This is to certify that the

dissertation entitled

ON THE APPLICATION OF RELEVANCE MEASURES
IN MECHANICAL DEDUCTION

presented by

James Stephen Soddy

has been accepted towards fulfillment
of the requirements for

Ph.D. degree in Computer Science

Carl V. Page
Major professor

Date June 4, 1982

ON THE APPLICATION OF RELEVANCE MEASURES IN MECHANICAL DEDUCTION

By

James Stephen Soddy

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Department of Computer Science

1982

ABSTRACT

ON THE APPLICATION OF RELEVANCE MEASURES
IN MECHANICAL DEDUCTION

By

James Stephen Soddy

In this thesis, the question of the relevance of one predicate to another is investigated in the light of choosing clauses as input to a resolution theorem prover. Several potential measures of relevance are surveyed and entropy is chosen for primary investigation. A procedure is developed for using such a measure to choose input clauses which can shorten the search for a proof. In a similar fashion, the question of the relevance of one attribute to another in a relational data base is explored, and a method is developed to use the entropic relevance measure for finding keys to the relation.

To Ginny

ACKNOWLEDGEMENTS

I am most grateful to Dr. Carl Page, the chairman of my guidance committee, for his encouragement, his professional and personal counsel, and his boundless patience through the development of this thesis. My thanks also go to Dr. Richard Dubes, Dr. John Forsyth, Dr. Leroy Kelly, and Dr. Hans Lee for serving on my guidance committee, for their excellent teaching, and for their helpful suggestions in their review of this work.

I am grateful to the Division of Engineering Research for generous financial support of this work.

Finally, I am indebted to my wife Virginia for her support through difficult years and for her careful preparation of this manuscript.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION AND BACKGROUND

1.1 INTRODUCTION

Mechanical theorem proving has many potential applications in the realm of artificial intelligence. Consider, for example, a deductive question answering system. Such a system consists of a data base containing facts and information relating those facts, along with some mechanism (a computer program) which allows the user to interrogate the system about those facts and relations. A deductive capability allows such a system to deduce answers which are not explicitly stored, if they follow logically from the information which is stored.

There is however, a practical difficulty which is encountered in this, or any other, application of mechanical theorem proving. The size of the search for a proof becomes so large that time and space limits are reached, or the search becomes unjustifiably expensive. A great deal of research effort has been expended in the direction of guiding and limiting the search for proofs.

An extensive study of theorem proving strategies at Stanford Research Institute [REBOH] concludes:

> "We believe that the most promising strategies and those to which most of the future research effort should be directed are those that are concerned with the semantics of clauses, predicates, and functions involved. The investigation of new syntactic-type strategies is of course not without importance, but we believe that the practical limit on the usefulness of such strategies has been reached."

Another study done at the University of Maryland [Wilson] concludes:

> "None of the inference systems tested enabled more than a marginal improvement in the overall power of unrestricted binary resolution. We suspect that further testing would cause the same conclusions to be reached about other refinements to binary resolution not tested here. ... We concur with the conclusions of Reboh that practical limits on syntactic type strategies are near, if not already acheived. ... We advocate investigation into ways for incorporating domain-dependent and problem-specific information into the unification process, the inference system, and the search strategy."

The question addressed in this research is how measures of "relevance" between predicates might be computed and employed in such a way as to be useful in directing the search for a proof. The best search strategies which are now available fall far short of human ability to select the "most promising paths" toward a proof. The role of the relevance measure is to aid in the selection of these paths. The measure should provide an ordering of the available information according to the liklihood that it will be employed in proving a given theorem. We might wish to interpret the statement 'A is relevant to B' as meaning that A is likely to be included in a deduction of B.

We show in this research that entropy is a reasonable

basis for computing a relevance measure; and suggest probability estimates as a method for capturing "expert knowledge" about relevance, We examine the structure of inter-predicate relationships. This examination serves to show both some strengths and limitations of our approach.

Included in this chapter are a brief description of resolution theorem proving, and a mechanism for its employment in question answering. In chapter 2 we survey several potential measures of relevance. Chapter 3 contains a discussion and example of a strategy for employing a relevance measure in a proof. In Chapter 4 we look at the object-predicate table in an effort to gain insight about the nature of the entropic relevance measure. In Chapter 5 some of the difficulties inherent in combining relevant evidence are considered. Finally, in Chapter 6, we compare the object-predicate table and the relational data base, and consider how the entropic relevance measure can be used in the analysis of the latter.

1.2 AN INTRODUCTION TO THEOREM PROVING USING RESOLUTION

The essential background on this topic may be found in Chang and Lee [1973]. We first establish some definitions essential to describing mechanical theorem proving techniques. A representation of formulas in which quantifiers and connectives do not explicitly appear is specified, and a method of proving theorems with the use of representation is demonstrated.

Before continuing further, it is necessary to state some basic definitions. They are as follows:

TERM: A constant, a variable, or an n-place function symbol followed by n terms.

(e.g. a, x, f(x), g(a, f(x)))

ATOM: (Atomic formula) An n-place predicate symbol followed by n terms.

LITERAL: An atom or the negation of an atom.

(e.g. Pa, -Pa, Qxy, -Qxf(x))

CLAUSE: The disjunction of zero or more literals.

CONNECTIVES: Implication '=>', disjunction 'V', conjunction '&', and negation '-'.

QUANTIFIERS: Universal '(x)' and existential 'Ex'.

Unless otherwise indicated, a,b,c represent constants; f,g,h, represent functions; x,y,z, represent universal variables, and upper case letters represent predicates.

If a formula consists of a string of quantifiers followed by M, where M is the remainder of the formula (called the matrix) and contains no quantifiers, then the formula is in PRENEX NORMAL FORM.

Every formula has an equivalent prenex normal form from which we can obtain its SKOLEM STANDARD FORM as follows:

1. The matrix is converted to conjunctive normal form. (i.e. a conjunction of disjunctions)
2. The existential quantifiers are eliminated by introducing SKOLEM FUNCTIONS. (i.e. The existential variables are replaced by functions of those universal variables which precede them.)

Two examples follow:

Formula 1: (x)(Ey)(Px V (Qxy & Rxy))

CNF: (x)(Ey)((px V Qxy) & (Px V Rxy))

SSF: (x)((Px V Qxf(x)) & (Px V Rxf(x)))

Formula 2: (Ex)(y)(Ez)(u)(v)(Ew)Pxyzuvw

SSF: (y)(u)(v)Payf(y)uvg(y,u,v)

Now, the only quantifiers in an SSF formula are universal. If we maintain a standard convention to distinguish universal variables from constants, we may then drop the remaining quantifiers. If we make the further convention that a set of clauses represents the conjunction of those clauses, we obtain the CLAUSEFORM of a formula. In this form, we would represent formula 1 as follows:

(Px V Qxf(x), Px V Rxf(x))

In a proof, each line will be a clause so that the symbol for disjunction may also be omitted. Formula 1 would appear in a proof as:

1. Px, Qxf(x)

2. Px, Rxf(x)

The RESOLUTION proof procedure is a mechanical procedure which will derive a contradiction (THE EMPTY CLAUSE) if, and only if, it is given an inconsistent set of clauses. Further, a first order predicate calculus formula is inconsistent if, and only if, the associated clauseform is an inconsistent set of clauses. Thus, to prove a theorem from a set of clauses representing the axioms and the negation of the theorem is inconsistent.

There are two rules of inference, the first of which

is SUBSTITUTION FOR UNIVERSAL VARIABLES. Two literals, L1 and L2, are said to be UNIFIABLE if there is a substitution s such that s applied to L1 yeilds the same result as s applied to L2. For example, the literal Px and the literal Pa are unifiable by the substitution a/x (a for x). The literals Pa and Pb are not unifiable because a and b are both constants.

The second rule allows us to form a new clause given two clauses C1 and C2 containing literals L1 and L2 respectively, and such that L1 is the negation of L2. For example:

C1: Pa, Qax, -Rc

C2: Rb, -Qax

Resolvent: Pa, -Rc, Rb

1.3 QUESTION ANSWERING USING RESOLUTION

Consider the application of the resolution proof procedure in the question answering environment. The ideas presented here are due to Cordell Green [1969] and may also be found in Hunt [1975]. A deductive QA system is capable of deducing the answer to a question even though that answer is not explicitly stored, as long as the answer is a logical consequent of facts and relations which are stored.

One approach to a deductive QA system is the use of a theorem prover. The facts and their relationships are stored as expressions in the first order predicate calculus. The system then treats a question as a theorem to be proved and in the process of finding the proof, generates the

answer to the question. This approach provides the user with the expressive capability of the predicate calculus, a language which is well defined, unambiguous, and very general. With a complete proof procedure, and sufficient time and space, an answer will always be found if sufficient information is present.

A possible dialogue between the user and a QA system is suggested by Green [1969] as follows:

1. The first statement is 'Smith is a man.'
   Input - STATEMENT: MAN(SMITH)
   Response - OK

2. Ask the question, 'Is Smith a man?'
   Input - QUESTION: MAN(SMITH)
   Response - ANSWER: YES

3. STATEMENT: (x)(MAN(x) -> ANIMAL(x))
   OK

4. Question: (Ey)ANIMAL(y)
   ANSWER: YES, y=SMITH

5. STATEMENT: (x)(ROBOT(x) -> MACHINE(x))
   OK

6. STATEMENT: ROBOT(ROB)
   OK

7. STATEMENT: (x)(MACHINE(x) -> -ANIMAL(x))
   OK

8. QUESTION: (x)ANIMAL(x)
   ANSWER: NO, x=ROB

9. STATEMENT: AT(SMITH,WORK) V AT(JONES, WORK)
   OK

10. QUESTION: (Ex)AT(x, WORK)
    ANSWER: YES, x=SMITH OR x=JONES

Evidently, we require more than the simple knowledge that our "theorem" has been proved. The system must keep track of the values of the variables which lead to the proof so that it can provide the answer which we are seeking.

Green accomplishes this by appending the 'Answer' predicate to the theorem to be proved. When a clause containing only the answer predicate is derived, the theorem prover views it as the empty clause and considers the theorem proved. The value of the variable in the answer predicate is the answer to the question. A demonstration of how this is used to obtain answer number 4 in the preceeding example is now given.

1. -ANIMAL(Y), ANSWER(Y)
   (When the question is negated, we obtain (Y)-ANIMAL(Y). The clauseform is found by deleting the quantifier and adding the ANSWER predicate.)

2. -MAN(X), ANIMAL(X)
   (Clauseform of statement 3.)

3. -MAN(X), ANSWER(X)
   (Resolvent of 1 and 2.)

4. MAN(SMITH)
   (Statement 2.)

5. ANSWER(SMITH)
   (SMITH/X in 3 and resolve with 4.)

The preceeding example is a simple illustration of how the question answering process might be carried out, given relevant information. (An extension of the "answer finding" technique may be found in Nilsson [1980].) However, the example sidesteps the problem of finding and ordering the relevant clauses, given a large base of facts and a question to answer from them. This problem requires that we employ a SELECTION STRATEGY which chooses the clauses required to solve the problem at hand, and a SEARCH STRATEGY which determines the order in which the inference mechanism employs these clauses in a search for proof.

The degree to which a QA system is practical in a large data base environment depends on the efficiency of these strategies. The effort which has gone into their study has been oriented almost exclusively to syntactic criteria such as the length of clauses, the occurrence of common predicates and constants, and the depth within the search tree. Some "semantic" strategies have been introduced which rely upon an arbitrary model of a set of clauses [KOWALSKI, 1969; LOVELAND, 1969].

FISHMAN [1973] proposed a "semantic closeness" measure based upon syntactic criteria, which he found was not noticeably helpful. He attributes this to the fact that the "... distance of the axioms is not at all an indication of their relevance to a particular query". The central purpose of this research is to compute a measure which does reflect the relevance of an axiom to a query.

1.4 IMPLEMENTATION CONSIDERATIONS

As we have previously observed, the size of the search space is a practical limitation in a computer implementation of a mechanical theorem prover. The most straightforward computer method is "unrestricted resolution" employing the "level-saturation (resolution) method" [CHANG, 1973]. To implement this procedure, we begin with the list of clauses representing the axioms and the negation of the theorem to be proved. This list constitutes level 0 of the proof. We perform all possible resolutions of the level 0 clauses,

thereby obtaining the level 1 list. We then perform all possible resolutions of the level 1 clauses with the level 0 clauses and with each other, thus obtaining the level 2 list. This process continues until the null clause appears in the list. As may be seen in the Chang [1973] example, this procedure introduces many redundant and irrelevant clauses into the proof.

Two syntactic strategies stand out for ease of implementation and general reduction of search size. A first approach to eliminating unwanted clauses is known as the "deletion strategy" [CHANG, 1973]. A clause C is subsumed by a clause D if there is a substitution s, such that s applied to D yeilds a subset of C. In other words, D subsumes C means D implies C. When a clause is generated which is a tautology or is SUBSUMED by another clause, it is deleted from the list. For example, the clause P V Q V R is subsumed by the clause P V Q. It is given in Chang [1973] that the deletion strategy is complete when employed with the level saturation method. The Maryland study [WILSON, 1976] found a restricted form of the deletion strategy, which allows substitution for single variables only, was not very costly while it significantly reduced the growth of the search space. Unrestricted deletion was found to be too costly in compute time.

A second strategy which may be employed in addition to deletion is the set-of-support strategy [CHANG, 1973]. Recall that the resolution procedure involves showing that a set of clauses S is inconsistent. A subset T of S is

called a set-of-support if S-T is consistent. The set-of-support strategy does not allow any resolutions between clauses in the set S-T. It is shown in Chang that the method is complete. The Maryland study [Wilson, 1976] indicates that resolution with set-of-support did perform significantly better than any of the other five inference systems tested.

We will employ deletion and set-of-support along with relevance based selection of clauses in the examples in Chapter 3 of this research. A study of these examples will aid in understanding the techniques discussed above.

CHAPTER 2

SOME MEASURES OF RELEVANCE

Recall that the major obstacle to the use of mechanical proof procedures is the very large number of paths which are developed in the search for a proof. The role of the relevance measure is to aid in the selection of the "most promising paths" to explore in the search for a proof. The relevance measure is required to lead to an ordering of the information available according to the liklihood that the information will be used in a proof of a theorem.

In this chapter, we consider possible measures of the relevance which one predicate has to another. Subsequently, it will be necessary to develop a strategy for the selection of clauses most relevant to a theorem. The measures considered here are probabilistic and statistical in nature. Their computation uses estimates of the joint and conditional probabilities of the properties or events in question. For the present, assume that the probability estimates are available as needed while we consider some of the possible measures of one predicate to another. The problem of estimating probabilities is considered in chapter 3.

The meaning of the word 'predicate' has some significant consequences regarding potential measures of relevance. A

predicate is a two valued function of one or more arguments, whose two values are normally interpreted as 'true' and 'false'. For example, G(x,y) might be interpreted as 'x is greater than y', and would be either true or false for a specific x,y pair. Hence, the values of a predicate are nominal, as opposed to being ratio, interval, or ordinal values. Since, however, the value of a predicate is binary, measures normally reserved for interval data (e.g. the product moment correlation) may be applied. In fact, any analysis method which is invariant under linear transformation may be applied to binary variables [ANDERBERG, 1973]. As a consequence, many kinds of measures are available and the choice must be based on the interpretation of a particular relevance measure, or how well it works in practice. In this chapter, we consider several classes of relevance measures in the light of the theorem proving application.

Since we are considering the relationship between two binary variables, we may summarize their relationship in a 2x2 contingency table. The discussion in this chapter centers around such a table. On the next page is a summary (Table 2.1) of some of the measures which we can compute from a contingency table. These and other possible measures are discussed in the material which follows.

TABLE 1

ASSOCIATION MEASURES BASED ON THE 2x2 CONTINGENCY TABLE

		S	
		1	0
R	1	A	B
	0	C	D

Correlation Coefficient

$$CC = (AB-BC) / SQRT[(A+B)(C+D)(A+C)(B+D)]$$

Cosine of the Angle, Θ, Between the Vectors

$$COS(\Theta) = [SQRT(A/(A+B)][SQRT(A/(A+C)]$$

Cross-ratio

$$X = BC / AD$$

Gamma

$$Q = (AD - BC)/(AD + BC)$$

Various means of the Conditional Probabilities

$$TM = \tfrac{1}{2}[A/(A+B) + A/(A+C)]$$

$$FM_1 = \tfrac{1}{4}[A/(A+B) + A/(A+C) + D/(B+D) + D/(C+D)]$$

$$FM_2 = \tfrac{1}{4}[B/(A+B) + B/(B+D) + C/(A+C) + C/(C+D)]$$

Concomitant Variation

$$CV = FM_1 - FM_2$$

TABLE 2

SAMPLE COMPUTATIONS

PRODUCT MOMENT CORRELATION = -0.485000

CONCOMITANT VARIATION = -0.485000

CHI SQUARED = 0.235225

CROSS RATIO = 989901.00

GAMMA = -0.999998

COSINE OF THE ANGLE = 0.010000

AVERAGE OF TWO COSINES = 0.005050

ARITHMETIC MEAN OF PROBABILITIES = 0.010000

MEAN OF FOUR PROBABILITIES = 0.257500

CONDITIONAL ENTROPY = 0.693550

UNCONDITIONAL ENTROPY = 0.918296

		S	
		1	0
R	1	1	99
	0	99	101

PRODUCT MOMENT CORRELATION = 0.580065

CONCOMITANT VARIATION = 0.603598

CHI SQUARED = 0.336476

CROSS RATIO = 3459.27

GAMMA = -0.999422

COSINE OF THE ANGLE = 0.587975

AVERAGE OF TWO COSINES = 0.582520

ARITHMETIC MEAN OF PROBABILITIES = 0.612857

MEAN OF FOUR PROBABILITIES = 0.801799

CONDITIONAL ENTROPY = 0.123203

UNCONDITIONAL ENTROPY = 0.177908

		S	
		1	0
R	1	11	14
	0	3	906

2.1 STATISTICAL MEASURES

Assume that the knowledge we have about two predicates is represented by a 2x2 contingency table which shows how many out of n objects yield values of true and false for each predicate. 'A' tells the number of objects about which both predicates are true, and so on.

		S	
		1	0
R	1	A	B
	0	C	D

All of our statistical measures of association of 'R' and 'S' may be computed using the values in this table. Also, we might to some extent infer their operational interpretations by examining these computations.

The predicates 'R' and 'S' are being applied to n objects and receiving a score of 0 or 1 on each object. (n = A+B+C+D). Consider the order of the objects to be fixed, and the result is a vector of scores for each predicate, say X for the first, and Y for the second. Given that the vectors are identical, then the interpretations of the predicates are identical in the given universe. If they are not identical, there are a number of ways to compute a measure of "similarity" or "dissimilarity" of these vectors. One method is to compute a measure of the angle between these vectors in n-space.

It can be shown that the cosine of this angle is given by the following:

$$\cos(\text{angle}) = A/\text{SQRT}[(A + B)(A + C)]$$

(This is equivalent to the inner product of the vectors divided by the product of their lengths.) If the means of the scores are subtracted from the original scores, and we compute the inner product of each vector with itself, and divide by n, we obtain the variance. Similarly, we can compute the covariance.

$$\text{VAR}(X) = X \cdot X/n$$

$$\text{VAR}(Y) = Y \cdot Y/n$$

$$\text{COV}(X,Y) = X \cdot Y/n$$

The product moment correlation is then computed:

$$R(X,Y) = \text{COV}(X,Y)/\text{SQRT}[(\text{VAR}(X))(\text{VAR}(Y))]$$

R(X,Y) may also be computed as the cosine of the angle between X' and Y' which represent X and Y normalized to zero mean and unit variance.

In terms of our 2x2 table:

$$R(X,Y) = (AD - BC)/\text{SQRT}[(A+B)(C+D)(A+C)(B+D)]$$

'AD - BC' represents the product of the number of matches (both zero or both one) minus the product of the numbers of mismatches. The factors in the denominator are the row and column sums. (e.g. 'A + B' represents the number of objects for which 'R' is true. The denominator is in fact the square of the geometric mean of these factors.)
It is given in Anderberg [1973] that the Chi-square statistic for the 2x2 contingency table is n times the square of the product moment correlation. Relative to the

Chi-square he further comments that it and related measures are useful as tests of hypotheses, but not so useful as measures of association. Similarly, Goodman and Kruskal [1954] state that they "have been unable to find any convincing published defense of Chi-square like statistics as measures of association." The correlation coefficient on the other hand is well established as a measure of association between two vectors of scores. Hogg and Craig [1970] explain that R(X,Y) is a measure of the intensity of concentration of the probability for X and Y about a line of the form Y = a + b(X). It would appear from the foregoing that the correlation coefficient would be a reasonable choice from among the statistical measures for association between predicates.

2.2 MEASURES BASED ON THE CROSS RATIO

Another group of measures useful in the analysis of 2x2 tables is based on the cross ratio (X = BC/AD). This is the ratio of mismatches to matches between the two predicates. One possible function of the cross ratio is $Q = (1-X)/(1+X)$. In terms of the table values:

$$Q = (AD - BC)/(AD + BC)$$

Q is independent of the marginal totals in a 2x2 table and is equivalent to a measure which Goodman and Kruskal [1954] call "gamma".

For the purpose of associating predicates, the cross ratio does not appear to be promising at all. That this

ratio is sensitive to concentration of values on a particular diagonal of the table is clearly seen in the sample computations. For purposes of predicate association, we might wish to detect concentrations anywhere in the table. This idea appears to be borne out when we look at the results of computing this and other measures of association for some particular tables.

2.3 MEASURES BASED ON CONDITIONAL PROBABILITIES

We have considered the cosine of the angle (Θ) between two vectors of scores as a measure of association. Note that it may be written as follows:

$$\cos(\Theta) = [\sqrt{A/(A+B)}][\sqrt{A/(A+C)}].$$

This is the geometric mean of two conditional probabilities. Namely, A/(A+B) is the probability of 'R' given 'S'. If we reversed the meanings of the zeroes and ones in the table, the corresponding computation would be:

$$\cos(\Theta_2) = [\sqrt{D/(B+D)}][\sqrt{D/(C+D)}]$$

The product of these two cosines is sometimes used as a measure of association in a 2x2 table.

An alternative approach is to employ arithmetic rather than geometric means of probabilities. We may define some as follows:

$$TM = \tfrac{1}{2}[A/(A+B) + A/(A+C)]$$

$$FM_1 = \tfrac{1}{4}[A/(A+B) + A/(A+C) + D/(B+D) + D/(C+D)]$$

$$FM_2 = \tfrac{1}{4}[B/(A+B) + B/(B+D) + C/(A+C) + C/(C+D)]$$

TM and FM_1 are average probabilities of matches in the

table. FM_2 is a corresponding mean of probabilities of mismatches. The difference,

$$CV = FM_1 - FM_2$$

is mathematically identical to the measure which Haralick [1975] calls the concomitant variation of events 'R' and 'S'. In his paper, he describes applications of this measure to form clusters based on association. This measure assumes the same range of values as the product moment correlation, and in the tables [TABLE 2] which were evaluated as trial computations, it shows equal to 'nearly equal' values.

2.4 MEASURES BASED ON OPTIMAL CLASS PREDICTION

The basic question asked here is: "If we know whether or not 'S' is true, what effect does this have on the probability that we can correctly guess whether 'R' is true?" Consider the following table, [AGARD, 1971]

		S	
		1	0
R	1	1	99
	0	99	101

If we do not know 'S'. we can guess 'R' correctly 2/3 of the time, by a strategy such as always guessing that 'R' is false. It happens that this remains exactly the same if we in fact DO know the truth value of 'S'! That is, the probability of a correct guess is given by:

$$1/3(99/100) + 2/3(101/200) = 2/3.$$

Although there is no increase, on the average, in any ability to predict 'R' given 'S', a knowledge of 'S' significantly increases my knowledge of the probability distribution of 'R'. Thus, I consider 'S' relevant to 'R' in a manner which measures of this type are unable to sense. This example will be considered further in the following section.

2.5 ENTROPIC MEASURES

When we prove a theorem, we may say that we have reduced our "uncertainty" about that theorem to zero. We might also say, about individual clauses used in a proof, that each makes a contribution toward our reduction in uncertainty about the theorem. This viewpoint leads to a conjecture, that the first clauses to be employed in an attempt to prove a theorem should be those which cause the greatest reduction in our uncertainty about the theorem. A rigorous description of this idea will require a quantitative definition of what is meant by "uncertainty". An approach is to assume that our uncertainty as to which of several events is going to occur is a function of the probabilities of those events. In what follows, let us assume that $H(P_1,P_2,...P_m)$ is such a function of several possibilities.

Robert Ash [1965] describes four requirements which might be placed on an uncertainty measure. The mathematical descriptions given by Ash might be interpreted as follows:

1. If we increase the number of equiprobable choices for the outcome of an experiment, we increase our uncertainty of the outcome.

   For example, $H(\frac{1}{2},\frac{1}{2})$ $H(\frac{1}{4},\frac{1}{4},\frac{1}{4},\frac{1}{4})$. In fact, one form of the function to be suggested yields 1 bit and 2 bits respectively for the measure of uncertainty in these two cases.

2. Our uncertainty as to the joint occurrence of two independent events equals the sum of our uncertainties as to their individual occurrences.

   For example, assume that E_1 and E_2 are independent events, each having probabilities $\frac{1}{2},\frac{1}{2}$. Then their joint distribution will be $\frac{1}{4},\frac{1}{4},\frac{1}{4},\frac{1}{4}$. (If the uncertainty of either event is 1 bit, the uncertainty of the joint event must be 2 bits.)

3. The average reduction in uncertainty which results from an observation does not depend upon whether we consider that observation in its entirety or broken into component parts.

   For example, consider choosing at random a digit by first deciding even or odd (1 bit) and then choosing which even or odd digit (2 bits), then the total uncertainty should be 3 bits.

4. The uncertainty measure must be a continuous function of the probabilities involved. That is, a small change in the probabilities must be associated with a small change in our uncertainty.

It is then proved [ASH, 1965] that there is only one function which satisfies the four given conditions. Given m possible outcomes of an experiment, having respective probabilities of P_1, P_2,...P_m, our uncertainty function is given by:

$$H(P_1, P_2,...P_m) = -C[\sum_i P_i(\text{Log}(P_i))]$$

where C is an arbitrary positive number, and the logarithm base is greater than one. This function is known in information theory as the entropy of an information source. The particular form of H to which the examples referred uses C=1 and logarithms base 2. As an information measure, 1

bit is equivalent to answering a yes/no question.

We may view any reduction in uncertainty as information gained about the outcome of an event, hence the uncertainty measure is a measure of the information which is gained by performing an experiment. To meet our needs, this measure is extended to tell us how much information is gained regarding an event Y if we are told that a second event X has occurred. If $P(X_i, Y_j)$ is the probability of the i'th possible outcome for X and the j'th possible outcome for Y, and $P(Y_j/X_i)$ is the corresponding conditional probability, then the conditional uncertainty of an event is defined as follows:

$$H(Y/X) = - \sum_{ij} P(X_i,Y_j)\text{Log}[P(Y_j/X_i)].$$

The conditional uncertainty is never larger than the absolute uncertainty, H(Y), further, there is equality provided that X and Y are independent. In general, the difference between the uncertainties is a measure of the information contained in X about Y. Thus, based upon some reasonable assumptions as to the nature of an uncertainty measure, we have arrived at conditional entropy to tell us how much information one event contains about another.

Returning to the example from the previous section, recall that knowledge of event S did not increase our ability to predict event R. H(R) = .9183 and H(R/S) = .6936 which shows approximately a 20% reduction in our uncertainty of R. Thus, conditional entropy detects the type of relevance which 'increase in predictability' fails to uncover.

Finally, we should consider how the correlation

coefficient is likely to compare with the entropy function as a relevance measure. Watanabe [1969] states that the correlation coefficient measures the degree of agreement between the values, whereas entropy measures the degree of departure from probabilistic independence. It is the case that probabilistic independence implies zero correlation, but the converse is not true. Thus the "entropic measure can uncover a relation ...that the correlation coefficient may not uncover." [Watanabe, 1969]

The discussion in this chapter combined with the sample computations suggests that entropy is a suitable basis for relevance measurement. Considerable support for this choice is added in Chapter 4 as we study the underlying structure which must be captured by the relevance measure. That the measure can indeed serve as a basis for selection of appropriate clauses from a larger set is demonstrated in the next chapter.

Chapter 3

RELEVANCE AND THEOREM PROVING

3.1 THE ROLE OF EXPERT KNOWLEDGE

We have already considered the idea that given a 2x2 table of association between two predicates, we can assign measures of association in various ways; in particular, we can compute the entropy of the table. Now, we will consider in more detail, the derivation of the original table and the significance of the entropy measure in this context.

Suppose that 'R' and 'S' represent either predicates or propositions. Then P(R&S) = A/n, P(R&-S) = B/n, and so on. The measures of association discussed in Chapter 2 are functions of relative frequency, so we may replace A, B, C, and D with the correct probabilities. Completion of the 2x2 table requires that we know P(R), P(S), and P(R/S). In general, exact values for these probabilities are not available. Thus, the question is reduced to whether or not workable approximations to the probabilities are available. We argue that such approximations might reasonably be available in the context of this application.

Recall that this work is an effort to provide a tool which may limit the size of search for proof in a mechanical

deduction. When skillful humans search for a proof, they employ a kind of judgement as to what information is most relevant and which paths are most promising. We might consider it desirable to incorporate some of that judgement in a mechanical proof environment. Now, let us consider a quote from Watanabe [1969].

> "Thus we are led to consider the notion of conditional probability (or its crude prototype) as the most primary and basic form of thought. ...In talking about the probability of a proposition B, if we knew (or determined the truth or falsehood of) all the facts relevant to B, the occurrence or non-occurrence of B would already be determined by these facts and there would hardly be any room for probabilistic guessing. On the other hand, if nothing relevant were known except logical tautologies, the value of the probability could not be decided at all, except perhaps by counting the possible cases the language happens to provide. ...The true usefulness of probability resides in the intermediate domain between these two extremes, and the value of probability depends critically on the relevant facts that are taken into account."

One approach to our problem is to accept human estimates of the unknown probabilities in an effort to obtain a reflection of human judgement. For the purposes of making these estimates, our "human theorem prover" need not provide correct values of these probabilities, but only the values which he uses in finding proofs. This is prehaps analagous to Jaynes' [1979] statement, concerning predicting the outcomes of experiments, that "we are asking only for predictions of EXPERIMENTALLY REPRODUCIBLE things; and for these all circumstances that are not under the experimenter's control must, of necessity, be irrelevant."

3.2 CONTEXT PROVIDES CONSTRAINTS

Let R1, R2, ...Rn represent a set of propositions or predicates and consider what is involved in constructing a table of probabilities. We might begin by estimating P(Ri), i=1,n. Keep in mind that these estimates are to be made in the context of a set of clauses (axioms) which includes relations among the predicates. This provides significant help in estimating the probabilities. First, it provides a context in which to decide which of the "relevant" facts to take into account. Second, it provides prima facie constraints on the estimates in a manner to be discussed shortly.

Once we have estimates of the P(Ri), we might next construct estimates of the P(Ri/Rj). Note that for each i,j pair, an estimate of P(Rj/Ri) will be computable from the previously estimated probabilities using:

$$[P(Ri/Rj)][P(Rj)] = [P(Rj/Ri)][P(Ri)].$$

This application of Baye's Theorem allows us to make a choice of which conditional probability is easier to estimate for each pair of propositions or predicates. For example, P(wearing jacket/ it is snowing) should be easier to estimate than P(it is snowing/ wearing jacket). We also have the option of estimating both conditional probabilities and cross checking the estimates for consistency, and then making the necessary revisions. Once again the axiom system provides constraints which are an aid in estimating the

the probabilities. Let us focus on the nature of these constraints.

Consider the fact that each clause is a disjunction, and that there is a strong relation between disjunction and implication. That is:

$$(R1 \rightarrow R2) \leftrightarrow (-R1 \vee R2).$$

Then, given the clause -R1 V R2, we can state the following:

$$P(R2/R1) = 1$$

$$P(R1/R2) = P(R1)/P(R2)$$

$$P(R1) \leq P(R2)$$

$$P(R1) + P(-R2) \leq 1$$

These relationships are quite easy to establish formally. Proofs of the first three, along with dozens of similar results can be found in Rudolf Carnap's work [1962].

To summarize the current position of this discussion, we need conditional probabilities to determine a measure of "relevance" and we might reasonably accept human estimates of these probabilities within the context of the constraints provided by a set of clauses. Further, we have considered several ways in which we might compute a measure of association based on those probabilities. We now wish to compute a number which reflects the relevance of one predicate to another in the sense that the first will contribute to a proof of the second.

We have a set of clauses (axioms) which we may refer to as our knowledge, K. To prove that some result 'R' is true, is to show that the entropy if R/K is zero. The entropy of 'R' measures our degree of uncertainty of 'R', or

the amount of information required to determine whether 'R' is true or false. It seems reasonable to say that those predicates are most relevant which, on the average, provide the most information about 'R'. It ahould be acknowledged that it does not logically follow that the most relevant group of predicates is composed of those predicates which are individually most relevant. A computational remedy of this difficulty requires obtaining the conditional entropy of 'R' relative to all pairs, triples, and so on, of the predicates which are present in 'K'. The result of this procedure is the kind of combinatorial explosion which we are here attempting to avoid. The alternative is to assume that by grouping the individually most relevant predicates we will obtain a reasonable approximation to the most relevant groups. Some support for this approach may be found in a report of the Advisory Group for Aerospace Research and Development [1971]:

> "Here, one assumes that the best n-variable predictors that are genuinely better than the best (n-1)-variable predictors are most likely to come from the best (n-1)-variable predictors genuinely better than the best (n-2)-variable predictors, and so forth. This assumption may not be true, but no better assumption exists which is still practical."

3.3 TWO EXAMPLES

Two examples illustrate the method which is being suggested here. For simplicity, this example will be in terms of proving theorems in the propositional calculus.

A series of tables are presented in the following pages which give:

1. A list of propositional variables with estimated probabilities and computed entropies.
2. A table of estimated conditional probabilities.
3. The resulting mutual entropies.
4. A list of axioms which serve as the context of the estimates and the basis for some sample proofs.

Two proofs are given for the first theorem, and three for the second. The proofs illustrate the effect of establishing different values for a "relevance level" parameter. A discussion of the methods involved follows the tables and proofs.

TABLE 3

PROPOSITIONS

S(it is snowing)	P(S) = .10, H(S) = .4590
J(wearing jacket)	P(J) = .50, H(J) = 1.000
C(comfortable)	P(C) = .90, H(C) = .4690
F(freezing)	P(F) = .40, H(F) = .9710
H(hot)	P(H) = .30, H(H) = .8813
B(bright sun)	P(B) = .20, H(B) = .7219
O(overcast)	P(O) = .60, H(O) = .9710
N(night)	P(N) = .50, H(N) = 1.000

TABLE 4

CONDITIONAL PROBABILITIES

	S	J	C	F	H	B	O	H
S	1.00	0.18	0.09	0.25	0.00	0.00	0.16	0.10
J	0.90	1.00	0.50	0.90	0.10	0.45	0.55	0.55
C	0.81	0.90	1.00	0.90	0.90	0.90	0.90	0.90
F	1.00	0.72	0.40	1.00	0.00	0.30	0.35	0.45
H	0.00	0.06	0.30	0.00	1.00	0.40	0.35	0.20
B	0.00	0.18	0.20	0.15	0.27	1.00	0.00	0.00
O	0.96	0.66	0.60	0.53	0.70	0.00	1.00	0.60
N	0.50	0.55	0.50	0.56	0.33	0.00	0.50	1.00

Table 5

TABLE OF MUTUAL ENTROPIES

	S	J	C	F	H	B	O	N
S	0	.4108*	.4360	.3245!	.4142*	.4349	.4129*	.4690
J	.9418	0	1.00	.6579!	.7801!	.9982	.9891	.9928
C	.4630	.3690	0	.4690	.4690	.4690	.4690	.4690
F	.8265*	.6289!	.9710	0	.6897!	.9632	.9597	.9634
H	.8265	.6614!	.8813	.6000!	0	.8730	.8681	.8464
B	.6878	.7201	.7219	.7141	.7136	0	.4!	.4855!
O	.9149	.9601	.9710	.9597	.9578	.3881!	0	.9710
N	1.000	.9928	1.000	.9924	.9651	.7636!	1.000	0

* Indicates that conditional entropy is 10% lower.
! Indicates that conditional entropy is 15% lower.

Table 6

TABLE OF AXIOMS AND THEIR CLAUSE FORMS

C1	S - F	-S V F
C2	F & J - C	-F V -J V C
C3	B - -O	-B V -O
C4	S - O	-S V O
C5	H - -F	-H V -F
C6	H & J - -C	-H V -J V -C
C7	B V O V N	B V O V N

Table 7

EXAMPLE 1 - TWO PROOFS

Theorem: Snow -> -Bright Sun (S -> -B)

Negation: S & B

Clauseform: T1: S

T2: B

PROOF 1 - Resolution with Set of Support

Resolvents - Level 1

R1: F (T1, C1)

R2: O (T1, C4)

R3: -O (T2, C3)

Resolvents - Level 2

R4: -J, C (R1, C2)

R5: -H (R1, C5)

R6: -B (R2, C3)

R7: -S (R3, C4)

B, N (R3, C7) (Subsumed by T2)

R8: Null (R3, R2)

PROOF 2 - Using Clauses Relevant at 10% Level (C3, C4)

Resolvents - Level 1

R1: O (T1, C4)

R2: -O (T2, C3)

Resolvents - Level 2

R3: -B (R1, C3)

R4: -S (R2, C4)

R5: Null (R2, R1)

Table 8

EXAMPLE 2 - THREE PROOFS

Theorem: Snow & Jacket -> Comfort

Clauseform: T1: S

T2: J

T3: -C

PROOF 1 - Resolution with Set of Support

Resolvents - Level 1

R1:	F	(C1, T1)
R2:	-F, C	(C2, T2)
R3:	-F, -J	(C2, T3)
R4:	O	(C4, T1)
	-H, -C	(C6, T2)

Resolvents - Level 2

R5:	-S, C	(C1, R2)
R6:	-S, -J	(C1, R3)
R7:	-J, C	(C2, R1)
R8:	-B	(C3, R4)
R9:	-H	(C5, R1)
	-H, -J, -F	(C6, R2)
R10:	-F	(T2, R3)
	-F	(T3, R2)
R11:	C	(R1, R2)
R12:	-J	(R1, R3)

Resolvents - Level 3

R13:	-S	(C1, R10)

(cont. following page)

Table 8 (cont.)

	-H, -J, -S	(C6, R5)
	-H, -J	(C6, R7)
	-H, -J	(C6, R11)
R14:	C	(T1, R5)
	-J	(T1, R6)
	-S	(T2, R6)
	C	(T2, R7)
R15:	Null	(T2, R12)

(24 Resolvents Generated)

PROOF 2 - Using Clauses Relevant at 10% Level (C1, C2, C5, C6)

Resolvents - Level 1

R1:	F	(C1, T1)
R2:	-F, C	(C2, T2)
R3:	-F, J	(C2, T3)
	-H, -C	(C6, T2)

Resolvents - Level 2

R4:	-S, C	(C1, R2)
R5:	-J, C	(C2, R1)
R6:	-H	(C6, T2)
	-H, -J, -F	(C6, R2)
R7:	-F	(T2, R3)
	-F	(T3, R2)
R8:	C	(R1, R2)
R9:	-J	(R1, R3)

(cont. following page)

Table 8 (cont.)

	-F, -H	(R2, R4)
Resolvents - Level 3		
R10:	-S	(C1, R7)
	-H, -J, -S	(C6, R4)
	-H, -J	(C6, R5)
	C	(T1, R4)
	C	(T2, R5)
R11:	Null	(T2, R9)
	(19 Resolvents Generated)	

PROOF 3 - Using Clauses Relevant at the 15% Level (C1, C2)

Resolvents - Level 1		
R1:	F	(C1, T1)
R2:	-F, C	(C2, T2)
R3:	-F, J	(C2, T3)
Resolvents - Level 2		
R4:	-S, C	(T1, R2)
R5:	-S, -J	(T1, R3)
R6:	-F	(T2, R3)
	-F	(T3, R2)
R7:	C	(R1, R2)
R8:	-J	(R1, R3)
Resolvents - Level 3		
	C	(T1, R4)
	-J	(T1, R5)

(cont. following page)

Table 8 (cont.)

R9: -S (T2, R5)

R10: Null (T2, R8)

(13 Resolvents Generated)

The preceding proofs represent a simulation of a procedure, which can be implemented on a computer, for resolution proofs using the set-of-support strategy. The procedure is the "level saturation method" [CHANG, 1973]. A new level is established by carrying out all possible resolutions of previous clauses with clauses which follow them in the current level. This procedure is repeated until the empty clause is generated.

The difference between multiple proofs of the same theorems above are the result of a screening of the axioms to be employed in the proof attempt. The screening consists of two steps. First, a list of predicates (propositions) which are relevant to N of more of the predicates in the theorem is constructed. For these proofs N = 2 is employed. In other words, we form a list of predicates relevant to at least two of the predicates in the theorem to be proved. We add to this list, the predicates which occur in the theorem itself. Then, we form a new list of the axioms which contains only those predicates on the first list.

We say that a predicate 'A' is relevant to a predicate 'B' if H(B/A) is a fixed percent (P) lower than H(B).

Characterized another way, 'A' must provide, on the average, a percent 'P' of the information required to determine 'B'. Thus, we have two parameters to set before a proof is attempted; they are the values of 'N' and 'P'. The proofs without relevance filtering may be viewed as P = 0%. A proof of each theorem is given with P = 10%, and a proof if the second is given w th P = 15%. (Note that there is no proof of the first with P = 15%.)

To illustrate, consider the second proof of the first theorem. The predicates in the theorem are 'S' and 'B'. A check of the table on page 33 shows that only clauses C3 and C4 are constructed using only these predicates. So, a proof is attempted and completed using axioms C3 and C4. This proof is notably shorter than the original proof with P = 0%. If we attempt the proof with P=15%, we find that there are no predicates which can be added to the two in the theorem, and there are no axioms containing only those predicates, and thus there is no proof.

The choice of the 10% and 15% levels here is ad hoc, but we can expect this to be close to the correct levels for other applications. There is an interesting analogy in a study of bacterial classification. Woese [1981] describes bacteria in terms of an RNA "dictionary" and forms an association coefficient based on the number of shared "words". On the basis of these coefficients, the bacteria are clustered into three groups. It is interesting to note that only in one case is there an association measure as high as 14% between groups, they are generally below 10%.

In only 4 cases is there an association measure as low as 13% within groups, all others are at least 15%. We are talking in a sense about "shared information" in both of these cases.

We may now make some observations regarding the effect of this relevance preselection method. First, since the clause selection does not interact with the proof attempt, it follows that neither the completeness nor the consistency of the theorem proving strategy will be affected. Further, since at P = 0% no clauses are eliminated, it follows that for some P level, the chosen clauses will be inconsistent if and only if the original set of clauses is inconsistent. It is however possible to choose a consistent subset of an inconsistent set for a proof attempt. This last may be an apparently undesirable aspect of our methodology. Yet, if we are to emulate some of the strength of human ability at proof, we may inherit some of its weakness. The method allows us to attempt to find a shorter proof without inhibitibg our ability to return to a larger set of clauses and employ any strategy desired.

In summary, we have demonstrated that if we can obtain a reasonable estimate of the appropriate conditional probabilities, we can effectively compute levels of information sharing between predicates. By restricting clause selection, using levels of information sharing as a criterion, we can select a good subset of the original clauses for proof attempts.

In chapter 4 we will be taking a more formal look at

the nature of the predicate interdependencies which are uncovered by the relevance measure. We will also consider the possible application of this technique to groups of predicates and clauses. We will demonstrate the nature of the interactions which our entropy computations are measuring. We will then be better able to see the strengths and limitations of the selection technique we have developed.

Chapter 4

RELEVANCE AND THE OBJECT PREDICATE TABLE

4.1 OBJECT RELATIONSHIPS

In this chapter, we are considering the question of mutual relevance of predicates in the light of Watanabe's Theorem of the Ugly Duckling" [Watanabe, 1969]. Briefly, this theorem states that (from a certain formal point of view) any two non-identical objects are equally similar or dissimilar as any other two. To establish these results formally, we must consider Watanabe's object-predicate table

To be as consistent as possible with Watanabe's notation, we will let $X = (x_1, x_2, \ldots x_m)$ be a set of objects and $Y = (y_1, y_2, \ldots y_n)$ be a collection of predicates. The proposition $y_j(x_i)$ which states that the object x_i affirms the predicate y_j is to be meaningful whether or not it is true. This requirement permits the application of all predicates to all objects so that we may have a complete table, and may be met by the modification of some predicates. For example, the predicate "is red" may be replaced by "has color, and if so is red" in the event that there is some element of X without color. The object-predicate table is a matrix of m rows and n columns whose element T(i,j) is

equal to 1 or 0 according to whether $y_j(x_i)$ is true or false, and $p(y_j)$ is the relative frequency of ones in column j. As a simple example, consider a table formed from two objects and two predicates.

	y_1	y_2
x_1	0	1
x_2	1	0

Observe that with two predicates there could be at most 4 distinct rows or "object types". If a table T does not contain two identical rows it is said to be irreducible with respect to X. Our example table contains two of the four possible object types and is irreducible.

We extend the table T by substituting Y* for Y. Y* is obtained by adding to Y all possible predicates which are logical combinations of the original y's. That is, we form new predicates from the old by the use of conjunction, disjunction and negation until no new predicates can be formed. Watanabe calls Y* the Boolean completion of Y. We might now consider the extension of our example table.

	y_1	y_2	y_3	y_4
x_1	0	1	0	1
x_2	1	0	0	1

Note that $y_3 = y_1 \,\&\, y_2$ while $y_4 = y_1 \vee y_2$, and no new predicates may be formed.

The theorem of the ugly duckling states that:

> "The number of those predicates y_j in a completed Boolean lattice Y* of predicates satisfied simultaneously by two non-identical objects x_i and x_k (of the list of objects X) is a fixed constant independent of the choice of the two objects." (The proof of this theorem is provided in APPENDIX A.)

Watanabe found a similar result concerning the number of predicates affirmed by one but not the other of the two objects, and the number of predicates simultaneously denied by both objects. Thus, we could come to the conclusion that "an ugly duckling and a swan are just as similar to each other as are two swans". (In this instance, similarity is based simply on the number of properties shared and not shared in the context of a scope of observation established by a set of predicates Y.)

Watanabe claims that a formal (syntactical) discussion of similarity must be based upon Y* and not upon a subset Y which might generate Y*. He argues that among the many subsets of predicates which might give rise to the completed Boolean lattice Y*, there is no <u>logical</u> ground to prefer one over another. We might ask, for example, on what logical basis should we prefer "red" to "neither yellow nor blue"?

It is the case that the theorem is no longer valid when we consider particular subsets Y of Y*. Thus, it follows that when we form groups of "similar" objects, we are actually applying some extra-logical weighting function to the various predicates which are shared by the objects.

Concerning the extension of Y to Y*, Watanabe observes that:

> "...everything we have done in the foregoing paragraphs can be done when we interchange the roles of X and Y, because the basic assumption of the entire discussion is that we are given a rectangular matrix with entries that are either 0 or 1. ...By the same method used in extending Y to Y*, we can extend X to X*. This amounts to adding all object types which were missing from the original table."

He goes on to suggest that in some cases we may

> "...have to exercise a great deal of imagination and mental flexibility to understand these 'fictitious' objects. Nonetheless, a formal discussion involving X* proves to be useful and productive."

4.2 PREDICATE RELATIONSHIPS

The formal interchangeability of X and Y is referred to as object-predicate reciprocity. Applying this, the theorem of the ugly duckling may be dualized to read:

> The number of those objects x_j in a completed Boolean lattice of objects X* simultaneously affirming two non-identical predicates y_i and y_k (of the list of predicates Y) is a fixed constant independent of the choice of the two predicates.

Thus, in this formal sense, it is not possible to group predicates on the basis of the number of object types of which they are simultaneously true. It follows then that when we form groups of "similar" predicates we are applying some extra-logical weighting function to the various object types which are shared by these predicates. In the paragraphs which follow, we will be dualizing Watanabe's

analysis of relationships between objects. Our purpose is to use his methods to gain insight into interactions of predicates.

Let Y' be a subset of Y consisting of r predicates ($y_{1'}$, $y_{2'}$,...$y_{r'}$). For each object, we have a set of values for these predicates which can be denoted by ($a_{1'}$, $a_{2'}$, ...$a_{r'}$), where $a_{j'}$ will be 0 or 1 depending on the corresponding table entry in the column under $y_{j'}$. The relative frequency with which each such combination occurs in the object-predicate table may be given by:

$$p(a_{1'}, a_{2'}, \ldots a_{r'}) = \sum_{i} \frac{1}{M} \prod_{j} d(a_{j'}, T(x_i, y_{j'})),$$

where d(a,b) is 1 if a = b, and 0 otherwise. With the help of this formula, we can define the entropy of our subset Y'.

$$S(Y') = - \sum_{a_{i'}} p(a_{1'}, a_{2'}, \ldots a_{r'}) \log[p(a_{1'}, a_{2'}, \ldots a_{r'})]$$

Finally, using the expression for entropy, we can define what Watanabe calls the "interdependence" of a set.

$$J(Y'; y_{1'}, y_{2'}, \ldots y_{r'}) = \sum_{j'} S(y_{j'}) - S(Y')$$

The notation indicates that we have partitioned Y' into individual predicates. Other partitions into subsets may be indicated after the semicolon. For example, to compute the interdependence between $y_{1'}$ and the rest of the set, we would use the following:

$$J(Y'; j_{1'}, (y_{2'}, y_{3'}, \ldots y_{r'})) = S(y_{1'}) + S(Y'-y_{1'}) - S(Y').$$

Observe that Watanabe's "interdependence", j, is the extension to larger sets of predicates of what we have called "decrease in entropy", or the "mutual information" of two predicates. Consider two examples intended to demonstrate the application of this concept.

The following table is isomorphic to Carnap's chess tournament example [CARNAP, 1962]. He used the example to demonstrate a pathological situation in measuring relevance. In his terminology, the predicate y_2 is "positively relevant" to the predicate y_1 in the sense that $p(y_1/y_2) > p(y_1)$. Similarly, y_3 is "positively relevant" to y_1. Yet the conjunction y_2 & y_3 is "negatively relevant" in the same sense that $p(y_1/y_2\&y_3) < p(y_1)$.

	y_1	y_2	y_3
x_1	1	1	1
x_2	1	1	0
x_3	1	1	0
x_4	1	0	1
x_5	1	0	1
x_6	0	1	1
x_7	0	1	1
x_8	0	0	0
x_9	0	0	0
x_{10}	0	0	0

Analysis of the interdependence of the predicates requires the following conditions:

$$S(y_1) = S(y_2) = S(y_3) = 1 \text{ bit},$$

$$S(y_1, y_2) = S(y_1, y_3) = S(y_2, y_3) = 1.9710 \text{ bits},$$

$$S(y_1, y_2, y_3) = 2.2464 \text{ bits}.$$

Hence, the pairwise interdependencies are very small:

$$J(y_1, y_2) = 2 - 1.9710.$$

Whereas, there is substantial interdependence in the group:

$$J(y_1, y_2, y_3) = 3 - 2.2464.$$

If we were trying to determine the truth of y_1, neither y_2 nor y_3 individually would be very helpful on the average. In combination however, they greatly reduce the uncertainty of y_1. That these complex interactions may be uncovered by the computation of entropies indicates some of the power of this method of analysis. Observe the conditional uncertainties (entropies):

$$S(y_1/y_2) = S(y_1, y_2) - S(y_2) = .97 \text{ bits},$$

$$S(y_1/y_2,y_3) = S(y_1, y_2, y_3) - S(y_2, y_3) = .28 \text{ bits}.$$

These are respectively a 3% and a 72% reduction in the uncertainty of y_1.

The second example, we will consider, is the dual of an example discussed quite extensively by Watanabe [1969].

	y_1	y_2	y_3	y_4
x_1	1	1	0	1
x_2	1	1	0	0
x_3	0	1	1	1
x_4	0	1	1	0
x_5	1	0	1	1
x_6	1	0	1	0
x_7	0	0	0	1
x_8	0	0	0	0

For this table, the entropy of each single predicate is 1 bit, the entropy of each pair of predicates is 2 bits, and the entropy of all but one of the triples is 3 bits. That special triple, (y_1, y_2, y_3) has an entropy of 2 bits. This means that no single predicate would be of use in determining the value of any other single predicate, and no pair containing y_4 would be useful in determining the value of any other predicate. However, a pair out of the special triple will always determine the value of the third member of the triple. The following are representative computations:

$$S(y_1/y_2) = S(y_1, y_2) - S(y_2) = 2 - 1,$$

$$S(y_1/y_2,y_3) = S(y_1, y_2, y_3) - S(y_2, y_3) = 2 - 2,$$

$$S(y_1/y_3,y_4) = S(y_1, y_3, y_4) - S(y_3, y_4) = 3 - 2.$$

Once again, we see entropic analysis revealing a complex structure through a straightforward computation. Watanabe observes:

> "It is interesting that none of the other mathematical methods so far proposed seems to be able to analyze a table as simple as Table 8.1 (this example)." [1969].

At this point, consider some of the implications of the preceding paragraphs. As a consequence of the Theorem of the Ugly Duckling, we know that the restricted object-predicate table contains non-logical information which is absent in the completed Boolean lattice. This information is inherent in the non-uniform probability distributions over the rows of the O-P table. And, this information is captured, at least in part, by interdependence analysis of the predicates.

The preceding examples were specifically intended to demonstrate the effectiveness of our methods for uncovering structure within the table. But, recall that we would not, in general, have such a table to analyze when we wished to prove a theorem. We argued in Chapter 3 that a reasonable approximation might be obtained for the probabilities of individual predicate symbols. It does not seem reasonable to argue that we could equally well obtain probability estimates for groups of predicates without an actual table. We might also wonder, even if such estimates were available, whether the search for structure in the table would correspond with a decrease in the size of a proof search.

Having observed that we can uncover the structure in

a group of predicates using this method of analysis, we might ask whether we could uncover the relationships in a set of clauses or other expressions built from predicates. In answer, consider this example.

	y_1	y_2	y_3
x_1	1	0	1
x_2	1	0	0
x_3	0	1	1
x_4	0	1	0

Observe that we have a set of three predicates, each having an entropy of 1 bit. The entropy of the clause $(y_1 \vee y_2)$ is 0. The entropy of the clause $(y_1 \vee y_3)$ is .81, showing that the entropy of the clauses cannot be computed from the individual entropies of the predicates. Computing the entrpoy of a clause requires a probability distribution on the clause itself. This is essentially the creation of a new predicate, as in our previous extension to Y*. Once again, it does not seem reasonable to argue that we could obtain good probability distributions for these new predicates without the O-P table.

We may now conclude two things. First, there is a structure underlying predicate interdependencies which is effectively uncovered by entropic analysis. Second, extension of the selection strategy developed in Chapter 3 requires that we have the equivalent of an object-predicate table available.

Chapter 5

RELEVANCE AND COMBINED EVIDENCE

In this chapter, we consider relevance measures in a somewhat different setting. Some deductive systems arrive at conclusions (recommendations) on the basis of probabilistic reasoning rather than implication. A good example is an expert system such as MYCIN [SZOLOVITS, 1978] which will diagnose bacterial infections and recommend treatment. This system is considered in greater detail in later paragraphs.

Expert systems introduce some new difficulties. A theorem proving strategy is helpful if it reduces the average effort required to find a proof. If there are exceptional cases for which the proof is lengthened by employing a particular heuristic, this will not destroy its overall value. When a proof is found, it is valid. On the other hand, if we employ a deduction strategy which is occasionally erroneous in an expert system, the overall value of the system may be diminished.

Some of the difficulties involved here have been mentioned previously. In Chapter 3 (Section 3.2, p. 29), we considered the possibility that the individually most relevant predicates may not be those which form the most relevant groups for the purpose of proving a theorem.

This same difficulty may be applied to the factors in a decision problem, Further, in Chapter 4 (Section 4.2, P. 48) we constructed a case in which each of two propositions were positively relevant to a third, yet their conjunction was negatively relevant. (Note, we can construct from this an example in which each of two propositions is positively relevant to a third while their disjunction is negatively relevant.) These cases pose difficulties when we attempt to logically combine observations which are individually supportive of an hypothesis.

It is appropriate here to consider why and how frequently these pathological cases may occur. Carnap [1962] details "possible relevance situations" in a general way. Interpreting his discussion in terms of probabilities, we have A as positively relevant to B if $p(B/A) > p(B)$ and A as negatively relevant to B if $p(B/A) < p(B)$. Carnap's relevance measure [$r(A,B) = p(A\&B) - p(A)p(B)$] is positive in the first case, and negative in the second. Using this measure. Carnap explores whether and how the relevance measures r(i,h) and r(j,h) determine the relevance measures of their combinations to h.

Carnap's discussion is based on the possible sign combinations (+,-,0) which various relevance measures can have in this situation. He considers seven possible combinations of i and j, namely: i&j, i&-j, -i&j, -i&-j, i, j, iVj. He finds that there are 75 possible sign combinations over the possible measures of relevance that each of these sentences may have to a sentence h. A list of these cases

is included as an appendix (Appendix B).

Most of the possible cases are intuitively reasonable. For example, if i is positively relevant to h and j is negatively relevant to h, then their conjunction or alternation could have +, -, or 0 relevance. Four of the 75 cases are somewhat counter to intuition. In one case, both i and j are positively relevant to h while their conjunction is negatively relevant. Also, cases like the last two occur for disjunction. If might be noted that if i and j are positively relevant to h, then at least one of their conjunction and disjunction must be positively relevant. We may wonder whether 4 out of 75 cases gives an indication of the frequency of occurence of pathological cases in actual practice.

Carnap gives a recipe for constructing each of the 4 cases [CARNAP, 1962]. The first one is described here, in detail. For a given h (assuming background information e which determines the probabilities) we take any three sentences (1), (2), and (3) satisfying the following conditions: The sentences are pairwise exclusive with respect to e; (1) is negatively relevant to h, its r-value being -r; both (2) and (3) are positive to h with r-value greater than r. If we take i as the disjunction of (1) and (2), and j as that of (1) and (3), then i and j both have positive r values found by adding those of the disjuncts. The conjunction of i and j is equivalent to (1), and hence has negative relevance to h.

It seems worthwhile, at this point, to include a des-

cription of Carnap's example, in order to provide increased insight into the foregoing discussion. In the example, we have ten chess players who participate in a chess tournament in New York City [CARNAP, 1962]. The group contains local and out of town players, junior and senior players, men and women, distributed as in the following arrangement.

	i(local)	-1(stranger)
j(junior)	M,W,W	M,M
-j(senior)	M,M	W,W,W

It is known that exactly one of these ten will win, and our evidence e indicates that each has an equal probability (.1) of winning. Further, we assume that additional evidence that some player or group cannot win will leave the remaining players with equal chances to win.

Let h represent the sentence 'a man wins'. Based on our evidence e, we have p(h) = 1/2. Now, suppose we receive a report that 'a local player wins', which may be based upon information that the strangers have been eliminated. This new evidence increases the probability of h to 3/5, and thus is positively relevant. If we had instead received a report that 'a junior player wins', we would likewise increase the probability ot h to 3/5. On the other hand, if we receive both reports, that is 'a local junior wins', we would diminish the probability of h to 1/3.

If we consider another hypothesis k, that is 'a woman wins', we can observe reversed relevance situations. An initial probability of 1/2 will be diminished by a report

of either i or j separately. A report of i&j will, however, be positively relevant.

If we are going to have a computer program emulate the behavior of a human expert, it seems reasonable to think that pathological cases of evidence combination must be identified. An individual program might successfully ignore or avoid these difficulties, but a general theory of expert systems must account for them.

Consider the Zadeh fuzzy-set rules. Namely, $p(A \vee B) = \max[p(A), p(B)]$, $p(A\&B = \min[p(A), p(B)]$. As rules for combining evidence, they may overconfirm or underconfirm the hypothesis. We will agree with Hart [1975] that when we are dealing with interdependent evidence, "the exact nature of these dependencies will rarely if ever be known." In general, however, following the fuzzy set rules may lead, as seen above, to accepting as positive some evidence which is actually negative and as negative some evidence which is actually positive. Further, in the light of the 75 cases, no single combining rule will avoid this possibility. We recommend that in the construction of an expert system, we call upon our human experts to identify, insofar as possible, those cases in which evidence applicable to an hypothesis will combine in unusual ways. We recommend further that the system include a mechanism for handling these cases when they occur.

If two rules which separately support a conclusion will tend to disconfirm it in combination, then this information can be built into those rules.

IF: 1) IT IS KNOWN THAT A LOCAL PLAYER WINS, AND
2) IT IS NOT KNOWN WHETHER A JUNIOR OR SENIOR WINS,

THEN: THERE IS EVIDENCE (.6) THAT A MAN WINS.

If there is a similar rule for juniors, neither rule will be activated inappropriately.

Note that this discussion is not intended as criticism of any existing programs. Rather, it is intended as a general analysis which may be useful in the construction of some future systems.

In the light of the foregoing discussion, let us briefly consider the MYCIN program. The purpose of this is to demonstrate a specific context for the type of reasoning mechanisms discussed in the preceeding paragraphs. The MYCIN program is an interesting and successful example of an expert system. It does, in part, employ the fuzzy set rules mentioned above.

The MYCIN program is designed to guide physicians in the appropriate treatment of bacterial infections. The primary knowledge base of the program is a set of independently stated rules of deduction. An example from the Szolovits article follows.

IF: 1) THE STAIN OF THE ORGANISM IS GRAM POSITIVE, AND,
2) THE MORPHOLOGY OF THE ORGANISM IS COCCUS, AND,
3) THE GROWTH CONFORMATION OF THE ORGANISM IS CHAINS,

THEN: THERE IS SUGGESTIVE EVIDENCE (.7) THAT THE IDENTITY OF THE ORGANISM IS STREPTOCOCCUS.

Computation of certainties in this program occurs at two levels. First, the program user must provide a degree of certainty for the antecedent individual conditions of

a rule. The certainties of the individual antecedents are combined using fuzzy-set rules to obtain an overall certainty of the antecedent. This measure of belief in the antecedent is multiplied by a certainty factor (.7 in the example) that the antecedent in the rule actually does imply the consequent. The resulting product is the measure of belief in the conclusion which is contributed by that rule. Second, if more than one rule contributes to the program's certainty of a fact, the measure of belief from the various rules are combined to yield the overall measure of belief in the given fact. If one rule gives us fact h with certainty CF1, and later another rule gives us fact h with certainty CF2, then the overall confidence CF = CF1 + CF2 - CF1xCF2.

In another system patterned after this model, it seems unlikely that there would be conflicting combinations within the individual rules. Caution would be indicated however, when it came to combining evidence contributed by different rules. Our human expert should look very carefully for pathological connections among different facts in the data base even where it is unreasonable to consider all interactions. If we can accomplish this, at least we will have all of our adjustments of confirmation going in the correct direction.

In the light of our earlier analysis of the object-predicate table, we may conclude with the following observations:

1) A combination of two facts is a new fact whose relevance to a conclusion is not a simple function

of the separate relevances.

2) It is unreasonable to consider all possible combinations of facts when providing guidance to a reasoning system.

3) It is worthwhile to ask our expert to consider how the most relevant (positive or negative) facts will affect a conclusion when acting in combination.

Chapter 6

THE USE OF ENTROPY IN UNCOVERING RELATIONAL STRUCTURE

Assume that we are given a set of objects and the values of various attributes which the objects possess. If we are to impose structure on this information in the form of a data model, we require information about the underlying relationships in the data. Even if we are given a data model, we may wish to study current relationships within the data with an eye toward validating or modifying the model. Entropy may be considered a valuable tool in the analysis of these relationships.

Let a relation be defined in an m x n table, with each row representing one of m objects (entities) and each column representing one of n attributes. Note that object-predicate tables previously discussed represent examples of such relations if we view the predicates as our attributes. What we are discussing here, is the relational data base type relation which is more general in that attribute values need not be binary. In this discussion, we are looking at the use of entropic relevance measures for identifying structure, and, in particular, for finding keys in this more general relation. An example of such a relation,

adopted from Tsichritzis [1977], follows.

HOMES1 (BUILDER, STYLE, PRICE)

Cadillac	Duplex	65000
Delzoto	Duplex	65000
Howlett	Bungalow	45000
Joint	Ranch	50000
Monza	Duplex	65000
Terex	Ranch	50000
Wimpey	Ranch	50000

The above example will help to illustrate the following definitions.

A subset P_1 of attributes A_{i_1}, A_{i_2}, A_{i_3}, ...A_{i_r} is a KEY if the value of every attribute in the set of attributes is functionally dependent upon P_i, and if P_i is minimal in the sense that none of its proper subsets has this property. In our example, builder is a key.

A subset of D_i of attributes is a DETERMINANT if the value of at least one attribute A_k, not belonging to D_i, is functionally dependent upon the values of D_i. In our example, style is a determinant of price.

A subset of attributes S_i is a SUBDETERMINANT of A_k if knowing values for the members of S_i will sometimes,

but not always, determines the value of A_k. Style is a subdeterminant of builder, as the only bungalow is built by Howlett.

An individual attribute A_k is a NUCLEAR attribute if it participates in every key. All other attributes are nonnuclear.

An individual attribute A_k is a PRIME attribute if it participates in at least one key. All other attributes are nonprime.

Note that not all keys are determinants. The set of nuclear attributes however, is a subset of every key and a subset of the set of prime attributes. Following the pattern used to compute interdependence of predicates in the object-predicate table, we may define the interdependence of the attributes in our relation.

Given a subset P_i of attributes, let $X_i = (a_{i_1}, a_{i_2}, \ldots a_{i_r})$ represent an assignment of values to the attributes in the order in which they occur in the table. Then, the probability $p(X_i)$ may be computed as the relative frequency of occurrence of this assignment of values in the appropriate columns of our table. This corresponds exactly to the definition given in Chapter 4 (Section 4.2, p. 46) for the table of binary values.

Given the probability defined above, we may compute $S(P_i)$, the entropy of the subset of attributes. The inter-

dependence of a subset of attributes $J(P_i)$ is then defined as the difference between the sum of the individual entropies and the entropy of a subset. This repeats the pattern of Watanabe's [1969] object interdependence computations.

$$S(P_i) = - \sum_{A_{i_j}} p(A_{i_1}, A_{i_2}, \ldots A_{i_r}) \log[p(A_{i_1}, A_{i_2}, \ldots A_{i_r})]$$

$$J(P_i) = \sum_j S(A_{i_j}) - S(P_i)$$

Consider an example which is equivalent to that found in Chapter 4 (Section 4.2, p. 49).

	a_1	a_2	a_3	a_4
x_1	1	1	0	1
x_2	1	1	0	0
x_3	0	1	1	1
x_4	0	1	1	0
x_5	1	0	1	1
x_6	1	0	1	0
x_7	0	0	0	1
x_8	0	0	0	0

As discussed earlier, each attribute has entropy equal 1 and each pair of attributes has entropy equal 2, while all other triples have entropy equal 3. The entropy of the entire relation is also 3. The interdependence of the first triple is 1+1+1-2 = 1 bit, the same as the interdependence in the relation. The interdependence of any other subset is 0. A nonzero value for j tells us that some of

the attributes included in a set can give us information about other attributes in that set. Consider the manner in which this helps us to find the keys.

Suppose that P_i is a determinant for A_k. Then we have $S(A_k/P_i)$ equal to 0, since total information destroys all uncertainty. Now since,

$$S(A_k/P_i) = S(A_k, P_i) - S(P_i)$$

we have

$$S(A_k, P_i) = S(P_i).$$

That is, adding A_k to the set of attributes P_i does not change the entropy of the set P_i. It also follows that the interdependence of the set is increased by the entropy of A_k when it is added to the set. This reflects the dependence of A_k on the rest of the set.

This situation is evident in our attribute triple (a_1, a_2, a_3) from our example. The entropy of a_1, a_2 is a determinant of a_3, adding a_4 to any pair of attributes will increase the entropy of the set. Thus, a_4 has no determinant. Thus, we may conclude that the set a_1, a_2, a_4 is a key to the relation, and that a_4 is a nuclear attribute.

The previous example provides a number of insights into the problem we are discussing. Since a nuclear attribute has no determinant, the deletion of a nuclear attribute from a set must decrease the entropy of the set. Thus, the nuclear attributes of a relation may be determined by singly deleting each attribute from the relation and computing the entropy of the resulting set. In the above example,

the deletion of a_4 left a triple of entropy 2, whereas the deletion of any other attribute left a triple of entropy 3. Thus, in one pass through the entropy list, we have determined that a_4 is the only nuclear attribute.

Once we have determined the set of nuclear attributes, we must consider the various ways in which it can be expanded into a key. Once again, this can be determined from the entropies of attribute sets. The essential fact is that the entropy of a key must equal the entropy of the relation. In the case that the entropy of the set of nuclear attributes does equal the entropy of the relation, it is the unique key. In other cases, additional attributes must participate in any key.

In the case of our example, we have analyzed the entropies of all of the subsets of attributes. We could see at the beginning that there were exactly three minimal subsets of entropy 3. These are all of the keys to the relation.

Consider the manner in which this approach might be used to uncover the structure of two examples given in Tsichritzis [1977]. The relations are based on information about a group of houses, and are given with keys underlined. First consider the previous example.

HOMES1 (BUILDER, STYLE, PRICE)

An analysis of the table (Chapter 6, p. 61) corresponding to this data model shows that there is one nuclear attribute. It also shows that the entropy of this nuclear attribute equals that of the relation, establishing it as a uni-

que key. Of course, we may also observe that the entropy of the remaining pair of attributes is less than that of the relation.

In the following example from Tsichritzis [1977],

HOUSES (ID, ADDRESS, LOT#, SUBDIVISION, STYLE, BUILDER)

an analysis of the corresponding table reveals that there are no nuclear attributes. An examination of the entropy of single attributes shows that two of these have entropy equaling that of the relation, thus each of these single attributes is a key. An analysis of the entropy of pairs of the remaining attributes shows that one of the eight possible pairs has entropy equal to that of the relation and, thus, is a key. At this point, only a pair of attributes would remain. These last two are nonprime attributes.

Consider one more example, which is equivalent to that in Chapter 4 (Section 4.2, p. 49).

	a_1	a_2	a_3
x_1	1	1	1
x_2	1	1	0
x_3	1	1	0
x_4	1	0	1
x_5	1	0	1
x_6	0	1	1
x_7	0	1	1
x_8	0	0	0
x_9	0	0	0
x_{10}	0	0	0

For this table, the entropy of each attribute is 1, the entropy of each pair is just less than 2, and the entropy of the relation is 2.2464 bits. Since the entropy of the relation is diminished by the deletion of any single attribute, we may conclude that every attribute is a nuclear attribute. The only key is the full attribute set.

To summarize, keyfinding using entropy involves two phases. In the first phase, the list of nuclear attributes is obtained. The second phase consists of searching for subsets of the remaining attributes which can be combined with the nuclear attributes to give us a set whose entropy equals that of the relation. Further, a guide to ordering the phase two search is that you consider first subsets of those attributes with the individually highest entropies. Consider an algorithm for the search based upon Nilsson's A* algorithm [NILSSON, 1980].

In the following, let K represent the nuclear attribute set of a relation R. Let A = R - K represent the set of remaining attributes. The Keysearch Algorithm may be described informally as follows:

1. Create a search graph G, consisting solely of the start node, K. Put K on a list called OPEN.
2. Create a list called CLOSED and a list called KEYS, both initially empty.
3. LOOP: if OPEN is empty, exit and return to KEYS.
4. Select the first node on OPEN, remove it from OPEN, and put it in CLOSED. Call this node n.
5. If n is a goal node, (I.E. if S(n) = S(R) and n

is not a superset of a set already on KEYS) add it to the list of KEYS.

6. If n is a superset of a key, go to LOOP.

7. Expand node n, generating the set, M, of its successors. Install the members of M as successors of n in G. Add the members of M to OPEN, if they have entropy greater than the entropy of n.

8. Reorder the list OPEN using evaluation function f (described below).

9. Go LOOP.

Now, we are searching for all of the minimal subsets of R whose entropy is the same as R. Each node in the search space represents a subset of R. When we expand a node, the set M represents each subset of R which can be obtained by adding one of the remaining attributes (of higher index) to those in the set represented at node n. The cost of reaching a node is the length of the path to the node which also equals the number of attributes added. One consequence of this is that the cost of reaching a node is independent of the path taken to it. Further, the use of indexing will prevent multiple generation of identical nodes.

To estimate the cost of reaching a goal node from a given node n, we consider the entropies of the attributes which remain to be added. The entropy of the union of two sets is less than or equal to the sum of their entropies. Consequently, when we add a new element to the set at n, the resulting increase in entropy of the set is at most

that of the attribute added. So, if we consider the difference between current entropy and goal entropy divided by the maximum of the remaining attribute entropies, we have an underestimate of the number of steps required to reach the goal. These considerations lead to the following evaluation function:

$$f(n) = g(n) + h(n)$$

$$g(n) = \text{cost of reaching node } n$$

$$h(n) = 1 + \text{FIX}[(S(R) - S(N))/S(a)].$$

S(a) represents the maximum of the remaining attribute entropies.

Since we know the exact value of g(n), and since h(n) underestimates the cost of reaching a goal, we are guaranteed to find a minimal cost path to a goal node. A minimal cost path is equivalent to a minimal size set and hence to a key [NILSSON, 1980]. Any later generation of superkeys is blocked by checking candidates against the KEYS list. The evaluation function guarantees that keys will be generated before their superkeys. The algorithm is guaranteed to terminate successfully, since the full relation R will be returned as a key if there is no proper subset of R which works. The algorithm will find all keys, since a node goes on CLOSED only if it is expanded or is a successor node to a key.

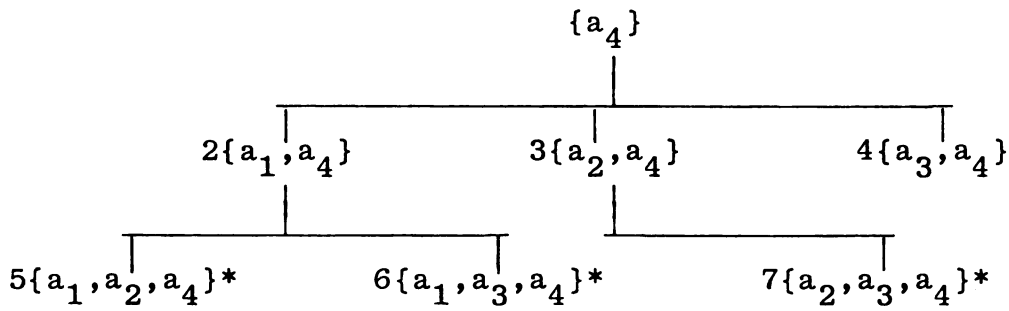
A version of the A* algorithm which would expand fewer nodes than this one would require a "more informed" evalua-

tion function than the one given. One such function is obtained if, instead of using the single maximum remaining entropy, we sum the remaining entropies in decreasing order until the needed difference is exceeded. The number of entropies added will give us an estimate at least as large as our h(n). It is doubtful that the additional computation required would be profitable.

Two examples are presented here in order to demonstrate the order in which the algorithm would generate the keys in our previous examples. First, consider the example presented on page 63. $K = \{a_4\}$.

Figure 1

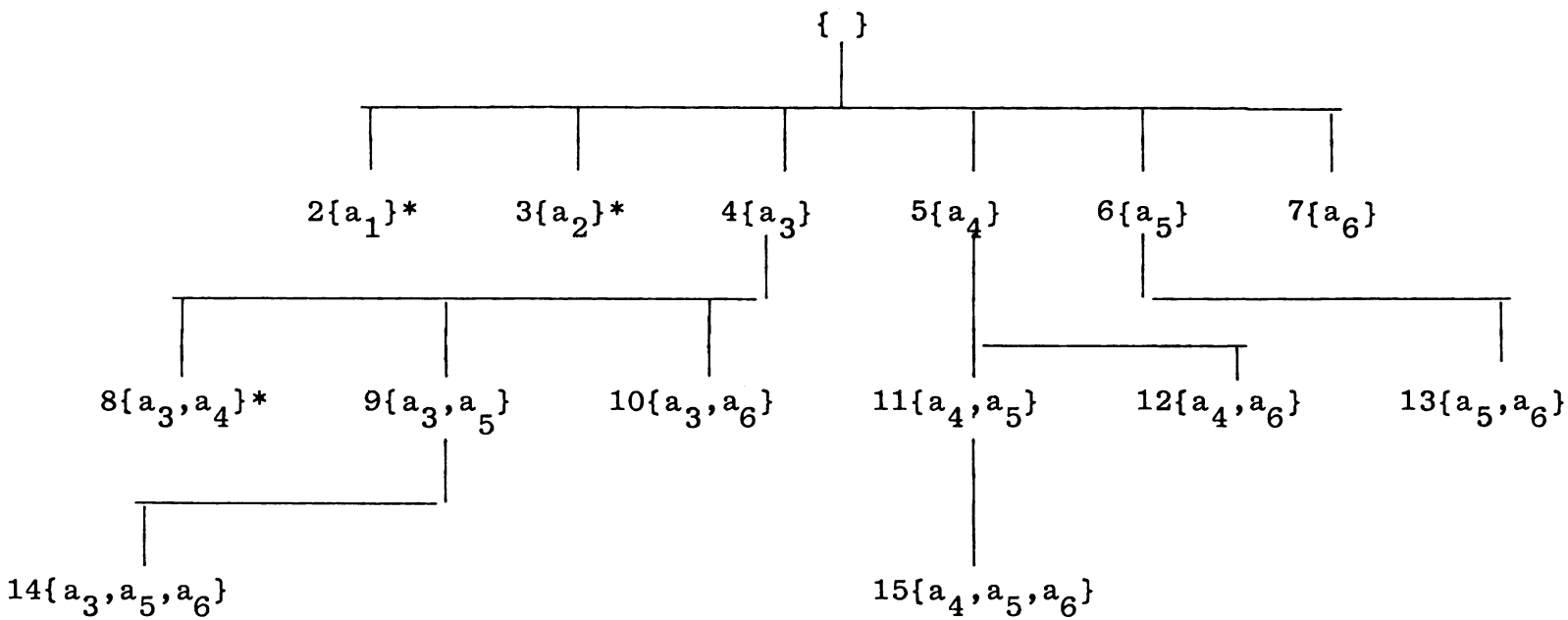
SAMPLE SEARCH TREE



In Figure 1, a total of seven nodes are generated to locate the three keys. Figure 2 (page 71) shows the search tree resulting from the application of the tree search algorithm to the six attributes of the HOUSES relation presented on page 66. Note that the nuclear set is empty.

Figure 2

SEARCH TREE FOR HOUSES RELATION



It is worthy of note that the analysis of a data base instance may be done in two stages. A complete analysis may be done on a sample of tuples. Any key of the entire set will be a key for the sample. The second stage will consist of verifying that keys found work in the whole data base. This approach reduces the size of the required entropy computations.

6.1 EVALUATION OF DATA BASE USAGE

Suppose that we are to analyze a relation which has been available over a period of time to a group of data base users. Further suppose that, in addition to the table defining the relation, we are given a record of the number of times each row of the table has been accessed. For example, consider the following:

	a_1	a_2	a_3	count
x_1	1	1	1	(1)
x_2	1	1	0	(2)
x_3	1	0	1	(2)
x_4	0	1	1	(2)
x_5	0	0	0	(3)

As before, let X_i represent an assignment of values to P_i, a subset of the attributes. We may now compute a new probability, $p^*(X_i)$, which represents the relative frequency of access of rows containing this particular

assignment of values in the appropriate columns of our table. We may also compute $S^*(P_i)$ and $J^*(P_i)$ by replacing p with p* in the previous definition. We might now say that we are working with a VIRTUAL data base which could be expanded into an actual data base. Observe that the expanded form of the current example is the previous example.

In the event that every row has been accessed at least once, analysis based on p* will reveal exactly the same set of keys as analysis based on p. This is a consequence of the fact that functional dependencies are not changed by the repetition of a tuple. In the event that some rows have never been accessed, they will not participate in the virtual data base. As a result, we may have changes in the functional dependencies over time. An analysis of the virtual data base reflects the actual usage which the corresponding data base has had and thus could serve as a guide to useful modifications of the data model.

An analysis of the virtual data base may provide other kinds of assistance to the data base manager as well. Given the keys to a relation, we would like to use them as efficiently as possible. This means that we would like to have our search lead us as directly as possible to a record containing the information sought. To accomplish this, we would direct the search, using first those attributes from the key which most quickly narrow the search. These are those with the highest entropy. In particular, if efficiency is to conform with actual usage, we wish to employ first those attributes with the highest entropy in the VIRTUAL

data base, if indeed these are different from the highest entropy attributes of the data base under study.

Consider again the example on page 72. Our analysis need not stop at the discovery that the only key is the full attribute set. We can observe that the entropy of the relation is much less than the sum of the entropies of the attributes comprising the key. This suggests that there is interdependence among these attributes which is not reflected in the functional dependencies. For example:

$$S(a_1/a_2,a_3) = S(a_1,a_2,a_3) - S(a_2,a_3) = .28 \text{ bits.}$$

This represents an average 72% reduction in the uncertainty of a_1 when we know a_2 and a_3. Similar reductions in the uncertainties of a_2 or a_3 occur when we are given the other two. Thus, for example, $D_i = a_2, a_3$ might be considered a "significant" subdeterminant of a_1. We could be guided by this to modify the data model so as to take advantage of this structure.

Consider a hypothetical relation, in the spirit of the Tsichritzis example, defined by:

HOUSES'(OWNER, ADDRESS, LOT#, SUB#, STYLE, BUILDER)

We might find in analyzing the data base that the OWNER attribute is able, on the average, to provide a 99% reduction in the uncertainty of the other attributes. We could then restructure our data so that HOUSES' admitted only

owners of one house, causing owner to become a key. A second relation HOUSES" could be defined admitting owners of more than one house.

Security information may also be gained from the entropic analysis of a data base. If the base contains some attribute information which is meant to be guarded from determination, the extent to which other attributes are "significant" subdeterminants may be computed from the mutual entropies.

If the efficiency of access and security of information are primary concerns of a data base manager, then the analysis tools developed in this chapter should be of assistance to him. A preliminary analysis of an instance of a data base may be done with a random sample of tuples to keep down computational cost. From this a complete picture of attribute interdependencies may be obtained.

Chapter 7

SUMMARY AND RECOMMENDATIONS

7.1 SUMMARY

The need for further employment of semantic information in deductive systems has led us to explore the use of relevance measures for clause selection. Preliminary exploration lead us to concentrate on the entropic relevance measures and their meaning. A strategy for the employment of these measures in choosing clauses for a deduction was developed and the object-predicate table underlying these measures was explored. The analogy between the object-predicate table and a relational data base was considered and applications to the study of data base structure were developed.

In Chapter 1, the basic concepts of resolution type theorem proving were introduced, and the manner in which it might be employed in question answering was considered.

In Chapter 2 we surveyed various measures of the relationship between two predicates and concluded that entropic measures are the most promising.

In Chapter 3 we developed a strategy for employing the relevance measures in the clause selection process and

illustrated this concept with some examples.

In Chapter 4 the object-predicate table underlying the relationships of the predicates was examined. We also considered the significance of Watanabe's theorem of the Ugly Duckling to this problem.

In Chapter 5 we considered the difficulties of combining relevant evidence, particularly in the light of Carnap's work on relevance.

Finally, in Chapter 6, we considered the application of entropic relevance measures in the analysis of data base structure and the development of keys.

7.2 RECOMMENDATIONS

Consider a few of the problems for which much better answers could be desirable. One class of problems stems from the fact that the object-predicate table is not actually available for our direct analysis in the theorem proving environment. We have suggested that for some applications we may use probability estimates obtained from a human "expert". In other circumstances [HART, 1975] more direct estimates of entropies would appear to be appropriate. More research on the best means for estimating the appropriate numbers is definately indicated.

Another problem which stems from the same source is the difficulty of estimating interactions of groups of predicates. Any insight which could be developed into this problem would be desirable. Also, further research into

the practical problems of combining evidence in a deductive system seems to be indicated.

The strategies developed in Chapter 3 pertain only to clause selection before the proof attempt is begun. Techniques which employ this information during the actual proof process might prove to be very useful. This problem is, of course, related to the problems already mentioned.

Lastly, we mention that terms such as "significant" have been used several times in this thesis. When is 15% mutual information significant, or when is 72% mutual information significant? It would appear that answers to questions of this kind must result from experimentation with implemented systems.

APPENDICES

APPENDIX A

Watanabe's Theorem of the Ugly Duckling

The number of those predicates y*, in a completed Boolean lattice Y* of predicates satisfied simultaneously by two nonidentical objects x_i and x_k (of the list of objects X) is a fixed constant independent of the choice of the two objects. By "nonidentical" is meant "belonging to two different types," that is, "corresponding to two different rows in the object-predicate table." To be able to speak of a completed lattice Y*, we must fix a "scope of observation". But, every time we change the scope we get a new Y*, and the theorem holds again in this new Y*.

Proof. Suppose that there are m different rows (object types) in the object-predicate table, which means that there are correspondingly m atoms in the lattice of Y*, and Y* has 2^m different members. Any predicate y* in Y* is a disjunction of a certain number of these atoms. A predicate shared by x_i and x_k is characterized by the fact that it contains the two atoms corresponding to these two object types. It can contain any number of the remaining (m - 2) atoms. There are 2^{m-2} different ways of taking some (or none) of the (m - 2) different atoms. Hence, 2^{m-2} different predicates are shared by these two objects, and this number, of course, does not depend on the choice of the two objects insofar as they belong to two different rows (nonidentical

objects).

More precisely, there are $\binom{m-2}{r}$ different ways of taking r out of the remaining (m - 2) predicates. Hence $\binom{m-2}{r}$ different predicates of dimension (r + 2) are shared by these two objects. Of course the number 2 mentioned above is obtained by $\sum_{r=0}^{m-2} \binom{m-2}{r}$.

APPENDIX B

Carnap's Relevance Situations for Two Observations

	(1)	(2)	(3)	(4)	(5)	(6)	(7)
T1a.	i.j	i.-j	-i.j	-i.-j	i	j	iVj
T1b.	h.k	h.-k	-h.k	-h.-k	h	k	hVk
No. 1	+	+	+	-	+	+	+
2	+	+	-	+	+	-	-
3A						+	
B	+	+	-	-	+	-	+
C						o	
4	+	-	+	+	-	+	-
5A					+		
B	+	-	+	-	-	+	+
C					o		
6A					+	+	
B					+	-	
C					+	o	
D					-	+	
E	+	-	-	+	-	-	-
F					-	o	
G					o	+	
H					o	-	
I					o	o	
7	+	-	-	-	+	+	+
8	-	+	+	+	-	-	-
9A					+	+	
B					+	-	
C					+	o	
D					-	+	
E	+	-	-	+	-	-	-
F					-	o	
G					o	+	
H					o	-	
I					o	o	

APPENDIX B (cont'd.)

	(1)	(2)	(3)	(4)	(5)	(6)	(7)
T1a.	i.j	i.-j	-i.j	-i.-j	i	j	iVj
T1b.	h.k	h.-k	-h.k	-h.-k	h	k	hVk
10A					+		
B	-	+	-	+	-	-	-
C					o		
11	-	+	-	-	+	-	+
12A						+	
B	-	-	+	+	-	-	-
C						o	
13	-	-	+	-	-	+	+
14	-	-	-	+	-	-	-
15	o	+	+	-	+	+	+
16	o	+	-	+	+	-	-
17	o	+	-	-	+	-	+
18	o	-	+	+	-	+	-
19	o	-	+	-	-	+	+
20	o	-	-	+	-	-	-
21	+	o	+	-	+	+	+
22	+	o	-	+	+	-	-
23	+	o	-	-	+	+	+
24	-	o	+	+	-	-	-
25	-	o	+	-	-	+	+
26	-	o	-	+	-	-	-
27	+	+	o	-	+	+	+
28	+	-	o	+	-	+	-
29	+	-	o	-	+	+	+
30	-	+	o	+	-	-	-
31	-	+	o	-	+	-	+
32	-	-	o	+	-	-	-
33	+	+	-	o	+	-	o
34	+	-	+	o	-	+	o
35	+	-	-	o	+	+	o
36	-	+	+	o	-	-	o
37	-	+	-	o	+	-	o
38	-	-	+	o	-	+	o
39	o	o	+	-	o	+	+
40	o	o	-	+	o	-	-
41	o	+	o	-	+	o	+
42	o	-	o	+	-	o	-

APPENDIX B (cont'd.)

	(1)	(2)	(3)	(4)	(5)	(6)	(7)
T1a.	i.j	i.-j	-i.j	-i.-j	i	j	iVj
T1b.	h.k	h.-k	-h.k	-h.-k	h	k	hVk
43	o	+	-	o	+	-	o
44	o	-	+	o	-	+	o
45	+	o	o	-	+	+	+
46	-	o	o	+	-	-	-
47	+	o	-	o	+	o	o
48	-	o	+	o	-	o	o
49	+	-	o	o	o	+	o
50	-	+	o	o	o	-	o
51	o	o	o	o	o	o	o

LIST OF REFERENCES

BIBLIOGRAPHY

[AGARD] Advisory Group for Aerospace Research and Development. "Conference Proceedings No. 94 on Artificial Intelligence." Request AGARD-CP-94-71, 7 Rue Ancelle 92 Neuilly Sur Seine, France, 1971.

[ANDERBERG] Anderberg, M. Cluster Analysis for Applications. Academic Press, New York, 1973.

[ASH] Ash, R. Information Theory. Interscience Publishers, New York, 1965.

[CARNAP] Carnap, R. Logical Foundations of Probability. 2nd Edition, University of Chicago Press, Chicago, 1962.

[CHANG] Chang, C.L., Lee, R.T.C. Symbolic Logic and Mechanical Theorem Proving. Academic Press, New York, 1973.

[FISHMAN] Fishman, D. "Experiments with a Resolution-Based Deductive Question Answering System and A Proposed Clause Representation for Parallel Search." Ph. D. Dissertation, University of Maryland, 1973.

[GOODMAN] Goodman, L., Kruskal, W. "Measures of Association for Cross Classification." American Statistical Association Journal, Vol. 54, 1959, pp. 123-163.

[GREEN] Green, C. "Theorem Proving by Resolution as a Basis for Question Answering Systems." In: Machine Intelligence 4. Meltzer and Michie (eds.) American Elsevier, New York, 1969.

[HARALICK] Haralick, R.M., Ripken, K. "An Associative-Categorical Model of Word Meaning." Artificial Intelligence, Vol. 6, 1975, pp. 75-99.

[HART] Hart, P. "A Computer Based Consultant for Mineral Exploration" (draft). Stanford Research Institute, Menlo Park, California, 1975.

[HOGG] Hogg, R., Craig, A. Introduction to Mathematical Statistics. 3rd. Edition, Macmillan, New York, 1970.

[HUNT] Hunt, E. Artificial Intelligence. Academic Press, New York, 1975.

[JAYNES] Jaynes, E. "Where Do We Stand on Maximum Entropy?" In: The Maximum Entropy Formalism, Levine, R., Tribus, M. (eds.), The MIT Press, Cambridge, Massachusetts, 1979.

[KOWALSKI] Kowalski, R., Hayes, P. "Semantic Trees in Automatic Theorem Proving." In: Machine Intelligence 4, 4, Meltzer and Michie (eds.), American Elsevier, New York, 1969.

[LEVINE] Levine, R., Tribus, M. (eds.) The Maximum Entropy Formalism. The MIT Press, Cambridge, Massachusetts, 1979.

[NILSSON] NILSSON, N. Principles of Artificial Intelligence. Tioga Press, Palo Alto, California, 1980.

[REBOH] Reboh, Raphael, Yates, Kling, Velarde. "Study of Automatic Theorem Proving Programs." Technical Note 75, Stanford Research Institute, Menlo Park, California, 1972.

[SZOLOVITS] Szolovits, P., Pauker, S.G. "Categorical and Probabilistic Reasoning in Medical Diagnosis." Artificial Intelligence, Vol. 11, 1978, pp. 115-144.

[TSICHRITZIS] Tsichritzis, D., Lochovsky, F. Data Base Management Systems. Academic Press, New York, 1977.

[WATANABE] Watanabe, S. Knowing and Guessing. John Wiley and Sons, Inc., New York, 1969.

[WILSON] Wilson, G.A., Minker, J. "Resolution. Refinements, and Search Strategies: A Comparative Study." IEEE Transactions on Computers, Vol. C-25, August, 1976, pp. 782-800.

[WOESE] Woese, C. "Archaebacteria." Scientific American, Vol. 244, No. 6, June, 1981.

REFERENCES

Cox, R. "Of Inference and Inquiry, An Essay in Deductive Logic." In: The Maximum Entropy Formalism, Levine, R., Tribus, M. (eds.), The MIT Press, Cambridge, Massachusetts, 1979.

Loveland, D. "Theorem Provers Combining Model Elimination and Resolution." In: Machine Intelligence 4, Meltzer and Michie (eds.), American Elsevier, New York, 1969.

Plaisted, D. "An Efficient Relevance Criterion for Mechanical Theorem Proving." In: The Proceedings of the First Annual National Conference on Artificial Intelligence, Menlo Park, California, 1980.

Skinner, C.W. "A Heuristic Approach to Inductive Inference in a Fact Retrieval System." Communications ACM, Vol. 17, 1974, pp. 77-86.