procedure Four

**Theorem 5** Let $S \in \mathcal{K}^*$ and $A \in \mathcal{P}$ ($\mathcal{P}$ is the set of atomic propositions).

$v^+(S \wedge A) \leq v^+(S) - v^-(S \wedge \sim A)$ and (9)

$v^-(S \wedge A) \geq v^-(S) - v^+(S \wedge \sim A)$ (10)

Proof. Clearly, $v(S\wedge A) = v(S) - v(S\wedge\sim A)$

Since v is arbitrary, $v^{+}(S\wedge A) \leq v_0(S) - v_0(S\wedge\sim A)$ for some specific $v_0$

$\Rightarrow v^{+}(S\wedge A) \leq v^{+}(S) - v^{-}(S\wedge\sim A)$ (Proof of (10) is similar.) □

Procedure four uses theorem 5 in the following manner. Suppose $S_e = A\wedge B\wedge C$. Entail the sentences $S_1 = A\wedge B$, $S_2 = A\wedge B\wedge\sim C$, $S_3 = A\wedge C$, $S_4 = A\wedge\sim B\wedge C$, $S_5 = B\wedge C$, and $S_6 = \sim A\wedge B\wedge C$. Now apply equation (9) to get:

$$v^{+}(S_e) \leq v^{+}(S_1) - v^{-}(S_2)$$

$$v^{+}(S_e) \leq v^{+}(S_3) - v^{-}(S_4)$$

$$v^{+}(S_e) \leq v^{+}(S_5) - v^{-}(S_6)$$

Equation (10) can be applied similarly.

For every atomic proposition A in $S_e$, two corresponding sentences are entailed: $(S_e - A)$, essentially $S_e$ with A removed, and $(S_e - A)\wedge\sim A$. The results are then combined as in theorem 5:

$$v^{+}(S_e) \leq v^{+}((S_e - A)) - v^{-}((S_e - A)\wedge\sim A)$$

$$v^{-}(S_e) \geq v^{-}((S_e - A)) - v^{+}((S_e - A)\wedge\sim A)$$

If b is the number of atomic propositions in $S_e$, procedure four entails 2b sentences. This could be a problem if $S_e$ has many atoms. However, most sentences are not (relatively) very 'long' and, more importantly, they do not grow in length as the knowledge base grows.

Procedure four is a 'blind' procedure in that, given a specific $S_e$, it entails the same 2b sentences no matter what. It does not examine $\mathcal{K}^{O}$ for help in deciding what to entail. If $\mathcal{K}^{O}$ is changed, the first three procedures will adapt accordingly because they use some heuristics. Procedure four, however, will not adapt. The reason for this design is at this time there could not be found a suitable mechanism for detecting when and to what theorem 5 should be applied. Therefore, the theorem is applied to all possibilities.

The need for procedure four became apparent from experimentation. It is a generalization of two procedures from earlier versions of ENTAIL. It often utilizes information not easily discernable to the human observer. One interesting application of theorem 5 is entailing S when ~S is known. S can be viewed as $S \wedge \mathbf{T}$.

$$v^{+}(S) = v^{+}(S \wedge \mathbf{T}) \leq v^{+}(\mathbf{T}) - v^{-}(\sim S) = 1 - v^{-}(\sim S) \tag{11}$$

$$v^{-}(S) = v^{-}(S \wedge \mathbf{T}) \geq v^{-}(\mathbf{T}) - v^{+}(\sim S) = 1 - v^{+}(\sim S) \tag{12}$$

Clearly, the final results in (11) and (12) are obvious. What is interesting is that theorem 5 can be used to obtain these results. This provides a method for using procedure four to calculate the validity of S directly from ~S which would otherwise require a special procedure of its own.

## Remarks

Algorithm ENTAIL is redundant in that many sentences will be entailed more that once. This is really an advantage, however, because work is not duplicated. Information learned from previous entailments will be reused thus resulting in a great savings of time. Another advantage is that the working knowledge base ($\mathcal{K}^{w}$) itself can be stored and used later. Suppose that five sentences need to be entailed. If the linear programming methods mentioned previously are used, the matrix M must be reconstructed for each sentence and five separate linear programming problems solved. ENTAIL, on the other hand, can save $\mathcal{K}^{w}$ from one sentence to the next. Thus, information already gained is never lost.

The complexity of the graph is order $n^{L}$ where n is the size of the original knowledge base $\mathcal{K}^{O}$ and L is the maximum depth allowed (the initial value of LEVEL). It should be noted that the complexity of the algorithm is not dependent on $\mathcal{K}^W$, the working knowledge base. The reason is that although ENTAIL makes use of $\mathcal{K}^W$, by design it only entails new sentences based on $\mathcal{K}^O$. Changes in $\mathcal{K}^W$ do not affect ENTAIL while changes in $\mathcal{K}^O$ do. This is justifiable because $\mathcal{K}^W$ really contains no 'new' information. Everything in $\mathcal{K}^W$ is dependent on $\mathcal{K}^O$. (Recall that this is the reason $V^O=t(\mathcal{K}^O)$ is called the anchor set for the graph G.)

Finally, at a first glance some of the techniques used might appear ad hoc. A closer look, however, at the first three procedures reveals that sentences in the knowledge base are viewed as being part of four different categories. $K^{p1}$ (sentences used by procedure one) contains sentences which are disjoint with $S_e$. $K^{p2}$ (sentences used by procedure two) contains sentences which are subsets of $S_e$. $K^{p3}$ includes sentences which are supersets of $S_e$ and also those which share common atoms with $S_e$. From this standpoint ENTAIL proceeds in an orderly fashion. Procedure four is not fully understood and needs to be studied further.

In just a few months of testing three procedures were removed, having been generalized with others. It is believed that with further experimentation the algorithm will generalize even further.

# CHAPTER IV

## APPROXIMATING CONDITIONAL VALIDITY

One very important function of a knowledge based system is to determine how a change in the validity of one sentence can affect another. This is termed conditional validity and is denoted by $v^-(S_e|S)$ and $v^+(S_e|S)$, the lower and upper validity of $S_e$ given S.

It will be shown how the graph G constructed by ENTAIL makes this calculation easy. But first two observations need to be made. A sentence $S' \in K$ is only directly affected by its immediate children. The validity of S' can only change if the validity of one of its children change. ($S_1$ is a child of S' if $(S',S_1)\in E$.) Secondly, recall that $(S',S_1)\in E$ does not mean $S_1$ did affect S', only that it could have affected S (in relation to this algorithm).

The algorithm presented below will calculate conditional upper and lower validities for all $S\in\mathcal{K}^W$. Let $S_n\in\mathcal{K}^O$. $S_n$ is the sentence with 'new' or changed upper and lower validity. $t:K\longrightarrow V$ is defined as before. Let $P(S)=\{S_p: (t(S_p),t(S))\in E\}$ called the parent set of S.

```
        x := Sn
        SET := P(x)
LOOP    if SET=ø then stop
        remove from SET any sentence S
        x := S
        call ENTAIL(x,LEVEL=1)
        if (av+(x|Sn)=av+(x) and av-(x|Sn)=av-(x))
                then goto LOOP
```

else do

| | |
|---|---|
| $av^+(x) := av^+(x \mid S_n)$ | If the validity of x has |
| $av^-(x) := av^-(x \mid S_n)$ | changed, there is the |
| SET := SET $\cap$ P(x) | possibility that some in P(x) |
| goto LOOP | might change. |

end

The changed validity of $S_n$ spreads or propagates upward from the vertex $t(S_n)$. If a sentence $S \in \mathcal{K}$ has its upper or lower validity modified due to $S_n$, there exists a directed path from $t(S_n)$ to $t(S)$ such that for every vertex j in the path, the upper or lower validity of $S_j = t^{-1}(j)$ was changed as well. Let $K_{cv}(S_n)$ denote the set of all $S \in \mathcal{K}$ such that the conditional validity of S based on $S_n$ is different from the 'normal' validity of S. $K_{cv}$ must be a connected subgraph of G. This explains why a shift in the validity of a sentence can affect only a very localized area of the knowledge base.

Often it is also useful to observe how a change in the validity of one sentence in $\mathcal{K}^O$ can affect the other sentences in $\mathcal{K}^O$. Up to this point, it was mentioned that ENTAIL would not attempt to entail sentences in $\mathcal{K}^O$. It would simply return the given validities. ENTAIL does, however, have a simple modification which allows it to entail any $S \in \mathcal{K}^O$ just as if $S \notin \mathcal{K}^O$. By allowing this, the algorithm presented in this section can be used to approximate $v^-(S_1 \mid S_2)$ and $v^+(S_1 \mid S_2)$ where both $S_1, S_2 \in \mathcal{K}^O$.

# CHAPTER V

## EMPIRICAL STUDIES

Thorough testing of ENTAIL presents two problems. First, the construction of large consistent varied knowledge bases is not easy. This is especially true if attempts are made to assign sentences validities near the bounding limits of consistency. The more difficult problem, however, is computing the exact validity of the sentence being entailed for comparison with those produced by the algorithm. The method used for computing the comparison is that given by Nilsson mentioned earlier which involves the matrix M and linear programming. M can have dimensions $n \cdot 2^n$ (where n is the number of sentences in the knowledge base) and thus, the linear programming problem will have, in general, n variables and $2^n$ constraints. Therefore, the construction of M, the conversion of M to a suitable linear programming problem (which requires extensive manipulation of the entire matrix M), and the solving of the linear programming problem itself all are prohibitively expensive in both time and memory requirements. Thus, the size of the knowledge bases used for testing had to be severely restricted.

It should be clear from the foregoing that the real problem in testing is not connected with ENTAIL but with computing the comparisons. In fact, ENTAIL did not need more than five seconds for any test run while the linear programming method required several minutes for even relatively small test cases.

The preceding discussion should also explain why the testing of ENTAIL would be slow and thus the results still preliminary. Over fifty knowledge bases have been tested and ENTAIL has calculated the exact correct validity function values for all but one of the test cases. It is not known yet why ENTAIL could not produce the correct values for this one knowledge base. The exact validity interval for this test case was [.251, .571] while ENTAIL returned [.129, .571]. This approximate interval is still correct (in that $.129 < v(S) < .571$ is still true) but the actual interval is smaller. This emphasizes why the interval estimate provided by ENTAIL is much more valuable than the point estimates mentioned previously. A meaningful decision can be made on the basis of this interval estimate even if it is not exact.

The number of sentences in the knowledge bases varied between 3 and 30 with the majority (40) being between 10 and 20. Only three knowledge bases needed a LEVEL parameter value of 3 before the exact results were obtained. All others needed only a LEVEL value of 2 or less. This raises the question of whether information deeper in the graph could affect the interval estimate. The above evidence supports the intuitive notion that the chance of the entailed sentence being affected by its descendants decreases rapidly as depth increases. This is also supported by the fact that the number of children of any vertex is order n (worst case). The breadth of children for any single vertex will increase linearly with the size of the knowledge base. Thus, it is not necessarily true that a larger knowledge base will require a deeper depth because much of the 'extra' information will be incorporated in the 'shallower' levels of the graph. This is extremely important because the amount of time required by ENTAIL increases exponentially with the value of LEVEL.

# CHAPTER VI

## CONCLUSION

The preceding discussion has presented a fast algorithm for the approximation of validity functions, an analysis of the algorithm's behavior, an application to condtional validities, and some empirical results.

One interesting problem left is that of finding what "gaps" exist in the algorithm. What kind of information will ENTAIL miss? The continued study of these gaps will provide understanding into exactly how the validity of one sentence affects another. Recall that in the graph G two sentences might be connected by many different paths. Each path implies, in a sense, a slightly different dependency between the two sentences. Could those dependencies be classified? In turn, might knowledge bases themselves be grouped based on the inherent dependencies of their member sentences? More importantly, is there some restriction which if imposed on a knowledge base would guarantee an exact solution by an algorithm such as ENTAIL?

Another topic of concern is the significance of the LEVEL parameter (the maximum depth ENTAIL is allowed to dive). What depth should be used in general? What kinds of sentences will need deeper depths to obtain accurate approximations? Is the needed depth dependent on the size of the knowledge base?

Some improvements in the algorithm might be the calculating of some error terms $\mathcal{E}^-$ and $\mathcal{E}^+$ such that $av^-(S_e) \leq v^-(S_e) \leq av^-(S_e) + \mathcal{E}^-$ and $av^+(S_e) \geq v^+(S_e) \geq av^+(S_e) - \mathcal{E}^+$, as well as designing a procedure to detect when a branch of the graph will not affect the approximation and therefore should not be traversed further. On a more practical level, it may only be important to know if $v^{-}(S_e)\geq a$, $v^{+}(S_e)\leq b$, or even that $\left|v^{-}(S_e)-v^{+}(S_e)\right|\leq c$ (for some bounds a, b, and c). The algorithm could easily be modified to handle all these cases.

In conclusion, it is hoped that this algorithm along with its analysis will provide a practical and efficient means for approximating validity functions and that the discussion in general will further the study of the effects of uncertainty in knowledge based systems.

# APPENDIX A

## Sample Knowledge Bases

ORIGINAL KNOWLEDGE BASE

| | | |
|---|---|---|
| CB^DY | ( 0.1986, | 0.1986) |
| D^C^A | ( 0.0287, | 0.0287) |
| ABC | ( 0.0849, | 0.0849) |
| AZ^R^B | ( 0.2000, | 0.2000) |
| ZY | ( 0.6000, | 0.6000) |
| RBA^ | ( 0.2030, | 0.2030) |
| CD^ | ( 0.4000, | 0.4000) |
| ESX | ( 0.9000, | 0.9000) |
| E^BR | ( 0.0660, | 0.0660) |
| Z^EA^ | ( 0.1700, | 0.1700) |
| YZSB^ | ( 0.2500, | 0.2500) |
| Y^ZAX^DB | ( 0.0200, | 0.0200) |
| R | ( 0.6550, | 0.6550) |
| XR^ | ( 0.3200, | 0.3200) |

NOTE: ^ stands for postfix negation. A^B^CD^ is equivalent to A'B'CD'.

NUMBER OF LEVELS = 2
NUMBER OF SENTENCES = 14
NUMBER OF SENTENCES ADDED = 13

SENTENCE TO BE ENTAILED

DY^C^E ( 0.0000, 0.3727)

WORKING KNOWLEDGE BASE

| | | | LEVEL |
|---|---|---|---|
| DY^C^E | ( 0.0000, | 0.3727) | 2 |
| C^DY^ | ( 0.0000, | 0.3727) | 1 |
| DC^ | ( 0.0000, | 0.3727) | 1 |
| Y^D | ( 0.0200, | 0.3727) | 1 |
| Y^ | ( 0.0200, | 0.4000) | 1 |
| C^ | ( 0.0287, | 0.4014) | 1 |
| E | ( 0.9000, | 0.9340) | 1 |
| D^Y^C^E | ( 0.0000, | 0.3800) | 1 |
| Y^C^E | ( 0.0000, | 0.4000) | 1 |
| DYC^E | ( 0.0000, | 0.3527) | 1 |
| DC^E | ( 0.0000, | 0.3727) | 1 |
| DY^CE | ( 0.0000, | 0.3727) | 1 |
| DY^E | ( 0.0000, | 0.3727) | 1 |
| DY^C^E^ | ( 0.0000, | 0.1000) | 1 |

NOTE: In the following partial trace,
E1 means procedure ENTAIL1 has started;
E3 means procedure ENTAIL3 has started;
E4 " " ENTAIL4 " "
E6 " " ENTAIL6 " "

ENTAIL1 corresponds to procedure one.
ENTAIL3 " " " two.
ENTAIL4 " " " three.
ENTAIL6 " " " four.

```
                                     LEVEL
ENTAIL   DY^C^E                        2
]E1
DY^C^E              ( 0.0000,   0.3727)   UPPER
]E3
]E4
]          ENTAIL   C^DY^                      1
```

```
]E1
C^DY^                    (  0.0000,    0.3727)   UPPER
]E3

]E4
]
]
]
]
]
]

]E6
]
]

]
]

]
]

C^DY^                    (  0.0000,    0.3727)
ENTAIL    DC^                        1
]E1
DC^                      (  0.0000,    0.3727)   UPPER
]E3

]E4
]
]

]E6
]
]

]
]

DC^                      (  0.0000,    0.3727)
ENTAIL    Y^D                        1
]E1
Y^D                      (  0.0000,    0.3727)   UPPER
]E3
]
Y^D                      (  0.0200,    1.0200)   LOWER

]E4
]
]
]
]

]E6
]
]

]
]
```

```
]          Y^D                    (  0.0200,    0.3727)
]          ENTAIL    Y^                    1
]          ]E1
]          Y^                     (  0.0000,    0.4000)   UPPER
]          ]E3
]          ]
]          Y^                     (  0.0200,    1.0200)   LOWER
]
]          ]E4
]
]          ]E6
]          ]
]
]          Y^                     (  0.0200,    0.4000)
]          ENTAIL    C^                    1
]          ]E1
]          C^                     (  0.0000,    0.4014)   UPPER
]          ]E3
]          ]
]          C^                     (  0.0287,    1.0287)   LOWER
]
]          ]E4
]
]          ]E6
]          ]
]
]          C^                     (  0.0287,    0.4014)
]          ENTAIL    E                     1
]          ]E1
]          E                      (  0.0000,    0.9340)   UPPER
]          ]E3
]          ]
]          E                      (  0.9000,    1.9000)   LOWER
]          ]
]
]
]          ]E4
]
]          ]E6
]          ]
]
]          E                      (  0.9000,    0.9340)
]E6
]          ENTAIL    D^Y^C^E               1
]          ]E1
]          D^Y^C^E                (  0.0000,    0.3800)   UPPER
]          ]E3
]
]          ]E4
]          ]
]          ]
]          ]
]          ]
]
1]         ]E6
]          ]     ENTAIL   DY^C^E               ALREADY DONE
]          ]
]
]          ]
]          ]
```

```
]
]

]
]
D^Y^C^E             (  0.0000,    0.3800)
ENTAIL    Y^C^E                 1
]E1
Y^C^E               (  0.0000,    0.4000)   UPPER
]E3

]E4
]

]E6
]
]

]
]

]
]
Y^C^E               (  0.0000,    0.4000)
ENTAIL    DYC^E                 1
]E1
DYC^E               (  0.0000,    0.3527)   UPPER
]E3

]E4
]
]
]
]

]E6
]
]

]         ENTAIL    DY^C^E                ALREADY DONE
]

]
]

]
]
DYC^E               (  0.0000,    0.3527)
ENTAIL    DC^E                  1
]E1
DC^E                (  0.0000,    0.3727)   UPPER
]E3

]E4
]
```

```
]          ]E6
]          ]
]          ]
]
]          ]
]          ]
]
]          ]
]          ]          ENTAIL    DC^                  ALREADY DONE
]
]          DC^E                 (  0.0000,    0.3727)
]          ENTAIL    DY^CE                   1
]          ]E1
]          DY^CE                (  0.0000,    0.3727)    UPPER
]          ]E3
]
]          ]E4
]          ]
]          ]
]          ]
]          ]
]
]          ]E6
]          ]
]          ]
]
]          ]
]          ]
]
]          ]          ENTAIL    DY^C^E               ALREADY DONE
]          ]
]
]          ]
]          ]
]
]          DY^CE                (  0.0000,    0.3727)
]          ENTAIL    DY^E                    1
]          ]E1
]          DY^E                 (  0.0000,    0.3727)    UPPER
]          ]E3
]
]          ]E4
]          ]
]          ]
]
]          ]E6
]          ]
]          ]
]
]          ]
]          ]
]
]          ]
]          ]          ENTAIL    Y^D                  ALREADY DONE
1]
]          DY^E                 (  0.0000,    0.3727)
]          ENTAIL    DY^C^E^                 1
]          ]E1
]          DY^C^E^              (  0.0000,    0.1000)    UPPER
]          ]E3
```

```
]
]           ]E4
]           ]
]           ]
]           ]
]
]           ]E6
]           ]
]           ]
]
]           ]
]           ]
]
]           ]
]           ]
]
]           ]           ENTAIL    DY^C^E                ALREADY DONE
]           ]           ENTAIL    C^DY^                 ALREADY DONE
]
]           DY^C^E^                ( 0.0000,    0.1000)
]           ENTAIL    C^DY^                  ALREADY DONE
DY^C^E                 ( 0.0000,    0.3727)
```

```
1F^BC                   (  0.0640,    0.0640)
 ADZ                    (  0.1620,    0.1620)
 BC^Z^A                 (  0.0260,    0.0260)
 BD                     (  0.2770,    0.2770)
 Z                      (  0.5000,    0.5000)
 ZCD                    (  0.2520,    0.2520)
 ACD^                   (  0.0470,    0.0470)
 FZG^                   (  0.2970,    0.2970)
 GRB                    (  0.0480,    0.0480)

 NUMBER OF LEVELS =   2
 NUMBER OF SENTENCES =   9
 NUMBER OF SENTENCES ADDED =   6

 ZRG                    (  0.0000,    0.5000)

 ZRG                    (  0.0000,    0.5000)     2
 ZG                     (  0.0000,    0.5000)     1
 GR                     (  0.0480,    0.7030)     1
 Z^RG                   (  0.0000,    0.5000)     1
 ZR^G                   (  0.0000,    0.5000)     1
 ZRG^                   (  0.0000,    0.5000)     1
 ZR                     (  0.0000,    0.5000)     1
1
1ENTAIL   ZRG                     2
 ]E1
 ZRG                    (  0.0000,    0.6770)   UPPER
 ]E3
 ]E4
 ]          ENTAIL   ZG                     1
 ]          ]E1
 ]          ZG                     (  0.0000,    0.6770)   UPPER
 ]          ]E3
 ]
 ]          ]E4
 ]          ]
 ]          ZG                     (  0.0000,    0.5000)   UPPER
 ]          ]E6
 ]          ]
 ]          ]
 ]
 ]          ]
 ]          ]          ENTAIL   Z                          ALREADY DONE
 ]
 ]          ZG                     (  0.0000,    0.5000)
 ]          ENTAIL   GR                     1
 ]          ]E1
 ]          GR                     (  0.0000,    0.7030)   UPPER
 ]          ]E3
 ]          ]
 ]          GR                     (  0.0480,    1.0480)   LOWER
 ]
 ]          ]E4
 ]          ]
 ]
 ]          ]E6
 ]          ]
 ]          ]
 ]
 ]          ]
```

```
]          ]
]
]          GR                    (  0.0480,    0.7030)
ZRG                   (  0.0000,    0.5000)    UPPER
]E6
]          ENTAIL   Z^RG                        1
]          ]E1
]          Z^RG                  (  0.0000,    0.5000)    UPPER
]          ]E3
]
]          ]E4
]          ]
]          ]
]          ]
]          ]
]          ]
]
]          ]E6
]          ]         ENTAIL   ZRG                     ALREADY DONE
]          ]         ENTAIL   GR                      ALREADY DONE
]
]          ]
]          ]
]
1]         ]
]          ]
]
]          Z^RG                  (  0.0000,    0.5000)
]          ENTAIL   GR                      ALREADY DONE
]          ENTAIL   ZR^G                        1
]          ]E1
]          ZR^G                  (  0.0000,    0.6550)    UPPER
]          ]E3
]
]          ]E4
]          ]
]          ZR^G                  (  0.0000,    0.5000)    UPPER
]          ]E6
]          ]
]          ]
]
]          ]         ENTAIL   ZRG                     ALREADY DONE
]          ]         ENTAIL   ZG                      ALREADY DONE
]
]          ]
]          ]
]
]          ZR^G                  (  0.0000,    0.5000)
]          ENTAIL   ZG                      ALREADY DONE
]          ENTAIL   ZRG^                        1
]          ]E1
]          ZRG^                  (  0.0000,    0.9520)    UPPER
]          ]E3
]
]          ]E4
]          ]
]          ]
]          ZRG^                  (  0.0000,    0.5000)    UPPER
]          ]E6
]          ]
```

```
  ]         ]
  ]
  ]         ]
  ]         ]
  ]
  ]         ]         ENTAIL   ZRG                    ALREADY DONE
  ]         ]
  ]
  ]         ZRG^                ( 0.0000,   0.5000)
  ]         ENTAIL   ZR                   1
  ]         ]E1
  ]         ZR                  ( 0.0000,   0.9740)   UPPER
  ]         ]E3
  ]
  ]         ]E4
  ]         ]
  ]         ZR                  ( 0.0000,   0.5000)   UPPER
  ]         ]E6
  ]         ]
  ]         ]
  ]
  ]         ]
  ]         ]         ENTAIL   Z                      ALREADY DONE
  ]
 1]         ZR                  ( 0.0000,   0.5000)
  ZRG                 ( 0.0000,   0.5000)
```

```
1F^BC                   (  0.0640,    0.0640)
 ADZ                    (  0.1620,    0.1620)
 BC^Z^A                 (  0.0260,    0.0260)
 BD                     (  0.2770,    0.2770)
 Z                      (  0.5000,    0.5000)
 ZCD                    (  0.2520,    0.2520)
 ACD^                   (  0.0470,    0.0470)
 FZG^                   (  0.2970,    0.2970)
 GRB                    (  0.0480,    0.0480)

 NUMBER OF LEVELS =   3
 NUMBER OF SENTENCES =   9
 NUMBER OF SENTENCES ADDED =  19

 ZRG                    (  0.0000,    0.2030)

 ZRG                    (  0.0000,    0.2030)     3
 ZG                     (  0.0000,    0.2030)     2
 G                      (  0.0480,    0.7030)     1
 Z^G                    (  0.0000,    0.5000)     1
 ZG^                    (  0.2970,    0.5000)     1
 GR                     (  0.0480,    0.7030)     2
 B                      (  0.2770,    1.0000)     1
 G^R                    (  0.0000,    0.9520)     1
 R                      (  0.0480,    1.0000)     1
 GR^                    (  0.0000,    0.6550)     1
 Z^RG                   (  0.0000,    0.5000)     2
 Z^                     (  0.5000,    0.5000)     1
 Z^R^G                  (  0.0000,    0.5000)     1
 Z^RG^                  (  0.0000,    0.5000)     1
 Z^R                    (  0.0000,    0.5000)     1
 ZR^G                   (  0.0000,    0.2030)     2
 ZR^G^                  (  0.0000,    0.5000)     1
 ZR^                    (  0.0000,    0.5000)     1
 ZRG^                   (  0.0000,    0.5000)     2
 ZR                     (  0.0000,    0.5000)     2
1
1ENTAIL   ZRG                     3
 ]E1
 ZRG                    (  0.0000,    0.6770)   UPPER
 ]E3
 ]E4
 ]         ENTAIL   ZG                     2
 ]         ]E1
 ]         ZG                     (  0.0000,    0.6770)   UPPER
 ]         ]E3
 ]
 ]         ]E4
 ]         ]         ENTAIL   G                     1
 ]         ]         ]E1
 ]         ]         G                     (  0.0000,    0.7030)   UPPER
 ]         ]         ]E3
 ]         ]         ]
 ]         ]         G                     (  0.0480,    1.0480)   LOWER
 ]         ]
 ]         ]         ]E4
 ]         ]
 ]         ]         ]E6
 ]         ]         ]
 ]         ]
```

```
]          ]          G                     (  0.0480,    0.7030)
]          ZG                    (  0.0000,    0.5000)   UPPER
]          ]E6
]          ]          ENTAIL   Z^G                      1
]          ]          ]E1
]          ]          Z^G                   (  0.0000,    0.5000)    UPPER
]          ]          ]E3
]          ]
]          ]          ]E4
]          ]          ]
]          ]          ]
]          ]          ]
]          ]          ]
]          ]
]          ]          ]E6
]          ]          ]          ENTAIL   ZG                         ALREADY DONE
]          ]          ]          ENTAIL   G                          ALREADY DONE
]          ]
]          ]          ]
]          ]          ]
]          ]
]          ]          Z^G                   (  0.0000,    0.5000)
]          ]          ENTAIL   G                   ALREADY DONE
]
]          ]          ENTAIL   ZG^                      1
]          ]          ]E1
]          ]          ZG^                   (  0.0000,    0.9520)    UPPER
]          ]          ]E3
]          ]          ]
]          ]          ZG^                   (  0.2970,    1.2970)    LOWER
]          ]
]          ]          ]E4
]          ]          ]
]          ]          ZG^                   (  0.0000,    0.5000)    UPPER
]          ]          ]E6
]          ]          ]
]          ]          ]
1]         ]
]          ]          ]          ENTAIL   ZG                         ALREADY DONE
]          ]          ]          ENTAIL   Z                          ALREADY DONE
]          ]
]          ]          ZG^                   (  0.2970,    0.5000)
]          ]          ENTAIL   Z                   ALREADY DONE
]          ZG                    (  0.0000,    0.2030)   UPPER
]          ZG                    (  0.0000,    0.2030)
]          ENTAIL    GR                    2
]          ]E1
]          GR                    (  0.0000,    0.7030)   UPPER
]          ]E3
]          ]          ENTAIL   B                        1
]          ]          ]E1
]          ]
]          ]          ]E3
]          ]          ]
]          ]          B                     (  0.0640,    1.0640)    LOWER
]          ]          ]
]          ]
]          ]          ]
]          ]          B                     (  0.2770,    1.2770)    LOWER
]          ]          ]          ENTAIL   GR                         ALREADY DONE
```

```
]         ]
]         ]
]         ]         ]E4
]         ]
]         ]         ]E6
]         ]         ]
]         ]
]         ]         B                   (  0.2770,    1.0000)
]         GR                  (  0.0480,    0.7710)   LOWER
]
]         ]E4
]
]         ]E6
]         ]         ENTAIL   G^R                     1
]         ]         ]E1
]         ]         G^R                 (  0.0000,    0.9520)   UPPER
]         ]         ]E3
]         ]
]         ]         ]E4
]         ]         ]
]         ]
]         ]         ]E6
]         ]         ]         ENTAIL   GR                    ALREADY DONE
]         ]         ]
]         ]
]         ]         ]
]         ]         ]
]         ]
]         ]         G^R                 (  0.0000,    0.9520)
]         ]         ENTAIL   R                       1
]         ]         ]E1
]         ]
]         ]         ]E3
]         ]         ]
]         ]         R                   (  0.0480,    1.0480)   LOWER
]         ]
]         ]         ]E4
1]        ]
]         ]         ]E6
]         ]         ]
]         ]
]         ]         R                   (  0.0480,    1.0000)
]
]         ]         ENTAIL   GR^                     1
]         ]         ]E1
]         ]         GR^                 (  0.0000,    0.6550)   UPPER
]         ]         ]E3
]         ]
]         ]         ]E4
]         ]
]         ]         ]E6
]         ]         ]
]         ]         ]
]         ]
]         ]         ]         ENTAIL   GR                    ALREADY DONE
]         ]         ]         ENTAIL   G                     ALREADY DONE
]         ]         GR^                 ( -0.6550,    0.6550)   UPPER
]         ]         GR^                 (  0.0000,    0.6550)
]         ]         ENTAIL   G                ALREADY DONE
]
```

```
]        GR                ( 0.0480,   0.7030)
ZRG               ( 0.0000,  0.2030)   UPPER
]E6
]        ENTAIL   Z^RG              2
]        ]E1
]        Z^RG              ( 0.0000,   0.5000)   UPPER
]        ]E3
]
]        ]E4
]        ]         ENTAIL   Z^                    1
]        ]         ]E1
]        ]         Z^                ( 0.0000,   0.5000)   UPPER
]        ]         ]E3
]        ]         ]
]        ]         Z^                ( 0.0260,   1.0260)   LOWER
]        ]
]        ]         ]E4
]        ]
]        ]         ]E6
]        ]         ]        ENTAIL   Z                   ALREADY DONE
]        ]         Z^                ( 0.5000,   0.5000)   LOWER
]        ]         Z^                ( 0.5000,   0.5000)
]        ]         ENTAIL   B              ALREADY DONE
]
]        ]E6
]        ]         ENTAIL   ZRG            ALREADY DONE
]        ]         ENTAIL   GR             ALREADY DONE
]
]        ]         ENTAIL   Z^R^G                 1
]        ]         ]E1
]        ]         Z^R^G             ( 0.0000,   0.5000)   UPPER
]        ]         ]E3
]        ]
]        ]         ]E4
]        ]
]        ]         ]E6
]        ]         ]
1]       ]         ]        ENTAIL   GR^                 ALREADY DONE
]        ]
]        ]         ]        ENTAIL   Z^RG                ALREADY DONE
]        ]         ]        ENTAIL   Z^G                 ALREADY DONE
]        ]
]        ]         ]
]        ]         ]
]        ]
]        ]         Z^R^G             ( 0.0000,   0.5000)
]        ]         ENTAIL   Z^G            ALREADY DONE
]
]        ]         ENTAIL   Z^RG^                 1
]        ]         ]E1
]        ]         Z^RG^             ( 0.0000,   0.5000)   UPPER
]        ]         ]E3
]        ]
]        ]         ]E4
]        ]         ]
]        ]
]        ]         ]E6
]        ]         ]
]        ]         ]        ENTAIL   G^R                 ALREADY DONE
]        ]
```

```
]          ]          ]
]          ]          ]
]          ]
]          ]          ]          ENTAIL   Z^RG                ALREADY DONE
]          ]          ]
]          ]
]          ]          Z^RG^                (  0.0000,   0.5000)
]          ]          ENTAIL    Z^R                     1
]          ]          ]E1
]          ]          Z^R                  (  0.0000,   0.5000)   UPPER
]          ]          ]E3
]          ]
]          ]          ]E4
]          ]          ]          ENTAIL   B                   ALREADY DONE
]          ]
]          ]          ]E6
]          ]          ]
]          ]          ]          ENTAIL   R                   ALREADY DONE
]          ]
]          ]          ]
]          ]          ]          ENTAIL   Z^                  ALREADY DONE
]          ]
]          ]          Z^R                  (  0.0000,   0.5000)
]
]          Z^RG                 (  0.0000,   0.5000)
]          ENTAIL    GR                   ALREADY DONE
]          ENTAIL    ZR^G                   2
]          ]E1
]          ZR^G                 (  0.0000,   0.6550)   UPPER
]          ]E3
]
]          ]E4
]          ZR^G                 (  0.0000, . 0.2030)   UPPER
]          ]E6
]          ]          ENTAIL    Z^R^G             ALREADY DONE
]          ]          ENTAIL    GR^               ALREADY DONE
]
1]         ]          ENTAIL    ZRG               ALREADY DONE
]          ]          ENTAIL    ZG                ALREADY DONE
]
]          ]          ENTAIL    ZR^G^                   1
]          ]          ]E1
]          ]          ZR^G^                (  0.0000,   0.9520)   UPPER
]          ]          ]E3
]          ]
]          ]          ]E4
]          ]          ]
]          ]          ZR^G^                (  0.0000,   0.5000)   UPPER
]          ]          ]E6
]          ]          ]
]          ]          ]
]          ]
]          ]          ]
]          ]          ]          ENTAIL   ZG^                 ALREADY DONE
]          ]
]          ]          ]          ENTAIL   ZR^G                ALREADY DONE
]          ]          ]
]          ]
]          ]          ZR^G^                (  0.0000,   0.5000)
]          ]          ENTAIL    ZR^                     1
```

```
]              ]              ]E1
]              ]              ZR^                        (  0.0000,    0.9520)   UPPER
]              ]              ]E3
]              ]
]              ]              ]E4
]              ]              ]
]              ]              ZR^                        (  0.0000,    0.5000)   UPPER
]              ]              ]E6
]              ]              ]
]              ]              ]
]              ]
]              ]              ]
]              ]              ]         ENTAIL   Z                        ALREADY DONE
]              ]
]              ]              ZR^                        (  0.0000,    0.5000)
]
]              ZR^G                      (  0.0000,    0.2030)
]              ENTAIL    ZG                      ALREADY DONE
]              ENTAIL    ZRG^                      2
]              ]E1
]              ZRG^                      (  0.0000,    0.9520)   UPPER
]              ]E3
]
]              ]E4
]              ZRG^                      (  0.0000,    0.5000)   UPPER
]              ]E6
]              ]            ENTAIL   Z^RG^              ALREADY DONE
]              ]            ENTAIL   G^R                ALREADY DONE
]
]              ]            ENTAIL   ZR^G^              ALREADY DONE
]              ]            ENTAIL   ZG^                ALREADY DONE
]
]              ]            ENTAIL   ZRG                ALREADY DONE
]              ]            ENTAIL   ZR                   1
]              ]            ]E1
]              ]            ZR                        (  0.0000,    0.9740)   UPPER
]              ]            ]E3
1]             ]
]              ]            ]E4
]              ]            ZR                        (  0.0000,    0.5000)   UPPER
]              ]            ]E6
]              ]            ]          ENTAIL   Z^R                       ALREADY DONE
]              ]            ]          ENTAIL   R                         ALREADY DONE
]              ]
]              ]            ]          ENTAIL   ZR^                       ALREADY DONE
]              ]            ]          ENTAIL   Z                         ALREADY DONE
]              ]
]              ]            ZR                        (  0.0000,    0.5000)
]
]              ZRG^                      (  0.0000,    0.5000)
]              ENTAIL    ZR                        2   DO MORE
]              ]E1
]
]              ]E3
]
]              ]E4
]
]              ]E6
]              ]            ENTAIL   Z^R                ALREADY DONE
]              ]            ENTAIL   R                  ALREADY DONE
```

```
]
]          ]          ENTAIL   ZR^                    ALREADY DONE
]          ]          ENTAIL   Z                      ALREADY DONE
]
]          ZR                   (  0.0000,    0.5000)
ZRG                   (  0.0000,    0.2030)
```

```
1F^BC                  (  0.0640,    0.0640)
 ADZ                   (  0.1616,    0.1616)
 DZ^                   (  0.4050,    0.4050)
 BC^ZA                 (  0.0258,    0.0258)
 B^D                   (  0.5330,    0.5330)
 S^DB                  (  0.1942,    0.1942)
 ZCD                   (  0.2519,    0.2519)
 ACD^                  (  0.0472,    0.0472)
 HJR^                  (  0.0116,    0.0116)
 A^CDZ^                (  0.1514,    0.1514)
 A^SJG^                (  0.0534,    0.0534)
 FZH                   (  0.2268,    0.2268)
 GR^B^                 (  0.0050,    0.0050)

 NUMBER OF LEVELS =   2
 NUMBER OF SENTENCES =  13
 NUMBER OF SENTENCES ADDED =   9

 ZRG                   (  0.0000,    0.5950)

 ZRG                   (  0.0000,    0.5950)     2
 Z                     (  0.2777,    0.5950)     1
 R                     (  0.0000,    0.9884)     1
 GR                    (  0.0000,    0.9416)     1
 G                     (  0.0050,    0.9466)     1
 Z^RG                  (  0.0000,    0.7223)     1
 ZR^G                  (  0.0000,    0.5950)     1
 ZG                    (  0.0000,    0.5950)     1
 ZRG^                  (  0.0000,    0.5950)     1
 ZR                    (  0.0000,    0.5950)     1
1
1ENTAIL   ZRG                    2
 ]E1
 ZRG                   (  0.0000,    0.5950)   UPPER
 ]E3
 ]E4
 ]         ENTAIL   Z                    1
 ]         ]E1
 ]         Z                   (  0.0000,    0.5950)   UPPER
 ]         ]E3
 ]         ]
 ]         Z                   (  0.1616,    1.1616)   LOWER
 ]         ]
 ]
 ]         ]
 ]         Z                   (  0.2519,    1.2519)   LOWER
 ]         ]
 ]
 ]         Z                   (  0.2777,    1.0000)   LOWER
 ]         ]E4
 ]
 ]         ]E6
 ]         ]
 ]
 ]         Z                   (  0.2777,    0.5950)
 ]         ENTAIL   R                    1
 ]         ]E1
 ]         R                   (  0.0000,    0.9884)   UPPER
 ]         ]E3
 ]
```

```
]          ]E4
]
]          ]E6
]          ]
]
]          R                    (  0.0000,   0.9884)
]          ENTAIL   GR                    1
]          ]E1
]          GR                   (  0.0000,   0.9416)   UPPER
]          ]E3
]
]          ]E4
]          ]
]
]          ]E6
]          ]
]          ]         ENTAIL   R                    ALREADY DONE
]
]          ]
]          ]
]
]          GR                   (  0.0000,   0.9416)
]          ENTAIL   G                     1
]          ]E1
]          G                    (  0.0000,   0.9466)   UPPER
]          ]E3
]          ]
]          G                    (  0.0050,   1.0050)   LOWER
]
]          ]E4
]
1]          ]E6
]          ]
]
]          G                    (  0.0050,   0.9466)
]E6
]          ENTAIL   Z^RG                  1
]          ]E1
]          Z^RG                 (  0.0000,   0.7223)   UPPER
]          ]E3
]
]          ]E4
]          ]
]          ]
]          ]
]          ]
]          ]
]          ]
]
]          ]E6
]          ]         ENTAIL   ZRG                  ALREADY DONE
]          ]         ENTAIL   GR                   ALREADY DONE
]
]          ]
]          ]
]
]          ]
]          ]
]
]          Z^RG                 (  0.0000,   0.7223)
```

```
 ]          ENTAIL   GR                        ALREADY DONE
 ]          ENTAIL   ZR^G                        1
 ]          ]E1
 ]          ZR^G                 (   0.0000,     0.5950)    UPPER
 ]          ]E3
 ]
 ]          ]E4
 ]          ]
 ]          ]
 ]
 ]          ]E6
 ]          ]
 ]          ]
 ]
 ]          ]          ENTAIL   ZRG                     ALREADY DONE
 ]          ]
 ]
 ]          ]
 ]          ]
 ]
 ]          ZR^G                 (   0.0000,     0.5950)
 ]          ENTAIL   ZG                          1
 ]          ]E1
 ]          ZG                   (   0.0000,     0.5950)    UPPER
 ]          ]E3
 ]
 ]          ]E4
 ]
 ]          ]E6
 ]          ]
 ]          ]          ENTAIL   G                       ALREADY DONE
1]
 ]          ]
 ]          ]          ENTAIL   Z                       ALREADY DONE
 ]
 ]          ZG                   (   0.0000,     0.5950)
 ]          ENTAIL   ZRG^                        1
 ]          ]E1
 ]          ZRG^                 (   0.0000,     0.5950)    UPPER
 ]          ]E3
 ]
 ]          ]E4
 ]          ]
 ]          ]
 ]
 ]          ]E6
 ]          ]
 ]          ]
 ]
 ]          ]
 ]          ]
 ]
 ]          ]          ENTAIL   ZRG                     ALREADY DONE
 ]          ]
 ]
 ]          ZRG^                 (   0.0000,     0.5950)
 ]          ENTAIL   ZR                          1
 ]          ]E1
 ]          ZR                   (   0.0000,     0.5950)    UPPER
 ]          ]E3
```

```
]
]         ]E4
]
]         ]E6
]         ]
]         ]          ENTAIL   R                     ALREADY DONE
]
]         ]
]         ]          ENTAIL   Z                     ALREADY DONE
]
]         ZR                   (   0.0000,    0.5950)
ZRG                  (   0.0000,    0.5950)
```

```
1E^BC                   (  0.1170,    0.1170)
 AEZJ^                  (  0.0580,    0.0580)
 DZ^                    (  0.4050,    0.4050)
 BC^ZA                  (  0.0250,    0.0250)
 E^D                    (  0.4460,    0.4460)
 S^DB                   (  0.1940,    0.1940)
 ZCJ                    (  0.1080,    0.1080)
 ACS^                   (  0.1740,    0.1740)
 EJ^                    (  0.2040,    0.2040)
 HJR^                   (  0.0110,    0.0110)
 A^CDZ^                 (  0.1510,    0.1510)
 A^SJG^                 (  0.0530,    0.0530)
 SJ                     (  0.1040,    0.1040)
 ZH                     (  0.2260,    0.2260)
 GR^B^                  (  0.0050,    0.0050)

 NUMBER OF LEVELS =    2
 NUMBER OF SENTENCES =  15
 NUMBER OF SENTENCES ADDED =  14

 ZRGJ                   (  0.0000,    0.5370)

 ZRGJ                   (  0.0000,    0.5370)     2
 ZJ                     (  0.1080,    0.5370)     1
 Z                      (  0.2260,    0.5950)     1
 JR                     (  0.0000,    0.7850)     1
 GR                     (  0.0000,    0.9420)     1
 JG                     (  0.0000,    0.7430)     1
 J                      (  0.1080,    0.7960)     1
 Z^RGJ                  (  0.0000,    0.6880)     1
 RGJ                    (  0.0000,    0.7430)     1
 ZR^GJ                  (  0.0000,    0.5370)     1
 ZGJ                    (  0.0000,    0.5370)     1
 ZRG^J                  (  0.0000,    0.5370)     1
 ZRJ                    (  0.0000,    0.5370)     1
 ZRGJ^                  (  0.0000,    0.4870)     1
 ZRG                    (  0.0000,    0.5950)     1
1
1ENTAIL   ZRGJ                   2
 ]E1
 ZRGJ                   (  0.0000,    0.5370)   UPPER
 ]E3
 ]E4
 ]          ENTAIL   ZJ                   1
 ]          ]E1
 ]          ZJ                  (  0.0000,    0.5370)   UPPER
 ]          ]E3
 ]          ]
 ]          ZJ                  (  0.1080,    1.1080)   LOWER
 ]
 ]          ]E4
 ]          ]
 ]          ]
 ]          ]
 ]          ]
 ]          ]
 ]          ]
 ]          ]
 ]          ]
 ]
```

```
 ]          ]E6
 ]          ]
 ]          ]
 ]
 ]          ]
 ]          ]
 ]
 ]          ZJ                    (  0.1080,    0.5370)
 ]          ENTAIL   Z                       1
 ]          ]E1
 ]          Z                     (  0.0000,    0.5950)   UPPER
 ]          ]E3
 ]          ]
 ]          Z                     (  0.0580,    1.0580)   LOWER
 ]          ]
 ]
 ]          ]
 ]          Z                     (  0.1080,    1.1080)   LOWER
 ]          ]
 ]          Z                     (  0.2260,    1.2260)   LOWER
 ]
 ]          ]E4
 ]
 ]          ]E6
 ]          ]
 ]
 ]          Z                     (  0.2260,    0.5950)
 ]          ENTAIL   JR                      1
 ]          ]E1
 ]          JR                    (  0.0000,    0.7850)   UPPER
 ]          ]E3
 ]
 ]          ]E4
 ]          ]
 ]          ]
 ]          ]
 ]          ]
 ]          ]
1]          ]
 ]
 ]          ]E6
 ]          ]
 ]          ]
 ]
 ]          ]
 ]          ]
 ]
 ]          JR                    (  0.0000,    0.7850)
 ]          ENTAIL   GR                      1
 ]          ]E1
 ]          GR                    (  0.0000,    0.9420)   UPPER
 ]          ]E3
 ]
 ]          ]E4
 ]          ]
 ]          ]
 ]
 ]          ]E6
 ]          ]
 ]          ]
```

```
]
]          ]
]          ]
]
]          GR                     (  0.0000,    0.9420)
]          ENTAIL    JG                     1
]          ]E1
]          JG                     (  0.0000,    0.7430)   UPPER
]          ]E3
]
]          ]E4
]          ]
]          ]
]          ]
]          ]
]          ]
]          ]
]
]          ]E6
]          ]
]          ]
]
]          ]
]          ]
]
]          JG                     (  0.0000,    0.7430)
]          ENTAIL    J                      1
]          ]E1
]          J                      (  0.0000,    0.7960)   UPPER
]          ]E3
]          ]
]          J                      (  0.1080,    1.1080)   LOWER
]          ]
]
]          ]
]
]          ]
]
1]
]          ]E4
]
]          ]E6
]          ]
]
]          J                      (  0.1080,    0.7960)
]E6
]          ENTAIL    Z^RGJ                  1
]          ]E1
]          Z^RGJ                  (  0.0000,    0.6880)   UPPER
]          ]E3
]
]          ]E4
]          ]
]          ]
]          ]
]          ]
]          ]
]          ]
]
]          ]E6
```

```
 ]          ]          ENTAIL   ZRGJ                    ALREADY DONE
 ]          ]
 ]
 ]          ]
 ]          ]
 ]
 ]          ]
 ]          ]
 ]
 ]          ]
 ]          ]
 ]
 ]          Z^RGJ                ( 0.0000,   0.6880)
 ]          ENTAIL   RGJ                    1
 ]          ]E1
 ]          RGJ                  ( 0.0000,   0.7430)   UPPER
 ]          ]E3
 ]
 ]          ]E4
 ]
 ]          ]E6
 ]          ]
 ]          ]          ENTAIL   JG                      ALREADY DONE
 ]
 ]          ]
 ]          ]          ENTAIL   JR                      ALREADY DONE
 ]
 ]          ]
 ]          ]          ENTAIL   GR                      ALREADY DONE
 ]
 ]          RGJ                  ( 0.0000,   0.7430)
 ]          ENTAIL   ZR^GJ                  1
 ]          ]E1
 ]          ZR^GJ                ( 0.0000,   0.5370)   UPPER
 ]          ]E3
 ]
 ]          ]E4
 ]          ]
1]          ]
 ]
 ]          ]E6
 ]          ]
 ]          ]
 ]
 ]          ]          ENTAIL   ZRGJ                    ALREADY DONE
 ]          ]
 ]
 ]          ]
 ]          ]
 ]
 ]          ]
 ]          ]
 ]
 ]          ZR^GJ                ( 0.0000,   0.5370)
 ]          ENTAIL   ZGJ                    1
 ]          ]E1
 ]          ZGJ                  ( 0.0000,   0.5370)   UPPER
 ]          ]E3
 ]
 ]          ]E4
```

```
]

]E6
]
]          ENTAIL   JG                    ALREADY DONE

]
]          ENTAIL   ZJ                    ALREADY DONE

]
]

ZGJ                  (  0.0000,    0.5370)
ENTAIL   ZRG^J                    1
]E1
ZRG^J                (  0.0000,    0.5370)   UPPER
]E3

]E4
]
]

]E6
]
]

]
]

]          ENTAIL   ZRGJ                  ALREADY DONE
]

]
]

ZRG^J                (  0.0000,    0.5370)
ENTAIL   ZRJ                      1
]E1
ZRJ                  (  0.0000,    0.5370)   UPPER
]E3

]E4
]

]E6
]
]          ENTAIL   JR                    ALREADY DONE

]
]          ENTAIL   ZJ                    ALREADY DONE

]
]

ZRJ                  (  0.0000,    0.5370)
ENTAIL   ZRGJ^                    1
]E1
ZRGJ^                (  0.0000,    0.4870)   UPPER
]E3
```

```
 ]         ]E4
 ]         ]
 ]         ]
 ]         ]
 ]         ]
 ]         ]
 ]         ]
 ]
 ]         ]E6
 ]         ]
 ]         ]
 ]
 ]         ]
 ]         ]
 ]
 ]         ]
 ]         ]
 ]
 ]         ]         ENTAIL   ZRGJ                 ALREADY DONE
 ]         ]
 ]
 ]         ZRGJ^              (  0.0000,    0.4870)
 ]         ENTAIL   ZRG                   1
 ]         ]E1
 ]         ZRG                (  0.0000,    0.5950)   UPPER
 ]         ]E3
 ]
 ]         ]E4
 ]         ]
 ]         ]
 ]
 ]         ]E6
 ]         ]
 ]         ]         ENTAIL   GR                   ALREADY DONE
 ]
 ]         ]
 ]         ]
 ]
1]         ]
 ]         ]
 ]
 ]         ZRG                (  0.0000,    0.5950)
 ZRGJ               (  0.0000,    0.5370)
```

# APPENDIX B

## Source Code for ENTAIL

```
ESET: PROC OPTIONS(MAIN);
 DCL 1 GB EXT,
       2 KNB(500,26),
         3 VALUE FIXED BIN(31),
         3 NEXTS FIXED BIN(31),
         3 NEXTA FIXED BIN(31),
       2 LEVEL(500) FIXED BIN(31),
       2 SENT(500) CHAR(20),
       2 STARTA(26) FIXED BIN(31),
       2 ENDA(26) FIXED BIN(31),
       2 STARTS(500) FIXED BIN(31),
       2 LASTSENT    FIXED BIN(31),
       2 BOUNDS(500),
         3 UPPER FLOAT(6),
         3 LOWER FLOAT(6),
       2 ORIGSENT FIXED BIN(31),
       2 NUMLEVEL FIXED BIN(31);
 DCL 1 S(10),
      2 ATOM FIXED BIN(31),
      2 VALUE FIXED BIN(31);
 DCL (UB,LB) FLOAT(6),
     INP CHAR(20),
     SNUM FIXED BIN(31);
 DCL  INPUT ENTRY EXT RETURNS(FIXED BIN(31)),
      ENTAIL ENTRY EXT;

 CALL INIT;
 ORIGSENT=INPUT(S);
LP: DO WHILE('1'B);
 PUT LIST('NUMBER OF LEVELS?');
 GET LIST(NUMLEVEL);
 CALL ENTAIL(S,NUMLEVEL,LB,UB,SNUM);
 CALL PRINTS;
 PUT EDIT('LB = ',LB,'UB = ',UB)(A,F(8,4),A,F(8,4));
 PUT LIST('ENTER NEW SENTENCE TO BE ENTAILED');
 GET EDIT(INP)(A(20));
 IF SUBSTR(INP,1,1)=' ' THEN LEAVE LP;
   PUT LIST(INP);
 CALL GETS(INP);
END;
 RETURN;

 INIT: PROC;
  STARTA=0;
  ENDA=0;
  STARTS=0;
  KNB.VALUE=0;
  LASTSENT=0;
  RETURN;
 END INIT;

 PRINTS: PROC;
  DCL OUT1 FILE STREAM OUTPUT PRINT,
      I FIXED BIN(31);
  DO I=1 TO ORIGSENT;
   PUT FILE(OUT1) SKIP EDIT(SENT(I),'(',BOUNDS(I).LOWER,', ',
      BOUNDS(I).UPPER,')')(A(20),A,F(8,4),A,F(8,4),A);
  END;
  PUT SKIP FILE(OUT1);
  PUT SKIP FILE(OUT1) EDIT('NUMBER OF LEVELS = ',NUMLEVEL)(A,F(3));
```

```
  PUT SKIP FILE(OUT1) EDIT('NUMBER OF SENTENCES = ',ORIGSENT)(A,F(3));
  PUT SKIP FILE(OUT1) EDIT('NUMBER OF SENTENCES ADDED = ',
                LASTSENT-SNUM)(A,F(3));
  PUT SKIP FILE(OUT1);
  PUT SKIP FILE(OUT1) EDIT(SENT(SNUM),'(',LB,', ',UB,')')
                    (A(20),A,F(8,4),A,F(8,4),A);
  PUT SKIP FILE(OUT1);
  DO I=SNUM TO LASTSENT;
   PUT FILE(OUT1) SKIP EDIT(SENT(I),'(',BOUNDS(I).LOWER,', ',
      BOUNDS(I).UPPER,')',LEVEL(I))(A(20),A,F(8,4),A,F(8,4),A,F(6));
  END;
  PUT FILE(OUT1) PAGE;
  RETURN;
 END PRINTS;
 GETS:PROC(INP);
  DCL INP CHAR(20),
      (I,J,K) FIXED BIN(31),
      HASH CHAR(26) INIT('ABCDEFGHIJKLMNOPQRSTUVWXYZ');
  J=INDEX(INP,' ')-1;
  K=1;
  DO I=1 TO J;
   S(K).ATOM=INDEX(HASH,SUBSTR(INP,I,1));
   IF (SUBSTR(INP,I+1,1)='^') THEN DO;
     S(K).VALUE=-1;
     I=I+1;
    END;
   ELSE  S(K).VALUE=1;
   K=K+1;
  END;
  S(K).ATOM=0;
 RETURN;
END GETS;
END ESET;
```

```
INPUT:PROC(S) RETURNS(FIXED BIN(31));
 DCL LET(29) FIXED BIN(31) INIT((29)0),
     HASH CHAR(29) INIT('ABCDEFGHIJKLMNOPQRSTUVWXYZ+*^'),
     ATOMSET EXT ENTRY,
     NPSENT(100) FIXED BIN(31),
     NSENT(100,20) FIXED BIN(31),
     CSENT(100) CHAR(20),
     NUMSENT FIXED BIN(31)INIT(0),
     EOF BIT(1) INIT('0'B),
     IN FILE INPUT STREAM,
     B(100) FLOAT(6);
  ON ENDFILE(IN) EOF='1'B;
 DCL INP CHAR(20),
     FLAG BIT(1) INIT('0'B),
     (I,J,N,K) FIXED BIN(31);
 DCL 1 S(*),
       2 ATOM  FIXED BIN(31),
       2 VALUE FIXED BIN(31);

 OPEN FILE(IN);
 DO I=1 TO 100;
   GET FILE(IN) EDIT(INP,B(I))(COL(1),A(20),COL(25),F(6,4));
   IF EOF THEN LEAVE;
   CSENT(I)=INP;
   NPSENT(I)=INDEX(CSENT(I),' ')-1;
   DO J=1 TO NPSENT(I);
     NSENT(I,J)=INDEX(HASH,SUBSTR(CSENT(I),J,1));
   END;
 END;
 NUMSENT=I-1;
 N=NUMSENT-1;
 CALL ATOMSET(NUMSENT,NPSENT,NSENT,CSENT,B,S);
RETURN(N);
END INPUT;
```

```
ATOMSET: PROC(NUMSENT,NPSENT,NSENT,CSENT,B,S);
 DCL ALPH(26) FIXED DEC(4,3) INIT((26)0),
     VAL FLOAT(6),
    B(*) FLOAT(6) CONNECTED,
    B2(NUMSENT) FLOAT(6),
    NPSENT(*) FIXED BIN(31) CONNECTED,
    CSENT(*) CHAR(20) CONNECTED,
    (TEMP,NUMSENT) FIXED BIN(31),
    EOF1 BIT(1) INIT('0'B),
    NSENT(*,*) FIXED BIN(31) CONNECTED,
    HASH CHAR(29) INIT('ABCDEFGHIJKLMNOPQRSTUVWXYZ+*^'),
    INVAR   FILE INPUT STREAM,
    OUTA    FILE OUTPUT STREAM,
     CH CHAR(1),
     (J,I) FIXED BIN(31);
DCL 1 S(*),
      2 ATOM  FIXED BIN(31),
      2 VALUE FIXED BIN(31);
DCL CREATES ENTRY EXT RETURNS(FIXED BIN(31));
 ON ENDFILE(INVAR) EOF1='1'B;

DCL 1 GB EXT,
      2 KNB(500,26),
        3 VALUE FIXED BIN(31),
        3 NEXTS FIXED BIN(31),
        3 NEXTA FIXED BIN(31),
      2 LEVEL(500) FIXED BIN(31),
      2 SENT(500) CHAR(20),
      2 STARTA(26) FIXED BIN(31),
      2 ENDA(26) FIXED BIN(31),
      2 STARTS(500) FIXED BIN(31),
      2 LASTSENT    FIXED BIN(31),
      2 BOUNDS(500),
        3 UPPER FLOAT(6),
        3 LOWER FLOAT(6),
      2 ORIGSENT FIXED BIN(31),
      2 NUMLEVEL FIXED BIN(31);

 OPEN FILE(INVAR);
 DO WHILE(^EOF1);
  GET FILE(INVAR) EDIT(CH,VAL)(COL(1),A(1),COL(10),F(5,3));
  ALPH(INDEX(HASH,CH))=VAL;
 END;
 CLOSE FILE(INVAR);
 PUT LIST('1 - CALC    2 - READ');
 GET LIST(J);
 DO I=1 TO NUMSENT-1;
  B2(I)=ATOMVAL(I);
  LEVEL(I)=1000;
  IF J=1 THEN B(I)=B2(I);
  TEMP=CREATES(S,1000);
  BOUNDS(TEMP).UPPER=B(I);
  BOUNDS(TEMP).LOWER=B(I);
 END;
 TEMP=ATOMVAL(NUMSENT);    /* PUTS SENTENCE TO BE ENTAILED IN S */
 CALL WRIT;
 RETURN;

WRIT: PROC;
 B(NUMSENT)=0;
```

```
  DO I=1 TO NUMSENT;
   PUT FILE(OUTA) EDIT(CSENT(I),B(I))(COL(1),A(20),COL(25),F(6,3));
  END;
  RETURN;
 END WRIT;

  ATOMVAL:PROC(K) RETURNS(FLOAT(6));
   DCL (L,K,J) FIXED BIN(31),
       DATA(20) FLOAT(6),
       TOP FIXED BIN(31);
   J=0;
   TOP=0;
   DO L=1 TO NPSENT(K);
     IF NSENT(K,L)<=26 THEN DO; TOP=TOP+1;
                                DATA(TOP)=ALPH(NSENT(K,L));
                                J=J+1;
                                S(J).ATOM=NSENT(K,L);
                                S(J).VALUE=1;
                            END;
     ELSE SELECT(NSENT(K,L));
            WHEN(27) DO; TOP=TOP-1;
                         DATA(TOP)=DATA(TOP)+DATA(TOP+1)-(DATA(TOP)*
                                   DATA(TOP+1));
                     END;
            WHEN(28) DO; TOP=TOP-1;
                         DATA(TOP)=DATA(TOP)*DATA(TOP+1);
                     END;
            WHEN(29) DO; DATA(TOP)=1-DATA(TOP);
                         S(J).VALUE=-1;
                     END;
          END;
   END;
  IF ^(TOP=1) THEN DO; PUT LIST('ERROR-ATOM');STOP; END;
  S(J+1).ATOM=0;
  RETURN(DATA(TOP));
 END ATOMVAL;
END ATOMSET;
```

```
ENTAIL: PROC(S,NOL,LB,UB,SNUM) RECURSIVE;
 DCL AVAIL(ORIGSENT) BIT(1),
     (SP,I,NOL,SNUM) FIXED BIN(31),
     (LB,UB) FLOAT(6);
 DCL ENTAIL1 ENTRY EXT,
     ENTAIL3 ENTRY EXT,
     ENTAIL4 ENTRY EXT,
     ENTAIL6 ENTRY EXT,
     CREATES ENTRY EXT RETURNS(FIXED BIN(31)),
     FIND    ENTRY EXT RETURNS(FIXED BIN(31)),
     OUT FILE STREAM OUTPUT PRINT;
 DCL 1 S(*),
       2 ATOM FIXED BIN(31),
       2 VALUE FIXED BIN(31);

 DCL 1 GB EXT,
       2 KNB(500,26),
         3 VALUE FIXED BIN(31),
         3 NEXTS FIXED BIN(31),
         3 NEXTA FIXED BIN(31),
       2 LEVEL(500) FIXED BIN(31),
       2 SENT(500) CHAR(20),
       2 STARTA(26) FIXED BIN(31),
       2 ENDA(26) FIXED BIN(31),
       2 STARTS(500) FIXED BIN(31),
       2 LASTSENT    FIXED BIN(31),
       2 BOUNDS(500),
         3 UPPER FLOAT(6),
         3 LOWER FLOAT(6),
       2 ORIGSENT FIXED BIN(31),
       2 NUMLEVEL FIXED BIN(31);

 IF S(1).ATOM=0 THEN DO;
   LB=1;  UB=1;
   RETURN;
  END;
 SP=10*(NUMLEVEL-NOL)+1;
 DO I=1 TO (SP-1) BY 10;
  PUT FILE(OUT) EDIT(']')(COL(I),A);
 END;
 AVAIL='1'B;
 I=FIND(S);
 IF I>0 THEN DO;
   SNUM=I;
   IF NOL>LEVEL(SNUM) THEN DO;
     PUT FILE(OUT) EDIT('ENTAIL   ',SENT(SNUM),NOL,'DO MORE')
                   (COL(SP),A,A,F(3),X(3),A);
     LEVEL(SNUM)=NOL;
     GOTO REDO;
    END;
   PUT FILE(OUT) EDIT('ENTAIL   ',SENT(SNUM),'ALREADY DONE')
           (COL(SP),A,A,A);
   LB=BOUNDS(SNUM).LOWER;
   UB=BOUNDS(SNUM).UPPER;
   RETURN;
  END;
 IF NOL=0 THEN DO;
   LB=0;   UB=1;
   SNUM=0;
   RETURN;
```

```
  END;
 SNUM=CREATES(S,NOL);
 PUT FILE(OUT) EDIT('ENTAIL    ',SENT(SNUM),NOL)(COL(SP),A,A,F(3));
REDO:
 SP=SP+1;
 DO I=1 TO (SP-1) BY 10;
  PUT FILE(OUT) EDIT(']')(COL(I),A);
 END;
 PUT FILE(OUT) EDIT('E1')(COL(SP),A);
 CALL ENTAIL1(SNUM);
 DO I=1 TO (SP-1) BY 10;
  PUT FILE(OUT) EDIT(']')(COL(I),A);
 END;
 PUT FILE(OUT) EDIT('E3')(COL(SP),A);
 CALL ENTAIL3(SNUM,NOL,AVAIL,S);
 DO I=1 TO (SP-1) BY 10;
  PUT FILE(OUT) EDIT(']')(COL(I),A);
 END;
 PUT FILE(OUT) EDIT('E4')(COL(SP),A);
 CALL ENTAIL4(SNUM,NOL,AVAIL,S);
 DO I=1 TO (SP-1) BY 10;
  PUT FILE(OUT) EDIT(']')(COL(I),A);
 END;
 PUT FILE(OUT) EDIT('E6')(COL(SP),A);
 CALL ENTAIL6(SNUM,NOL,S);
 LB=BOUNDS(SNUM).LOWER;
 UB=BOUNDS(SNUM).UPPER;
 SP=SP-1;
 DO I=1 TO (SP-1) BY 10;
  PUT FILE(OUT) EDIT(']')(COL(I),A);
 END;
 PUT FILE(OUT) EDIT(SENT(SNUM),'(',LB,', ',UB,')')
          (COL(SP),A(20),A,F(8,4),A,F(8,4),A);
 RETURN;
END ENTAIL;
```

```
CREATES: PROC(S,NOL) RETURNS (FIXED BIN(31));
 DCL (I,NOL) FIXED BIN(31),
     CH CHAR(1),
     MT CHAR(20) VAR INIT(''),
     HASH CHAR(26) INIT('ABCDEFGHIJKLMNOPQRSTUVWXYZ');
 DCL 1 S(*),
       2 ATOM FIXED BIN(31),
       2 VALUE FIXED BIN(31);

 DCL 1 GB EXT,
       2 KNB(500,26),
         3 VALUE FIXED BIN(31),
         3 NEXTS FIXED BIN(31),
         3 NEXTA FIXED BIN(31),
       2 LEVEL(500) FIXED BIN(31),
       2 SENT(500) CHAR(20),
       2 STARTA(26) FIXED BIN(31),
       2 ENDA(26) FIXED BIN(31),
       2 STARTS(500) FIXED BIN(31),
       2 LASTSENT    FIXED BIN(31),
       2 BOUNDS(500),
         3 UPPER FLOAT(6),
         3 LOWER FLOAT(6),
       2 ORIGSENT FIXED BIN(31),
       2 NUMLEVEL FIXED BIN(31);


 LASTSENT=LASTSENT+1;
 STARTS(LASTSENT)=S(1).ATOM;
 I=1;
 DO WHILE(S(I).ATOM^=0);
  IF ENDA(S(I).ATOM)=0 THEN DO;
    STARTA(S(I).ATOM)=LASTSENT;
    ENDA(S(I).ATOM)=LASTSENT;
   END;
  ELSE DO;
    KNB(ENDA(S(I).ATOM),S(I).ATOM).NEXTS=LASTSENT;
    ENDA(S(I).ATOM)=LASTSENT;
   END;
  KNB(LASTSENT,S(I).ATOM).NEXTA=S(I+1).ATOM;
  KNB(LASTSENT,S(I).ATOM).VALUE=S(I).VALUE;
  KNB(LASTSENT,S(I).ATOM).NEXTS=0;
  CH=SUBSTR(HASH,S(I).ATOM,1);
  MT=MT]]CH;
  IF S(I).VALUE=-1 THEN MT=MT]]'^';
  I=I+1;
 END;
 LEVEL(LASTSENT)=NOL;
 BOUNDS(LASTSENT).LOWER=0;
 BOUNDS(LASTSENT).UPPER=1;
 SENT(LASTSENT)=MT;
 RETURN(LASTSENT);
END CREATES;
```

```
ENTAIL1: PROC(SNUM);    /* CORRESPONDS TO PROCEDURE ONE */
 DCL (SNUM,K;J,I,LISTPTR,LAST2,LIST(ORIGSENT)) FIXED BIN(31),
     LIST2(ORIGSENT) FIXED BIN(31),
     (ONLIST(ORIGSENT),FLAG) BIT(1),
     UPDATE ENTRY EXT,
     (SUM,HOLD) FLOAT(6);

 DCL 1 GB EXT,
       2 KNB(500,26),
         3 VALUE FIXED BIN(31),
         3 NEXTS FIXED BIN(31),
         3 NEXTA FIXED BIN(31),
       2 LEVEL(500) FIXED BIN(31),
       2 SENT(500) CHAR(20),
       2 STARTA(26) FIXED BIN(31),
       2 ENDA(26) FIXED BIN(31),
       2 STARTS(500) FIXED BIN(31),
       2 LASTSENT    FIXED BIN(31),
       2 BOUNDS(500),
         3 UPPER FLOAT(6),
         3 LOWER FLOAT(6),
       2 ORIGSENT FIXED BIN(31),
       2 NUMLEVEL FIXED BIN(31);


 LISTPTR=0;
 SUM=0;
 ONLIST='0'B;
 J=STARTS(SNUM);
 DO WHILE(J^=0);
  K=STARTA(J);
  DO WHILE((K^=0)&(K<=ORIGSENT));
   IF (KNB(K,J).VALUE^=0)&(KNB(K,J).VALUE^=KNB(SNUM,J).VALUE)
      &(^ONLIST(K))  THEN DO;
       LISTPTR=LISTPTR+1;
       LIST(LISTPTR)=K;
       ONLIST(K)='1'B;
    END;
    K=KNB(K,J).NEXTS;
  END;
  J=KNB(SNUM,J).NEXTA;
 END;
 FLAG='0'B;
 LAST2=0;
 I=1;
 DO WHILE((LAST2^=0)](I<=LISTPTR));
  DO WHILE(I<=LISTPTR);
   IF DIFF(LIST(I)) THEN DO;
     FLAG='1'B;
     LAST2=LAST2+1;
     LIST2(LAST2)=I;
    END;
   I=I+1;
  END;
  HOLD=0;
  IF FLAG THEN DO;
    FLAG='0'B;
    DO J=1 TO LAST2;
     HOLD=HOLD+BOUNDS(LIST(LIST2(J))).LOWER;
    END;
```

```
    END;
   IF HOLD>SUM THEN SUM=HOLD;
   I=LIST2(LAST2)+1;
   LAST2=LAST2-1;
  END;
  SUM=1-SUM;
  HOLD=0;
  CALL UPDATE(SNUM,HOLD,SUM);
  RETURN;

  DIFF:PROC(A) RETURNS(BIT(1));
   DCL (A,B,C,Z) FIXED BIN(31),
        FLAG  BIT(1);
   DO Z=1 TO LAST2;
    B=LIST(LIST2(Z));
    C=STARTS(A);
    FLAG='0'B;
    DO WHILE((C^=0)&(^FLAG));
     IF KNB(A,C).VALUE=((-1)*KNB(B,C).VALUE) THEN FLAG='1'B;
     C=KNB(A,C).NEXTA;
    END;
    IF ^FLAG THEN RETURN('0'B);
   END;
   RETURN('1'B);
  END DIFF;
 END ENTAIL1;
```

```
ENTAIL3: PROC(SNUM,NOL,AVAIL,S) RECURSIVE;      /* CORRESPONDS TO */
 DCL (SNUM,NOL,I,SN,J,K,LASTS) FIXED BIN(31),   /* PROCEDURE TWO  */
     AVAIL(*) BIT(1),
     (TEMP,UB,LB) FLOAT(6),
     (TRY(ORIGSENT),FLAG) BIT(1);
 DCL UPDATE ENTRY EXT,
     ENTAIL ENTRY EXT;
 DCL 1 S(*),
       2 ATOM FIXED BIN(31),
       2 VALUE FIXED BIN(31);
 DCL (LISTPTR,LAST2,LIST(ORIGSENT)) FIXED BIN(31),
     LIST2(ORIGSENT) FIXED BIN(31),
     (SUM,HOLD) FLOAT(6);

 DCL 1 GB EXT,
       2 KNB(500,26),
         3 VALUE FIXED BIN(31),
         3 NEXTS FIXED BIN(31),
         3 NEXTA FIXED BIN(31),
       2 LEVEL(500) FIXED BIN(31),
       2 SENT(500) CHAR(20),
       2 STARTA(26) FIXED BIN(31),
       2 ENDA(26) FIXED BIN(31),
       2 STARTS(500) FIXED BIN(31),
       2 LASTSENT    FIXED BIN(31),
       2 BOUNDS(500),
         3 UPPER FLOAT(6),
         3 LOWER FLOAT(6),
       2 ORIGSENT FIXED BIN(31),
       2 NUMLEVEL FIXED BIN(31);

 LISTPTR=0;
 TEMP=0;
 TRY='0'B;
 I=STARTS(SNUM);
 DO WHILE(I^=0);
  J=STARTA(I);
  DO WHILE((J^=0)&(J<=ORIGSENT));
   IF (AVAIL(J))&(^TRY(J)) THEN DO;
     K=STARTS(SNUM);
     TRY(J)='1'B;
     FLAG='0'B;
     DO WHILE((K^=0)&(^FLAG));
       IF KNB(J,K).VALUE^=KNB(SNUM,K).VALUE THEN FLAG='1'B;
       K=KNB(SNUM,K).NEXTA;
     END;
     IF ^FLAG THEN DO;
       K=STARTS(J);
       LASTS=0;
       AVAIL(J)='0'B;
       DO WHILE(K^=0);
        IF KNB(SNUM,K).VALUE=0 THEN DO;
          LASTS=LASTS+1;
          S(LASTS).VALUE=KNB(J,K).VALUE;
          S(LASTS).ATOM=K;
          S(LASTS+1).ATOM=0;
         END;
        K=KNB(J,K).NEXTA;
       END;
       LISTPTR=LISTPTR+1;
```

```
      LIST(LISTPTR)=J;
      CALL ENTAIL(S,NOL-1,LB,UB,SN);
      UB=1-LB+BOUNDS(J).UPPER;
      LB=BOUNDS(J).LOWER;
      CALL UPDATE(SNUM,LB,UB);
     END;
    END;
    J=KNB(J,I).NEXTS;
   END;
   I=KNB(SNUM,I).NEXTA;
  END;

SUM=0;
FLAG='0'B;
LAST2=0;
I=1;
DO WHILE((LAST2^=0)](I<=LISTPTR));
 DO WHILE(I<=LISTPTR);
  IF DIFF(LIST(I)) THEN DO;
    FLAG='1'B;
    LAST2=LAST2+1;
    LIST2(LAST2)=I;
   END;
  I=I+1;
 END;
 HOLD=0;
 IF FLAG THEN DO;
   FLAG='0'B;
   DO J=1 TO LAST2;
    HOLD=HOLD+BOUNDS(LIST(LIST2(J))).LOWER;
   END;
  END;
 IF HOLD>SUM THEN SUM=HOLD;
 I=LIST2(LAST2)+1;
 LAST2=LAST2-1;
END;
HOLD=1;
CALL UPDATE(SNUM,SUM,HOLD);
RETURN;

DIFF:PROC(A) RETURNS(BIT(1));
 DCL (A,B,C,Z) FIXED BIN(31),
     FLAG    BIT(1);
 DO Z=1 TO LAST2;
  B=LIST(LIST2(Z));
  C=STARTS(A);
  FLAG='0'B;
  DO WHILE((C^=0)&(^FLAG));
   IF KNB(A,C).VALUE=((-1)*KNB(B,C).VALUE) THEN FLAG='1'B;
   C=KNB(A,C).NEXTA;
  END;
  IF ^FLAG THEN RETURN('0'B);
 END;
 RETURN('1'B);
END DIFF;
 END ENTAIL3;
```

```
ENTAIL4: PROC(SNUM,NOL,AVAIL,S) RECURSIVE;
  /* CORRESPONDS TO PROCEDURE THREE */
 DCL (SNUM,NOL,SN,I,J,K,LASTS,LIST(ORIGSENT),LASTL,LIST2(ORIGSENT),
      LISTJ(ORIGSENT),LAST2) FIXED BIN(31),
     AVAIL(*) BIT(1),
     (JUB,LB,UB,TEMP) FLOAT(6);
 DCL ENTAIL ENTRY EXT,
     UPDATE ENTRY EXT,
     FIND   ENTRY EXT RETURNS(FIXED BIN(31));
 DCL 1 S(*),
      2 ATOM FIXED BIN(31),
      2 VALUE FIXED BIN(31);

 DCL 1 GB EXT,
       2 KNB(500,26),
         3 VALUE FIXED BIN(31),
         3 NEXTS FIXED BIN(31),
         3 NEXTA FIXED BIN(31),
       2 LEVEL(500) FIXED BIN(31),
       2 SENT(500) CHAR(20),
       2 STARTA(26) FIXED BIN(31),
       2 ENDA(26) FIXED BIN(31),
       2 STARTS(500) FIXED BIN(31),
       2 LASTSENT    FIXED BIN(31),
       2 BOUNDS(500),
         3 UPPER FLOAT(6),
         3 LOWER FLOAT(6),
       2 ORIGSENT FIXED BIN(31),
       2 NUMLEVEL FIXED BIN(31);

 LASTL=0;
 I=STARTS(SNUM);
 DO WHILE(I^=0);
  J=STARTA(I);
  DO WHILE((J^=0)&(J<=ORIGSENT));
   IF AVAIL(J) THEN DO;
     K=STARTS(J);
     LASTS=0;
     AVAIL(J)='0'B;
     DO WHILE(K^=0);
      IF KNB(SNUM,K).VALUE^=0 THEN DO;
        LASTS=LASTS+1;
        S(LASTS).ATOM=K;
        S(LASTS).VALUE=KNB(SNUM,K).VALUE;
        S(LASTS+1).ATOM=0;
       END;
      K=KNB(J,K).NEXTA;
     END;
     SN=FIND(S);
     IF SN=0 THEN
                  CALL ENTAIL(S,NOL-1,LB,UB,SN);
     IF SN>0 THEN CALL ADDLIST(SN,J);
    END;
    J=KNB(J,I).NEXTS;
   END;
   I=KNB(SNUM,I).NEXTA;
  END;
  UB=1;
  DO I=1 TO LASTL;
   IF BOUNDS(LIST(I)).UPPER<UB THEN UB=BOUNDS(LIST(I)).UPPER;
```

```
 END;
 LAST2=0;
 LB=0;
 I=1;
 DO WHILE((LAST2^=0)](I<=LASTL));
  DO WHILE(^COVERED(SNUM)&(I<=LASTL));
   IF ^COVERED(LIST(I)) THEN DO;
     LAST2=LAST2+1;
     LIST2(LAST2)=I;
    END;
   I=I+1;
  END;
  IF COVERED(SNUM) THEN DO;
    TEMP=0;
    DO J=1 TO LAST2;
     TEMP=TEMP+BOUNDS(LIST(LIST2(J))).LOWER;
    END;
    TEMP=TEMP-(LAST2-1);
    IF TEMP>LB THEN LB=TEMP;
    IF SAME THEN DO;
     JUB=SHARE;
     TEMP=0;
     DO J=1 TO LAST2;
      TEMP=TEMP+BOUNDS(LISTJ(LIST2(J))).LOWER;
     END;
     TEMP=TEMP-(LAST2-1)*JUB;
     IF TEMP>LB THEN LB=TEMP;
    END;
    LAST2=LAST2-1;
   END;
  ELSE DO;
    I=LIST2(LAST2)+1;
    LAST2=LAST2-1;
   END;
 END;
 CALL UPDATE(SNUM,LB,UB);
 RETURN;

SAME: PROC RETURNS(BIT(1));
 DCL (Z,I) FIXED BIN(31);
 Z=STARTS(SNUM);
 DO WHILE(Z^=0);
  DO I=1 TO LAST2;
   IF KNB(LISTJ(LIST2(I)),Z).VALUE=(-1)*KNB(SNUM,Z).VALUE THEN
                RETURN('0'B);
  END;
  Z=KNB(SNUM,Z).NEXTA;
 END;
 RETURN('1'B);
END SAME;

SHARE: PROC RETURNS(FLOAT(6));
 DCL (JLB,JUB) FLOAT(6),
    (I,J,K) FIXED BIN(31),
    FLAG BIT(1);
 K=0;
 I=STARTS(LISTJ(LIST2(1)));
 DO WHILE(I^=0);
  FLAG='0'B;
  IF (KNB(SNUM,I).VALUE=0) THEN DO;
```

```
    DO J=2 TO LAST2 WHILE(^FLAG);
     IF (KNB(LISTJ(LIST2(J)),I).VALUE ^=
        KNB(LISTJ(LIST2(1)),I).VALUE) THEN FLAG='1'B;
    END;
    IF ^FLAG THEN DO;
      K=K+1;
      S(K).ATOM=I;
      S(K).VALUE=KNB(LISTJ(LIST2(1)),I).VALUE;
     END;
   END;
  I=KNB(LISTJ(LIST2(1)),I).NEXTA;
 END;
 S(K+1).ATOM=0;
 CALL ENTAIL(S,NOL-1,JLB,JUB,I);
 RETURN(JUB);
END SHARE;
ADDLIST: PROC(N,J);
 DCL (N,J) FIXED BIN(31);
 LASTL=LASTL+1;
 LIST(LASTL)=N;
 LISTJ(LASTL)=J;
 RETURN;
END ADDLIST;
COVERED: PROC(N) RETURNS(BIT(1));
 DCL (M,N,A) FIXED BIN(31),
     FLAG BIT(1);
 A=STARTS(N);
 DO WHILE(A^=0);
  FLAG='0'B;
  DO M=1 TO LAST2 WHILE(^FLAG);
   IF KNB(LIST(LIST2(M)),A).VALUE=KNB(N,A).VALUE THEN FLAG='1'B;
  END;
  IF ^FLAG THEN RETURN('0'B);
  A=KNB(N,A).NEXTA;
 END;
 RETURN('1'B);
END COVERED;
END ENTAIL4;
```

```
ENTAIL6: PROC(SNUM,NOL,S) RECURSIVE;  /* CORRESPONDS TO PROCEDURE */
 DCL (SNUM,SN,NOL,J,I,K) FIXED BIN(31),  /*  FOUR  */
     (LB,UB,LB1,UB1) FLOAT(6);
 DCL UPDATE ENTRY EXT,
     ENTAIL ENTRY EXT;

 DCL 1 S(*),
       2 ATOM FIXED BIN(31),
       2 VALUE FIXED BIN(31);

 DCL 1 GB EXT,
       2 KNB(500,26),
         3 VALUE FIXED BIN(31),
         3 NEXTS FIXED BIN(31),
         3 NEXTA FIXED BIN(31),
       2 LEVEL(500) FIXED BIN(31),
       2 SENT(500) CHAR(20),
       2 STARTA(26) FIXED BIN(31),
       2 ENDA(26) FIXED BIN(31),
       2 STARTS(500) FIXED BIN(31),
       2 LASTSENT    FIXED BIN(31),
       2 BOUNDS(500),
         3 UPPER FLOAT(6),
         3 LOWER FLOAT(6),
       2 ORIGSENT FIXED BIN(31),
       2 NUMLEVEL FIXED BIN(31);


 K=1;
 DO WHILE('1'B);
  I=STARTS(SNUM);
  J=1;
  DO WHILE(I^=0);
   S(J).ATOM=I;
   S(J).VALUE=KNB(SNUM,I).VALUE;
   I=KNB(SNUM,I).NEXTA;
   J=J+1;
  END;
  S(J).ATOM=0;
  S(K).VALUE=S(K).VALUE*(-1);
  IF (K=J) THEN LEAVE;
  CALL ENTAIL(S,NOL-1,LB1,UB1,SN);
  DO I=K TO (J-1);
   S(I).ATOM=S(I+1).ATOM;
   S(I).VALUE=S(I+1).VALUE;
  END;
  CALL ENTAIL(S,NOL-1,LB,UB,SN);
  UB=UB-LB1;
  LB=LB-UB1;
  CALL UPDATE(SNUM,LB,UB);
  K=K+1;
 END;
 RETURN;
END ENTAIL6;
```

```
UPDATE: PROC(SNUM,LB,UB);
 DCL (SNUM,SP,I) FIXED BIN(31),
     (LB,UB) FLOAT(6),
     (FLAG1,FLAG2) BIT(1),
     OUT FILE STREAM OUTPUT PRINT;

 DCL 1 GB EXT,
       2 KNB(500,26),
         3 VALUE FIXED BIN(31),
         3 NEXTS FIXED BIN(31),
         3 NEXTA FIXED BIN(31),
       2 LEVEL(500) FIXED BIN(31),
       2 SENT(500) CHAR(20),
       2 STARTA(26) FIXED BIN(31),
       2 ENDA(26) FIXED BIN(31),
       2 STARTS(500) FIXED BIN(31),
       2 LASTSENT    FIXED BIN(31),
       2 BOUNDS(500),
         3 UPPER FLOAT(6),
         3 LOWER FLOAT(6),
       2 ORIGSENT FIXED BIN(31),
       2 NUMLEVEL FIXED BIN(31);

 FLAG1='0'B;  FLAG2='0'B;
 SP=10*(NUMLEVEL-LEVEL(SNUM))+1;
 DO I=1 TO (SP-1) BY 10;
  PUT FILE(OUT) EDIT(']')(COL(I),A);
 END;
 IF BOUNDS(SNUM).LOWER<LB THEN DO;
    BOUNDS(SNUM).LOWER=LB;    FLAG1='1'B;
   END;
 IF BOUNDS(SNUM).UPPER>UB THEN DO;
    BOUNDS(SNUM).UPPER=UB;   FLAG2='1'B;
   END;
 IF(FLAG1]FLAG2) THEN DO;
   PUT FILE(OUT) EDIT(SENT(SNUM),'(',LB,', ',UB,')')
          (COL(SP),A(20),A,F(8,4),A,F(8,4),A);
   IF FLAG1 THEN PUT FILE(OUT) EDIT('LOWER')(X(3),A);
   IF FLAG2 THEN PUT FILE(OUT) EDIT('UPPER')(X(3),A);
  END;
 RETURN;
END UPDATE;
```

# BIBLIOGRAPHY

## Books

Schafer, G.A.. Mathematical Theory of Evidence. Princeton: Princeton University Press, 1979.

Shortliffe, E. H.. Computer Based Medical Consultations: MYCIN. New York: Elsevier, 1976.

## Articles

Duda, R.O., Hart, P.E., and Nilsson, N.J.. "Subjective Bayesian Methods for Rule-base Inference Systems," in: Proceedings 1976 National Computer Conference, AFIPS 45 (1976) pp. 1075-1082.

Halpern, J.Y. and Rabin, M.O.. "A Logic to Reason About Likelihood," Artificial Intelligence, 32 (1987) pp. 379-405.

Nilsson, N.J.. "Probabilistic Logic," Artificial Intelligence, 28 (1986) pp. 71-87.

Santos, E.S. and Santos, E. Jr.. "Reasoning With Uncertainty in a Knowledge Based System," in: Proceedings the Seventeenth International Symposium on Multiple-Valued Logic, (May 1987) pp. 75-81.

Zadeh, L.A.. "Fuzzy Logic and Approximate Reasoning," Synthese, 30(1975) pp. 407-428.