1

MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS
STANDARD REFERENCE MATERIAL 1010a
(ANSI and ISO TEST CHART No. 2)

National Library
of Canada

Bibliothèque nationale
du Canada

Canadian Theses Service

Service des thèses canadiennes

Ottawa, Canada
K1A 0N4

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

RUNES

Reasoning in Uncertain Nested Evidence Spaces

by

John Altoft, B.Sc.

A thesis submitted to the
Faculty of Graduate Studies and Research
in partial fulfillment of
the requirements for the degree of
Master of Engineering

Ottawa-Carleton Institute for Electrical Engineering
Faculty of Engineering
Department of Systems and Computer Engineering
Carleton University
December 1990

© copyright
1990, John R. Altoft

National Library
of Canada

Bibliothèque nationale
du Canada

Canadian Theses Service Service des thèses canadiennes

Ottawa, Canada
K1A 0N4

The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

ISBN 0-315-68888-2

Canada

The undersigned hereby recommends to
The Faculty of Graduate Studies and Research
acceptance of this thesis

RUNES
Reasoning in Uncertain Nested Evidence Spaces

submitted by John R. Altoft, B.Sc.
in partial fulfillment of the requirements for the degree of
Master of Engineering

A handwritten signature in black ink, appearing to read "N.W. Dumas", is written over a horizontal line.

Thesis Supervisor

A handwritten signature in black ink, appearing to read "V. Zyglidopoulos", is written over a horizontal line.

Chair, Department of Systems and Computer Engineering

Department of Systems and Computer Engineering
Faculty of Engineering
Carleton University
January, 1991

Abstract

Classification and diagnosis problems are considered in the context of evidential reasoning. Techniques for Shafer-Dempster reasoning over a hierarchy of hypotheses are examined. The RUNES system of nested subdomains of inference is proposed. RUNES uses an AND-OR hierarchy to achieve a concise yet clear representation. The computationally efficient algorithm for Shafer-Dempster belief functions over disjunctive trees, devised by Shafer and Logan, is extended to AND-OR trees without compromising its efficiency. The RUNES design is realized in the Smalltalk programming language. The approach is tested on a classification problem in which the types of trees are identified from their leaf and fruit attributes. The problem domain is cast as an AND-OR tree of 113 nodes which is equivalent to 723 million hypotheses over the possible attribute values. In simulated use the RUNES implementation of this problem proves to be efficient, effective, and robust to erroneous evidence or uncertainty in the specification of the solutions.

Keywords: Evidential Reasoning, Shafer-Dempster theory, hierarchies of hypotheses, belief functions, classification, diagnosis.

Acknowledgements

I would like to thank my advisor, Nick Dawes, for his guidance throughout this work and for his reminders to aim high. I am very grateful for the efforts of my friends Brian Barry and Glen Little, for their critical reviews of this document, and Jeff McAffer for his help and insights into the world of PostScript. I also wish to thank The Defence Research Establishment Ottawa for supporting my Masters program.

I have a special thanks for my children Karin, Alec and Erik, who despite their young age, were very tolerant of my absence, and who could rejuvenate their father with their hugs and kisses. And a very special thanks to my much loved wife Carmen for her encouragement and support. She worked very hard at keeping the world from intruding on me while I was working and I could not have done it without her.

Table Of Contents

Acceptance Sheet	ii
Abstract	iii
Acknowledgements	iv
Table Of Contents	v
List of Tables	vii
List of Figures	viii
Glossary Of Terms	x
Introduction	1
Chapter 1: The Classification Problem	4
1.1 Dempster-Shafer Theory	4
1.2 Related Work	6
1.3 Complexity in Real Classification Problems	11
Chapter 2: RUNES	17
2.1 Objective and Scope	17
2.2 Design	18
2.3 Mapping a Problem into RUNES	20
2.4 Maintaining Beliefs in AND-OR Trees	25
2.4.1 Details of the Belief Calculation	25
2.5 Inferring the Solution	31
2.5.1 Details of Tracing a RUNES Tree	32

Chapter 3: An Implementation of RUNES	59
3.1 Class Structure	39
3.2 Realization of AND-OR Trees	42
3.3 The Classifier.....	43
3.4 Query Generation Algorithm.....	45
3.5 The Best Path Browser.....	49
3.6 Example Problem Domains.....	52
3.7 The Tree Editor.....	65
Chapter 4: Analysis and Results.....	67
4.1 Testing Methodology	67
4.2 Simple Problem Domain.....	69
4.3 Realistic Problem Domain	70
4.4 Representation and Computational Requirements.....	96
Chapter 5: Summary	99
5.1 Concluding Remarks.....	99
5.2 Further Work	100
5.2.1 Solution Space Concepts	100
5.2.2 Independence of Evidence Spaces	101
5.2.3 Defaults and Prior Beliefs.....	102
5.2.4 Heuristics for Inference Control	103
5.2.5 Complexity in Evidence	103
5.2.6 Parallelism.....	104
Appendix A: Dempster-Shafer Theory	106
Appendix B: RUNES Algorithm for Nested Evidence Spaces.....	113
Appendix C: Test Results.....	122
References.....	151

List of Tables

TABLE	DESCRIPTION	PAGE
4.1	Distribution of Tree Types in the Forest and the Sample.....	84
4.2	Effectiveness using All Detailed Evidence plus Erroneous Evidence	87
4.3	Effectiveness using mixed Detailed Evidence and Erroneous Evidence.....	90
4.4	Ranking Solution Set Members using Prior Probabilities	93

List of Figures

FIGURE	DESCRIPTION	PAGE
1.1	Independent Attribute Classes	12
1.2	Composite Disjunctive Hierarchy of Attributes	13
2.1	Observable Attributes and Attribute Classes	23
2.2	Combining Evidence Spaces	23
2.3	RUNES Tree for Simple Example	24
2.4	Belief Function Reduction	38
3.1	RUNES Class Hierarchy	40
3.2	RUNES Classifier and Browser Classes	41
3.3	The Classifier	44
3.4	The Best Path Browser	51
3.5	The Tree Classification Problem	53
3.6	Needle Leaf <AND> Subtree	54
3.7	Cone_Fruit <AND> Subtree	55
3.8	Broad Leaf <AND> Subtree	56
3.9	Once_Compound <AND> Subtree	57
3.10	BL Attachment <OR> Subtree	57
3.11	BL Body_Shape <OR> Subtree	58
3.12	BL_Tip_Shape <OR> Subtree	58

FIGURE	DESCRIPTION	PAGE
3.13	BL_Base_Shape <OR> Subtree	59
3.14	BL_Edge <OR> Subtree	59
3.15	BL_Base_Symmetry <OR> Subtree.....	60
3.16	BL_Lobing <OR> Subtree.....	60
3.17	Classifier Showing Tree Classification Problem.....	61
3.18	Degree of Leaf Compounding.....	63
3.19	The Tree Attribute Editor	66
4.1	Update Times for Dempster's Rule.....	71
4.2	Growth in Hypotheses with Dempster's Rule	72
4.3	Update Times for RUNES with Supporting Evidence	75
4.4	Update Times for RUNES with Refuting Evidence.....	76
4.5	Update Times versus Depth of Supporting Evidence in RUNES Tree	78
4.6	Update Times versus Depth of Refuting Evidence in the RUNES Tree	79
4.7	Time to Trace the RUNES Tree.....	80
4.8	Growth of Hypotheses in the Trace Tree.....	82
4.9	Plausibility Difference versus Strength of Additional Erroneous Evidence	88
4.10	Plausibility Difference versus Strength of Mixed Erroneous Evidence	91
A.1	Dempster's Rule using a Unit Square	110

Glossary Of Terms

Basic Probability Assignment

a function which distributes a unit mass over a hypothesis space.

Belief

also called support, is the minimum portion, of a total belief of one, which must be (i.e. has already been) accorded to a proposition given the evidence.

Belief Function

a function which assigns a belief to each hypothesis in a hypothesis space.

Belief Interval

an interval whose lower bound is the belief in a hypothesis and whose upper bound is the plausibility of that hypothesis.

bpa

see Basic Probability Assignment

$\mathcal{C}$

see Coarsening.

Coarsening

a new frame of discernment created by grouping the members of an existing frame.
A coarsening will be denoted by the symbol: $\mathcal{C}$.

Confusion

a measure of uncertainty indicating the degree to which one's belief is distributed over many focal hypotheses.

Dempster's Rule

the rule for forming a belief function corresponding to the pooling of independent bodies of evidence. The symbol for the binary function denoting Dempster's Rule is: $\oplus$. (eg. $\text{Bel}_1 \oplus \text{Bel}_2$).

Dichotomous Belief Functions

a belief function is dichotomous with dichotomy $\{A, \sim A\}$ if it has no focal elements other than A , $\sim A$ and Θ .

Dissonance

a measure of uncertainty indicating the degree to which one's belief is attributed to incompatible focal hypotheses.

Evidence

observation/measurement/information concerning the state of the problem.

Focal Element

a hypothesis which has a nonzero assignment from the Basic Probability Assignment.

Frame of Discernment

the set of all possible answers to a question or a class of problems under consideration. A frame of discernment will be denoted by the symbol: Θ .

Note: Dempster-Shafer theory assumes one and only one of the members of the set is the correct answer and that the answer is indeed a member of the set.

$\mathcal{H}$

see Hypothesis Space.

Hypothesis

a subset of the frame of discernment.

Hypothesis Space

a set of hypotheses derived from the frame of discernment, over which the reasoning is to proceed. The minimal space is the set of all singletons from the frame of discernment. A hypothesis space will be denoted by the symbol: $\mathcal{H}$.

Mass

the assignment given a hypothesis by the Basic Probability Assignment.

Mass function

alias for Basic Probability Assignment.

Nonspecificity

a measure of uncertainty indicating the degree to which one's belief is attributed to hypotheses of high cardinality. In other words, ambiguity resulting from focal hypotheses which include substantial portions of the frame of discernment.

$\mathcal{P}$

see Partition.

Partition

given a frame of discernment Θ , a partition is a set of disjoint non-empty subsets of Θ whose union equals Θ . A partition can itself be regarded as a frame of discernment. A partition will be denoted by the symbol: $\mathcal{P}$.

Plausibility

the maximum portion, of a total belief of one, that could possibly be attained by proposition, given the evidence. The plausibility of a hypothesis is one minus the belief in its negation.

Θ

see Frame of Discernment.

$\mathcal{R}$

see Refinement.

Refinement

a new frame of discernment created by subdividing the members of an existing frame. A refinement will be denoted by the symbol: $\mathcal{R}$.

$\mathcal{M}(A)$

the set of nodes immediate below A; A's children.

Simple Support Function

a belief function which has at most one focal element other than Θ .

Singleton

a subset of the Frame of Discernment with only a single member; in other words a hypothesis with no ambiguity about the proposed solution.

Support

alias for Belief.

Introduction

In the latter half of the 1980's the expert systems approach to problem solving appeared ready to leave the laboratory and enter the world of real applications. However, many applications found that the simple problem representations used in the laboratory models did not scale-up. Complexity in both representation and computation proved their undoing. Techniques were required to manage the complexity of real problems.

The expert system approach tries to emulate the human approach to problem solving, at least to the degree that we think that we understand it. In the case of classification, humans seem to instinctively form categories and generalized concepts [Lakoff87]. It has long been standard practice for humans to address classification problems by devising a hierarchical taxonomy. These class-subclass hierarchies are often referred to as 'ISA' (standing for 'is a') or 'kind-of' (for 'is kind of') hierarchies. These taxonomies allow the problem solving to proceed at various levels of abstraction depending on the nature and the availability of information.

A common children's game, "20 Questions", demonstrates the power of the hierarchical approach. The preferred strategy for determining the unknown entity is to begin with very general abstractions and to successively refine the guess (i.e. move from class to subclass) until the entity is discovered. The power of the approach is attested to by the fact that it is often possible to classify an arbitrary entity using no more than twenty queries which only receive answers of "true" or "false".

This thesis is focused on the application of the Dempster-Shafer theory of belief functions to the problems of entity classification or fault diagnosis. In particular we will address those problems which have structure that can be hierarchically organized. Dempster-Shafer theory is a mathematically consistent technique for estimating an upper and lower bound for our belief in a hypothesis given uncertain evidence [Dempster66,68, Shafer76a,87a]. Some of the techniques proposed herein may be applicable to hierarchical structures which use the Bayesian model of belief functions [Pearl86,88]. The Dempster-Shafer approach was selected over the Bayesian model as it can explicitly represent uncertainty (i.e. lack of information). The Dempster-Shafer approach also removes the requirement that a belief of p in a proposition implies a belief of $1-p$ in the negation of that proposition. For a comparison of the two techniques in the context of hierarchical classification structures refer to the papers by Lee [Lee88] and Pearl [Pearl86a]. Unfortunately Dempster-Shafer theory is computationally prohibitive, even for problems of moderate size, if applied in an unrestricted manner [Orponnen90].

The objective of this research was to devise a representation for classification and diagnostic problems which was both concise and expressive and, at the same time, computationally efficient. Secondary issues to be considered were: measures or heuristics to evaluate the state of the analysis, the use of prior beliefs, and the suitability of the design for parallel implementation. In Section 2.2 and 2.3 we introduce a new representation of nested evidence spaces based on an AND-OR hierarchy which is significantly more concise and expressive than disjunctive hierarchies. Efficient updating of a Dempster-Shafer belief function over an AND-OR tree is achieved, with linear computational complexity, by extending Shafer and Logan's original algorithm (refer to Section 2.4).

This work does not represent a complete expert system design, instead it focuses on the representation of evidence and the computation of Dempster-Shafer beliefs over the evidence and by implication, over the solutions. The prototype system which implements the RUNES concepts has two modes of operation: passive interactive and active query. In the passive interactive mode the user volunteers evidence at any time and focussed on any node in the RUNES representation and the system responds by computing the effect of the new evidence on the current problem description (refer to Section 3.3). In the active query mode RUNES takes control of the session and poses systematic questions to the user about the attributes of the solution, again computing the effect of each piece of evidence as it becomes available. This approach to evidence gathering is described in Section 3.4.

Although the RUNES system shares several features with other systems from the domains of concept clustering and concept formation, such as: solutions grouped into categories, categories defined by sets of attribute values and hierarchical structure; the central purpose of RUNES is different. In concept clustering and concept formation the principal thrust of the research has been in the area of deducing the categories and their structure given descriptions of objects (solutions) in terms of their attribute values. Gennari, Langley and Fisher[89] present CLASSIT, a recent system for concept formation and they also review the features of its predecessors. In general these systems assume that: there is no a priori structure for the categories; the objects are described by object attribute pairs; and the object descriptions are unambiguous, complete (or nearly so) and uncontested. By contrast, the RUNES system is designed to exploit a static a priori category structure to enhance the classification performance in domains where the object-attribute pairs are ambiguous, incomplete, contested or otherwise uncertain.

Chapter 1.

The Classification Problem

The RUNES approach was designed for a class of problems known as diagnostic or classification problems. Although classification and diagnosis each provide a slightly different context for their respective problems they both share an underlying structure. Classifying means to arrange in classes, where a class is a grouping of objects possessing the same characteristics. Diagnose means to ascertain from symptoms the nature of a disease or fault. Both processes are based on reasoning about attributes of the solution.

Another feature common to many classification and diagnostic problems is the presence of uncertainty in either observation and measurement or in the relationship between the observed attribute and the solutions. The following section will introduce a mathematical theory of evidence which supports belief functions with an explicit representation of uncertainty.

1.1.Dempster-Shafer Theory

Dempster-Shafer theory is a system which represents degrees of belief numerically and explicitly allows for uncertainty in those beliefs. The theory originated with Dempster's proposal that probabilities be replaced with intervals consisting of a lower probability and an upper probability [Dempster 67]. The lower bound, now

commonly referred to as the *belief*¹, corresponds to the minimum probability which must be accorded to the proposition. The upper bound, now called the *plausibility*, is the maximum probability that could possibly be attained by the proposition. Dempster provided a method to combine multiple beliefs which is now known as Dempster's Rule. This rule of combination requires that the participating belief functions be based on independent (i.e. distinct) bodies of evidence, although he admits:

"The mechanism adopted here [Dempster's Rule of combination] assumes *independence* of the sources, a concept whose real world meaning is not so easily described as its mathematical definition."

Shafer, a student of Dempster, provided a solid mathematical foundation for Dempster's belief intervals when he developed a *Mathematical Theory of Evidence* [Shafer 76]. This approach is now commonly referred to as Dempster-Shafer theory or Evidential Reasoning. A brief introduction to Dempster-Shafer theory will be given here; for a more detailed introduction refer to Appendix A.

If we represent our total belief as a unit mass which can be distributed over the hypotheses of the hypothesis space then the traditional Bayesian approach only allows mass to be assigned to the singleton hypotheses from the frame of discernment². As a consequence when we attribute a portion of our belief, say m , to a hypothesis A then we must attribute $1-m$ to $\neg A$ (i.e. the negation of A) or a subset of $\neg A$, that is, the mass must be distributed over the remaining singletons. Dempster-Shafer theory however

¹ Some references call the lower bound the *support*.

² The frame of discernment is the set of all possible solutions within the problem domain. A hypothesis is a subset of the frame of discernment and it represents the proposition that the solution to the current problem is a member of this subset. The hypothesis space is the power set of the frame of discernment, that is, it is the set of all possible subsets of the frame of discernment.

allows not only the singleton hypotheses from the frame of discernment but all possible subsets of the frame of discernment. Now when we attribute a portion of our belief to hypothesis A it is no longer required that the balance be attributed to members of A's complement. The balance of our belief could be attributed to any hypothesis, which may include all of A, part of A or none of A.

A Bayesian belief function is a valid Dempster-Shafer belief function; the reverse however is not necessarily true [Shafer76a]. Probability Measures and Dempster-Shafer Measures are both examples of Fuzzy Measures [Dubois&Prade80, Shafer87a]. Klir and Folger [Klir&Folger88] give an overview of Fuzzy Measures and include an extensive list of references.

Most of the difficulties in applying Dempster-Shafer theory to real problems come from the computational complexity of Dempster's Rule of combination for belief functions. The rule can be simply described as taking each focal element of one belief function in combination with each focal element of the other belief function. The combination involves intersecting the focal elements and taking the product of the weights of belief on these foci. Any resulting weight of belief which is associated with a null focal element must be removed by a normalization process. Unfortunately Orponnen has shown that Dempster's Rule is #P-complete³ making its general application in real problems computationally prohibitive [Orponnen90].

1.2. Related Work

Barnett devised an efficient technique to compute Dempster's rule for applications

³ #P-complete (read "number-P-complete") the hardest examples of a class of computationally difficult problems involving an enumeration process [Garey79].

which can be represented by a very restricted hypothesis space and which use only simple support functions⁴ to represent the evidence [Barnett81]. The restricted hypothesis space must consist of only the frame of discernment and all its singleton hypotheses. (i.e. no intermediate or general abstractions are supported). Each simple support function must have either a singleton or its complement as its focus. Barnett's algorithm has a computational complexity which is linearly proportional to the cardinality of the frame of discernment.

Gordon and Shortliffe proposed reducing the general Dempster-Shafer hypothesis space from the power set of the frame of discernment to only those hypothesis which are of semantic interest and which form a single disjunctive hierarchy [Gordon&Shortliffe85a]. The root of the hierarchy represents the entire frame of discernment and the hierarchy terminates in the singleton hypotheses at the leaf positions. They also require that the evidence derived from the observations should be represented by simple support functions in order to reduce the complexity of the calculations. The first step of their algorithm produces, for each node, a simple support function containing all confirming evidence focused on the node and a second simple support function containing all refuting evidence focused on the node. The next step combines all of the confirming support functions for all nodes in the hierarchy into a single belief function. The last step consists of incrementally incorporating the refuting support functions from all nodes of the hierarchy. Gordon and Shortliffe did not state the computational complexity of their algorithm but Shafer and Logan report that they were able to implement it in a form which was linear in the number of nodes in the tree

⁴ Simple support functions are belief functions with only two focal hypotheses where one of these hypotheses is the frame of discernment. Essentially these functions focus a portion of our belief on a single hypothesis and leave the remaining portion of our belief uncommitted.

[Shafer&Logan87b].

Unfortunately the Gordon and Shortliffe method does have a further weakness in that it cannot, in all cases, calculate the plausibilities of the hypotheses. The problem occurs when dealing with evidence which is focused on the complement of a hypothesis in the hierarchical space. Evidence of this kind may generate support for a hypothesis which is not represented in the restricted hypothesis space. When this occurs the Gordon and Shortliffe method calls for an approximation which attributes that support to the hypothesis in the hierarchy which is the smallest superset of the desired hypothesis. The approximation results in computed plausibilities which may be greater than the actual values but are never less than the actual values. This approximation restricts the usefulness of the plausibility measure. If the computed plausibility of a hypothesis is small then the actual plausibility must also be small and it is implausible that the solution is a member of that hypothesis. If however, the computed plausibility is high one cannot infer that the actual plausibility is also high and therefore one cannot, in this case, make a confident decision based on plausibility.

Shafer and Logan proposed a technique [Shafer&Logan87b] to handle hierarchically organized hypothesis spaces like those described by Gordon and Shortliffe. Their technique however, handles negated evidence without an approximation thereby producing a correct representation of the plausibilities of the hypotheses in the hierarchy. A feature of Shafer and Logan's approach is that all computations are local to the node with the changes being propagated, as messages, up or down the hierarchy. Their technique could be described as a four step algorithm. The first step is the same as that of Gordon and Shortliffe, combine all evidence with a common focus only in this case a single dichotomous support function is used instead of two simple support functions. The second step propagates the result of the observed evidence up the tree by determining the

effect that offspring nodes have on their parent node. The next step uses the constraint that the solution must be within the frame of discernment of the root node, to resolve the final beliefs of the immediate children of the root node. Finally, the fourth step propagates what amounts to a normalization process down the tree resulting in all nodes obtaining a belief value consistent with all of the evidence. The computational complexity of their algorithm is linear in the number of nodes in the hierarchy (as is Gordon and Shortliffe's method). Shafer and Logan's algorithm forms the basis of the belief maintenance system within RUNES and will be elaborated in the following sections.

Shenoy and Shafer explore further the idea of local computation with update messages in what they call *Qualitative Markov* trees [Shenoy&Shafer86,90]. In this work they generalize Pearl's Bayesian causal trees [Pearl85] into a form usable with the more general Dempster-Shafer belief functions. Pearl's causal trees exploit the conditional independence of one set of nodes from another set, given the node which joins the two sets. In Qualitative Markov trees it is the partitions (i.e. the frames of discernment) represented by the sets of nodes which must be conditional independent, not the probabilities, thus their use of the adjective *qualitative*.

A related area of work which has application to the classification problem is the work by Jaynes on *Maximum Entropy* theory [Jaynes57a, 57b, 79]. This theory was developed as a result of the work by Shannon on information theory, uncertainty and entropy [Shannon48]. Jaynes proposed that Shannon's entropy measure could be used as a tool of statistical inference to define the values for probabilities. Jaynes' work focused on the use of Maximum Entropy theory to define the probability distribution over the members of the frame of discernment in the least presumptive manner given the constraints imposed by the evidence. Although Maximum Entropy theory restricts itself

to solutions of the form of traditional probability distributions over the members of the frame of discernment, it does not restrict the constraints imposed by the evidence. The evidence is free to place constraints on the probability of any hypothesis over the frame of discernment.

Measures of uncertainty or, conversely, information are of potential interest for heuristic control of inferencing. Shannon's entropy measure is confined to domains where one's belief is only distributed over the singletons from the frame of discernment (eg. Bayesian belief functions). However the success of Shannon's and Jaynes' techniques inspired the development of new measures, based on generalizations of Shannon's entropy, which are applicable to more general belief functions and their associated hypothesis spaces [Klir&Folger88, Yager87]. Klir and Folger describe uncertainty as having several subtly different aspects. First there is the notion of *nonspecificity* which represents the ambiguity resulting from belief in hypotheses with high cardinality (i.e. hypotheses representing multiple indistinguishable alternatives). Next there is *dissonance* which represents the uncertainty which results from combining conflicting evidence. And finally there is the concept of *confusion* which is a measure of the lack of focus in the belief function. High levels of confusion are related to belief functions with many foci even if the belief function is consonant⁵. These three uncertainty measures were examined to determine their usefulness as metadata for the query generation algorithms used in the RUNES system.

Next there is the work by Yen on extensions to Dempster-Shafer theory to include a priori beliefs [Yen86, 89]. In Yen's approach each piece of evidence is

⁵ Consonant belief functions have no internal conflict. All focal hypotheses of the belief function are nested, one within the other, forming one chain from the most general hypothesis to the most specific one.

weighted inversely proportional to its prior probability and then combined using Dempster's rule. The resulting belief function, after all evidence has been combined, is then weighted proportional to its prior probabilities and normalized. This method produces results which are consistent with Bayes' theorem with conditional independence assumptions. Yen also described a method to provide probabilistic mappings from evidence to solution hypotheses where the mass associated with a focal element from the evidential support function is distributed, according to a probability distribution, over a set of solution hypotheses.

Lastly, a summary of the related work would be remiss without mention of Pearl whose extensive work [Pearl85,86a,86b,87,88,90] in the area of Bayesian evidential reasoning over networks and hierarchies has prompted many other investigations of evidential reasoning. In particular, Pearl's work on local computation of belief functions laid the ground work for Shafer and Logan's belief revision algorithm for disjunctive hierarchies which is an essential part of the RUNES approach.

1.3. Complexity in Real Classification Problems

The hierarchical classification approaches described in the preceding section were all designed for a single disjunctive hierarchy. Classification problems in the real world rarely fit neatly into a single disjunctive hierarchy. It is usually the case that the objects being classified will have several types of attributes which are not hierarchically related to one another. For example, objects can have the attributes of shape and colour, Figure 1.1. Shape is not a specialization nor a generalization of colour, therefore any disjunctive hierarchy involving both attributes must make an arbitrary decision to subordinate one to the other.

Figure 1.1: Independent Attribute Classes

This method of producing a single disjunctive hierarchy where one does not naturally exist has the very undesirable side effect of restricting the hypotheses that can be represented by the hierarchy. For example, if colour is subordinated to shape, then the colour subtree would be replicated below each terminal node of the shape subtree (Figure 1.2). Assume that "circle" and "square" are two possible values for the shape attribute and that "red" and "blue" are possible values for the colour attribute, then the hierarchy will contain nodes for "circle" and "square", and somewhere below circle it will contain "red-circle" and "blue-circle". The colour attributes will be similarly repeated below square. In this case, the resulting hierarchy has no representations for the hypotheses involving both general shape concepts and any colour concept, only hypotheses with specific shape attributes (i.e. terminal nodes in the shape subtree) can have colour attributes. A similar situation would exist for general colour concepts if shape is subordinated to colour instead.

Figure 1.2: Composite Disjunctive Hierarchy of Attributes

A related problem is evidence collection and combination when the hierarchy contains replicated subtrees. Continuing with the problem domain with attributes of shape and colour where colour has been subordinated to shape, we find that evidence about colour is no longer independent of our belief in the shape. Consider evidence which suggests that the colour of the unknown object is red. In order to use the techniques of either Gordon and Shortliffe or Shafer and Logan, the evidence must be represented by a support function focused on a single node in the tree. Unfortunately there are at least two nodes in the tree implied by the concept "red", those being "red-circle" and "red-square". One cannot simply add the evidence to each node separately as this violates the requirement of Dempster's Rule for independence between pieces of evidence. It would also create dissonance in the belief function where none should exist.

In order to cast the colour evidence as a support function that is acceptable to the hierarchical representation the colour evidence must include shape information (i.e. "red-circle" or "red-square"). Clearly that approach is very unsatisfactory as we must give up the independence of attribute measurement that is otherwise possible. In addition attributes higher in the tree get preferential treatment as they are less dependent on the specification of other attributes. Indeed it would be difficult to proceed with evidence combination in any fashion except top-down, and even then belief revision may be unduly complicated by the interplay of the preferred attributes.

Creating a single disjunctive hierarchy by arbitrarily subordinating attributes which have no intrinsic hierarchical relationship, results in the subordinated attribute subtree being replicated. This replication leads to extremely large representations for problems that by human standards would be considered small. Consider a problem domain with six attribute types, each represented by a binary subtree with eight terminal values. A disjunctive hierarchy produced by the method described above would have 501,735 nodes of which 262,144 would be terminal nodes. If one were presented with the disjunctive hierarchy alone it would take some effort to find the otherwise simple problem hiding within. Clearly a more concise representation would be beneficial to even moderately sized problems and essential for larger problems.

If a single disjunctive hierarchy is not a good representation for most real classification problems, what is? To explore that question let us return to the game of "20 Questions". What are the techniques used by humans when playing 20 Questions, and what do they tell us about our underlying representation for the problem? As stated previously the players usually use a successive refinement technique, but closer examination reveals that there are actually two strategies at work here. There is the obvious partitioning of the solution space into disjoint subdomains; for example,

mammals can be partitioned into "humans", "whales", "dogs", etc. This corresponds to the disjunctive hierarchies that are discussed above. The second line of questioning involves multiple alternate partitions of the same subdomain. For example, the subdomain of mammal might be partitioned into land dwelling mammals and water dwelling mammals. Another partitioning of the subdomain of mammals might be into carnivore mammals, herbivore mammals, and omnivorous mammals. The point is that our structure may have multiple alternate representations for the same subset of solutions.

The disjunctive partitioning of a hypothesis (set of potential solutions) reduces the ambiguity by simply grouping the current choices into disjoint hypotheses. The multiple alternate partitions of the same hypothesis using attributes from different attribute classes achieves the same reduction of ambiguity through partitioning within each attribute class but the largest reduction in ambiguity comes from combining attributes across attribute classes. Again using the above mammal example, choosing "land dwelling mammal" defines one element of a partition of the set of mammals while choosing "carnivore mammal" selects an element of a different partition. Now if "carnivore mammal" and "land dwelling mammal" are taken together the hypothesis that this defines is just the intersection of the two partition elements. Conjoining the attribute classes is a simple matter of taking the intersection of the partition elements and associating them with the union of the attribute values.

The disjunctive partitioning, therefore, is a process of specialization which reduces ambiguity by superseding an existing attribute value with a more precise attribute value from the same attribute class (which has a subset superset relationship with the existing attribute). The conjunction of attribute classes, however, is a specification process which reduces ambiguity through the composition of attribute values from different attribute classes.

Both of these strategies use classical category theory⁶ to indirectly manipulate sets of solutions by acting on sets of attribute values. The categories of "human", "whale", "dog", etc.; actually have no special status in the representation other than their zoological origin, which makes them seem less adhoc. Therefore the subdomain, or solution category, of "mammal" can be represented as having a set of alternatives for partitioning, each based on a different attribute class. Notice also that these attributes do not necessarily extend to the entire solution space (i.e. inanimate objects would not be partitioned according to their place of dwelling).

Gordon and Shortliffe proposed what they call "hierarchical hypothesis spaces" which arranged solutions into a single disjunctive hierarchy. Shafer and Logan maintained the disjunctive hierarchy but called it "hierarchical evidence" thereby shifting the emphasis from reasoning about solutions to reasoning about evidence. However, Shafer and Logan still represented their trees as terminating with solutions in the leaf positions. None of the proposed techniques make use of composition of attributes.

In addition, only Yen's work explicitly recognized the difference between solutions and evidence. Solutions are not observed they are implied by the evidence. The attributes are observed and much of the reasoning takes place over these attributes with the solutions playing a relatively passive role.

⁶ The use of categories in human reasoning is complex and poorly understood. Lakoff provides an excellent introduction into category theory [Lakoff87]. The term "category" when used with respect to the RUNES system or machine representations will assume the classical definition of categories, that is a crisp set of entities sharing one or more common attributes.

Chapter 2.

RUNES

2.1. Objective and Scope

Previous experience with real classification problems indicated that the use of a single disjunctive hierarchy was too restrictive, and the use of Dempster's Rule in an unconstrained hypothesis space was computationally prohibitive. The objective of this research was to devise an Evidential Reasoning system for classification problems which was both concise and expressive in its representation and, at the same time, efficient in its computation of the beliefs. The research was to focus on problem domains which exhibited some hierarchical structure in the relationship between the elements of the evidence. Secondary issues to be considered were: measures or heuristics to evaluate the state of the analysis, prior beliefs, and the suitability of the design for parallel implementation. The scope of this work does not include the subsystem responsible for explanation and it only includes a primitive query generation facility.

This work assumes that Dempster-Shafer theory will be used for the belief system. The merits of this technique versus other techniques (most notably the Bayesian approach) have been, and continue to be, debated in the literature [Zadeh84,86, Pearl86a, Kyburg87a, Shafer87a, Lee88]. The Dempster-Shafer approach was selected for its explicit representation of uncertainty and its intuitive appeal.

The RUNES design reported herein has several underlying assumptions, some of

which are a direct result of adopting the Dempster-Shafer approach and some of which merely allow the work to be scoped to a reasonable task. The fundamental assumptions rooted in Dempster-Shafer theory are:

- 1) the frame of discernment of the problem is finite and exhaustive.
- 2) there is exactly one correct answer.
- 3) the foci of support functions representing the evidence are crisp sets¹.

The pragmatic assumptions are:

- 1) the evidence is represented by simple or dichotomous support functions.
- 2) the observable attributes can be represented as discrete values.
- 3) number of attribute types is finite.
- 4) the categories of solutions defined by sets of attribute values are crisp sets.

Fortunately many interesting problems can be addressed while respecting these constraints.

2.2. Design

The RUNES design has two fundamental assumptions: first, that solutions are not directly observed, they are implied by the evidence and second, that a significant portion of the reasoning required to obtain a solution is concerned with combining the evidence into a single coherent, or at least preferred, view.

RUNES is based on the idea that categories of solutions (i.e. a hypothesis in the solution space) can be formed from the frame of discernment for a problem domain. Each category is defined by the set of attribute values common to all members of that category. The attribute values must be taken from the set of values which the observable

¹ Classical set theory, membership is not graded (i.e. not Fuzzy sets).

evidence can assume, if the categories are to be useful concepts in the inference process. Within the RUNES system these attribute values are referred to as evidential elements.

The evidential elements in RUNES are grouped into sets which define all possible outcomes for a test or an observation. These sets are the frames of discernment for the tests. Associated with each evidential frame of discernment is a hypothesis space which, in the RUNES system, is referred to as an evidence space. Evidence spaces can support their own hierarchically organized hypothesis spaces. In some cases a hypothesis in the evidence space (i.e. a subset of the evidential frame) will have several independent attributes of its own. In these cases the same techniques can be used to produce separate evidence spaces for each of the contributing attributes. This factoring of the evidence spaces into subspaces can be applied as often as is needed.

Since every set of evidential elements defines a category of solutions, the frame of discernment for a class of attributes defines a category. In some cases, this category will be the entire frame of discernment for the problem which implies that every possible solution exhibits at least one evidential element from this attribute class. We will see later that these attribute classes have a special status.

Thus far, RUNES has been described as a system with an explicit distinction between solutions and evidence, that it clusters observable evidence values into attribute classes, and that sets of evidential values define categories of solutions, but it still requires a structure to support the interaction of evidence. The structure that RUNES uses is a hierarchy of nested evidence spaces. The hierarchy is a simple tree structure which allows both disjunctive and conjunctive nodes. An evidence space is a hypothesis space over the frame of discernment of an attribute class. The positioning of the evidence spaces within the hierarchy (i.e. the nesting) is determined by the relationships between

their associated categories in the solution space. In other words, attribute classes which have a category of solutions equal to the entire solution frame, appear at the root of the RUNES hierarchy. Those attribute classes which have solution categories which include only a fraction of the possible solutions will appear toward the leaves of the hierarchy.

The RUNES system makes certain assumptions about the organization of the hierarchy. An evidence space is a hierarchically organized disjunction of evidential hypotheses, evidential elements and conjunctions of embedded evidence spaces. Within the RUNES design the root of an evidence space is called an *EvidenceSpace* (or a *TopEvidenceSpace*). An evidential hypothesis and an evidential element are called a *Hypothesis* and an *EvidentialElement* respectively, and the conjunctive node is called an *AndNode*. Only the roots of evidence spaces are conjoined and they are always connected by conjunction. The allowed structures and the process of creating a RUNES structure will be described in subsequent sections.

2.3. Mapping a Problem into RUNES

When mapping a problem domain into a RUNES tree the goal is to create a structured evidence space representing the observable evidence values and hypotheses over these values which are of semantic or pragmatic interest. Semantically interesting hypotheses are those which are in common usage in the problem domain and have some conceptual meaning to humans operating in that problem domain. Hypotheses of pragmatic interest arise from the nature of the observation process or the machine representation of the problem domain and do not necessarily correspond to natural concepts in the problem domain. Recall that evidence spaces define the values and allowed hypotheses for an evidence class and that evidence spaces are nested according to their relationship to the solution space of the problem domain.

As with many problem-solving endeavors, building a RUNES tree structure is usually an iterative process. The following procedures will serve as a guide:

1) Identify all of the observable attributes for all possible solutions within the problem domain. Group the attributes by type to form a frame of discernment for each attribute type. There will be an evidence space (i.e. a subdomain of inference) associated with each of these attribute types. In the RUNES system, the evidence spaces must be hierarchically organized. An evidence space with minimal structure would consist of a disjunctive root node whose $\delta\delta$ contained one terminal node for each member of the frame of discernment. Often, however, evidence spaces will contain intermediate hypotheses of semantic or pragmatic interest. Each node in the subtree represents a subset of the possible attribute values. Each node also has an associated category of solutions (i.e. a hypothesis in the solution space of the problem) defined by those attribute values.

2) One or more of the evidence spaces will have an associated category of solutions which equals the entire frame of discernment of the problem domain. These subtrees form the evidence spaces closest to the root of the RUNES hierarchy. If there is only one such evidence space its root is the root of the RUNES tree. If there are several, then a conjunctive root is created and they form its $\delta\delta$.

3) Any evidence spaces which are associated with categories of solutions which are proper subsets of the frame of discernment of the problem domain, must attach to nodes in one of the upper level subtrees. Given an evidence space which must be attached, the point of attachment is found by locating the node in the current RUNES hierarchy which has a category of solutions equal in membership to the category defined by the evidence space being joined. If the node at the attachment point is not a root node and not a terminal node, then it becomes a conjunctive root node whose $\delta\delta$ contains both

the evidence space being attached and a new evidence space created from the existing subtree below the attachment point. If the node at the attachment point is already a conjunctive root, then the evidence space being attached simply joins the $\mathcal{M}$ of that root. If the node at the attachment point is a terminal node, then the attaching evidence space is just a further disjunctive refinement of the existing terminal node and its subtree should be attached without distinguishing it as a separate evidence space. This process continues until all evidence spaces have been incorporated into a single RUNES tree structure.

The procedure for constructing a RUNES tree will now be illustrated using a simple example. This problem has thirteen solutions in its frame of discernment:

1. Thread	6. SugarCube	11. Coaster
2. SolidGlassBall	7. CardboardBox	12. GlassPaperWeight
3. SoapBubble	8. PostageStamp	13. GlassDisk
4. BowlingBall	9. WindowPane	
5. Balloon	10. Aquarium	

Figure 2.1 shows the nine observable attributes of this problem domain and the solutions (indicated by numbers) which have each attribute. The same figure also shows the grouping of attribute values into frames of discernment for an attribute class. Figure 2.2 shows the addition of a semantically interesting hypothesis space over each attribute class.

The next step is to collect the evidence space or spaces which have an associated category of solutions equal to the entire frame of discernment for the problem. In this example there is only one, the *Shape* evidence space, therefore it will be the root of the RUNES tree. Next the remaining evidence spaces are added by comparing the category

Figure 2.1: Observable Attributes and Attribute Classes

Figure 2.2: Combining Evidence Spaces

of solutions associated with their roots with those of the nodes in the current RUNES tree (based on *Shape*). In this example, *Transparency* has the same category of solutions as *MultiDimensional* so *MultiDimensional* becomes a conjunctive node, its subtree becomes a new evidence space (with a root node created for it) and the *Transparency* evidence space becomes a child of *MultiDimensional*. Likewise, the evidence space *Interior* has the same category of solutions as *3_Dimensional*, so *3_Dimensional* is also converted to allow *Interior* to attach. The resulting RUNES tree for this simple example is shown in Figure 2.3.

Figure 2.3: RUNES Tree for Simple Example

2.4. Maintaining Beliefs in AND-OR Trees

The belief maintenance mechanism within RUNES is a derivative of the technique developed by Shafer and Logan for Dempster-Shafer reasoning in purely disjunctive hierarchies [Shafer&Logan87b]. Shafer and Logan's algorithm features locally controlled calculations and belief propagation by passing messages up and down the tree. The essence of this approach is based on two observations. First that when determining the degrees of belief in a node A and its children, resulting from the evidence bearing on nodes below A, it is sufficient to consider the degrees of belief in each child separately. Secondly, the evidence which bears on nodes above A or down other branches of the tree, affects our belief in the children of A only inasmuch as it affects our belief in A itself. This means that this evidence can affect the absolute value of the belief in the children of A but not their standing relative to each other.

In the following section we will see that these two observations hold for conjunctive nodes as well, if the evidence spaces which are being conjoined are independent. The RUNES enhancements to the disjunctive algorithm are then discussed.

2.4.1. Details of the Belief Calculation

Given the above observations, Shafer and Logan devised an approach which is based on this equation:

$$\text{Bel}_{\Theta \downarrow} = \text{Bel}_{A \downarrow} \oplus \text{Bel}_A \oplus \text{Bel}_{A \uparrow}$$

where:

$\text{Bel}_{A \downarrow}$ - belief function given evidence focussed on nodes below A.

Bel_A - belief function given evidence focussed directly on A.

$\text{Bel}_{A \uparrow}$ - belief function given evidence focussed on nodes above A

or on nodes down other branches.

$\text{Bel}_{\Theta \downarrow}$ - belief function given all evidence.

and $\oplus$ denotes Dempster's Rule for combining beliefs. Refer to Appendix B for a description of belief function syntax and usage.

The Shafer-Logan algorithm has two distinct phases which exploit the interaction of evidence in a hierarchically organized evidence space. The first phase begins at the terminal nodes of the tree and progresses toward the root accumulating evidence as it proceeds. At the end of the first phase all nodes in the tree have their children in correct relative relationship to each other but their final belief, in the context of all of the evidence, has not been determined. The second pass proceeds from the root back down the tree as each node incorporates the effect of evidence bearing on nodes outside of its subtree. The algorithm as they presented it, is implemented as six subroutines² which determine, for each node below the root node, the values of beliefs representing various combinations of evidence focused on themselves, nodes above or nodes below. A brief overview of the algorithm will be presented here; for a complete description of the algorithm refer to Appendix B.

In the first phase of the algorithm Subroutines 1 and 2 are used by all nodes below the root node to propagate beliefs up the tree. These routines assume that all evidence with a common focus has been combined into a single dichotomous belief for each node (i.e. Bel_A for each node A). Subroutine 1 determines $Bel_{A\downarrow}(A)$ and $Bel_{A\downarrow}(\neg A)$ given $(Bel_B \oplus Bel_{B\downarrow})(B)$ and $(Bel_B \oplus Bel_{B\downarrow})(\neg B)$ for all B where B is a child of A. For a terminal node B, $Bel_{B\downarrow}$ is vacuous; which is why the calculation proceeds from the terminal nodes to the root node. Subroutine 2 calculates $(Bel_A \oplus Bel_{A\downarrow})(A)$ and $(Bel_A \oplus Bel_{A\downarrow})(\neg A)$.

² The numbering of the subroutines within this paper are identical to those in Shafer and Logan's original paper to facilitate referencing.

When $(\text{Bel}_A \oplus \text{Bel}_{A\downarrow})$ has been determined for all A in the bl of the root node then Subroutine 3 is used to calculate $\text{Bel}_{\Theta\downarrow}(A)$ and $\text{Bel}_{\Theta\downarrow}(\neg A)$ for all A . Subroutine 3 implements the assumption that the frame of discernment, represented by the root node, must contain the solution.

Phase two of the algorithm is implemented by Subroutines 4, 5 and 6 as the beliefs based on all of the evidence are propagated back down the tree. Subroutine 5 determines the values of $\text{Bel}_{A\downarrow}(B)$, $\text{Bel}_{A\downarrow}(\neg B)$ and $\text{Bel}_{A\downarrow}(B \cup \neg A)$ given $(\text{Bel}_C \oplus \text{Bel}_{C\downarrow})(C)$ and $(\text{Bel}_C \oplus \text{Bel}_{C\downarrow})(\neg C)$ for all C where B and C are children of A . Subroutine 6 calculates $\text{Bel}_{\Theta\downarrow}(B)$ and $\text{Bel}_{\Theta\downarrow}(\neg B)$ given $\text{Bel}_{A\downarrow}(B)$, $\text{Bel}_{A\downarrow}(\neg B)$, $\text{Bel}_{A\downarrow}(B \cup \neg A)$, $\text{Bel}_{A\downarrow}(A)$, $\text{Bel}_{A\downarrow}(\neg A)$, and the results of Subroutine 4; where B is a child of A . Subroutine 4 calculates the effect of evidence focussed on a node, above it, or down other branches; $(\text{Bel}_{A\downarrow} \oplus \text{Bel}_A)(A)$ and $(\text{Bel}_{A\downarrow} \oplus \text{Bel}_A)(\neg A)$ given $\text{Bel}_{\Theta\downarrow}(A)$, $\text{Bel}_{\Theta\downarrow}(\neg A)$, $\text{Bel}_{A\downarrow}(A)$, and $\text{Bel}_{A\downarrow}(\neg A)$.

Subroutine 5 is similar to Subroutine 3 except that the assumption that the solution must be within the hypothesis represented by A cannot be made a priori. $\text{Bel}_{A\downarrow}(B \cup \neg A)$ is the belief which devolves onto B as the belief in A increases. The behavior of Subroutines 4, 5 and 6 approaches that of Subroutine 3 as the belief in A increases toward a certainty.

Let us now reconsider, in the context of a conjunctive node in a RUNES tree, the two key observations of Shafer and Logan concerning belief combination in hierarchies. First there is the observation that the children of A can be considered separately in relation to the evidence bearing on nodes below A . Examination of this property in the case of disjunctive nodes reveals that it is the result of the requirement that every node have only one parent, the requirement that the evidence be represented by simple or

dichotomous support functions and the fact that Dempster's Rule is associative. This means that every piece of evidence bearing on nodes below A can be uniquely attributed to one of A 's children and that the resulting belief is independent of the order of combination of the evidence. Since all nodes in a RUNES tree have only one parent and the evidence is restricted to simple or dichotomous support functions, RUNES trees meet the requirements and the first observation holds for them as well.

Examination of Shafer and Logan's second observation concerning evidence focussed on nodes above A or down other branches of the tree, reveals that it too is a result of the single parent requirement, combined with the fact that sibling nodes in Shafer and Logan's tree are disjoint. These two features constrain the nodes above A or down other branches to fall into one of two categories: those nodes which contain A as a proper subset and those which do not intersect A . It is not possible to have a node which does not have A as a proper subset yet still has a nonempty intersection. As a result, no evidence focussed on nodes above A or down other branches can influence our belief in the children of A other than to support or refute our belief that the solution is indeed among the solutions represented by A and its children.

In the case where A is a RUNES conjunctive node, the children of A are not disjoint, in fact they represent exactly the same hypothesis as A . Recall that the children of a conjunctive node all represent the same hypothesis in the solution space of the problem, but that each proceeds to partition that hypothesis using a different attribute type. Each child of a conjunctive node is a distinct evidence space.

Let A be the parent of B , where A is a RUNES conjunctive node. Then Shafer and Logan's second observation holds for RUNES trees if the evidence focussed on nodes above B and down other branches does not imply a belief in some but not all of B 's children. Node A is disjoint with its siblings therefore the nodes above A and down other

branches conform to this requirement. Node A is an equivalent hypothesis to node B therefore A confers no preferences on the children of B. That leaves only the siblings of B. The siblings of B conform to the requirements if they represent independent partitions of the solution space hypothesis represented by A. Two partitions are independent if no hypothesis from one partition (other than the vacuous hypothesis) contains any hypothesis from the other partition.

The RUNES system assumes that conjoined evidence spaces are independent and that Shafer and Logan's second observation therefore applies. This is not always the case for real problems, with the result that the beliefs produced by the RUNES system may deviate from the correct values. If two hypotheses from independent evidence spaces are conjoined then the belief in the resulting hypothesis is simply the product of the beliefs in the original hypotheses. Similarly the plausibility is also the product of the plausibilities of the origin hypotheses. Omitting the trivial case for beliefs which are certainties, conjunction of independent hypotheses always results in a belief which is smaller than that of any of the arguments. However, if the evidence spaces are not independent the resulting belief can be equal to or larger than that of one or all of the arguments. Assuming independence, as RUNES does, can lead to under-estimating the belief and the plausibility. The independence assumption will be re-examined later in this report.

The RUNES algorithm for computing the beliefs of nested hierarchical evidence spaces uses the Shafer-Logan algorithm with some variations. First let us identify those parts which are the same. The *EvidentialElement* and *Hypothesis* nodes of RUNES are the same as Shafer-Logan's terminal nodes and non-terminal, non-root nodes respectively. They use Subroutines 1 and 2 to update themselves from their children and they use Subroutines 4, 5, and 6 to update their children. The *TopEvidenceSpace* nodes are very similar to a Shafer-Logan root node in that they represent the entire solution

frame, therefore their belief is always assumed to be a certainty (i.e. their frame must contain the solution) and they do not require their children to update them. Both use Subroutine 3 to update their children.

The RUNES algorithm differs from the Shafer-Logan algorithm in that it also handles conjunctive nodes. The *AndNode* represents the conjunction of evidence gathered in different evidence spaces where these spaces share a common category of solutions. In other words, the *AndNode* and each of its children represent the same hypothesis in the solution space of the problem. Since the children of the conjunctive node do not represent disjoint hypothesis, it is inappropriate to use Subroutines 1 and 2 to update the conjunctive node from its children. Instead each evidence space under the conjunctive node combines all evidence within its subtree into a single dichotomous belief focussed on the root of the space. These beliefs summarize the evidence for and against the proposition that the solution is in the hypothesis that the *AndNode* represents. The *AndNode* then combines these beliefs using Dempster's Rule to produce $Bel_{A\downarrow}(A)$, and $Bel_{A\downarrow}(\neg A)$ where A is the conjunctive node.

The *AndNode* plays a dual role in a RUNES tree. Its primary role is to perform the conjunction of the evidence spaces below it. In this role it determines the combined effect of all evidence focused on nodes within the evidence spaces below it. The *AndNode's* secondary role is only active when it is imbedded in another evidence space (i.e. it has a parent node in some upper evidence space). This role involves representing its hypothesis within the upper evidence space. From the point of view of the *AndNode's* parent node, the *AndNode* appears to be a terminal node in the upper evidence space. As a node in the upper evidence space it can have evidence focussed directly on it (Bel_A) and it will have values for $Bel_{\Theta\downarrow}(A)$ and $Bel_{\Theta\downarrow}(\neg A)$ computed for it by its parent (i.e. Subroutines 4, 5, and 6). It will also project the combined evidence from the evidence

spaces below it into the upper evidence space, and it will project the plausibility and belief in its hypothesis down into the evidence spaces below. From the upper evidence space, the combine evidence from below the *AndNode* appears to all be focused on the *AndNode*.

The children of an *AndNode* are the roots of distinct evidence spaces but unless the *AndNode* is the root of the entire tree, the children do not represent a category of solutions equal to the entire frame of discernment for the problem. When they do not represent the entire frame of discernment they can not, as Shafer and Logan's root does, assume that their frame includes the solution; therefore Subroutine 3 can not be used to update their children. In these cases the roots of the evidence spaces must use Subroutines 4, 5, and 6 to update their children in the same manner as other disjunctive nodes. The evidence space roots must also support their *AndNode* by performing the above mentioned calculation of Bel_{A_i} .

2.5. Inferring the Solution

The RUNES tree is a concise and flexible representation for structuring and reasoning about the observable evidence but it must also have a means of deriving a belief over the hypotheses of the solution space of the problem. This is accomplished by determining a *best path* down the RUNES tree and representing it as a *Trace tree*. The best path through a disjunctive node is determined by choosing the child with the best combination of belief and plausibility.

For a RUNES conjunctive node the best path can be determined by taking the Cartesian product of the conjoined evidence spaces to produce a disjunctive space and then simply selecting the best joint hypothesis using the same methods as for the

disjunctive nodes. In practice, one would not want to completely expand the conjunction and search for a best branch because of the potentially large space that could result. The next section details efficient techniques for determining the solution by tracing a path down the RUNES tree.

2.5.1. Details of Tracing a RUNES Tree

Although the best path through a disjunctive node was described as the child with the best combination of belief and plausibility, the method used for the ranking of the choices must be considered [Kyburg87b]. Since the state of our belief in a node is expressed as an interval, there is not always an exact ordering. RUNES currently uses a simple ranking scheme based on most plausible first, reserving the beliefs for tie breaking only.

The best path through a RUNES conjunctive node can be determined by taking the Cartesian product of the plausible members of the conjoined evidence spaces to produce a disjunctive space and then simply selecting the best joint hypothesis using the same methods as for the disjunctive nodes.

The Trace tree is a dynamically generated, purely disjunctive hierarchy. As a disjunctive hierarchy, the Trace tree enjoys the ease of ranking of choices and the ability to prune unpromising subtrees without causing any difficulties with evidence combination since it is never directly involved in that process. In addition, the construction of the Trace tree can be guided by the current beliefs in the attributes as it is dynamically generated after the evidence has been combined in the RUNES tree. For example, evidential hypotheses which have an associated vacuous belief are not elaborated, since there is no information below those hypotheses. Likewise, an evidential hypothesis which is implausible need not be elaborated since none of its offspring can be

more plausible than it is.

Previous sections discussed representing the problem as a single disjunctive hierarchy by choosing a fixed order in which to elaborate the attribute classes. It was suggested that the complete and sequential elaboration of attribute classes was one way to build the tree. This is depth-first elaboration; each child in the composite hierarchy differs from its parent by the specialization of one attribute (evidential hypothesis). An alternate approach is parallel elaboration of the attributes. In this case a child differs from its parent by the specialization of all attributes which are not already evidential elements. Clearly the depth-first approach results in a tree with many more nodes. RUNES uses parallel elaboration for Trace tree construction in order to produce a very compact representation of the solution. Other methods for elaboration, including user interactive selection of the attributes to elaborate, are possible within the RUNES system. Using this technique and those mentioned in the previous paragraph, it should be possible to represent the important aspects of a large hypothesis space with a small Trace tree.

The Trace tree is composed of *RunesTraceNodes*. Each of these nodes has an associated set of evidential hypotheses or elements in the RUNES tree. The node represents the hypothesis which is the conjunct of all evidence in its evidence set. The node also has an associated category of solutions defined by the attributes in the evidence set (i.e. the intersection of the categories defined by each attribute). It is the *RunesTraceNodes* which map the beliefs from the RUNES tree to the solution space by combining the beliefs in the attributes in its evidence set and attributing the result to the category of solutions in its solution set. If the evidence spaces which the attributes are drawn from are independent, the calculation of the resulting belief or plausibility is simply the product of the beliefs or plausibilities of the attributes. However, in real problems strict independence between the evidence spaces is not to be expected and a

technique to correct for some of the correlation will be discussed later in this section.

The best solution (unique or otherwise), given the current evidence, is the category of solutions associated with the *RunesTraceNode(s)* which has the highest belief. Note that the category may contain more than one solution and if it does it signifies that either there is insufficient evidence currently available to differentiate the choices or the choices are inherently ambiguous. In the first case, the collection of additional evidence should resolve the ambiguity. In the second case, the solutions have a completely overlapping set of observable attributes. Solutions which are inherently ambiguous cannot be distinguished by any test or observation known to the system (i.e. the attribute(s) which do distinguish them are not observable by the system).

The RUNES system allows inherently ambiguous solutions to be enumerated but it does not provide any criteria to distinguish one from the other. There is an advantage to allowing the solutions to be enumerated instead of representing them as a single concept and that is that the ambiguity need not be detected during the knowledge acquisition phase. Each solution can be described individually to the system without regard to those which preceded it or those that will follow it.

Returning to the problem concerning the lack of strict independence between evidence spaces, let us first clarify what is meant by independence. Independence in this context means that no hypothesis in any evidence space has a solution category which is a subset of any solution category for any hypothesis in another evidence space (excepting the root hypothesis). In other words, the correlation of interest is taking place in the solution space of the problem. Recall that when the evidence spaces are not strictly independent the belief determined by simple multiplication may under-estimate the actual value. In order to compute the correct belief the process must consider the actual

solution categories themselves and use Dempster's Rule or an equivalent combination technique. However, the calculation must insure that all evidence is included and that it appears only once. The beliefs held by nodes in the RUNES tree are global beliefs, that is they are each the result of all of the available evidence. Since the beliefs are not based on independent bodies of evidence they cannot be simply combined by Dempster's Rule.

What is needed is a technique which can take advantage of the partial belief functions already present in the RUNES tree and thus avoid complete recombination of all evidence using Dempster's Rule (after all, the avoidance of that computation was the entire purpose of this work). The key to an efficient solution is to again apply Shafer and Logan's observations concerning factoring of the belief function.

$$\text{Bel}_{\Theta \downarrow} = \text{Bel}_{B_0} \oplus \text{Bel}_B \oplus \text{Bel}_{B \downarrow}$$

We wish to determine $\text{Bel}_{\Theta \downarrow}(B)$ and $\text{Bel}_{\Theta \downarrow}(\neg B)$ for some *RunesTraceNode* B whose parent trace node is A. Notice that:

$$(\text{Bel}_B \oplus \text{Bel}_{B \downarrow}) = \{ \oplus (\text{Bel}_e \oplus \text{Bel}_{e \downarrow}) \mid e \in \text{Evidence Set for } B \}$$

for any node, B, in the Trace tree. This allows us to determine $\text{Bel}_{\Theta \downarrow}(B)$ and $\text{Bel}_{\Theta \downarrow}(\neg B)$ for any child, B, of the root trace node; since for these nodes Bel_{B_0} , is vacuous.

For any other node, B, in the Trace tree, the values of $\text{Bel}_{\Theta \downarrow}(B)$ and $\text{Bel}_{\Theta \downarrow}(\neg B)$ can be determined once $\text{Bel}_{\Theta \downarrow}(A)$ and $\text{Bel}_{\Theta \downarrow}(\neg A)$ are known, where $B \in \mathcal{M}(A)$. The computation is performed using Shafer and Logan's Subroutines 4 and 6 given $\text{Bel}_{\Theta \downarrow}(A)$, $\text{Bel}_{\Theta \downarrow}(\neg A)$, $\text{Bel}_{A \downarrow}(A)$, $\text{Bel}_{A \downarrow}(\neg A)$ and given $\text{Bel}_{A \downarrow}(B)$, $\text{Bel}_{A \downarrow}(\neg B)$ and $\text{Bel}_{A \downarrow}(B \cup \neg A)$.

The complete belief function resulting from all of the evidence collected in the RUNES tree can also be factored from the perspective of A, the parent of B, as follows:

$$\text{Bel}_{\Theta \downarrow} = \text{Bel}_{A_0} \oplus \text{Bel}_A \oplus \text{Bel}_{A \downarrow}$$

$$= (\text{Bel}_{A_0} \oplus \text{Bel}_A)$$

$$\oplus \{ \text{Bel}_{E \downarrow} \mid E \in \text{Evidence Set for } A \}$$

$$= (\text{Bel}_{A_0} \oplus \text{Bel}_A)$$

$$\oplus [\oplus \{ \text{Bel}_{E_n \downarrow} \mid E_n \in \text{Evidence Set for } A \text{ and } e_n \in \mathcal{M}(E_n)$$

$$\text{and } e_n \in \text{Evidence Set for } B \}]$$

Notice that $\text{Bel}_{A \downarrow}(A)$, $\text{Bel}_{A \downarrow}(\neg A)$, $\text{Bel}_{A \downarrow}(B)$, $\text{Bel}_{A \downarrow}(\neg B)$ and $\text{Bel}_{A \downarrow}(B \cup \neg A)$ are all determined by $\text{Bel}_{A \downarrow}$ which could be constructed from $\oplus \{ \text{Bel}_{E \downarrow} \mid E \in \text{Evidence Set for } A \}$ except that the nodes in the RUNES tree (the E's) reduce the $\text{Bel}_{E \downarrow}$ to foci of E , $\neg E$, and Θ , which is too coarse to resolve the interaction of the children (e_n) and thus the children of A (node B in this case). However, the RUNES nodes e_n maintain sufficient data to construct a usable $\text{Bel}_{E \downarrow}$ with six focal elements e_n , $\neg e_n$, $(e_n \cap E_n)$, $(e_n \cup \neg E_n)$, $\neg E_n$ and Θ . These belief functions when combined, are able to resolve B, $\neg B$, and $(B \cup \neg A)$.

It is therefore possible to compute the belief of any trace node, without resorting to complete recomputation, given that all its ancestors back to the root of the Trace tree are computed first. However, computing the combination of all $\text{Bel}_{E_n \downarrow}$ for large evidence sets will again suffer the computational complexity of Dempster's Rule unless we deliberately restrict the growth of the belief function. As we have just seen, as few as six foci are needed to capture the essence of a belief function from the perspective of a node in the RUNES tree. Figure 2.4 illustrates how a belief function can be reduced by viewing it from the perspective of a single node. Applying this to a node, B, in the Trace

tree, the belief function Bel_{A_i} need only resolve the following six hypotheses: A , $\neg A$, B , $(\neg B \cap A)$, $(B \cup \neg A)$, and Θ . With this goal in mind all of the $\text{Bel}_{E_{n,i}}$ are combined iteratively with the resulting belief function being reduced, after each iteration, to the six discerning foci³. This approach has computational complexity that is linear in the number of evidence nodes being combined.

³The discerning hypotheses change with each iteration.

Combining Evidence Focused on or below X
with Evidence Focused on or below Y.

	Y	$\sim Y$	Θ
X	null	X	X
$\sim X$	Y	$\sim X \sim Y$	$\sim X$
Θ	Y	$\sim Y$	Θ

Including Evidence Focused on or below Z.

	Z	$\sim Z$	Θ
X	null	X	X
Y	null	Y	Y
$\sim X$	Z	$\sim X \sim Z$	$\sim X$
$\sim Y$	Z	$\sim Y \sim Z$	$\sim Y$
$\sim X \sim Y$	Z	$\sim X \sim Y \sim Z$	$\sim X \sim Y$
Θ	Z	$\sim Z$	Θ

Resulting Complete Belief Function
{all evidence below P}

P is the parent of X, Y and Z.

Figure 2.4 Belief Function Reduction

Chapter 3.

An Implementation of RUNES

RUNES was implemented in Digitalk V/286 which is a version of Smalltalk for IBM style 80286 or 80386 computers. Smalltalk is an object-oriented language which features: objects defined by classes, a superclass subclass hierarchy, inheritance of structure and procedure by subclasses, and computation by message passing. Objects are a combination of state (data) and procedures to manipulate the state. The procedures are invoked by sending a message (a symbolic request for services) to the object. Smalltalk was the origin of the concepts of windows, pop-up menus, and mouse driven interfaces.

3.1. Class Structure

The RUNES system uses the class inheritance features of Smalltalk to implement a set of modular components which can be assembled to form the RUNES And-Or hierarchy. Figure 3.1 depicts the class hierarchy of the RUNES tree classes. The *AbstractNode* class defines the basic nature of hierarchical nodes. This includes the basic state, such as: the root of my hierarchy, my parent, my children, etc. The *RunesNode* class adds the state and procedures associated with the categories of solutions in the problem domain. The *ShaferLoganRoot* and *ShaferLoganNode* classes implement the Shafer-Logan algorithms for hierarchical evidence. All of the above classes are abstractions which provide code modularity; they are never instantiated in a running system. The remaining classes are used to actually construct a RUNES tree. These classes

implement the modifications to the Shafer-Logan algorithms to support nested evidence spaces. They also implement the protocol to support the query process.

Figure 3.1: RUNES Class Hierarchy

The *TopEvidenceSpace* class is a specialization of *ShaferLoganRoot* which is responsible for implementing the consequences of the assumption that the top frame of discernment is exhaustive, that is, that the solution must be within its frame. The conceptually similar *EvidenceSpace* class is also the root of a tree (actually a subtree) but since its frame of discernment is not exhaustive it is, for the purposes of implementation, more like a *ShaferLoganNode* than a *ShaferLoganRoot*. It does however implement some procedures associated with being the root of an evidence space.

The *Hypothesis* class is very close to the original *ShaferLoganNode* in that it represents a simple disjunction of possibilities and the *EvidentialElement* class is simply a version of *ShaferLoganNode* which is optimized for the case where it has no children.

The *AndNode* class has a dual role and its position in the class hierarchy reflects its secondary role as evidential element in some upper evidence space. In this role it behaves much like a *Hypothesis* or an *EvidentialElement*. It also implements its primary role as conjunctive operator over the evidence spaces below it.

Figure 3.2 illustrates the class structure for the browsers and classifiers used in RUNES. In Smalltalk a browser is a generic name for a multipaned window which provides a graphical user interface to some application. In this case the browsers provide the user interfaces for the tree parameter library, the Trace tree, and the RUNES tree.

Figure 3.2: RUNES Classifier and Browser Classes

The *TreeEditor* provides the interface for the parameter library while the Trace tree interface is implemented by the *RunesPathBrowser*. The RUNES tree has two

custom browsers designed for each of the two problem domains tested. The *SimpleClassifier* is used for the simple example problem while the *TreeClassifier* is used for the tree classification problem. The remaining classes provide state and procedures which are shared by their subclasses.

3.2. Realization of AND-OR Trees

The top root of a RUNES tree must be an instance of either *TopEvidenceSpace* or *AndNode*. If the top root is an *AndNode* then all members of its *ℳ* must be instances of *TopEvidenceSpace*. Instances of *TopEvidenceSpace* cannot appear elsewhere in the RUNES tree since their solution category must equal the entire frame of discernment for the problem. The *ℳ* of a *TopEvidenceSpace* can consist of a mixture of instances of *Hypothesis*, *EvidentialElement*, or *AndNode*.

The remainder of a RUNES tree is composed of instances of *EvidenceSpace*, *Hypothesis*, *EvidentialElement*, and *AndNode*. Instances of *EvidenceSpace* can have any of the other three types in its *ℳ*. Instances of *Hypothesis* can have *Hypothesis*, *EvidentialElement* or *AndNode* types in its *ℳ*. Instances of *AndNode* can have only *EvidenceSpace* types in its *ℳ* (unless it's the top root), and instances of *EvidentialElement* have no children.

This may appear unduly complex but they are merely implementation details which need not be visible to the user, nor for that matter to the knowledge engineer. The current implementation of RUNES does not require that the terminal nodes of the tree be declared as *EvidentialElements* when they are added to the tree. Instead they are added as *Hypothesis* nodes and only when the tree is complete are the terminal nodes converted to *EvidentialElements*. It is certainly feasible to allow the tree to be constructed from

generic conjunctive and disjunctive nodes with the conversion to RUNES nodes performed automatically by the system after the tree is complete.

3.3. The Classifier

The 'Classifier' is a basic Smalltalk browser consisting of three panes. Figure 3.3 shows the Classifier displaying the RUNES tree for the Simple Example problem (Figure 2.4). The large upper-most pane provides a list of the nodes in the RUNES tree. The list depicts the hierarchical organization by placing children directly after their parent and indenting the entries proportional to their depth in the hierarchy. Conjunctive nodes are marked by the suffix <AND>, while the roots of evidence spaces are marked by the suffix <OR>. Any of the listed nodes can be selected by 'clicking' on them with the mouse. A selected entry will display a list of its component names in the lower left pane. Selecting one of these names will cause the data associated with that component to be described in the lower right pane. In Figure 3.3 the root of the RUNES tree is selected as well as the solutionSet component and the lower right pane displays the hypothesis which is also the frame of discernment for this example problem.

There is support for large trees, in that the Classifier can hide all nodes which do not have <AND> or <OR> suffixes. It can also move down the tree using the currently selected node as the root of a subtree for display and, of course, it can backup along the same route.

There are 'pop-up' action menus associated with each of the panes which allow the user to enter evidence, invoke the query mode of operation, run a fixed set of tests, open a 'best path' browser, reset the state of the tree, etc. Manual entry of evidence is performed by selecting, in the upper pane, the node which will be the focus of the

Browsing: Example_Tree	
Example_Tree(0) <OR>	
1_Dimensional(1) MultiDimensional(2) <AND> Transparency(3) <OR> Transparent(4) Opaque(5) Dimensionality(6) <OR> 2_Dimensional(7) Rectangle(8) Circle(9) 3_Dimensional(10) <AND> 3D_Shape(11) <OR> Sphere(12) Cube(13) 3D_Interior(14) <OR> Solid(15) Hollow(16)	
name	HypSet{255 31 }
solutionSet	(Thread, SolidGlassBall, SoapBubble, BowlingBall,
nonspecificityNorm	Balloon, SugarCube, CardboardBox, PostageStamp,
confusionNorm	WindowPane, Aquarium, Coaster, GlassPaperWeight,
dissonanceNorm	GlassDisk,)
nonspecificity	
confusion	

Figure 3.3: The Classifier

evidence, then selecting, from the menu, a weight of evidence for or against that node. The menu provides four weights of evidence: strong, moderate, weak and very weak corresponding to values of 0.9, 0.6, 0.3 and 0.1 respectively. The classifier also supports a rudimentary query mode which prompts the user for evidence. The query algorithm is described in the following section.

The effect of the evidence can be viewed by selecting a node in the upper pane and selecting the name of one of its belief components in the lower left pane which will cause its current value to appear in the lower right pane. If the user wishes to see the state of the diagnosis in terms of a solution, then selecting 'browse best path' from the menu will cause the system to compute a trace tree and open a 'Best Path' browser to view it. The Best Path browser will be described in a subsequent section.

3.4. Query Generation Algorithm

The bulk of the RUNES system is associated with the maintenance of a belief function over an entire collection of evidence. It can produce a belief value and a plausibility value for any subset of evidential attributes, thereby inferring the beliefs for the subset of the solutions which have those attribute values. Obviously this is not a complete expert system. A simple query generation capability was provided to allow the examination of heuristics for inference control.

In a hierarchically organized hypothesis space, query generation often involves choosing the best path from the root down into the tree until there is insufficient evidence to choose the next branch; a query can then be posed based on the choices at that point. Shafer suggested that a belief value (i.e. support) greater than 0.5 was a good heuristic for selecting the best path in a disjunctive hierarchy since any node meeting this criterion

was guaranteed to exceed the values of all its siblings and indeed an hypothesis formed from its siblings [Shafer&Logan87b]. However when no single node has a belief greater than 0.5 and a choice is required, then selection on the basis of highest plausibility is reasonable. Notice that whenever a node does have a support greater than 0.5 it will also always have the greatest plausibility, suggesting perhaps that plausibility is slightly more general as a heuristic.

Let us now consider how a simple query generation can be implemented for a RUNES tree. The disjunctive nodes can be handled in either manner discussed above. The implemented version of RUNES uses plausibility to rank sibling nodes and if the highest plausibility node has a low belief value then a query over these siblings is attempted. The query is not guaranteed as the query system has a simple technique to avoid asking the same question repeatedly, and this query may have been posed once already.

The conjunctive nodes in the RUNES tree add a new dimension to the query process and the heuristics mentioned above do not apply to these nodes. Unlike a disjunctive node where the solution lies below one, and only one, of its children; the solution lies below *all* of the children of a conjunctive node. The query generator will most likely visit all of the children before the problem is completely solved but the order it uses can affect how quickly the system converges onto a short list of potential solutions¹. In RUNES, all of the children of a conjunctive node are disjunctive nodes which form the root of an evidence space. Being disjunctive nodes the standard plausibility heuristic can be used within each evidence space resulting in one candidate

¹ If there are more conjoined evidence spaces than is necessary to uniquely specify a solution or the evidence spaces are not strictly independent then all spaces need not be visited. In these cases intelligent selection of the spaces to query can reduce the number of queries required to reach an answer.

query for each child of the conjunctive node.

The RUNES system considered two methods for selecting a query from potential queries available from a conjunctive node. The first method is based on normalized² uncertainty measures for each evidence space. The uncertainty measures: *Nonspecificity*, *Dissonance* and *Confusion*; each provide a single scalar which corresponds to a different aspect of the uncertainty in the belief function over a hypothesis space. A normalized Nonspecificity with a value near one indicates that most of the belief is focused on hypotheses of high cardinality (i.e. general concepts not terminal values). A high value for the normalized Confusion indicates that the belief is widely dispersed across many focal hypotheses (perhaps conflicting hypotheses). When the normalized Dissonance approaches one, most of the belief is focused on many incompatible hypotheses.

The uncertainty measure heuristic considered for RUNES, although not extensively tested, was to use high values for the sum of the normalized Nonspecificity and Confusion to indicate the need for more evidence gathering. Basically the evidence space with the largest sum won the right to pose the next query. The normalized Dissonance was not used although high values could be used to either suggest more evidence be gathered to break the deadlock, or the existing evidence in that space could be re-examined or discounted.

The second approach to query selection was based on maximum discernment of the current list of best solutions. Using the standard plausibility and support heuristic for disjunctive nodes, choose the best evidential node (as deep as the available information allows) in each of the conjoined evidence spaces and intersect their associated solution

² The uncertainty measures: Nonspecificity, Dissonance and Confusion; are easily normalized into the range of zero to one, given that the structure of the hypothesis space is static.

space categories to produce a current list of best solutions. The queries are then ranked according to how well they *discern* the list of best solutions. A query which has responses able to distinguish each member of the list uniquely is ranked ahead of a query which only discerns subsets of the list.

The current implementation of RUNES has a rudimentary notion of a query which is represented as a node from the RUNES tree. The node can be any disjunctive node and a valid response to the query is evidence for or against one of its children, or a response indicating no information available. Given this query structure, the maximum discernment is achieved when there is never more than one member of the best solutions list associated with any individual child of the query node.

RUNES implements a simple discernment rating algorithm which returns unnormalized numerical values wherein a lower value indicates better discernment. The algorithm can be summarized as follows:

If (query node is a terminal node)

Then (return worst rating).

queryIntersection := solution category of query node $\cap$ current best solutions.

If (queryIntersection = $\emptyset$)

Then (return worst rating).

queryMatch := (cardinality of current best solutions) / (cardinality of queryIntersection).

For (all children of query node)

Do (card_n := cardinality of (solution category of child $\cap$ current best solutions)).

aveCard := (average of non-zero card_n).

maxCard := (maximum value of card_n).

discernmentRating := (aveCard + maxCard) * queryMatch

The value, *queryMatch*, rates the general appropriateness of the query to the list of current best solutions. Then all of the children of the query node are evaluated for their ability to discern individual members of the best solutions list. Finally an overall rating is produced.

Early versions of the RUNES system used the uncertainty measures heuristic for query selection but later versions switched to the discernment heuristic. The uncertainty technique is less goal oriented in that it tries to maintain a uniform balance of uncertainty (or information) between all evidence spaces and the state of the solution is a side effect. Whenever one evidence space advances its understanding (i.e. reduces its uncertainty) the query generator switches its focus to a less resolved evidence space. This basically produces a breadth first query generator. The later versions of the query generator, with the discernment heuristic, also shifted their focus frequently but in this case it was solution oriented and the shift represented an attempt to capitalize on the most discriminating query available. Example transcripts of query sessions have been included in Appendix C.

3.5. The Best Path Browser

The 'Best Path' browser is a two pane Smalltalk browser where the upper pane is a list of nodes depicting the hierarchy of the *Trace tree* (Figure 3.4). Recall that the Trace tree is a purely disjunctive hierarchy constructed from the RUNES tree. Each child represents the refining of the evidential attributes of its parent. All paths (i.e. all combinations of evidential attributes) with a plausibility greater than a selectable minimum plausibility are displayed. The system's first choice for the solution is the terminal node of the first displayed path. In Figure 3.4 the first choice, "Opaque &

Hollow & Cube", is the shown by the selection bar. Alternate plausible solutions, "Opaque & Solid & Cube", etc.; are shown as other branches. The lower the plausibility of a solution, the lower it appears on the list. Attributes with the "_" suffix indicate evidence space root nodes. The presence of these nodes in the trace instead of an evidential element or hypothesis indicates that there is no significant information within their space.

Selecting one of the nodes in the trace will cause a text description of that node to appear in the lower pane. The description contains the belief and plausibility of the set of solutions which it represents along with a short description of the evidential attributes which were conjoined to form this node. In the case of "Opaque & Hollow & Cube" in Figure 3.4, the set of solutions has a single member, "CardboardBox" and a plausibility of 1.0 indicating that no other solution has any degree of support from the evidence (although as indicated in the upper list, several are still plausible to some degree).

Figure 3.4: The Best Path Browser

3.6. Example Problem Domains

The RUNES implementation was tested on two example problem domains. The first problem domain was the simple example used previously to demonstrate the technique for constructing a RUNES tree. The problem has thirteen solutions, common objects, in its frame of discernment. It has five distinct evidence spaces containing nine evidential elements and three evidential hypotheses for a total of seventeen nodes in the tree, Figure 2.4. It has the ability to represent 27 intermediate hypotheses not including the entire frame and the solutions themselves.

This simple problem serves as a benchmark for verifying correct operation of the algorithms. It has no inherently ambiguous solutions, no solutions with missing attributes, and the evidence spaces are independent with respect to the solution space.

The second example problem domain is a real problem, the classification of trees and shrubs using observable evidence concerning their leaf and fruit structure. The Peterson Field Guide Series: *A Guide to Trees and Shrubs* by G.A. Petrides, served as the source of expertise for this problem. The book provides many decision trees which use observable features to first group trees into categories with common basic attributes and then attempts to distinguish between the members of a category using other features.

Figures 3.5 - 3.16 depict the RUNES evidence spaces which were used to represent a portion of the knowledge in the field guide. Nodes which are joined by disjunction are connected by a single vertical bar whereas those which are conjoined have a double vertical bar connecting them. As previously mentioned, the *AndNodes* mark themselves by appending <AND> to their names and the disjunctive root nodes append <OR> to their names.

Figure 3.5: The Tree Classification Problem

Figure 3.6: Needle_Leaf <AND> Subtree

Figure 3.7: Cone_Fruit <AND> Subtree

Figure 3.8: Broad_Leaf <AND> Subtree

Figure 3.9: Once_Compound <AND> Subtree

Figure 3.10: BL_Attachment <OR> Subtree

Figure 3.11: BL_Body_Shape <OR> Subtree

Figure 3.12: BL_Tip_Shape <OR> Subtree

Figure 3.13: BL_Base_Shape <OR> Subtree

Figure 3.14: BL_Edge <OR> Subtree

Figure 3.15: BL_Base_Symmetry <OR> Subtree

Figure 3.16: BL_Lobing <OR> Subtree

Figure 3.17 shows the Classifier browsing a portion of the implemented RUNES tree. The Classifier shows the evidential element, "Flat_Needle(5)", has been selected along with the component, "solutionSet", and the lower right pane displays the associated solution space category. The implemented leaf structure attributes represent about one half of the available characteristics and the implemented fruit structure contains only basic characteristics. The structure of many additional attributes such as: twig, branch, bud and bark characteristics were examined but were not included in the problem implementation.

Unlike the simple problem this one has all of the features which make real problems difficult to implement. As mentioned above, many of the attributes used in the guide were not encoded in the implementation of this problem and as a result, inherently

Browsing: Tree	
Tree(0) <AND> Leaf_Category(1) <OR> Scale_Like_Leaf(2) Needle_Leaf(3) <AND> Needle_Shape(4) <OR> Flat_Needle(5) Needle_Not_Flat(6) 3_Sided_Needle(7) 4_Sided_Needle(8) Needle_Attachment(9) <OR> Needles_Bunched(10) Needles_In_Bundles(11) 2_Needles_Per_Bundle(12) 3_Needles_Per_Bundle(13) 5_Needles_Per_Bundle(14) Needles_On_A_Spur(15) Needles_Not_Bunched(16) Needles_On_Stalks(17) Needles_Not_On_Stalks(18) Broad_Leaf(19) <AND> BL_Type(20) <OR> BL_Type_1(21) BL_Type_2(22)	
name	HypSet{64 24 0 0 0 64 8 0 0 0 0 0 0 0 0 0 0 }
solutionSet	(AmericanYew, Baldcypress, BalsamFir,
support	(EasternHemlock, FraserFir,)
localSupport	
plausibility	
localPlausibility	
bel	

3.17 Classifier Showing Tree Classification Problem

ambiguous solutions were not uncommon. The guide book is not a catalog of features, but rather an algorithm for classifying trees and as such it did not include attribute values for all characteristics of all trees, simply those required at that point in their algorithm. This problem also has a significant amount of correlation between attribute classes (eg. cone fruit implies not broad leaves). In addition several species of trees had multiple possible attribute values for the same attribute type (eg. an evergreen tree with most bundles having three needles but some with two needles). The attributes with continuous values also created difficulty as they had to be represented as discrete values. Finally there is a potential problem not with representation but with usage by non-experts. Several of the attributes make rather subtle distinctions, for example: oblong versus elliptical, or triangular versus egg shaped, seem like reasonably different types for leaf body shapes until one tries to describe a maple leaf or an oak leaf in these terms. One is asked to ignore the deep lobing, the tip shape and the base shape (all of which are handled by other evidence spaces) and describe the body of the leaf in terms of an ideal shape. Clearly novice users, without the benefit of convention or experience, may find this a difficult task.

Of particular difficulty was the hierarchical representation of characteristics with continuous values (eg. the typical length of the leaf or size of the fruit) or those with a range of discrete values (eg. the number of leaflets on a compound leaf). In both cases the difficulty was not a matter of producing a discrete range of values but one of representing hypotheses over those values. Consider the evidence space "Once_Compound_Degree", Figure 3.9, which discerns the number of leaflets on a compound leaf. There are many examples of once compound leaves with two or three leaflets; five leaflets is also common, and higher degrees of compounding usually prefer an odd number of leaflets. The difficulty lies in the various degrees of compounding

Figure 3.18 Degree of Leaf Compounding

possible for a single type of tree. For example, the Blue Ash can have 7 to 11 leaflets, the American Mountain Ash can have 11 to 17 leaflets and the Black Walnut can have 7 to 17 leaflets. Figure 3.18 illustrates the hypothesis space required to properly represent these trees and it is obviously not the simple disjunctive tree that RUNES evidence spaces expect.

The current implementation compromises by choosing a single disjunctive hierarchy (Figure 3.8) and allowing trees to enumerate multiple attribute values within a single attribute class. For example, in order to cover the range from 7 to 17, the Black Walnut would list both "Leaf_Count_6_to_11" and "Leaf_Count_GreaterThan_11" as its attribute values in the evidence space, "Once_Compound_Degree". The rationale for this decision was that in most of these cases the variation seemed to reflect differences between instances of the same type of tree not variations within a single instance

(although that occasionally occurred as well). Thus it was reasonable to assume that given any particular instance only one observation value would be produced (therefore no dissonance is generated in the evidence space) and the correct tree would be in the associated solution category for whichever value was chosen. This approach has proved workable but is clearly a somewhat undesirable approximation as it undermines the assumed disjunction of the associated solution categories.

With respect to the potential impact on the accuracy of the belief calculation, the two most significant difficulties of the tree classification problem were: incomplete a priori knowledge about tree attributes and the lack of independence between the evidence spaces with respect to the solution space. The pragmatic solution to the first problem has to represent the attribute of the solution by a general evidential hypothesis in place of an evidential element when the precise value for that attribute was unknown. This resulted in the solution being a plausible proposition for any of the evidential elements in the subtree of that evidential hypothesis. Although the cause is different, the representation with its associated deficiencies is analogous to the situation for compound leaflets, described above.

An indication of the degree of correlation between evidence spaces with respect to the solution space can be obtained by examining the number of terminal solutions represented by the RUNES tree and the number of tree types that it was designed to represent. The entire RUNES tree for the tree classification problem has 17 evidence spaces, 68 evidential elements, and represents 723 million hypotheses of which 48,438,437 are terminal hypotheses, that is they represent all the possible combinations of evidential elements. Clearly there are more terminal hypotheses possible than there are types of trees known to the system (142 tree types in this implementation) and indeed there may not be that many species of trees documented anywhere. Many if not most of

these hypotheses correspond to combinations of attributes which are physically plausible but unrealized (eg. a tree with broad leaves and cone fruit). This is a direct manifestation of the correlation between the evidence spaces. The correlation between the evidence spaces makes the computation of the correct belief and plausibility of the solution difficult (Refer to Section 3.4) but as the results will show it had little effect on finding the correct solution.

3.7. The Tree Editor

The data entry task for the attribute descriptions of the trees in the tree classification problem prompted the creation of the 'Tree Editor', Figure 3.19 , which is a variation of the Classifier browser. The Tree Editor has three panes with the top pane displaying the RUNES tree structure, but the bottom left pane displays a list of all tree names known to the system. The bottom right pane shows the observable attributes of the tree whose name is currently selected (lower left pane). The user of the editor can select an evidential element or hypothesis from the RUNES hierarchy in the top pane and associate it with the tree name selected in the lower left pane. If an evidential hypothesis is used then all values which it represents are added to the selected tree. The user also has the option to remove any attribute previously assigned to the selected tree.

The RUNES implementation allows for the solutions to be held separately from the evidence space structure and loaded prior to use. The Tree Editor uses this feature to store the library of trees and attributes separately from the evidence space structure. This separation of solutions and their attributes from the structure of the hypothesis spaces over those attributes has facilitated changes to the structure of the hypothesis spaces without requiring modification of the tree library (as long as evidential attributes are neither deleted nor renamed).

Editing: Tree

Tree(0) <AND>

- Leaf_Category(1) <OR>
 - Scale_Like_Leaf(2)
 - Needle_Leaf(3) <AND>
 - Needle_Shape(4) <OR>
 - Flat_Needle(5)
 - Needle_Not_Flat(6)
 - 3_Sided_Needle(7)
 - 4_Sided_Needle(8)
 - Needle_Attachment(9) <OR>
 - Needles_Bunched(10)
 - Needles_In_Bundles(11)
 - 2_Needles_Per_Bundle(12)
 - 3_Needles_Per_Bundle(13)
 - 5_Needles_Per_Bundle(14)
 - Needles_On_A_Spur(15)
 - Needles_Not_Bunched(16)
 - Needles_On_Stalks(17)
 - Needles_Not_On_Stalks(18)
 - Broad_Leaf(19) <AND>
 - BL_Type(20) <OR>
 - ...

AtlanticWhiteCedar	Set(
AutumnWillow	'Needles_Not_On_Stalks'
Baldcypress	'Cone_Scales_Without_Horns'
Balsampine	'Cone_Scales_Not_Stiff'
BalsamPoplar	'Many_Cone_Scales'
BaltimoreAsh	'Flat_Needle').
BasketWillow	

Figure 3.19: The Tree Attribute Editor

Chapter 4.

Analysis and Results

4.1. Testing Methodology

The testing of the implemented RUNES system was divided into four segments focussing on: accuracy of the belief function computation, execution speed of the algorithm, effectiveness in use with a real problem, and lastly preliminary testing of heuristics for query generation.

The accuracy tests consisted of comparing the belief function computed by the RUNES system with the belief function calculated using Dempster's Rule in an unrestricted hypothesis space. Please note that these tests are to determine the accuracy of the belief calculation *for those hypotheses that the tested RUNES structure represents*. In other words, it is assumed that the structure as built, captures the essential hypotheses of the problem domain and that it is sufficient to determine the belief and the plausibility of these hypotheses. Clearly there will be many hypotheses which the structure does not represent and which may be present in the unrestricted hypothesis space, but this does not constitute incorrect behavior. If each hypothesis in the RUNES tree has the same belief and the same plausibility as that hypothesis has in the unrestricted hypothesis space, then the system is deemed to be functioning correctly.

For the accuracy tests only the simple problem domain example was used so that the question of computational correctness was not clouded by the issues of independence of evidence spaces or incomplete a priori information.

The speed of execution was tested to verify that the RUNES algorithm was able to achieve a computational complexity that was linear in the number of nodes. For the purposes of comparison, the Dempster's Rule computation in an unrestricted hypothesis space was also tested. It must be stressed that neither the RUNES algorithm nor, in particular, the Dempster's Rule algorithm were optimized for speed, as the objective of these tests was only to establish the order of complexity of these algorithms.

The third set of tests was designed to evaluate the effectiveness of the RUNES system approach when used in a real problem domain. The problem domain was the tree classification problem and the "use" of the system was simulated. Testing involved the generation of a forest with a prescribed distribution of tree species randomly distributed within. A test sample of 100 trees was picked at random from the forest and identified by the RUNES tree classifier. The primary performance measures were:

- 1) the ranking of the correct solution category.
- 2) the cardinality of the correct solution category.
- 3) the plausibility difference between the correct solution category and its closest rival.

where the "correct solution category" means that category which contains the tree type which is actually being observed. The cardinality of the solution category is an important measure as the RUNES system does not provide any formal method for ranking the members of the solution category. The extreme example of this is the solution category containing all possible trees, technically this is a correct answer in that it contains the

solution, but it is of no utility. The measure of the plausibility difference is included to assess how pronounced the win or loss was.

A limited amount of post testing analysis was performed on the results of the effectiveness tests to evaluate the benefits of including an a priori probability distribution for the types of trees in the forest. The use of the a priori probabilities was restricted to removing implausible types from the solution category and ranking the remaining types (within the category only).

The last tests involved observing the behavior of the query generation subsystem. The purpose of these tests was to demonstrate that the RUNES tree could be used as a guide for structured query generation and that simple measures could be provided by the RUNES tree to focus the query process.

4.2. Simple Problem Domain

The computation of the belief function over the RUNES tree and the Trace tree were both verified by a series of accuracy tests wherein the belief and plausibility values of hypotheses in these trees were compared to those for the same hypotheses in a belief function (referred to as the control belief function) computed by Dempster's Rule. Three sets of evidential data were used: the first was consonant supporting evidence, the second was consonant refuting evidence and the third was random evidence. Each solution from the frame of discernment of the problem was tested with each type of evidence.

Testing proceeded as follows: first a member of the frame of discernment was selected and a collection of evidence was simulated for it, then the evidence was added one piece at a time to both the RUNES tree and the control belief function, and finally after each piece of evidence was added a Trace tree was constructed and the belief and

plausibility of every node in the RUNES tree and the Trace tree were compared to those in the control belief function.

The results from all three evidence types and over all solutions were the same; no detectable difference between the beliefs and plausibilities of the RUNES or Trace trees and those of the control belief function. The belief computation performed by RUNES was equivalent to Dempster's Rule.

4.3. Realistic Problem Domain

The results of testing the RUNES implementation of a classifier for trees containing 17 attribute classes and descriptions of 142 tree types were very good. The time to update the belief function over the RUNES tree is independent of the amount of evidence accumulated, unlike Dempster's Rule which requires exponentially increasing processing time. Throughout the testing RUNES usually ranked the hypothesis, containing the actual solution, either first or second. The RUNES implementation also proved to be very robust in the presence of erroneous evidence, demonstrating a linear decline in classification performance as the amount of misinformation increased.

The effect of the computational complexity of Dempster's Rule in an unrestricted hypothesis space is clearly evident in Figure 4.1. The time required to update the basic probability assignment is growing exponentially with the addition of new evidence. The term new evidence is used to indicate that the evidence being added has a different focal hypothesis than any piece of evidence already combined by the belief function. For this test all evidence is consonant supporting evidence, meaning that there is no disagreement in the evidence and all evidence is represented as simple support functions focussed on single nodes in the RUNES tree (plus Θ). The rapidly increasing processing requirements

Figure 4.1: Update Times for Dempster's Rule

Figure 4.2: Growth in Hypotheses with Dempster's Rule

are a direct reflection of the exponential growth rate for focal hypotheses in the belief function, Figure 4.2. As each piece of evidence supporting a new focal hypotheses is added to the existing belief function the number of hypotheses approximately doubles¹. This is as expected since the evidence consists of two focal hypotheses: one being Θ which does not change the existing set of focal hypotheses when the intersection is performed, and the second focal element is a hypothesis not already represented in the belief function so its intersection with the set of focal hypotheses creates a new set of hypotheses². Both Figure 4.1 and 4.2 represent test runs which were terminated whenever the number of focal hypotheses exceeded 512. Most runs had to be terminated in this manner otherwise the number of focal hypotheses increased into the several thousands and the computer ran out of available memory.

The time required to compute the belief function for the RUNES tree as a function of the accumulation of evidence is shown in Figure 4.3. This test used exactly the same consonant supporting evidence as was used for the test of Dempster's Rule. The RUNES tree results however show no correlation between processing time and the amount of evidence already accumulated. This result is due to the fixed set of allowed focal hypotheses, which are the nodes of the RUNES tree itself. Note that the horizontal lines in the data are an artifact of the system clock in the computer which produces minimum timing steps of 50 or 60 ms. The required processing time remains uncorrelated to the accumulation of evidence when refuting consonant support functions are instead used to represent the evidence, Figure 4.4. Refuting consonant evidence refers to evidence which is nonconflicting but whose foci are some aspect of the negation

¹ The number of focal hypotheses would exactly double if the evidence spaces of this problem were strictly independent.

² Since the evidence is consonant, none of the intersections result in the null set.

of the solution. For example, if the solution had the attribute value, *red*, for the attribute class, *colour*, and the remaining possible values were *blue* and *green*; then consonant refuting evidence would refute blue with one piece of evidence and refute green with another piece of evidence.

Figure 4.3: Update Times for RUNES with Supporting Evidence

Figure 4.4: Update Times for RUNES with Refuting Evidence

The implemented RUNES system uses deferred computation to improve its performance by only computing those values required to maintain the state of the belief function plus any values requested. When new evidence is added the entire tree is notified that previous belief values are no longer valid, and the contribution of the new evidence is propagated up the tree to the root. The downward propagation of normalized final beliefs is deferred until a node is requested to state its belief or plausibility. At that time the beliefs of all ancestors of that node, and the sibling nodes of those nodes, is calculated. In the next two tests the node which is the focus of the evidence is asked for its belief after the evidence was added, forcing the computation of beliefs for all ancestors of the focal node (and all of their siblings). Figure 4.5, illustrates that for consonant confirming evidence the required processing time is a linear function of the depth of the node in the RUNES hierarchy. Figure 4.6 demonstrates that this is also true for refuting consonant evidence.

As mentioned in preceding chapters, inferring a solution involves combining the evidence in the RUNES tree and then finding the best path down the tree. The best path, as well as its most plausible alternatives, are represented by the creation of a Trace tree. Figure 4.7 shows the processing time required to create a Trace tree versus the depth in the RUNES tree of the focal node for the last evidential update. In this test the Trace tree was calculated after each piece of evidence was added. Clearly the required processing time tends to increase with evidence focused deeper in the RUNES tree, but the data does not distinguish the nature of the function relating them. The clustering of data points into two groups is caused by the difference in complexity of the subtrees representing evergreens, the lower cluster, and broad leafed trees, the upper cluster.

Figure 4.5: Update Times versus Depth of Supporting Evidence in RUNES Tree

Figure 4.6: Update Times versus Depth of Refuting Evidence in the RUNES Tree

Figure 4.7: Time to Trace the RUNES Tree

Since the focal hypotheses of the Trace tree are not predetermined nor are they expressly limited in number, a test was conducted to investigate the growth of the number of Trace tree hypotheses as a function of the accumulation of new evidence, Figure 4.8. There is clearly a trend toward larger numbers of hypotheses as the amount of evidence increases, but again the data are inconclusive as to the exact relationship between them. The expectation is that for consonant evidence the number of hypotheses increases monotonically with the increase in the amount of evidence but that the rate of growth decreases. The reason for this speculation is that the Trace tree refines all conjoined attributes in parallel therefore the tree can accommodate many new pieces of evidence with the addition of a single node to the bottom of the tree. Then, assuming there are a modest number of conjunctive nodes scattered throughout the RUNES tree, the rate of growth of the Trace tree should decline as it will utilize more parallelism as it becomes deeper.

Figure 4.8: Growth of Hypotheses in the Trace Tree

2 of/de 2

MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS
STANDARD REFERENCE MATERIAL 1010a
(ANSI and ISO TEST CHART No. 2)

The preceding performance tests establish that the RUNES system did achieve its goal of high computational performance when compared to the use of Dempster's Rule. The next sequence of tests will demonstrate the effectiveness of the RUNES implementation for the tree classification problem. In the tests, the system is measured under simulated use by a non-expert.

First a forest of 1600 trees was generated with a predetermined probability distribution for the tree types. Then a random sample of 100 trees was selected for the subsequent test runs. Table 4.1 lists the probability distribution for the tree types in the entire forest and the distribution of types in the random sample. The trees represent types which are common to upland forests in Ontario. The White and Red Pines are a popular choice for reforestation and the distributions were intended to simulate this activity. The skewed distribution was deliberate, since one objective was to test the usefulness of prior probability information and a near uniform distribution would render the test uninformative. Note that although there were only ten tree types in the sample and in the forest, the Tree Classifier was using all 142 possible tree types when it processed them.

For all effectiveness tests, the evidence was divided into three categories: basic evidence, detailed evidence and erroneous evidence. The basic evidence represents gross attribute characteristics typical of nodes high in the RUNES tree (i.e. near the root). Examples of basic evidence are: *broad leaves* versus *needles*, or *cone fruit* versus *berries*. It was assumed that even a non-expert would have little trouble correctly identifying these attributes, therefore they never appear as erroneous data and they have relatively strong levels of support (i.e. random selection from 0.5 to 0.95). Detailed evidence consisted of any correct attribute of the solution not included in the basic evidence. Examples of detailed evidence are: *oblong* versus *elliptical* leaf body shape, or cone scales with *stout horns* versus *weak horns*. Since the detailed evidence represents subtle

distinctions and judgement calls by the user, its strength of evidence was only weak to moderate (i.e. random selection from 0.1 to 0.4). Erroneous evidence has assumed to result from misidentification of detailed evidence. It was simulated by choosing a node which was the focus of a piece of detailed evidence (eg. oblong) and then selecting one of its sibling nodes as the focus of the erroneous evidence (eg. elliptical). The strength of the erroneous evidence varied with the test and will be identified in the test descriptions.

Tree Type	Percent of Forest	Percent of Sample
White Pine	50	38
Red Pine	20	25
Jack Pine	5	2
Sugar Maple	5	8
Black Maple	5	8
Quaking Aspen	5	4
Bigtooth Aspen	5	9
American White Birch	2	2
White Ash	2	1
Heartleaf Willow	1	3

Table 4.1: Distribution of Tree Types in the Forest and the Sample

There were eight separate test runs performed under two sets of test conditions, to measure the effectiveness of the RUNES implementation. The eight runs consisted of seven with fixed strengths for the erroneous evidence and one with random strengths for

the erroneous evidence. The fixed values ranged from 0.0 to 0.6 in steps of 0.1. The random strengths were uniformly distributed in the range 0.1 to 0.4 (designated by R^* in the tables). The first set of test conditions had all basic and all detailed evidence included with the strengths randomly selected from the ranges previously described. The erroneous evidence was then added to it. The results from all 100 trees in the forest sample, for the eight tests with erroneous evidence, are summarized in Table 4.2.

The *Survival Rate* indicates the percentage of the trees from the sample which had the correct solution represented by some node somewhere in the Trace tree. The threshold of the Trace tree was set to include any hypothesis with a plausibility greater than 0.1, therefore in the cases where the solution did not survive, its plausibility was below 0.1.

The next three rows contain the average, the minimum, and the maximum *Plausibility Difference* for the node from the Trace tree which contains the solution (hereafter referred to as the solution node). The plausibility difference is the plausibility of the solution node less the plausibility of the hypothesis which is its chief rival. When the solution node has the highest plausibility, the chief rival is the node in second place and the difference is a positive number. However, if the solution node is not the most plausible then the node which is the most plausible, is the chief rival and the difference will be a negative number. The plausibility difference from the first set of tests is shown in graphical form in Figure 4.9. The dashed line in the graph indicates zero plausibility difference, or the point at which the solution node and its rival have equal plausibilities. Above the dashed line the solution node was the first choice and below the dashed line there was one or more hypotheses with higher plausibilities. The solid line, indicating a linear best fit for the data points, demonstrates that RUNES is robust in the presence of erroneous evidence. Each vertically oriented cluster of 100 points is one pass through the

entire sample of trees. Note that only the strength placed on each erroneous piece of evidence is varied between runs thus the internal structure of the clusters is very similar.

All subsequent rows of the table are compiled from only those cases where the correct solution was represented somewhere in the Trace tree. The rows five through seven in the table list the average, the minimum, and the maximum *Ranking* of the solution node. Recall that RUNES ranks choices by decreasing plausibility and only uses belief values to rank choices of equal plausibility. The results show that the average ranking was never worse than 1.48 and the survival rate was 0.97 or better. This means that 97% of the time the solution node was in the Trace tree and usually it was ranked either first or second.

The last three rows of the table contain the average, the minimum, and the maximum cardinality of the solution set for the solution node. Recall that the Trace node has a set of evidence attributes which define a category of solutions in the solution space of the problem. The category of solutions may contain more than a single member depending on the evidence collected and the inherent ambiguity of the solutions. The results in Table 4.2 demonstrate that the addition of erroneous evidence to an otherwise complete description of the problem has little effect on the cardinality of the solution set. The erroneous evidence affects our belief in the solution set by introducing rival hypotheses but its effects within a solution set are weak.

Measures	Strength of Erroneous Evidence							
	0.0	0.1	0.2	0.3	0.4	0.5	0.6	R*
Survival Rate	1.0	1.0	1.0	1.0	1.0	0.97	0.97	1.0
Average Plausibility Difference	0.98	0.55	0.45	0.37	0.31	0.3	0.37	0.42
Minimum Plausibility Difference	0.0	0.0	0.0	-0.04	-0.06	-0.11	-0.15	-0.09
Maximum Plausibility Difference	1.0	0.93	0.86	0.79	0.7	1.0	1.0	0.9
Average Ranking	1.0	1.0	1.07	1.38	1.48	1.39	1.28	1.24
Minimum Ranking	1	1	1	1	1	1	1	1
Maximum Ranking	1	1	2	3	3	3	2	3
Average Cardinality	2.04	2.04	2.04	2.04	2.04	2.19	2.72	2.02
Minimum Cardinality	1	1	1	1	1	1	1	1
Maximum Cardinality	6	6	6	6	6	7	7	5

Table 4.2: Effectiveness using All Detailed Evidence plus Erroneous Evidence

Figure 4.9: Plausibility Difference versus Strength of Additional Erroneous Evidence

The second set of test conditions again used all the basic evidence with the same random strengths however this time the detailed evidence was not guaranteed to be complete. The addition of either (correct) detailed evidence or erroneous evidence was controlled by random selection. The probability was 80% in favour of the detailed evidence with the balance made up by the erroneous evidence. When detailed evidence was chosen the strength was, as before, either fixed in the range of 0.0 to 0.6 or randomly generated in the range of 0.1 to 0.4. The results for the same sample of 100 trees are contained in Table 4.3. Clearly the effect of incomplete detailed evidence has increased the influence of the erroneous evidence but not greatly. The survival rate of the solution node in the Trace tree was the most affected; dropping from 97% to 87% in the worst case. The average ranking of the solution node and its average plausibility difference were barely changed. The plausibility difference is illustrated in Figure 4.10; note that the slope of the best fit line is within one percent of the slope from the previous series of tests. The vertical dispersion (i.e. in plausibility difference) however has noticeably increased.

Measures	Strength of Erroneous Evidence							
	0.0	0.1	0.2	0.3	0.4	0.5	0.6	R*
Survival Rate	1.0	0.89	0.89	0.89	0.89	0.89	0.87	0.91
Average Plausibility Difference	0.92	0.42	0.37	0.33	0.3	0.27	0.26	0.43
Minimum Plausibility Difference	0.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
Maximum Plausibility Difference	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Average Ranking	1.0	1.0	1.12	1.31	1.43	1.43	1.41	1.18
Minimum Ranking	1	1	1	1	1	1	1	1
Maximum Ranking	1	1	2	3	3	3	3	3
Average Cardinality	2.98	2.85	2.85	2.85	2.85	2.85	2.97	2.8
Minimum Cardinality	1	1	1	1	1	1	1	1
Maximum Cardinality	11	7	7	7	7	7	7	7

Table 4.3: Effectiveness using mixed Detailed Evidence and Erroneous Evidence

Figure 4.10: Plausibility Difference versus Strength of Mixed Erroneous Evidence

The RUNES design does not include any means to directly exploit knowledge of a priori probability distributions. It has been shown that prior beliefs cannot be handled in the same manner as observed evidence [Shastri88,89, Yen86,89] and Yen has suggested a modification to Dempster-Shafer theory to include prior beliefs [Yen86]. Incorporation of prior beliefs in RUNES will be discussed in a subsequent section.

Although RUNES does not use prior beliefs some examination of the potential benefits was attempted. A simple test was devised in which the solution set of a solution node was post-processed using prior probability distributions to rank the solutions within the set. No beliefs for individual members were generated by this technique, however, since Dempster-Shafer theory dictates that the belief in the solution set belongs to the set as a whole and should not be distributed (indifferently or otherwise) to the members of the set. The technique for processing the solution set consisted of two steps. First, the impossible or improbable solutions, as determined by their prior probabilities, were removed from the solution set. In practice this is a simple thresholding technique to remove members which are improbable and it could easily be improved to inhibit removing improbable members if there were no better candidates in the set. The second step involves ranking the remaining members of the solution set according to their prior probabilities.

The solution sets from the solution nodes in the second test series (Table 4.3) were post-processed using this simple algorithm and the results appear in Table 4.4. The probability distribution for the forest (Table 4.1) was used for the prior probability distribution and the probability distribution of the sample was assumed to be unknown. The average cardinality of the solution sets before processing with prior probabilities was very close to three whereas it was 1.4 to 1.5 after processing. The maximum cardinality of the solution sets decreased to 3 compared to a high of 11 before processing.

Measures	Strength of Erroneous Evidence							
	0.0	0.1	0.2	0.3	0.4	0.5	0.6	R*
Average Cardinality with Priors	1.52	1.42	1.42	1.42	1.42	1.42	1.45	1.45
Minimum Cardinality with Priors	1	1	1	1	1	1	1	1
Maximum Cardinality with Priors	3	3	3	3	3	3	3	3
Average Ranking with Priors	1.09	1.06	1.06	1.06	1.06	1.06	1.06	1.04
Minimum Ranking with Priors	1	1	1	1	1	1	1	1
Maximum Ranking with Priors	3	3	3	3	3	3	3	2

Table 4.4: Ranking Solution Set Members using Prior Probabilities

The average ranking of the correct solution within the solution set has 1.06 to 1.09 indicating that it was nearly always ranked first. Note that this use of prior probabilities provides a means of ranking the otherwise ambiguous choices within the solution set but that it does not affect the ranking nor the belief of the solution node. These preliminary tests suggest that prior probabilities could be used to rank the members of the solution set or indeed any hypothesis with a solution space category containing multiple members.

The final set of tests of the RUNES implementation was focused on testing the simple discernment heuristic for query generation. The transcripts from two complete test runs are included in Appendix C. The first transcript contains the results for a query generator which does not use the discernment heuristic whereas the second transcript contains the results for one which does use it. By the third query the non heuristic query generator had reduced the solution set cardinality to 11 and the heuristic query generator had reduced it to 15. The non heuristic method then poses two queries, which have no effect on the cardinality of the solution set, before it finally finds one which does reduce it (to 7). The heuristic method however, reduces the cardinality of the solution set with each of its next two queries (to 8 and then 7). The inherent ambiguity among the various types of maple trees then dominates the problem and neither approach makes rapid progress toward the final solution set (which still has a cardinality of 5).

A short excerpt from another query transcript is included here to illustrate the discernment heuristic at work. At this point in the session the classifier has already determined that the *Leaf_Category* is *Needle_Leaf* and that the *Needle_Attachment* is *Needles_In_Bundles*. The current best description indicates that it has yet to determine the *Needle_Shape* and *Fruit_Category* and that it has not determined the number of needles per bundle. The discernment rating for each of these three potential queries is

listed in the transcript immediately after the current candidates list. The *Needles_In_Bundles* query has the best discernment of the current candidates (lowest number) and thus it is selected, and in this case the answer of 5 needles per bundle is sufficient to solve the problem by uniquely identifying the tree as a White Pine.

Tree Classifier Problem

White-Pine, Heuristic Query Generator

{Excerpt from session}

Begin query session: (Nov 7, 1990 12:04:21)

Current best descriptions:

Needle_Shape_? & Needles_In_Bundles & Fruit_Category_?
Bel:0.3 Pl: 1.0

Current Candidates:

(JackPine, LoblollyPine, LongleafPine, MountainPine, PitchPine,
RedPine, ScotchPine, ScrubPine, ShortleafPine, SwampPine, WhitePine,)

***** Discernment of: Needles_In_Bundles(11) is: 11.666667

**** Discernment of: Needle_Shape(4) <OR> is: 22.0

***** Discernment of: Fruit_Category(86) <OR> is: 22.0

Pose query over children of: Needles_In_Bundles(11)

Choose one to support or refute:

2_Needles_Per_Bundle(12)

3_Needles_Per_Bundle(13)

5_Needles_Per_Bundle(14)

answer: 5_Needles_Per_Bundle(14)

is supported with weight: 0.6

Current best descriptions:

Needle_Shape_? & 5_Needles_Per_Bundle & Fruit_Category_?...
Bel:0.6 Pl: 1.0

Current Candidates: (WhitePine,)

***** SOLUTION *****

RUNES Trace Node: Needle_Shape_? & 5_Needles_Per_Bundle & Fruit_Category_?
 SUPPORT = 0.6, PLAUSIBILITY = 1.0)
 (evidentialSupport= 0.59328, evidentialPlausibility= 1.0)

SOLUTIONS:

=====

(WhitePine,)

Contributing Evidence:

=====

There is no evidence in the EvidenceSpace: Fruit_Category

There is no evidence in the EvidenceSpace: Needle_Shape

5_Needles_Per_Bundle with support: 0.6 and plausibility: 1.0
 in the EvidenceSpace: Needle_Attachment

HypSet{00000000000000000000640}(WhitePine,)

4.4. Representation and Computational Requirements

The algorithm for computing the belief function over the RUNES tree was developed from Shafer and Logan's algorithm [Shafer&Logan87b]. Shafer and Logan included some analysis of their algorithm in which they stated that for a disjunctive tree with N nonterminal nodes and a branching factor of F , the algorithm was linear in the number of nodes (i.e. $\{a + bNF\}$). They empirically verified this claim with tests of trees with values of F up to 5 and values of N up to 30,000.

Continuing with Shafer and Logan's definitions of F and N the number of terminal nodes, N_T , and the total number of nodes in the tree, T , are:

$$N_T = NF - N + 1.$$

$$T = NF + 1.$$

A RUNES tree, unlike Shafer and Logan's tree, also contains conjunctive nodes which connect the evidence spaces for each attribute class into a single hierarchy. Let us now assume that the N described above is the number of nonterminal nodes in a single evidence space and that there are E distinct evidence spaces of this size over the same set of solutions. Then if a single disjunctive hierarchy (required by Shafer and Logan) is

created by subordinating and replicating evidence spaces the resulting tree has:

$$N_T = (NF - N + 1)^E .$$

$$T = \left\lceil \frac{F}{F - 1} \right\rceil * [(NF - N + 1)^E - 1] + 1$$

Given that F and N are fixed the size of the disjunctive tree is growing exponentially with the number of evidence spaces, E . An equivalent RUNES tree would use a conjunctive node to join the evidence spaces resulting in a tree with:

$$N_T = E(NF - N + 1) .$$

$$T = E(NF + 1) + 1 .$$

The RUNES representation is much more compact, having a growth rate that is linear in the number of evidence spaces. Assuming both tree representations were using algorithms which were linear in the number of nodes in the tree, the RUNES representation would clearly be more efficient as it has drastically fewer nodes. Moreover RUNES does not sacrifice the ability to represent a rich hypothesis space to achieve this performance. Since the RUNES tree explicitly represents any evidential hypothesis

which could be the focus of an observation, no fidelity in the belief function over the tree is lost. Furthermore, the creation of the Trace tree from the RUNES tree allows any valid combination of evidential hypotheses resulting in an effective hypothesis space of size:

$$H = (NF + 1)^E .$$

Whereas in the disjunctive hierarchy approach the hypothesis space is limited to the nodes of the disjunctive tree (approximately $N^E * F * (F - 1)^{E-1}$).

Chapter 5.

Summary

5.1. Concluding Remarks

Gordon and Shortliffe referred to their approach as a *hierarchical hypothesis space* to describe their disjunctive hierarchy of hypotheses over the set of all possible solutions. Shafer and Logan preferred to described their disjunctive tree as *hierarchical evidence* , stating that the evidence bears directly on the nodes of the tree but that the belief in other hypotheses may be calculated as well. Both approaches however, failed to distinguish the difference between evidence and solutions which exists in all but trivial problems. Solutions are not generally observed or measured in any direct manner; instead they are inferred by the correspondence between their attributes and the attributes observed from the unknown entity. It is therefore more appropriate to form the hypothesis space over the sets of observable attributes and combine all evidence in this structure before inferring a solution. This is the first principle of the RUNES system.

The two other systems mentioned above are the best known examples of previous applications of Dempster-Shafer theory to structured hypothesis spaces. Both approaches restricted themselves to a single disjunctive hierarchy. Such structures are not amenable to real problems with many distinct attribute types which have no intrinsic subclass-superclass relationship. Forcing the attribute types into a single disjunctive structure by enumerating the combinations is awkward at best and often prohibitive since the

hypothesis space increases exponentially with the number of attribute types. The RUNES hierarchy provides both conjunctive and disjunctive nodes. This results in a concise and natural representation where the distinct attribute types can have separate conjoined subtrees. The RUNES tree increases at a rate that is linear with the number of attribute types.

The RUNES system of nested evidence spaces has a more expressive representation than the previous approaches resulting in a more compact data structure. Although concise, the representation does not sacrifice clarity as it is based on the familiar and-or tree structure. RUNES implements a Dempster-Shafer belief system with extensions to Shafer and Logan's original algorithm to support conjunctive nodes. The performance of the belief calculations within RUNES is excellent due to a computational complexity that is linear in the number of nodes in the tree and a tree structure with size linearly proportional to the number of attribute types. In addition the RUNES system displayed a linearly decreasing performance in response to increasing amounts of erroneous evidence indicating that RUNES is robust with no indications of brittleness.

5.2. Further Work

5.2.1. Solution Space Concepts

The current design of the RUNES system does not provide for structure within the solution space; it only structures the evidence spaces. A structured solution space is however not precluded. Two potential forms of structure in the solution space were identified during the implementation of the tree classifier problem. The first is the obvious addition of solution space categories (concepts) which need not be a direct manifestation of observable attribute values. One could envision a system of categories

which grouped trees according to their potential uses in landscaping. These categories need not be disjoint. This form of solution space structure could be implemented in a fashion loosely similar to the trace tree, in that it would have its own structure but it would derive its beliefs by referencing nodes in the RUNES tree structure.

The second form of solution space structure could be called solution microstructure. Several times during the knowledge acquisition phase of the tree classifier, a tree was described as having either attribute value A or B or both, where A and B were mutually exclusive values within the RUNES evidence space structure. In most cases the distinction between A and B for other trees was real and useful, making the option of merging A and B into a single attribute value unattractive. It appears to be possible to resolve this problem by providing these anomalous cases with a solution space substructure which properly handles the disjunction of their multiple representations thus allowing them to benefit from observations which support any of their possible forms.

5.2.2. Independence of Evidence Spaces

The lack of independence between evidence spaces with respect to their solution categories in the solution space is a significant problem in real applications if the accuracy of belief and the plausibility of the solution are important. Lacking independence, the complicated procedure described in section 2.4 must be used in place of simple multiplication. It is important to note however that the implementation of the Trace tree used the assumption of independence to find the best path and it was always correct if the evidence was consonant. Therefore it seems reasonable to expect that independence can be assumed for the purpose of finding the best solution. This is very important since it would imply that the procedure for accurate belief computations with

correlated evidence spaces can be selectively applied to those nodes known to be of interest.

If the application domain has only a small amount of correlation between evidence spaces it may be advantageous to test the spaces for independence and mark those which are not independent. The belief computation for the solution could then use simple multiplication when combining those spaces not marked as correlated.

5.2.3. Defaults and Prior Beliefs

The investigation into the use of prior probabilities in RUNES was preliminary but encouraging. Prior probabilities were only used to resolve residual ambiguity in the solutions provided by RUNES. Clearly prior probabilities warrant further investigation into techniques to properly integrate them with the RUNES system.

The work by Yen [Yen86, 89] on modification to Dempster-Shafer theory to incorporate prior beliefs may be applicable to RUNES. In Yen's system each piece of evidence is weighted inversely proportional to its prior probability and then combined using Dempster's rule. The resulting belief function, after all evidence has been combined, is then weighted proportional to its prior probabilities and normalized. Now one normally assumes that the prior beliefs that one wishes to incorporate are the prior beliefs in the solutions to the problem, however, RUNES performs its belief consolidation over the evidential hypothesis spaces not the solution space. If Yen's approach were to be used it must use the prior beliefs in the evidential attributes.

Yen also described a method to provide probabilistic mappings from evidence to solution hypotheses where the mass associated with a focal element from the evidential support function is distributed, according to a probability distribution, over a set of

solution hypotheses. Although this may be used to handle the cases where solutions have more than one attribute value for a single attribute class, it is thought to be a sub-optimal solution. For example, if a tree is characterized as having mostly 3 needles per bundle but occasionally has 2 needles per bundle, then evidence focused on 2 needles per bundle would have a small probability value for this tree. Evidence focused on 3 needles per bundle would have a corresponding larger probability value for this tree. The problem with this approach being used in the manner which Yen suggests is that the mass of a single piece of evidence is distributed across a number of disjoint hypotheses which creates dissonance where there should only be ambiguity (i.e. nonspecificity)

The incorporation of default values into RUNES was considered briefly but no technique compatible with Dempster-Shafer theory could be devised.

5.2.4. Heuristics for Inference Control

Heuristics and metadata for the controlling of the inference process in RUNES were examined. The use of uncertainty measures to evaluate the state of the inference provides very concise summaries and further work on their application as metadata for the controlling process is warranted.

The use of prior beliefs in the solutions to rank the members of an otherwise ambiguous solution set was demonstrated. This approach could be used within the inferencing control process to focus its attention on the most plausible members of the current list of best solutions. This would be particularly beneficial during the initial stages of the analysis when the list of best solutions can be very large.

5.2.5. Complexity in Evidence

During the knowledge acquisition phase for the tree classification problem it

became painfully clear that the problem required a much richer representation for evidence. Many times attributes were described with qualifiers like: 'mostly', 'usually', and 'sometimes'. Often the attributes were continuous valued, such as the length of the needles or cones. These experiences suggested that some technique, similar to fuzzy sets which allow for a degree of membership, may be required to map real world observations into the evidence spaces. It may not be possible to implement a rich representation for evidence while maintaining the assumption that observations can be represented by simple or dichotomous support functions.

If a particular evidence space thwarts all attempts to cast it as a structured hypothesis space then perhaps a hybrid system could be constructed. The RUNES tree would provide the overall structure in which the difficult evidence space would appear as a node like the current *AndNodes*. Within that evidence space, unrestricted computation using Dempster's Rule would be implemented with the results supporting or refuting the parent node in the RUNES tree. When restricted to these typically small frames of discernment the unrestricted hypothesis space may be manageable. However it is unclear as to whether the Trace tree could be modified to support evidential hypotheses from an unstructured space although the evidential elements would certainly still be usable.

5.2.6. Parallelism

The work reported herein did not include a study of parallel implementations of the RUNES system however some thoughts on parallel implementations have been included here. The RUNES algorithm has two distinct phases: the upward propagation combining evidence, and the downward normalization of beliefs. Except for the computationally insignificant downward propagation of a message indicating that the current beliefs are now invalid, the first phase is characterized by a single line of

computation (child to parent) from the node which has been updated, to the root of the structure. This does not lend itself nicely to parallel implementation. The normalization phase, however, fans out from the root of the structure to all nodes in the tree and does involve a significant proportion of the overall computation. Per node parallelism would naturally apply to this phase. However, the gains from a parallel implementation may not be as great as it might first appear. If the system is only interested in the plausible branches the normalization process need not continue into implausible subtrees as the plausibility of a child is never greater than the plausibility of its parent. In the limit, this pruning technique could reduce the downward normalization phase to a single line of computation from parent to most plausible child.

An alternative to fine grain parallelism (i.e. each node in parallel) is parallelism at the level of evidence spaces, assuming the source of evidence is capable of parallel operation. There seems to be no reason why multiple update messages could not proceed asynchronously up the tree. The downward normalization phase is however more difficult to implement asynchronously. If, for short periods, the local computation of belief can satisfy the requirements of the query generator; then synchronous normalization could take place at fixed or variable intervals. Perhaps recently calculated global beliefs could serve as estimates until the next normalization phase. It may be possible to make normalization a requested service which is only triggered when the belief within an evidence space changes significantly. When one evidence space, E_1 , changes significantly; the evidence space, E_2 , in which it is embedded can choose to handle the normalization locally if the change in E_1 had minimal effect on the state of E_2 , or propagate the change upward and wait for normalization from above.

Appendix A

Dempster-Shafer Theory

Dempster-Shafer theory is a mathematical theory of evidence and plausible reasoning which deals with weights of support and numerical degrees of belief. The approach was conceived by Arthur Dempster [Dempster66] and later elaborated by Glen Shafer [Shafer76a,87a]. It features an explicit representation of ignorance or uncertainty. Shafer describes the basic concept of the theory as follows:

"The additive degrees of belief of the Bayesian theory correspond to an intuitive picture in which one's total belief is susceptible to division into various portions, and that intuitive picture has two fundamental features. First, to have a degree of belief in a proposition is to commit a portion of one's belief to it. And secondly, whenever one commits only a portion of one's belief to a proposition, one must commit the remainder to its negation. The obvious way to obtain a more flexible and realistic picture is to discard the second of these features while retaining the first."

Removing the requirement to commit the remainder of one's belief to the negation of a hypothesis implies that the state of our belief in a proposition can no longer be captured by a single scalar value (i.e. a probability). Dempster-Shafer theory uses two scalar values which can range from zero to one and which together describe an interval of belief for a proposition. The lower bound of the interval is referred to as the *belief* in

the proposition¹ while the upper bound is called the *plausibility* of the proposition. The belief in a proposition is the portion of one's total belief which is already committed to the proposition. The plausibility is that portion of one's total belief which has not been committed to the negation of the proposition.

The width of the belief interval, that is the plausibility minus the belief, is the explicit representation of ignorance or uncertainty. A complete lack of information is called a *vacuous belief* and is represented by a belief of zero and a plausibility of one. Vacuous beliefs have a degree of uncertainty equal to one. A proposition which is a certainty has a belief of one and a plausibility of one. Similarly, an impossibility will have a belief and plausibility of zero. Both certainties and impossibilities have degrees of uncertainty equal to zero.

Dempster-Shafer theory defines the set of possible solutions from the problem domain to be the *frame of discernment* of the problem. The members of the frame of discernment are assumed to be mutually exclusive and the frame is assumed to be exhaustive (i.e. complete). For example, the frame of discernment for a single die would be the set { 1, 2, 3, 4, 5, 6 }. Dempster-Shafer theory assumes that exactly one member of the frame of discernment corresponds to the truth.

A proposition or hypothesis is represented by a subset of the frame of discernment. The *hypothesis space* is the set of all possible hypotheses, that is, the set of all subsets of the frame of discernment (i.e. the power set of the frame of discernment). For example, the hypothesis space for a single coin is the set { $\emptyset$, {heads}, {tails}, {heads,tails} }. Note that although the null set is included, the assumption that the frame

¹Some references will refer to the lower bound as the support for the proposition.

of discernment is exhaustive will preclude our assigning any belief to this proposition. The symbol Θ will be used to denote the hypothesis for the entire frame of discernment and $\mathcal{H}$ will denote the entire hypothesis space over Θ .

Dempster-Shafer theory defines a *basic probability number*, $m(A)$, which represents that portion of our total belief which is assigned to exactly A . The value $m(A)$ can be thought of as that portion of our belief which is constrained to stay within A , that is within the subset of solutions which A represents, but that is otherwise free to move to any subset of A . This freedom of movement or underspecification of the distribution of our total belief is the representation of our ignorance. A function is called a *basic probability assignment* (bpa) if it satisfies:

$$m: \mathcal{H} \rightarrow [0, 1].$$

$$m(\emptyset) = 0.$$

$$\sum_{A \subseteq \Theta} m(A) = 1$$

A belief function assigns to each hypothesis a measure of our total belief. For A and B members of $\mathcal{H}$, the number $\text{Bel}(A)$ can be viewed as our degree of belief in the proposition that the truth is one of the members of the subset A . $\text{Bel}(A)$ is the measure of the total probability mass constrained to lie within A .

$$\text{Bel}(A) = \sum_{B \subseteq A} m(B)$$

The subset B of Θ is called a *focal element* of Bel if $m(B) \neq 0$. The vacuous belief function is the simplest belief function with only one focal element equal to Θ . A *simple support function* is a belief function with at most one focal element not equal to Θ (and

optionally one focus equal to Θ). A *dichotomous* belief function can have two foci not equal to Θ where one focus is the negation of the other.

A plausibility function is also a measure of our total belief. For A and B members of $\mathcal{H}$, the number $Pl(A)$ is the degree of belief in the proposition that the truth is not constrained to lie within $\neg A$.

$$Pl(A) = \sum_{B \cap A \neq \emptyset} m(B)$$

$$\text{or } Pl(A) = 1 - Bel(\neg A).$$

Given several belief functions based on distinct bodies of evidence but over the same frame of discernment, Dempster's rule of combination can be used to compute their orthogonal sum, which will be a new belief function based on the combined evidence. Consider two simple support functions, S_1 and S_2 , with foci A and B respectively, where $S_1(A) = s_1$, and $S_2(B) = s_2$. The basic probability assignments associated with each support function are then: $m_1(A) = s_1$, $m_1(\Theta) = 1 - s_1$; and $m_2(B) = s_2$, $m_2(\Theta) = 1 - s_2$. For this simple example Dempster's rule of combination, $S = S_1 \oplus S_2$, can be illustrated using a *unit square*, Figure A.1. The unit square has a length, width and area equal to one. The length and width represent each of the bpa's being combined and the area represents the resulting bpa.

	B	Θ
A	$A \cap B$	$A \cap \Theta = A$
Θ	$B \cap \Theta = B$	Θ

Figure A.1: Dempster's Rule using a Unit Square

This simple example has four interesting cases to consider. First, there is the case where $A = B$. This represents two different pieces of evidence supporting the same hypothesis. In this case all quadrants except the lower right quadrant are equal to A while the lower right quadrant equals Θ . The result is:

$$m(A) = 1 - (1 - s_1)(1 - s_2) \quad m(\Theta) = (1 - s_1)(1 - s_2)$$

or

$$S(A) = 1 - (1 - s_1)(1 - s_2)$$

The second interesting case is $A \cap B \neq \emptyset$, $A \neq B$. For this case all four quadrants represent different hypotheses. The resulting support function is:

$$S(C) = \begin{cases} 0 & \text{if } C \text{ does not contain } A \cap B \\ s_1 s_2 & \text{if } C \text{ contains } A \cap B \text{ but neither } A \text{ nor } B \\ s_1 & \text{if } C \text{ contains } A \text{ but not } B \\ s_2 & \text{if } C \text{ contains } B \text{ but not } A \\ 1 - (1 - s_1)(1 - s_2) & \text{if } C \text{ contains both } A \text{ and } B \text{ but } C \neq \Theta \\ 1 & \text{if } C = \Theta \end{cases}$$

for all $C \in \Theta$.

The third case is where A is a subset of B . In this case the upper two quadrants are focused on A and the lower left quadrant has a focus of B . The resulting support function is:

$$S(C) = \begin{cases} 0 & \text{if } C \text{ does not contain } A \\ s_1 & \text{if } C \text{ contains } A \text{ but not } B \\ 1 - (1 - s_1)(1 - s_2) & \text{if } C \text{ contains } B \text{ but } C \neq \Theta \\ 1 & \text{if } C = \Theta \end{cases}$$

for all $C \in \Theta$.

The last interesting case is where $A \cap B = \emptyset$. In this case the upper left quadrant represents a mass which is focussed on the null set. Since the definition of belief functions expressly forbids this, Dempster's rule provides for the renormalization of the resulting bpa such that the sum of the masses for its focal elements is once again equal to one. The resulting support function, after normalization, is:

$$S(C) = \begin{cases} 0 & \text{if } C \text{ does not contain neither } A \text{ nor } B \\ \kappa s_1(1 - s_2) & \text{if } C \text{ contains } A \text{ but not } B \\ \kappa s_2(1 - s_1) & \text{if } C \text{ contains } B \text{ but not } A \\ \kappa[1 - (1 - s_1)(1 - s_2)] & \text{if } C \text{ contains } A \text{ and } B \text{ but } C \neq \Theta \\ 1 & \text{if } C = \Theta \end{cases}$$

for all $C \in \Theta$ where $\kappa = (1 - s_1 s_2)^{-1}$.

Since Dempster's rule considers each focal element of one belief function with each focal element of the other belief function and produces a bpa with all possible intersections of the two sets of foci, the approach can be computationally expensive if not prohibitive for problems of moderate size. An analysis of the computational complexity of Dempster's rule has performed by Orponen [Orponen90] and it has found to be #P-complete.

The disjunctive hierarchies discussed in this report are an attempt to reduce the computational complexity by restricting the allowed foci of the belief function to those represented by the tree. These hierarchies can be viewed as nested *partitions*. A partition is a set of disjoint subsets of the frame of discernment whose union equals that frame. In a disjunctive hierarchy the children of a node form a partition over the frame of discernment of the parent node. Obviously multiple different partitions of the same frame of discernment are possible. Two partitions, $\mathcal{P}_1$ and $\mathcal{P}_2$, of the same frame of discernment are independent if no member of $\mathcal{P}_1$ is a subset of a member of $\mathcal{P}_2$ and no member of $\mathcal{P}_2$ is a subset of a member of $\mathcal{P}_1$.

Appendix B

RUNES Algorithm for Nested Evidence Spaces

The following notation will be used throughout:

Θ	- the root node of the tree (the frame of discernment).
$A \ B \ C$	- nodes in a tree (subsets of Θ).
$\neg A$	- the negation of A (nodes which are neither directly above
nor directly below A in the tree).
Bel	- a belief function over all nodes of a tree.
Pl	- a plausibility function over all nodes of a tree.
$Bel(A)$	- the belief in node A (subset A).
$Bel_{\Theta \downarrow}$	- belief function given all evidence.
$Bel_{A \downarrow}$	- belief function given only evidence focussed on nodes below A.
Bel_A	- belief function given only evidence focussed on directly A.
$Bel_{A \uparrow}$	- belief function given only evidence focussed on nodes above A
or on nodes down other branches.
$Bel_1 \oplus Bel_2$	- Dempster's Rule for combining belief functions.
$(Bel_1 \oplus Bel_2)(A)$	- the belief in node A after combining belief functions.
$\{ \oplus Bel_i \mid i = 1 \text{ to } n \}$	$= Bel_1 \oplus Bel_2 \oplus \dots \oplus Bel_n$

Shafer and Logan [Shafer&Logan87b] noted that all evidence¹ focused on nodes in a hierarchy could be factored as follows:

$$\text{Bel}_{\Theta\downarrow} = \text{Bel}_{A\downarrow} \oplus \text{Bel}_A \oplus \text{Bel}_{A\downarrow}$$

Where A is any node except Θ and $\oplus$ denotes Dempster's Rule.

The six subroutines presented by Shafer and Logan calculate the following scalar values which capture the state of belief in the hypotheses represented by the tree:

$\text{BelZero}(B) = \text{Bel}_B(B)$	$\text{notBelZero}(B) = \text{Bel}_B(\neg B)$
$\text{BelBelow}(B) = \text{Bel}_{B\downarrow}(B)$	$\text{notBelBelow}(B) = \text{Bel}_{B\downarrow}(\neg B)$
$\text{BelDown}(B) = (\text{Bel}_B \oplus \text{Bel}_{B\downarrow})(B)$	$\text{notBelDown}(B) = (\text{Bel}_B \oplus \text{Bel}_{B\downarrow})(\neg B)$
$\text{BelLocal}(B) = \text{Bel}_{A\downarrow}(B)$	$\text{notBelLocal}(B) = \text{Bel}_{A\downarrow}(\neg B)$
$\text{Bel}^*(B) = \text{Bel}_{A\downarrow}(B \cup \neg A)$	
$\text{Bel}(B) = \text{Bel}_{\Theta\downarrow}(B)$	$\text{notBel}(B) = \text{Bel}_{\Theta\downarrow}(\neg B)$

Where A is any node except Θ and A is the parent of B.

The values of primary interest are $\text{Bel}(B)$ and $\text{notBel}(B)$ which are the beliefs for and against B given all of the evidence. Two other values, $\text{BelLocal}(B)$ and $\text{notBelLocal}(B)$, are of interest for use by algorithms, such as query generators, which need to rank sibling nodes but do not need to know the actual final beliefs. These two local beliefs represent the belief in B given all of the evidence focused on nodes in the

¹ Assuming the evidence is represented as simple or dichotomous support functions.

subtree below B's parent, including all evidence focused on nodes in any other evidence space contained in that subtree. This insures that all evidence which discerns B from its siblings is considered².

The first phase of Shafer and Logan's approach uses Subroutines 1 and 2 for all nodes below the root node to propagate beliefs up the tree. These routines assume that all evidence focussed on any node (eg. node B) has been combined to produce BelZero(B) and notBelZero(B) which represent a single dichotomous belief focussed on that node (this process will be discussed later in this section). Subroutines 1 and 2 are applied to nodes in the tree starting at the leaves and working back toward the root of the tree.

Subroutine: 1

$$\text{BelBelow}(A) = 1.0 - K_1$$

$$\text{notBelBelow}(A) = K_1 * \prod_{B \in \mathcal{M}(A)} \frac{\text{notBelDown}(B)}{1.0 - \text{BelDown}(B)}$$

where

$$K_1 = \left[1.0 + \sum_{B \in \mathcal{M}(A)} \frac{\text{BelDown}(B)}{1.0 - \text{BelDown}(B)} \right]^{-1}$$

Note: If A is a leaf node in the tree then BelBelow(A) = notBelBelow(A) = 0.

²This assumes that the hypotheses in the evidence spaces above B and down other branches are uncorrelated to the hypotheses represented by B and its siblings. Refer to the section on Independence of Evidence Spaces.

Subroutine: 2

$$\text{BelDown}(A) = 1.0 - K_2 * (1.0 - \text{BelZero}(A)) * (1.0 - \text{BelBelow}(A))$$

$$\text{notBelDown}(A) = 1.0 - K_2 * (1.0 - \text{notBelZero}(A)) * (1.0 - \text{notBelBelow}(A))$$

$$\text{where } K_2 = [1.0 - \text{BelZero}(A) * \text{notBelBelow}(A) - \text{notBelZero}(A) * \text{BelBelow}(A)]^{-1}$$

Note: If A is a leaf node in the tree then:

$$\text{BelDown}(A) = \text{BelZero}(A) \text{ and } \text{notBelDown}(A) = \text{notBelZero}(A).$$

When BelDown and notBelDown have been determined for all nodes in the ΔL of the root node then Subroutine 3 is used to calculate the values of Bel and notBel for each of these nodes. Subroutine 3 implements the assumption that the frame of discernment represented by the root node must contain the solution. Any combination of evidence which suggests the solution is outside the frame of discernment of the problem, as represented by the root node, is assumed to be conflict, resulting from dissonant evidence, which can be removed by normalization.

Subroutine: 3

For each $A \in \mathcal{M}(\Theta)$:

$$\text{Bel}(A) = 1.0 - K_3 \left[1.0 + \sum_{\substack{B \in \mathcal{M}(\Theta) \\ B \neq A}} \frac{\text{BelDown}(B)}{1.0 - \text{BelDown}(B)} \cdot \prod_{\substack{B \in \mathcal{M}(\Theta) \\ B \neq A}} \frac{\text{notBelDown}(B)}{1.0 - \text{BelDown}(B)} \right]$$

$$\text{notBel}(A) = 1.0 - K_3 \left[\frac{1.0 - \text{notBelDown}(A)}{1.0 - \text{BelDown}(A)} \right]$$

where

$$K_3 = \left[1.0 + \sum_{C \in \mathcal{M}(\Theta)} \frac{\text{BelDown}(C)}{1.0 - \text{BelDown}(C)} \cdot \prod_{C \in \mathcal{M}(\Theta)} \frac{\text{notBelDown}(C)}{1.0 - \text{BelDown}(C)} \right]^{-1}$$

Phase two of the algorithm is implemented by Subroutines 4, 5 and 6 as the beliefs based on all of the evidence are propagated back down the tree. Subroutine 4 calculates the effect of evidence focussed on a node A, above A, or down other branches. Subroutine 6 calculates $\text{Bel}(B)$ and $\text{notBel}(B)$ for any node B in the $\mathcal{M}$ of A. As the belief in A increases toward a certainty the combined behavior of Subroutines 4, 5 and 6 approaches that of Subroutine 3.

Subroutine: 4

$$\text{BelUp}(A) = 1.0 - K_4 \left[\frac{1.0 - \text{Bel}(A)}{1.0 - \text{BelBelow}(A)} \right]$$

$$\text{notBelUp}(A) = 1.0 - K_4 \left[\frac{1.0 - \text{notBel}(A)}{1.0 - \text{notBelBelow}(A)} \right]$$

where

$$K_4 = \left[\frac{1.0 - \text{Bel}(A)}{1.0 - \text{BelBelow}(A)} + \frac{1.0 - \text{notBel}(A)}{1.0 - \text{notBelBelow}(A)} - \frac{1.0 - \text{Bel}(A) - \text{notBel}(A)}{1.0 - \text{BelBelow}(A) - \text{notBelBelow}(A)} \right]^{-1}$$

Subroutine 5 determines the values of $\text{BelLocal}(B)$, $\text{notBelLocal}(B)$ and $\text{Bel}^*(B)$ for all B in the $\mathcal{L}$ of A . Subroutine 5 does not require Subroutine 4 to proceed it and in fact Subroutines 4 and 6 can be deferred until values for either $\text{Bel}(B)$ or $\text{notBel}(B)$ are required. Subroutine 5 is similar to Subroutine 3 except that the assumption that the solution must be within the hypothesis represented by A cannot be made a priori. $\text{Bel}^*(B)$ is an expression of the belief which may devolve onto B , in whole or in part depending on the final value of $\text{bel}(A)$.

Subroutine: 5

For each $B \in \mathcal{M}(A)$:

$$\text{BelLocal}(B) = 1.0 - K_5 \left[1.0 + \sum_{\substack{C \in \mathcal{M}(A) \\ C \neq B}} \frac{\text{BelDown}(C)}{1.0 - \text{BelDown}(C)} \right]$$

$$\text{notBelLocal}(B) = 1.0 - K_5 \left[\frac{1.0 - \text{notBelDown}(B)}{1.0 - \text{BelDown}(B)} \right]$$

$$\text{BelStar}(B) = 1.0 - K_5 \left[1.0 + \sum_{\substack{C \in \mathcal{M}(A) \\ C \neq B}} \frac{\text{BelDown}(C)}{1.0 - \text{BelDown}(C)} - \prod_{\substack{C \in \mathcal{M}(A) \\ C \neq B}} \frac{\text{notBelDown}(C)}{1.0 - \text{BelDown}(C)} \right]$$

where: $K_5 = \left[1.0 + \sum_{C \in \mathcal{M}(A)} \frac{\text{BelDown}(C)}{1.0 - \text{BelDown}(C)} \right]^{-1}$

Subroutine: 6

For each $B \in \mathcal{L}(A)$:

$$\text{Bel}(B) = K_6 \left[\text{BelUp}(A) * (\text{BelStar}(B) - \text{notBelBelow}(A)) \right. \\ \left. + \text{BelLocal}(B) * (1.0 - \text{BelUp}(A) - \text{notBelUp}(A)) \right].$$

$$\text{notBel}(B) = 1.0 - K_6 * (1.0 - \text{notBelUp}(A)) * (1.0 - \text{notBelLocal}(B)).$$

$$\text{where } K_6 = \left[1.0 - \text{BelBelow}(A) * \text{notBelUp}(A) - \text{notBelBelow}(A) * \text{BelUp}(A) \right]^{-1}.$$

As stated previously, the conjunctive nodes in RUNES can not use Subroutine 1 to calculate the values of $\text{BelBelow}(A)$ and $\text{notBelBelow}(A)$ as it assumes that the children of node A are disjoint. Fortunately Dempster's Rule can be directly applied in this situation since all children represent the same hypothesis. The only requirement is that the children of a conjunctive node be able to express the combined effect of all evidence focused on nodes below them in the form of a dichotomous belief function focused on the hypothesis that they, and the conjunctive node, represent. Note that in the RUNES system the children of conjunctive nodes are always disjunctive nodes and that Subroutine 1 will provide the children with precisely the values required by the conjunctive node.

Recall that $\text{BelZero}(A)$ and $\text{notBelZero}(A)$ represent the combined effect of all evidence focused directly on the node A . Since all evidence is represented by simple or dichotomous support functions (i.e. they have foci of: A , $\neg A$ or Θ) these values can be determined by the direct application of Dempster's Rule without risk of a combinatorial explosion³.

³ Note also that this computation is related to the number of pieces of evidence not the number of potential solutions nor the number of conceivable hypotheses.

Appendix C

Test Results

Tree Classification Problem

Silver Maple - Simple Query Generator

Begin query session: (Nov 6, 1990 09:58:12)

Pose query over children of: Leaf_Category(1) <OR>

Choose one to support or refute:

Scale_Like_Leaf(2)

Needle_Leaf(3) <AND>

Broad_Leaf(19) <AND>

answer: Broad_Leaf(19) <AND>

is supported with weight: 0.9

Pose query over children of: BL_Type(20) <OR>

Choose one to support or refute:

BL_Simple(21)

BL_Compound(22)

answer: BL_Simple(21)

is supported with weight: 0.9

Current best descriptions:

BL_Simple & BL_Attachment_? & BL_Body_Shape_? & BL_TipShape_? &
BL_Base_Shape_? & BL_Base_Symmetry_? & BL_Edge_? & BL_Lobing_? &

Appendix C

123

Fruit_Category_?... Bel: 0.9 Pl: 1.0

Current Candidates: HypSet(more any 25 members)

Pose query over children of: BL_Attachment(36) <OR>

Choose one to support or refute:

Alternate_Attachment(37)

Not_Alternate_Attachment(38)

answer: Not_Alternate_Attachment(38)

is supported with weight: 0.9

Current best descriptions:

BL_Simple & Not_Alternate_Attachment & BL_Body_Shape_? & BL_TipShape_?
& BL_Base_Shape_? & BL_Base_Symmetry_? & BL_Edge_? & BL_Lobing_? &
Fruit_Category_?... Bel: 0.81 Pl: 1.0

Current Candidates: HypSet(BasketWillow, BlackMaple, CommonCatalpa,
CommonLilac, MountainMaple, NorwayMaple, PrincessTree, RedMaple, SilverMaple,
StripedMaple, SugarMaple,)

Pose query over children of: Not_Alternate_Attachment(38)

Choose one to support or refute:

Opposite_Attachment(39)

Whorled_Attachment(40)

answer: Opposite_Attachment(39)

is supported with weight: 0.9

Current best descriptions:

BL_Simple & Opposite_Attachment & BL_Body_Shape_? & BL_TipShape_? &
BL_Base_Shape_? & BL_Base_Symmetry_? & BL_Edge_? & BL_Lobing_? &
Fruit_Category_?... Bel: 0.81 Pl: 1.0

Appendix C

124

Current Candidates: HypSet(BasketWillow, BlackMaple, CommonCatalpa, CommonLilac, MountainMaple, NorwayMaple, PrincessTree, RedMaple, SilverMaple, StripedMaple, SugarMaple,)

Pose query over children of: BL_Base_Symmetry(67) <OR>

Choose one to support or refute:

Even_Base(68)

Uneven_Base(69)

answer: Even_Base(68)

is supported with weight: 0.9

Current best descriptions:

BL Simple & Opposite_Attachment & BL_Body_Shape_? & BL_TipShape_? &
BL_Base_Shape_? & Even_Base & BL_Edge_? & BL_Lobing_? & Fruit_Category_?...
Bel: 0.729 Pl: 1.0

Current Candidates: HypSet(BasketWillow, BlackMaple, CommonCatalpa, CommonLilac, MountainMaple, NorwayMaple, PrincessTree, RedMaple, SilverMaple, StripedMaple, SugarMaple,)

Pose query over children of: BL_Lobing(81) <OR>

Choose one to support or refute:

Not_Lobed(82)

Lobed(83)

answer: Lobed(83)

is supported with weight: 0.9

Current best descriptions:

BL Simple & Opposite_Attachment & BL_Body_Shape_? & BL_TipShape_? &
BL_Base_Shape_? & Even_Base & BL_Edge_? & Lobed & Fruit_Category_?... Bel:
0.81 Pl: 1.0

Current Candidates: HypSet(BlackMaple, MountainMaple, NorwayMaple, RedMaple,

SilverMaple, StripedMaple, SugarMaple,)

Pose query over children of: Lobed(83)

Choose one to support or refute:

Fan_Lobed(84)

Feather_Lobed(85)

answer: Fan_Lobed(84)

is supported with weight: 0.3

Current best descriptions:

BL_Simple & Opposite_Attachment & BL_Body_Shape_? & BL_TipShape_? &
BL_Base_Shape_? & Even_Base & BL_Edge_? & Fan_Lobed & Fruit_Category_?...
Bel: 0.851553 Pl: 1.0

Current Candidates: HypSet(BlackMaple, MountainMaple, NorwayMaple, RedMaple,
SilverMaple, StripedMaple, SugarMaple,)

Pose query over children of: BL_TipShape(50) <OR>

Choose one to support or refute:

Pointed_Tip(51)

Not_Pointed_Tip(55)

answer: Pointed_Tip(51)

is supported with weight: 0.9

Current best descriptions:

BL_Simple & Opposite_Attachment & BL_Body_Shape_? & Pointed_Tip &
BL_Base_Shape_? & Even_Base & BL_Edge_? & Fan_Lobed & Fruit_Category_?...
Bel: 0.19683 Pl: 1.0

Current Candidates: HypSet(BlackMaple, MountainMaple, NorwayMaple, RedMaple,
SilverMaple, StripedMaple, SugarMaple,)

Pose query over children of: Pointed_Tip(51)

Choose one to support or refute:

Long_Pointed_Tip(52)

Moderately_Pointed_Tip(53)

Short_Pointed_Tip(54)

answer: Short_Pointed_Tip(54)

is refuted with weight:: 0.6

Current best descriptions:

BL_Simple & Opposite_Attachment & BL_Body_Shape_? & Pointed_Tip &
BL_Base_Shape_? & Even_Base & BL_Edge_? & Fan_Lobed & Fruit_Category_?...
Bel: 0.19683 Pl: 1.0

Current Candidates: HypSet(BlackMaple, MountainMaple, NorwayMaple, RedMaple,
SilverMaple, StripedMaple, SugarMaple,)

Pose query over children of: BL_Edge(70) <OR>

Choose one to support or refute:

Smooth_Edge(71)

Not_Smooth_Edge(72)

answer: Not_Smooth_Edge(72)

is supported with weight: 0.9

Current best descriptions:

BL_Simple & Opposite_Attachment & BL_Body_Shape_? & Pointed_Tip &
BL_Base_Shape_? & Even_Base & Not_Smooth_Edge & Fan_Lobed &
Fruit_Category_?... Bel: 0.177147 Pl: 1.0

Current Candidates: HypSet(BlackMaple, MountainMaple, NorwayMaple, RedMaple,
SilverMaple, StripedMaple, SugarMaple,)

Pose query over children of: Not_Smooth_Edge(72)

Choose one to support or refute:

Wavy_Edge(73)

Toothed_Edge(76)

answer: Toothed_Edge(76)

is supported with weight: 0.9

Current best descriptions:

BL_Simple & Opposite_Attachment & BL_Body_Shape_? & Pointed_Tip &
BL_Base_Shape_? & Even_Base & Toothed_Edge & Fan_Lobed & Fruit_Category_?...
Bel: 0.177147 Pl: 1.0

Current Candidates: HypSet(MountainMaple, NorwayMaple, RedMaple, SilverMaple,
StripedMaple, SugarMaple,)

Pose query over children of: Toothed_Edge(76)

Choose one to support or refute:

Single_Toothed_Edge(77)

Double_Toothed_Edge(80)

answer: no evidence for choice

Pose query over children of: BL_Body_Shape(41) <OR>

Choose one to support or refute:

Elongated_Body(42)

Not_Elongated_Body(47)

answer: Not_Elongated_Body(47)

is supported with weight: 0.9

Current best descriptions:

BL_Simple & Opposite_Attachment & Not_Elongated_Body & Pointed_Tip &
BL_Base_Shape_? & Even_Base & Toothed_Edge & Fan_Lobed & Fruit_Category_?...

Bel: 5.7927069e-1 Pl: 1.0

Current Candidates: HypSet(MountainMaple, NorwayMaple, RedMaple, SilverMaple, StripedMaple, SugarMaple,)

Pose query over children of: Not_Elongated_Body(47)

Choose one to support or refute:

Egg_Shaped_Body(48)

Triangular_Body(49)

answer: no evidence for choice

Pose query over children of: BL_Base_Shape(58) <OR>

Choose one to support or refute:

Tapered_Base(59)

Not_Tapered_Base(63)

answer: Tapered_Base(59)

is refuted with weight:: 0.9

Current best descriptions:

BL_Simple & Opposite_Attachment & Not_Elongated_Body & Pointed_Tip &
Not_Tapered_Base & Even_Base & Toothed_Edge & Fan_Lobed & Fruit_Category_?...

Bel: 1.5394942e-1 Pl: 1.0

Current Candidates: HypSet(MountainMaple, NorwayMaple, RedMaple, SilverMaple, StripedMaple, SugarMaple,)

Pose query over children of: Not_Tapered_Base(63)

Choose one to support or refute:

Heart_Shaped_Base(64)

Rounded_Base(65)

Square_Base(66)

answer: Heart_Shaped_Base(64)

is supported with weight: 0.3

Current best descriptions:

BL_Simple & Opposite_Attachment & Not_Elongated_Body & Pointed_Tip &
Heart_Shaped_Base & Even_Base & Toothed_Edge & Fan_Lobed &
Fruit_Category_?... Bel: 3.1333407e-1 Pl: 1.0

Current Candidates: HypSet(MountainMaple, NorwayMaple, RedMaple, SilverMaple,
StripedMaple, SugarMaple,)

Pose query over children of: Fruit_Category(86) <OR>

Choose one to support or refute:

Cone_Fruit(87) <AND>
Berry_Fruit(98)
Nut_Fruit(101)
Dry_Seed_Fruit(107)
Fleshy_Fruit(111)
Other_Fruit(112)

answer: Dry_Seed_Fruit(107)

is supported with weight: 0.9

Current best descriptions:

BL_Simple & Opposite_Attachment & Not_Elongated_Body & Poied_Tip &
Heart_Shaped_Base & Even_Base & Toothed_Edge & Fan_Lobed & Dry_Seed_Fruit...
Bel: 0.177147 Pl: 1.0

Current Candidates: HypSet(MountainMaple, NorwayMaple, RedMaple, SilverMaple,
StripedMaple, SugarMaple,)

Pose query over children of: Dry_Seed_Fruit(107)

Choose one to support or refute:

Single_Dry_Seed(108)
Paired_Dry_Seeds(109)

Appendix C

130

Dry_Seed_Pod(110)

answer: Paired_Dry_Seeds(109)

is supported with weight: 0.9

Current best descriptions:

BL_Simple & Opposite_Attachment & Not_Elongated_Body & Pointed_Tip &
Heart_Shaped_Base & Even_Base & Toothed_Edge & Fan_Lobed &
Paired_Dry_Seeds... Bel: 0.177147 Pl: 1.0

Current Candidates: HypSet(MountainMaple, NorwayMaple, RedMaple, SilverMaple,
StripedMaple, SugarMaple,)

Reconsidering previous questions

Pointed_Tip(51) has new evidence.

Pose query over children of: Pointed_Tip(51)

Choose one to support or refute:

Long_Pointed_Tip(52)

Moderately_Pointed_Tip(53)

Short_Pointed_Tip(54)

answer: Moderately_Pointed_Tip(53)

is supported with weight: 0.6

Current best descriptions:

BL_Simple & Opposite_Attachment & Not_Elongated_Body &
Moderately_Pointed_Tip & Heart_Shaped_Base & Even_Base & Toothed_Edge &
Fan_Lobed & Paired_Dry_Seeds... Bel: 0.177147 Pl: 1.0

Current Candidates: HypSet(MountainMaple, NorwayMaple, RedMaple, SilverMaple,
StripedMaple, SugarMaple,)

Reconsidering previous questions

Toothed_Edge(76) has new evidence.

Pose query over children of: Toothed_Edge(76)

Choose one to support or refute:

Single_Toothed_Edge(77)

Double_Toothed_Edge(80)

answer: Single_Toothed_Edge(77)

is supported with weight: 0.3

Current best descriptions:

BL_Simple & Opposite_Attachment & Not_Elongated_Body &
Moderately_Pointed_Tip & Heart_Shaped_Base & Even_Base & Single_Toothed_Edge
& Fan_Lobed & Paired_Dry_Seeds... Bel: 0.118098 Pl: 1.0

Current Candidates: HypSet(MountainMaple, NorwayMaple, RedMaple, SilverMaple,
StripedMaple, SugarMaple,)

Pose query over children of: Single_Toothed_Edge(77)

Choose one to support or refute:

Coarse_Toothed_Edge(78)

Fine_Toothed_Edge(79)

answer: Fine_Toothed_Edge(79)

is refuted with weight: 0.3

Current best descriptions:

BL_Simple & Opposite_Attachment & Not_Elongated_Body &
Moderately_Pointed_Tip & Heart_Shaped_Base & Even_Base & Coarse_Toothed_Edge
& Fan_Lobed & Paired_Dry_Seeds... Bel: 0.0354294 Pl: 1.0

Appendix C

132

Current Candidates: HypSet(MountainMaple, NorwayMaple, RedMaple, SilverMaple, SugarMaple,)

Reconsidering previous questions

QUIT .

***** SOLUTION *****

RUNES Trace Node:

BL_Simple & Opposite_Attachment & Not_Elongated_Body
& Moderately_Pointed_Tip & Heart_Shaped_Base & Even_Base
& Coarse_Toothed_Edge & Fan_Lobed & Paired_Dry_Seeds

SUPPORT = 0.0354294, PLAUSIBILITY = 1.0)

(evidentialSupport= 2.8697814e-3, evidentialPlausibility= 1.0)

SOLUTIONS:

=====

HypSet(MountainMaple, NorwayMaple, RedMaple, SilverMaple, SugarMaple,)

Contributing Evidence:

=====

Paired_Dry_Seeds with support: 0.9 and plausibility: 1.0
in the EvidenceSpace: Fruit_Category

HypSet(AmericanWhiteBirch, AshleafMaple, BlackBirch, BlackMaple, GrayBirch,
MountainMaple, NorthernPricklyAsh, NorwayMaple, RedMaple, RiverBirch,
SilverMaple, StripedMaple, SugarMaple, SwampBirch, YellowBirch).

Not_Elongated_Body with support: 0.9 and plausibility: 1.0
in the EvidenceSpace: BL_Body_Shape

HypSet(more any 25 members).

Even_Base with support: 0.9 and plausibility: 1.0
in the EvidenceSpace: BL_Base_Symmetry

HypSet(more any 25 members)

BL_Simple with support: 0.9 and plausibility: 1.0
in the EvidenceSpace: BL_Type

HypSet(more any 25 members)

Heart_Shaped_Base with support: 0.3 and plausibility: 1.0
in the EvidenceSpace: BL_Base_Shape

HypSet(more any 25 members)

Moderately_Pointed_Tip with support: 0.6 and plausibility: 1.0
in the EvidenceSpace: BL_TipShape

HypSet(more any 25 members)

Opposite_Attachment with support: 0.9 and plausibility: 1.0
in the EvidenceSpace: BL_Attachment

HypSet(AshleafMaple, BaltimoreAsh, BasketWillow, BlackAsh, BlackMaple,
BlueAsh, CommonCatalpa, CommonLilac, GreenAsh, Horsechestnut,
MountainMaple, NorwayMaple, OhioBuckeye, PrincessTree, PumpkinAsh,
RedAsh, RedMaple, SilverMaple, StaghornSumac, StripedMaple, SugarMaple,
SweetBuckeye, WhiteAsh, WingedSumac,)

Fan_Lobed with support: 0.3 and plausibility: 1.0
in the EvidenceSpace: BL_Lobing

HypSet(BlackJackOak, BlackMaple, MountainMaple, NorwayMaple, RedMaple,
SilverMaple, StripedMaple, SugarMaple, SwampOak, Sycamore, WhitePoplar,)

Coarse_Toothed_Edge with support: 0.09 and plausibility: 1.0
in the EvidenceSpace: BL_Edge

HypSet(more any 25 members)

Tree Classification Problem

Silver Maple - Heuristic Query Generator

Begin query session: (Nov 6, 1990 14:12:31)

***** Discernment of: Leaf_Category(1) <OR> is: 8.0
***** Discernment of: Fruit_Category(86) <OR> is: 7.0

Pose query over children of: Fruit_Category(86) <OR>

Choose one to support or refute:

- Cone_Fruit(87) <AND>
- Berry_Fruit(98)
- Nut_Fruit(101)
- Dry_Seed_Fruit(107)
- Fleshy_Fruit(111)
- Other_Fruit(112)

answer: Dry_Seed_Fruit(107)

is supported with weight: 0.9

***** Discernment of: Dry_Seed_Fruit(107) is: 40.333333
***** Discernment of: Leaf_Category(1) <OR> is: 48.0

Pose query over children of: Dry_Seed_Fruit(107)

Choose one to support or refute:

- Single_Dry_Seed(108)
- Paired_Dry_Seeds(109)
- Dry_Seed_Pod(110)

answer: Paired_Dry_Seeds(109)

is supported with weight: 0.9

Current best descriptions:

Leaf_Category_? & Paired_Dry_Seeds... Bel: 0.9 Pl: 1.0

Current Candidates: HypSet(AmericanWhiteBirch, AshleafMaple, BlackBirch, BlackMaple, GrayBirch, MountainMaple, NorthernPricklyAsh, NorwayMaple, RedMaple, RiverBirch, SilverMaple, StripedMaple, SugarMaple, SwampBirch, YellowBirch,)

***** Discernment of: Leaf_Category(1) <OR> is: 20.0

Pose query over children of: Leaf_Category(1) <OR>

Choose one to support or refute:

Scale_Like_Leaf(2)

Needle_Leaf(3) <AND>

Broad_Leaf(19) <AND>

answer: Broad_Leaf(19) <AND>

is supported with weight: 0.9

Current best descriptions:

Broad_Leaf & Paired_Dry_Seeds... Bel: 0.9 Pl: 1.0

Current Candidates: HypSet(AmericanWhiteBirch, AshleafMaple, BlackBirch, BlackMaple, GrayBirch, MountainMaple, NorthernPricklyAsh, NorwayMaple, RedMaple, RiverBirch, SilverMaple, StripedMaple, SugarMaple, SwampBirch, YellowBirch,)

***** Discernment of: Broad_Leaf(19) <AND> is: 30.0

***** Discernment of: BL_Type(20) <OR> is: 20.5

***** Discernment of: BL_Attachment(36) <OR> is: 15.5

***** Discernment of: BL_Body_Shape(41) <OR> is: 21.5

***** Discernment of: BL_TipShape(50) <OR> is: 21.5

***** Discernment of: BL_Base_Shape(58) <OR> is: 19.5

***** Discernment of: BL_Base_Symmetry(67) <OR> is: 22.5

Appendix C

136

***** Discernment of: BL_Edge(70) <OR> is: 22.5

***** Discernment of: BL_Lobing(81) <OR> is: 15.5

Pose query over children of: BL_Attachment(36) <OR>

Choose one to support or refute:

Alternate_Attachment(37)

Not_Alternate_Attachment(38)

answer: Not_Alternate_Attachment(38)

is supported with weight: 0.9

Current best descriptions:

BL_Type_? & Not_Alternate_Attachment & BL_Body_Shape_? & BL_TipShape_?
& BL_Base_Shape_? & BL_Base_Symmetry_? & BL_Edge_? & BL_Lobing_? &
Paired_Dry_Seeds... Bel: 0.81 Pl: 1.0

Current Candidates: HypSet(AshleafMaple, BlackMaple, MountainMaple,
NorwayMaple, RedMaple, SilverMaple, StripedMaple, SugarMaple,)

***** Discernment of: Not_Alternate_Attachment(38) is: 12.0

***** Discernment of: BL_Base_Shape(58) <OR> is: 11.0

***** Discernment of: BL_Type(20) <OR> is: 11.0

***** Discernment of: BL_TipShape(50) <OR> is: 12.0

***** Discernment of: BL_Edge(70) <OR> is: 12.0

***** Discernment of: BL_Base_Symmetry(67) <OR> is: 12.0

***** Discernment of: BL_Body_Shape(41) <OR> is: 11.0

***** Discernment of: BL_Lobing(81) <OR> is: 11.0

Pose query over children of: BL_Base_Shape(58) <OR>

Choose one to support or refute:

Tapered_Base(59)

Not_Tapered_Base(63)

answer: Tapered_Base(59)

is refuted with weight: 0.9

Current best descriptions:

BL_Type_? & Not_Alternate_Attachment & BL_Body_Shape_? & BL_TipShape_?
& Not_Tapered_Base & BL_Base_Symmetry_? & BL_Edge_? & BL_Lobing_? &
Paired_Dry_Seeds... Bel: 0.729 Pl: 1.0

Current Candidates: HypSet(BlackMaple, MountainMaple, NorwayMaple, RedMaple,
SilverMaple, StripedMaple, SugarMaple,)

***** Discernment of: Not_Tapered_Base(63) is: 9.3333333
***** Discernment of: Not_Alternate_Attachment(38) is: 10.5
***** Discernment of: BL_TipShape(50) <OR> is: 10.5
***** Discernment of: BL_Type(20) <OR> is: 10.5
***** Discernment of: BL_Edge(70) <OR> is: 10.5
***** Discernment of: BL_Base_Symmetry(67) <OR> is: 10.5
***** Discernment of: BL_Lobing(81) <OR> is: 10.5
***** Discernment of: BL_Body_Shape(41) <OR> is: 10.5

Pose query over children of: Not_Tapered_Base(63)

Choose one to support or refute:

Heart_Shaped_Base(64)
Rounded_Base(65)
Square_Base(66)

answer: Heart_Shaped_Base(64)

is supported with weight: 0.3

Current best descriptions:

BL_Type_? & Not_Alternate_Attachment & BL_Body_Shape_? & BL_TipShape_?
& Heart_Shaped_Base & BL_Base_Symmetry_? & BL_Edge_? & BL_Lobing_? &
Paired_Dry_Seeds... Bel: 0.243 Pl: 1.0

Current Candidates: HypSet(BlackMaple, MountainMaple, NorwayMaple, RedMaple,
SilverMaple, StripedMaple, SugarMaple,)

***** Discernment of: Not_Alternate_Attachment(38) is: 10.5

Appendix C

***** Discernment of: BL_TipShape(50) <OR> is: 10.5
 ***** Discernment of: BL_Type(20) <OR> is: 10.5
 ***** Discernment of: BL_Edge(70) <OR> is: 10.5
 ***** Discernment of: BL_Base_Symmetry(67) <OR> is: 10.5
 ***** Discernment of: BL_Lobing(81) <OR> is: 10.5
 ***** Discernment of: BL_Body_Shape(41) <OR> is: 10.5

Pose query over children of: Not_Alternate_Attachment(38)

Choose one to support or refute:

Opposite_Attachment(39)

Whorled_Attachment(40)

answer: Opposite_Attachment(39)

is supported with weight: 0.9

Current best descriptions:

BL_Type_? & Opposite_Attachment & BL_Body_Shape_? & BL_TipShape_? &
 Heart_Shaped_Base & BL_Base_Symmetry_? & BL_Edge_? & BL_Lobing_? &
 Paired_Dry_Seeds... Bel: 0.243 Pl: 1.0

Current Candidates: HypSet(BlackMaple, MountainMaple, NorwayMaple, RedMaple,
 SilverMaple, StripedMaple, SugarMaple,)

***** Discernment of: BL_Type(20) <OR> is: 10.5
 ***** Discernment of: BL_TipShape(50) <OR> is: 10.5
 ***** Discernment of: BL_Edge(70) <OR> is: 10.5
 ***** Discernment of: BL_Base_Symmetry(67) <OR> is: 10.5
 ***** Discernment of: BL_Body_Shape(41) <OR> is: 10.5
 ***** Discernment of: BL_Lobing(81) <OR> is: 10.5

Pose query over children of: BL_Type(20) <OR>

Choose one to support or refute:

BL_Simple(21)

BL_Compound(22)

answer: BL_Simple(21)

is supported with weight: 0.9

Current best descriptions:

BL_Simple & Opposite_Attachment & BL_Body_Shape_? & BL_TipShape_? &
Heart_Shaped_Base & BL_Base_Symmetry_? & BL_Edge_? & BL_Lobing_? &
Paired_Dry_Seeds... Bel: 0.729 Pl: 1.0

Current Candidates: HypSet(BlackMaple, MountainMaple, NorwayMaple, RedMaple,
SilverMaple, StripedMaple, SugarMaple,)

***** Discernment of: BL_Base_Symmetry(67) <OR> is: 10.5

***** Discernment of: BL_Edge(70) <OR> is: 10.5

***** Discernment of: BL_Body_Shape(41) <OR> is: 10.5

***** Discernment of: BL_Lobing(81) <OR> is: 10.5

***** Discernment of: BL_TipShape(50) <OR> is: 10.5

Pose query over children of: BL_Base_Symmetry(67) <OR>

Choose one to support or refute:

Even_Base(68)

Uneven_Base(69)

answer: Even_Base(68)

is supported with weight: 0.9

Current best descriptions:

BL_Simple & Opposite_Attachment & BL_Body_Shape_? & BL_TipShape_? &
Heart_Shaped_Base & Even_Base & BL_Edge_? & BL_Lobing_? &
Paired_Dry_Seeds... Bel: 0.2187 Pl: 1.0

Current Candidates: HypSet(BlackMaple, MountainMaple, NorwayMaple, RedMaple,
SilverMaple, StripedMaple, SugarMaple,)

***** Discernment of: BL_Body_Shape(41) <OR> is: 10.5

***** Discernment of: BL_Edge(70) <OR> is: 10.5

***** Discernment of: BL_Lobing(81) <OR> is: 10.5

***** Discernment of: BL_TipShape(50) <OR> is: 10.5

Pose query over children of: BL_Body_Shape(41) <OR>

Choose one to support or refute:

Elongated_Body(42)

Not_Elongated_Body(47)

answer: Not_Elongated_Body(47)

is supported with weight: 0.9

Current best descriptions:

BL_Simple & Opposite_Attachment & Not_Elongated_Body & BL_TipShape_? &
Heart_Shaped_Base & Even_Base & BL_Edge_? & BL_Lobing_? &
Paired_Dry_Seeds... Bel: 0.729 Pl: 1.0

Current Candidates: HypSet(BlackMaple, MountainMaple, NorwayMaple, RedMaple,
SilverMaple, StripedMaple, SugarMaple,)

***** Discernment of: Not_Elongated_Body(47) is: 14.0

***** Discernment of: BL_Lobing(81) <OR> is: 10.5

***** Discernment of: BL_TipShape(50) <OR> is: 10.5

***** Discernment of: BL_Edge(70) <OR> is: 10.5

Pose query over children of: BL_Lobing(81) <OR>

Choose one to support or refute:

Not_Lobed(82)

Lobed(83)

answer: Lobed(83)

is supported with weight: 0.9

Current best descriptions:

BL_Simple & Opposite_Attachment & Not_Elongated_Body & BL_TipShape_? &
Heart_Shaped_Base & Even_Base & BL_Edge_? & Lobed & Paired_Dry_Seeds... Bel:
0.729 Pl: 1.0

Current Candidates: HypSet(BlackMaple, MountainMaple, NorwayMaple, RedMaple, SilverMaple, StripedMaple, SugarMaple,)

***** Discernment of: Not_Elongated_Body(47) is: 14.0

***** Discernment of: Lobed(83) is: 10.5

***** Discernment of: BL_TipShape(50) <OR> is: 10.5

***** Discernment of: BL_Edge(70) <OR> is: 10.5

Pose query over children of: Lobed(83)

Choose one to support or refute:

Fan_Lobed(84)

Feather_Lobed(85)

answer: Fan_Lobed(84)

is supported with weight: 0.3

Current best descriptions:

BL_Simple & Opposite_Attachment & Not_Elongated_Body & BL_TipShape_? &
Heart_Shaped_Base & Even_Base & BL_Edge_? & Fan_Lobed & Paired_Dry_Seeds...
Bel: 0.19683 Pl: 1.0

Current Candidates: HypSet(BlackMaple, MountainMaple, NorwayMaple, RedMaple, SilverMaple, StripedMaple, SugarMaple,)

***** Discernment of: Not_Elongated_Body(47) is: 14.0

***** Discernment of: BL_TipShape(50) <OR> is: 10.5

***** Discernment of: BL_Edge(70) <OR> is: 10.5

Pose query over children of: BL_TipShape(50) <OR>

Choose one to support or refute:

Pointed_Tip(51)

Not_Pointed_Tip(55)

answer: Pointed_Tip(51)

is supported with weight: 0.9

Current best descriptions:

BL_Simple & Opposite_Attachment & Not_Elongated_Body & Pointed_Tip &
Heart_Shaped_Base & Even_Base & BL_Edge_? & Fan_Lobed & Paired_Dry_Seeds...
Bel: 0.177147 Pl: 1.0

Current Candidates: HypSet(BlackMaple, MountainMaple, NorwayMaple, RedMaple,
SilverMaple, StripedMaple, SugarMaple,)

***** Discernment of: Not_Elongated_Body(47) is: 14.0

***** Discernment of: Pointed_Tip(51) is: 8.3333333

***** Discernment of: BL_Edge(70) <OR> is: 10.5

Pose query over children of: Pointed_Tip(51)

Choose one to support or refute:

Long_Pointed_Tip(52)

Moderately_Pointed_Tip(53)

Short_Pointed_Tip(54)

answer: Short_Pointed_Tip(54)

is refuted with weight: 0.6

Current best descriptions:

BL_Simple & Opposite_Attachment & Not_Elongated_Body & Pointed_Tip &
Heart_Shaped_Base & Even_Base & BL_Edge_? & Fan_Lobed & Paired_Dry_Seeds...
Bel: 0.177147 Pl: 1.0

Current Candidates: HypSet(BlackMaple, MountainMaple, NorwayMaple, RedMaple,
SilverMaple, StripedMaple, SugarMaple,)

***** Discernment of: Not_Elongated_Body(47) is: 14.0

***** Discernment of: BL_Edge(70) <OR> is: 10.5

Pose query over children of: BL_Edge(70) <OR>

Choose one to support or refute:

Smooth_Edge(71)

Not_Smooth_Edge(72)

answer: Not_Smooth_Edge(72)

is supported with weight: 0.9

Current best descriptions:

BL_Simple & Opposite_Attachment & Not_Elongated Body & Pointed Tip &
Heart_Shaped_Base & Even_Base & Not_Smooth_Edge & Fan_Lobed &
Paired_Dry_Seeds... Bel: 0.2187 Pl: 1.0

Current Candidates: HypSet(BlackMaple, MountainMaple, NorwayMaple, RedMaple,
SilverMaple, StripedMaple, SugarMaple,)

***** Discernment of: Not_Elongated_Body(47) is: 14.0

***** Discernment of: Not_Smooth_Edge(72) is: 9.5

Pose query over children of: Not_Smooth_Edge(72)

Choose one to support or refute:

Wavy_Edge(73)

Toothed_Edge(76)

answer: Toothed_Edge(76)

is supported with weight: 0.9

Current best descriptions:

BL_Simple & Opposite_Attachment & Not_Elongated Body & Pointed Tip &
Heart_Shaped_Base & Even_Base & Toothed_Edge & Fan_Lobed &
Paired_Dry_Seeds... Bel: 0.177147 Pl: 1.0

Current Candidates: HypSet(MountainMaple, NorwayMaple, RedMaple, SilverMaple,
StripedMaple, SugarMaple,)

***** Discernment of: Not_Elongated_Body(47) is: 12.0
 ***** Discernment of: Toothed_Edge(76) is: 9.0

Pose query over children of: Toothed_Edge(76)

Choose one to support or refute:

Single_Toothed_Edge(77)

Double_Toothed_Edge(80)

answer: no evidence for choice

***** Discernment of: Not_Elongated_Body(47) is: 12.0

Pose query over children of: Not_Elongated_Body(47)

Choose one to support or refute:

Egg_Shaped_Body(48)

Triangular_Body(49)

answer: no evidence for choice

***** No Discerning Queries remain *****

***** Discernment of: Not_Smooth_Edge(72) is: 9.0
 ***** Discernment of: BL_Edge(70) <OR> is: 9.0
 ***** Discernment of: Pointed_Tip(51) is: 8.0
 ***** Discernment of: BL_TipShape(50) <OR> is: 9.0
 ***** Discernment of: Lobed(83) is: 9.0
 ***** Discernment of: BL_Lobing(81) <OR> is: 9.0
 ***** Discernment of: BL_Body_Shape(41) <OR> is: 9.0
 ***** Discernment of: BL_Base_Symmetry(67) <OR> is: 9.0
 ***** Discernment of: BL_Type(20) <OR> is: 9.0
 ***** Discernment of: Not_Alternate_Attachment(38) is: 9.0
 ***** Discernment of: Not_Tapered_Base(63) is: 8.0
 ***** Discernment of: BL_Base_Shape(58) <OR> is: 9.0

Appendix C

145

***** Discernment of: BL_Attachment(36) <OR> is: 9.0
***** Discernment of: Leaf_Category(1) <OR> is: 8.0
***** Discernment of: Dry_Seed_Fruit(107) is: 8.0
***** Discernment of: Fruit_Category(86) <OR> is: 7.0
***** Discernment of: Toothed_Edge(76) is: 9.0
***** Discernment of: Not_Elongated_Body(47) is: 12.0

Reconsidering previous questions

Pointed_Tip(51) has new evidence.

Pose query over children of: Pointed_Tip(51)

Choose one to support or refute:

Long_Pointed_Tip(52)

Moderately_Pointed_Tip(53)

Short_Pointed_Tip(54)

answer: Moderately_Pointed_Tip(53)

is supported with weight: 0.6

Current best descriptions:

BL_Simple & Opposite_Attachment & Not_Elongated_Body &
Moderately_Pointed_Tip & Heart_Shaped_Base & Even_Base & Toothed_Edge &
Fan_Lobed & Paired_Dry_Seeds... Bel: 0.177147 Pl: 1.0

Current Candidates: HypSet(MountainMaple, NorwayMaple, RedMaple, SilverMaple,
StripedMaple, SugarMaple,)

***** No Discerning Queries remain *****

***** Discernment of: Pointed_Tip(51) is: 8.0
***** Discernment of: Not_Smooth_Edge(72) is: 9.0
***** Discernment of: BL_Edge(70) <OR> is: 9.0
***** Discernment of: Pointed_Tip(51) is: 8.0

Appendix C

***** Discernment of: BL_TipShape(50) <OR> is: 9.0
 ***** Discernment of: Lobed(83) is: 9.0
 ***** Discernment of: BL_Lobing(81) <OR> is: 9.0
 ***** Discernment of: BL_Body_Shape(41) <OR> is: 9.0
 ***** Discernment of: BL_Base_Symmetry(67) <OR> is: 9.0
 ***** Discernment of: BL_Type(20) <OR> is: 9.0
 ***** Discernment of: Not_Alternate_Attachment(38) is: 9.0
 ***** Discernment of: Not_Tapered_Base(63) is: 8.0
 ***** Discernment of: BL_Base_Shape(58) <OR> is: 9.0
 ***** Discernment of: BL_Attachment(36) <OR> is: 9.0
 ***** Discernment of: Leaf_Category(1) <OR> is: 8.0
 ***** Discernment of: Dry_Seed_Fruit(107) is: 8.0
 ***** Discernment of: Fruit_Category(86) <OR> is: 7.0
 ***** Discernment of: Toothed_Edge(76) is: 9.0
 ***** Discernment of: Not_Elongated_Body(47) is: 12.0

 Reconsidering previous questions

 Toothed_Edge(76) has new evidence.

 Pose query over children of: Toothed_Edge(76)

 Choose one to support or refute:

 Single_Toothed_Edge(77)

 Double_Toothed_Edge(80)

 answer: Single_Toothed_Edge(77)

 is supported with weight: 0.3

Current best descriptions:

 BL_Simple & Opposite_Attachment & Not_Elongated_Body &
 Moderately_Pointed_Tip & Heart_Shaped_Base & Even_Base & Single_Toothed_Edge
 & Fan_Lobed & Paired_Dry_Seeds... Bel: 0.118098 Pl: 1.0

Current Candidates: HypSet(MountainMaple, NorwayMaple, RedMaple, SilverMaple,
 StripedMaple, SugarMaple,)

***** Discernment of: Single_Toothed_Edge(77) is: 9.0

 Pose query over children of: Single_Toothed_Edge(77)

Choose one to support or refute:

Coarse_Toothed_Edge(78)

Fine_Toothed_Edge(79)

answer: Fine_Toothed_Edge(79)

is refuted with weight:: 0.3

Current best descriptions:

BL_Simple & Opposite_Attachment & Not_Elongated_Body &
 Moderately_Pointed_Tip & Heart_Shaped_Base & Even_Base & Coarse_Toothed_Edge
 & Fan_Lobed & Paired_Dry_Seeds... Bel: 0.0354294 Pl: 1.0

Current Candidates: HypSet(MountainMaple, NorwayMaple, RedMaple, SilverMaple,
 SugarMaple,)

***** No Discerning Queries remain *****

***** Discernment of: Single_Toothed_Edge(77) is: 8.5
 ***** Discernment of: Toothed_Edge(76) is: 7.5
 ***** Discernment of: Pointed_Tip(51) is: 6.6666667
 ***** Discernment of: Not_Smooth_Edge(72) is: 7.5
 ***** Discernment of: BL_Edge(70) <OR> is: 7.5
 ***** Discernment of: Pointed_Tip(51) is: 6.6666667
 ***** Discernment of: BL_TipShape(50) <Ok> is: 7.5
 ***** Discernment of: Lobed(83) is: 7.5
 ***** Discernment of: BL_Lobing(81) <OR> is: 7.5
 ***** Discernment of: BL_Body_Shape(41) <OR> is: 7.5
 ***** Discernment of: BL_Base_Symmetry(67) <OR> is: 7.5
 ***** Discernment of: BL_Type(20) <OR> is: 7.5
 ***** Discernment of: Not_Alternate_Attachment(38) is: 7.5
 ***** Discernment of: Not_Tapered_Base(63) is: 6.6666667
 ***** Discernment of: BL_Base_Shape(58) <OR> is: 7.5
 ***** Discernment of: BL_Attachment(36) <OR> is: 7.5
 ***** Discernment of: Leaf_Category(1) <OR> is: 6.6666667

Appendix C

148

***** Discernment of: Dry_Seed_Fruit(107) is: 6.6666667
***** Discernment of: Fruit_Category(86) <OR> is: 5.8333333
***** Discernment of: Toothed_Edge(76) is: 7.5
***** Discernment of: Not_Elongated_Body(47) is: 10.0

Reconsidering previous questions

Leaf_Category(1) <OR> has new evidence.

Pose query over children of: Leaf_Category(1) <OR>

Choose one to support or refute:

Scale_Like_Leaf(2)

Needle_Leaf(3) <AND>

Broad_Leaf(19) <AND>

answer: QUIT

***** SOLUTION *****

RUNES Trace Node:

BL_Simple & Opposite_Attachment & Not_Elongated_Body
& Moderately_Pointed_Tip & Heart_Shaped_Base & Even_Base
& Coarse_Toothed_Edge & Fan_Lobed & Paired_Dry_Seeds

SUPPORT = 0.0354294, PLAUSIBILITY = 1.0)

(evidentialSupport= 2.8697814e-3, evidentialPlausibility= 1.0)

SOLUTIONS:

=====

HypSet(MountainMaple, NorwayMaple, RedMaple, SilverMaple, SugarMaple,)

Contributing Evidence:

=====

Paired_Dry_Seeds with support: 0.9 and plausibility: 1.0
in the EvidenceSpace: Fruit_Category

HypSet(AmericanWhiteBirch, AshleafMaple, BlackBirch, BlackMaple, GrayBirch,
MountainMaple, NorthernPricklyAsh, NorwayMaple, RedMaple, RiverBirch,
SilverMaple, StripedMaple, SugarMaple, SwampBirch, YellowBirch)

Not_Elongated_Body with support: 0.9 and plausibility: 1.0
in the EvidenceSpace: BL_Body_Shape

HypSet(more any 25 members)

Even_Base with support: 0.9 and plausibility: 1.0
in the EvidenceSpace: BL_Base_Symmetry

HypSet(more any 25 members)

BL_Simple with support: 0.9 and plausibility: 1.0
in the EvidenceSpace: BL_Type

HypSet(more any 25 members)

Heart_Shaped_Base with support: 0.3 and plausibility: 1.0
in the EvidenceSpace: BL_Base_Shape

HypSet(more any 25 members)

Moderately_Pointed_Tip with support: 0.6 and plausibility: 1.0
in the EvidenceSpace: BL_TipShape

HypSet(more any 25 members)

Opposite_Attachment with support: 0.9 and plausibility: 1.0
in the EvidenceSpace: BL_Attachment

HypSet(AshleafMaple, BaltimoreAsh, BasketWillow, BlackAsh, BlackMaple, BlueAsh, CommonCatalpa, CommonLilac, GreenAsh, Horsechestnut, MountainMaple, NorwayMaple, OhioBuckeye, PrincessTree, PumpkinAsh, RedAsh, RedMaple, SilverMaple, StaghornSumac, StripedMaple, SugarMaple, SweetBuckeye, WhiteAsh, WingedSumac,)

Fan_Lobed with support: 0.3 and plausibility: 1.0
in the EvidenceSpace: BL_Lobing

HypSet(BlackJackOak, BlackMaple, MountainMaple, NorwayMaple, RedMaple, SilverMaple, StripedMaple, SugarMaple, SwampOak, Sycamore, WhitePoplar,)

Coarse_Toothed_Edge with support: 0.09 and plausibility: 1.0
in the EvidenceSpace: BL_Edge

HypSet(more any 25 members)

References

[Barnett81]

Barnett, J.A. "Computational Methods For A Mathematical Theory Of Evidence."
Proc. 7th Int'l Joint Conf. AI (1981):86-875.

[Dempster66]

Dempster, A.P. "Upper and Lower Probabilities Induced by a Multivalued Mapping."
Ann. Math. Stat. 38 (1966):325-339.

[Dempster68]

Dempster, A.P. "A Generalization of Bayesian Inference." *J. R. Stat. Soc. Ser. B*,
Vol. 30, (1968):205.

[Dubois&Prade80]

Dubois, D. and Prade, H. "Fuzzy Measures: Probabilities/Possibilities." *Fuzzy Sets
and Systems: Theory and Applications*, Academic Press, 1980.

[Dubois&Prade86]

Dubois, D. and Prade, H. "The Principle of Minimum Specificity as a Basis for
Evidential Reasoning." *Int. Conf. on Information Processing and Management of
Uncertainty in Knowledge-Based Systems* 30 June - 4 July 1986, Paris, (1987):75-84.

[Garey&Johnson79]

Garey, M.R. and Johnson, D.S. *Computers and Intractability, A Guide to the Theory of NP-Completeness*. New York: W.H. Freeman and Co., 1979.

[Gennari,Langley&Fisher89]

Gennari, J.H., Langley, P. and Fisher, D. "Models of Incremental Concept Formation." *Artificial Intelligence* 40 (1989):11-61.

[Gordon&Shortliffe85a]

Gordon, J. and Shortliffe, E.H. "A Method For Managing Evidential Reasoning In Hierarchical Hypothesis Spaces." *Artificial Intelligence* 26 (1985):323-358.

[Gordon&Shortliffe85b]

Gordon, J. and Shortliffe, E.H. "The Dempster-Shafer Theory of Evidence." *Rule-Based Expert Systems: The MYCIN Experiments of The Stanford Heuristic Programming Project*, eds. B.G. Buchanan and E.H. Shortliffe, Addison-Wesley, 1985.

[Jaynes57a]

Jaynes, E.T. "Information Theory and Statistical Mechanics." *Physical Review* 106, No.4, (May 1957):620-630.

[Jaynes57b]

Jaynes, E.T. "Information Theory and Statistical Mechanics. II." *Physical Review* 108, No.2, (Oct. 1957):171-190.

[Jaynes79]

Jaynes, E.T. "Where Do We Stand On Maximum Entropy?" *The Maximum Entropy Formalism*, Massachusetts Institute of Technology, (1979):15-118.

[Jaynes82]

Jaynes, E.T. *E.T. Jaynes: "Papers on Probability, Statistics and Statistical Physics"*.
ed. R.D. Rosenkrantz, Dordrecht, Holland: D. Reidel Pub., 1982.

[Klir&Folger88]

Klir, G.J. and Folger, T.A. *Fuzzy Sets, Uncertainty and Information*. State University
of New York, Binghamton: Prentice Hall, 1988.

[Kyburg87a]

Kyburg, H.E. "Bayesian and Non-Bayesian Evidential Updating." *Artificial
Intelligence* 31 (1987):271-293.

[Kyburg87b]

Kyburg, H.E. "Representing Knowledge and Evidence for Decision." *Int. Conf. on
Information Processing and Management of Uncertainty in Knowledge-Based
Systems*, 30 June - 4 July 1986, Paris, (1987):30-40.

[Lakoff87]

Lakoff, G. *"Women, Fire and Dangerous Things" What Categories Reveal about the
Mind*. Chicago: The University of Chicago Press, 1987.

[Lee88]

Lee, C.H. "A Comparison of Two Evidential Reasoning Schemes." *Artificial
Intelligence* 35 (1988):127-14.

[Orponen90]

Orponen, P. "Dempster's Rule of Combination is #P-complete." *Artificial Intelligence*
44 (1990):245-253.

[Pearl85]

Pearl, J. *Fusion, Propagation, and Structuring in Bayesian Networks*. tech report No. CSD-850022, Computer Science Department, University of California, Los Angeles, USA, 1985.

[Pearl86a]

Pearl, J. "On Evidential Reasoning in a Hierarchy of Hypotheses." *Artificial Intelligence* 28 (1986):9-15.

[Pearl86b]

Pearl, J. "Fusion, Propagation, and Structuring in Belief Networks." *Artificial Intelligence* 29 (1986):241-288.

[Pearl87]

Pearl, J. "Distributed Revision of Composite Beliefs." *Artificial Intelligence* 33 (1987):173-215.

[Pearl88]

Pearl, J. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo: Morgan Kaufmann Pub. Inc., 1988.

[Pearl90]

Pearl, J. "Bayesian and Belief-Function Formalisms for Evidential Reasoning: A Conceptual Analysis." *Readings in Uncertain Reasoning*, eds. G. Shafer and J. Pearl, San Mateo: Morgan Kaufmann Pub. Inc., pp.540-574, 1990.

[Shafer76a]

Shafer, G. *A Mathematical Theory of Evidence*. Princeton University Press, 1976.

[Shafer76b]

Shafer, G. "Hierarchical Evidence." *The Second Conference on Artificial Intelligence Applications - The Engineering of Knowledge-Based Systems*, IEEE Computer Society Press, (1985):16-25.

[Shafer84]

Shafer, G. "Probability Judgement in Artificial Intelligence and Expert Systems." *The Calculus of Uncertainty in Artificial Intelligence and Expert Systems*, Proceedings of a Conf. 28-29 Dec. 1984, (1984):7-76.

[Shafer87a]

Shafer, G. "Belief Functions and Possibility Measures." *The Analysis of Fuzzy Information* Vol.1, ed. J.C. Bizdek, CRC Press, 1987.

[Shafer&Logan87b]

Shafer, G. and Logan, R. "Implementing Dempster's Rule For Hierarchical Evidence." *Artificial Intelligence* 33 (Nov 1987):271-298.

[Shannon48]

Shannon, C.E. "A Mathematical Theory of Communication." *Bell Syst. Tech. J.* 27 (1948):379-423.

[Shastri88]

Shastri, L. *Semantic Networks: An Evidential Formalization and its Connectionist Realization*. San Mateo: Morgan Kaufmann Pub., 1988.

[Shastri89]

Shastri, L. "Default Reasoning in Semantic Networks: A Formalization of Recognition and Inheritance." *Artificial Intelligence* 39 (1989):283-355.

[Shenoy&Shafer86]

Shenoy, P.P. and Shafer, G. "Propagating Belief Functions using Local Computations." *IEEE Expert* 1, No.3, (1986):43-52.

[Shenoy&Shafer90]

Shenoy, P.P. and Shafer, G. "Axioms for Probability and Belief-Function Propagation." *Readings in Uncertain Reasoning*, eds. G. Shafer and J. Pearl, San Mateo: Morgan Kaufmann Pub., pp.575-610, 1990.

[Yager83]

Yager, R.R. "Entropy and Specificity in a Mathematical Theory of Evidence." *Int. J. General Systems* 9 (1983):249-260.

[Yager85]

Yager, R.R. "On the Relationship of Methods of Aggregating Evidence in Expert Systems." *Cybernetics and Systems: An Int. J.* 16 (1985):1-21.

[Yager86]

Yager, R.R. "Arithmetic and other Operations on Dempster-Shafer Structures." *Int. J. Man-Machine Studies* 25 (1986):357-366.

[Yager87]

Yager, R.R. "On Dempster-Shafer Framework and New Combination Rules." *Information Sciences* 41 (1987):93-137.

[Yen86]

Yen, J. "A Reasoning Model Based on Extended Dempster-Shafer Theory." *Proceedings of the 5th National Conference on Artificial Intelligence*, 1986, Philadelphia, Pa., (1986):125-131.

[Yen 89]

Yen, J. "GERTIS: A Dempster-Shafer Approach To Diagnosing Hierarchical Hypotheses." *Communications of the ACM* 32, No.5, (May 1989):573-585.

[Zadeh84]

Zadeh, L. "Review of A Mathematical Theory of Evidence, by Glenn Shafer." *The AI Magazine* 5, No.3, (1984):81.

[Zadeh86]

Zadeh, L. "A Simple View of the Dempster-Shafer Theory of Evidence and its Implication for the Rule of Combination." *The AI Magazine*, (Summer 1986):85-90.

END

03·03·92

FIN