A Probabilistic Framework for Explanation

Solomon Eyal Shimony
Ph.D. Dissertation

Department of Computer Science
Brown University
Providence, Rhode Island, 02912

Technical Report No. CS-91-57

August 1991

A Probabilistic Framework for Explanation

Technical Report CS-91-57
Computer Science Department
Brown University
Providence, RI

by
Solomon Eyal Shimony
B.Sc. Technion, 1982
M.Sc. Ben Gurion University, 1986
M.Sc. Brown University, 1989

Vita

Solomon (Shlomo) Eyal Shimony was born in Jerusalem, Israel on September 12, 1960. He graduated Cum Laude with a B.Sc. in Electronic Engineering from the Technion, Israeli Institute of Technology in 1982. He received a Master's degree in Computer Science (graphics) from Ben-Gurion University in 1986, and a Master's degree in Computer Science (artificial intelligence) from Brown University in 1989. He currently holds an academic staff position at Ben-Gurion University, Be'er-Sheva, Israel. His research interests are application of probability theory to artificial intelligence, and real-world applications of artificial intelligence. He is a member of AAAI.

Abstract

We present a wish list for a domain-independent theory of explanation under uncertainty, based on the research literature. We examine the prevalent explanation schemes in light of this list, and show why none of the currently used schemes are adequate for domain-independent explanation. In particular, we look at maximum a-posteriori (MAP) explanations (which suffer from the overspecification problem) and posterior probabilities (which may select inconsistent explanations).

We also consider one non-probabilistic scheme, cost-based abduction, but relate it to the probabilistic work by providing a probabilistic semantics for it, using Bayesian belief networks. We show that although this is a useful explanation scheme (a similar scheme, least-cost proofs, is used in Hobbs' and Stickel's TACITUS natural language understanding program), it makes independence assumptions that are not valid in the general case.

We then suggest a new theory, "irrelevance-based MAPs", a scheme where explanations are partial models, such that irrelevant variables are left unassigned. We propose and evaluate four instances of irrelevance-based MAPs: independence-based MAPs, delta independence-based MAPs, generalized delta independence-based MAPs and quasi independence-based MAPs. We show that these schemes have the advantages of MAP explanations, but do not exhibit the overspecification problem.

A best-first algorithm that generates explanations for the first two above schemes is presented. Implementations of the algorithm exhibit reasonable run-time behavior, over a toy-domain set of problems, despite the fact that the problem of finding explanations is NP-hard. We demonstrate empirically that our independence-based MAP algorithm shows reasonable performance for up to medium size problems, by experimenting on a set of randomly generated belief networks. Finally, we discuss alternate algorithms for computing irrelevance-based explanations: linear systems of inequalities, and stochastic simulation.

Acknowledgments

I would like to thank my advisor Eugene Charniak for his great support and for inspiring the beginnings for this thesis. His perseverance in tolerating my mumbling comments and flashes of idiocy are commendable. He waded through completely incomprehensible first drafts of my papers and this thesis heroically, and his help might even make this work marginally readable...

Thanks to my other committee members, Tom Dean and Philip Klein, for reducing the number of glaring errors in this thesis by a significant number, not to say that a lot of those do not still menace this text...

The Brown University Computer Science Department is an extraordinary experience for a graduate student. Be it providing ample and much required hardware and software, or an inspiring atmosphere for graduate studies. Also commendable is the opportunity for graduate students to get experience in performing tasks that are traditionally only available to faculty members, an extremely useful preparation for a future academic job.

Of great help was the AI student community at the department. In particular, I would like to mention Lynn Stein, on her advice regarding my early research at the department, and Felix Yen, Randy Callistri, and Eugene Santos Jr., who gave me a slight nudge in the right direction towards finding my thesis research topic. To the WIMP research group, Glenn Carroll and Robert Goldman, for convincing me that I probably would rather not sink into the quagmire of somebody else's software, and for other fruitful discussions.

Thanks are also due to all the lunch Bridge (and other card game) players at the department, who helped make my stay at the Brown more enjoyable, and of longer duration... To the department's secretarial and technical support staff, for their continued availability and helpfulness, goes my gratitude.

Kudos to my wife for suffering my manic-depressive states generated by lapses of will during my years at Brown, and to my parents for waiting (almost) stoically for my return from the North American continent...

This work has been supported in part by the National Science Foundation under grants IST 8416034 and IST 8515005 and Office of Naval Research under grant N00014-79-C-0529. The author was funded by a Brown University fellowship, a Corinna Borden Keen Fellowship, and various RA and TA appointments, some of which (CS52) he would dearly like to forget...

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Explanation, finding causes for observed facts (or evidence), is frequently encountered within Artificial Intelligence. For example, some researchers (see [Hobbs and Stickel, 1988], [Charniak and Goldman, 1988], [Stickel, 1988]) view understanding of natural language text as finding the facts (in an internal representation) that would explain the existence of the given text.

In automated medical diagnosis (for example the work of [Cooper, 1984], [Shachter, 1986], and [Peng and Reggia, 1987]), one wants to find the disease or set of diseases that explain the observed symptoms. In vision processing, recent research formulates the problem in terms of finding some set of objects that would explain the given image.

In vision processing, recent research formulates the problem in terms of finding some set of objects that would explain the given image. See, for example, [Geeman and Geeman, 1984] an important paper in the area, but also [Modestino and Zhang, 1989] on finding the most probable model describing a picture, and region analysis work in [Feldman and Yakimovsky, 1974].

Scientific theories are models that attempt to fit (or "predict") the given observations. Recent research in probabilistic reasoning (see [Sher, 1990]) attempt to formalize the formation of scientific theories in terms of a theory of explanation, using a maximum likelihood approach.

1.1 Toward a Comprehensive Theory

Following the method of many researchers (such as cited above), we characterize finding an explanation, as follows: given world knowledge in the form of (usually causal) rules, and observed facts (a formula), determine what needs to be assumed in order to *predict* the evidence. Additionally, we would like to select an explanation that is "optimal" in some sense.

Viewed in this way, many recognition problems can be naturally reformulated as explanation problems. That may be one reason for the prevalence of explanation within artificial intelligence. But even though many domain-dependent solutions to the problem have been explored in the literature, there is no good comprehensive theory of explanation that is sufficiently general. The main aim of this thesis is to provide a framework for such a theory in the probabilistic context.

1.1.1 What is a Good Explanation?

We begin by posing the question: what makes a good explanation? Various criteria have been raised in the literature, both for evaluating the goodness of an explanation and for evaluating an explanation system as a whole. We start out by compiling a list of desirables that a good explanation system should have (these requirements appear in papers by various researchers in the field). This list of desirables is unachievable in full, because some of the requirements may be mutually contradictory, but ideally, we would want an explanation scheme to have all of these

features, at least to some extent. A good explanation is a set of facts that has the following properties:

1. Plausibility, or likelihood: how plausible is the explanation in the world, or how likely (in a probabilistic setting)?
2. Simplicity: in essence "Occam's razor".
3. Predictiveness: the explanation should predict the evidence.
4. Relevance: this term is complicated, and not well understood. We will elaborate on this requirement later.
5. Consistency: the explanation should be internally consistent.
6. Degree of cover: as much of the evidence as possible should be explained by facts that are *distinct* from the observations.
7. Appropriate specificity: the explanation should be appropriately specific. Usually, the most specific explanation is to be preferred (for example when we have a choice of actions in a is-a hierarchy to explain some state of the world), but that is not always the case. The correct specificity depends on the domain, and on how facts in the world are represented. Therefore, we will not examine the specificity issue in this paper, except at the very end.
8. Completeness: we would like the explanation to be as detailed as possible.

Several tradeoffs are inherent within this list. For example, completeness contravenes relevance and simplicity, because if the explanation is very detailed, then it is complicated, and some parts of it may even be irrelevant. We wish to examine how current schemes of explanation fare with respect to this list. After pointing out the shortcomings of existing schemes, we suggest improved methods that overcome the problems inherent to the existing schemes.

One way of balancing the tradeoffs would be to assign numbers indicating "goodness" along each of the dimensions (the desirables). But then we may ask: how do we combine the numbers? We are not seeking to directly optimize a single explanation. Rather, we are searching for a system that provides better overall performance along as many desirables as possible, preferably all of them. Thus, we are looking for a system that performs reasonably well in all dimensions, and has no cases (problem instances) where it performs disastrously along one or more dimensions. Some explanation systems are based on other criteria (the "coherency" criterion, for example. We will discuss this criterion later), but we can still try to evaluate them in light of our compiled list.

1.1.2 Shortcomings of Existing Systems

We look at existing systems of explanation and show why they are insufficient, both in terms of our criteria and in terms of problems inherent to these systems of explanation. We claim that standard proof-theoretic schemes are not sufficiently general in that they do not take uncertainty into account, and also fail to sufficiently grade the quality of the explanations.

Proof theoretic schemes that use weights, such as least-cost proofs, neither deal with negation correctly, nor have semantics. We provide a probabilistic semantics for one such system (cost-based abduction), and examine its performance in the probabilistic setting. We show that the undesirable behavior of the system in the presence of negation carries over into the probabilistic setting, and is compounded by the invalid independence assumptions made by the system, such that inconsistent explanations may result.

We then explore current probabilistic schemes for explanation: posterior probabilities, and maximum a-priori models (MAP). We show that these systems are not adequate because they

fail to take into account the fact that an explanation has to be *relevant*. In particular, posterior probabilities may include irrelevant facts if they have high a-priori probability, and MAPs suffer from the overspecification problem, which may in turn lead to an anomalous explanation.

The MAP scheme, at least, does not suffer from potential inconsistencies. We use that as a starting point, and argue that by using a *partial* maximum a-priori model as an explanation, we can solve the overspecification problem. We use the intuition that we are *not interested* in the facts that are *irrelevant* to our observed facts. We assume for simplicity that the world knowledge is represented as a probability distribution in the form of a belief network.

1.2 Improved Explanation Schemes

We propose three ways to define the class of partial models (or assignments) that we are interested in, i.e. to decide what is irrelevant. The first attempt, independence-based assignments, uses statistical independence as a criterion for irrelevance. We then define irrelevance-based partial MAP as the highest probability irrelevance-based assignment. We show that although in some cases it alleviates the overspecification problem, it is still unstable, and sometimes assigns values to variables that we would think of as irrelevant.

The two other criteria for deciding irrelevance are more liberal in recognizing facts as irrelevant. δ-independence is a criterion that specifies that a fact is irrelevant if the given facts are independent of it within a tolerance of δ. Quasi-independence is an improved version that states that we are not interested in the value of a variable if it cannot sufficiently affect the likelihood of the explanation. We argue that quasi-independence is better than δ-independence, because it takes into account the prior probability of facts when deciding the irrelevance issue.

1.2.1 Implementation Issues

Having defined the theoretical criteria for a good explanation, and a system that works well according to the criteria (quasi-independence), we now need a way to *implement* the system. We need an effective algorithm to compute the best explanation given the world knowledge and the observed facts. By *effective* we do not mean a necessarily *efficient* algorithm, as that may be impossible. That is because the explanation problem, even in the simplest case (proof-theoretic explanation), is NP-hard. We do mean an algorithm that for practical problems is computable in reasonable time.

In this thesis, we prove that the above explanation criteria have important properties that facilitate design of an effective algorithm. Two such properties are essential to developing an effective algorithm:

1. The irrelevance-based (or δ-independence based, or quasi-independence based, respectively) partial assignment should be easily *recognizable*.

2. The probability of the above partial assignments should be easily computable.

We show that the criteria for irrelevance and the overall probability of the assignment can be computed using information local to the variables in the belief network, and is thus linear-time computable. In the case of δ-independence, this is only true in approximation, and we need to bound the error and show that it is negligible in practice. In order to do that, we propose validation experiments, using randomly generated belief networks. We implement effective algorithms for the first two explanation systems.

1.2.2 Validation

For an overall validation of our proposed explanation systems, we will show why our schemes perform at least as well as existing schemes for AND/OR belief networks, such as those generated

by the WIMP natural language understanding program. We will also perform timing experiments to show that the execution time of our algorithms is comparable to that of algorithms for existing systems of explanation, on the same class of problems. We will also construct a toy domain, and show that our schemes performs well, on explanations from this domain, where many other systems fail.

1.2.3 Contributions

In short, we contribute the following results in this thesis:

1. Provide a framework for evaluating explanation systems.
2. Provide probabilistic semantics for schemes of explanation that are originally non-probabilistic.
3. Define the meta-scheme of irrelevance-based explanation, to solve problems inherent in MAP based explanation. We investigate three sub-classes of irrelevance-based explanation:
   (a) Independence-based partial MAPs.
   (b) δ-independence based partial MAPs.
   (c) Quasi-independence based partial MAPs.
4. Generalize irrelevance-based explanations to allow for disjunctive assignments.
5. Prove the following important properties of partial assignments:
   (a) Independence-based partial assignments can be recognized in linear time.
   (b) The probability of an independence-based partial assignment is linear-time computable.
   (c) δ-independence based partial assignments can be recognized in linear time.
   (d) Reasonable bounds for the probability δ-independence based partial assignments can be computed in linear time.
6. Design effective algorithms for finding explanations in the proposed schemes, specifically independence-based MAPs and δ-independence-based MAPs.
7. We suggest an alternate algorithms for finding irrelevance-based explanations, through reduction to linear systems of inequalities, based on the ideas in [Santos Jr., 1991a], [Santos Jr., 1991d], [Santos Jr., 1991b], and [Santos Jr., 1991c].

1.3 Thesis Outline

We introduced the problem of explanation, and how it is used in various fields of Artificial Intelligence. We suggested a wish list that explanation systems should hope to achieve.

In chapter 2, we survey existing explanation systems. Pointing out the desirability of having a system that can cope with uncertainty, we examine probabilistic schemes for explanation and evaluate them in light of the wish list. We will show that existing systems fall short of the requirements, and we thus need to construct a better system.

Chapter 3 presents cost-based abduction, our variant of weighted abduction, and provide probabilistic semantics for it. Chapter 4 discusses the idea of irrelevance-based explanations, and one instantiation of it, independence-based explanations. Chapter 5 presents two improved irrelevance criteria, instances of irrelevance-based explanations: δ-independence and quasi-independence. We then allow assignment of disjunctions of values to variables, rather than just a single value, and show that this is a useful generalization of our earlier irrelevance-based explanation schemes.

Chapter 6 discusses attempts to use other basic algorithms for finding irrelevance-based explanation, and displays comparative timing results for the algorithms. The summary discusses the contributions of the proposed thesis, and suggests directions for future research.

Three appendixes are provided at the end of the thesis. Appendix A is a review of probability theory, independence and probabilistic networks. All the chapters of the thesis assume familiarity with belief networks, and thus we advise a reader unfamiliar with the concept to read appendix A first. Appendix B provides proofs that are too lengthy to include in the running text. Appendix C explains our notation and provides a notation semantics table.

Chapter 3

Cost-Based Abduction

An useful way of considering partial models is to start with a working non-probabilistic system for explanation, find a probabilistic semantics for it and evaluate the resulting model. We moved in that direction in a our paper ("Probabilistic Semantics for Cost-Based Abduction", [Charniak and Shimony, 1990b]), where we started off with Hobbs and Stickel least-cost proof [Hobbs and Stickel, 1988], modified that into cost-based abduction, and then proceeded to provide probabilistic semantics for the latter (work which was completed in [Shimony, 1990]). This chapter reviews the work done in [Charniak and Shimony, 1990b] and [Shimony, 1990]), and puts it in perspective with respect to our model of explanation.

3.1 Cost-Based Abduction and WAODAGs

A rule based explanation system with assumability costs (cost-based abduction system) has rules of the form:

$$R: \ (p_1 \ \wedge \ p_2 \ \wedge \ ... \ \wedge \ p_n)^{C_R} \ \rightarrow \ q$$

with costs $c(p_i)$ for each conjunct, and a cost C_R for applying the rule. A conjunct has the same cost in all the rules where it appears on the left hand side (LHS). The cost of proving q with this rule is the cost of all the conjuncts assumed, plus the cost of the rule. For the rest of this section and the next one, we assume without loss of generality that all rule costs are 0. We can do this by adding a p_0 (that appears nowhere else) to the LHS, with a cost $c(p_0) = c_R$. We want to find a minimal cost proof for some fact set $\mathcal{E}$ ("the evidence").

We now formalize the minimum cost proof problem as a minimization problem on a weighted AND/OR DAG (acronym WAODAG). We use a three valued logic (values $\{T, F, U\}$), augmented by symbols for keeping track of assumed nodes, versus implied nodes. The values we actually use are $Q = \{T^A, T, U, F^A, F\}$, where U stands for unassigned (intuitively: either true or false, but still undetermined), T for true, F for false, and the A superscript stands for "assumed". We use $u \searrow v$ to say that u is an immediate parent of v.

Definition 1 *A WAODAG is a 4-tuple (G, c, r, s), where:*

1. G is a connected directed acyclic graph, G = (V, E).
2. c is a function from $\{V \times Q\}$ to the non-negative reals, called the ***cost function.*** For values T, F, U, we have zero cost. $c(v) \equiv c(v, T^A)$.
3. r is a function from V to {AND, OR}, called the *label.* A node labeled AND is called an AND node, etc.

4. s is an AND node with outdegree 0 (*evidence* node).

Definition 2 *A truth assignment for a WAODAG is a function $\mathcal{A}$ from V to Q. A truth assignment is a (possibly partial)* model *iff the following conditions hold:*

1. If v is a root node (a node with in-degree 0) then $\mathcal{A}(v) \in \{T^A, U, F^A\}$.
2. If v is a non-root node, then it can only be assigned values consistent with its parents and its label (AND or OR), and if its parents do not uniquely determine the node's truth value, it can have any value in $\{T^A,\ F^A,\ U\}$.

The exact details of consistency are pursued in [Charniak and Shimony, 1990b], but should be obvious from the well-known definitions of AND and OR in 3-valued logic. Note that in our DAG, an OR node is true if at least one of its *parents* is true, as in belief networks, but *not* as commonly used for search AND/OR trees. A non-root node may still be assumed true if its parents determine that it has to be true.

Intuitively, an assignment is a model if the AND/OR constraints are obeyed. A node v where $\mathcal{A}(v) = T^A$ in an assignment, is called an *assumed true* node with respect to the assignment. Likewise for other values of $\mathcal{A}(v)$.

Definition 3 *A model for a WAODAG is* satisfying *iff $\mathcal{A}(s) \in \{T^A,\ T\}$.*

Definition 4 *The cost of an assignment $\mathcal{A}$ for a WAODAG is the sum*

$$C(\mathcal{A}) = \sum_{v \in V} c(v, \mathcal{A}(v)) \tag{3.1}$$

The Best Selection Problem is the problem of finding a minimal cost (possibly not unique) satisfying model for a given WAODAG. The Given Cost Selection Problem is that of finding a satisfying model with cost less than or equal to a given cost. Note that in a partial model, assuming a node false is useless, as such an assumption cannot contribute towards a satisfying model.

Theorem 1 *The Given Cost Selection Problem is NP-complete.*

The theorem is easily proved via a reduction from Vertex Cover, as shown in appendix B. The form of the proof provided shows that the following special cases are still NP-hard:

- Two level WAODAGs.
- WAODAGs with log(n) depth (where n is the number of nodes in the WAODAG), and in-degree at most 2.

The Best Selection Problem is clearly at least as hard as the Given Cost Selection Problem, because if we had a minimal cost satisfying model, we can find its cost in $O(|V|)$, and give an answer to the Given Cost Selection Problem. Thus, the Best Selection Problem is NP-hard.

We will now make the connection between the graphs and the rule based system. We assume that exactly all possibly relevant rule and fact instances are given. How that may be achieved is beyond the scope of this thesis.

Theorem 2 *The Best Selection Problem subsumes the problem of finding a minimal cost proof for the rule-based system with assumability costs, assuming that the rule based system is acyclic.*

Informal proof: by constructing a WAODAG (i.e. constructing the graph G, and assigning labels and costs) for the rule instance set, as follows:

Rules		Literal	Cost
R_1 :	sb $\rightarrow$ say(bank1)	rb	3
R_2 :	rb $\rightarrow$ say(bank1)	sb	2
R_3 :	rb $\wedge$ w $\rightarrow$ say(water5)	w	2
R_4 :	p $\wedge$ w $\rightarrow$ say(water5)	p	4

Table 3.1: Rules for cost-based abduction example

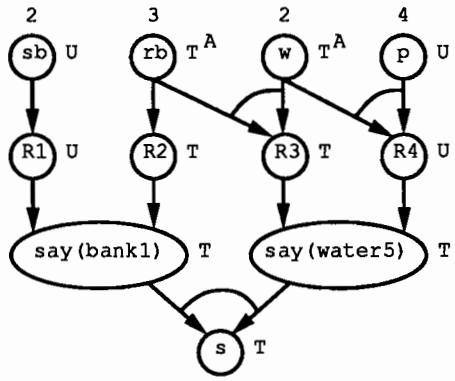

Figure 3.1: WAODAG for our example rules

1. For each literal in any rule[1] R's LHS, construct OR node v, and set $c(v, T^A)$ to the cost of the literal in the system. For each literal appearing only on the RHS of rules, construct an OR node v, with $c(v, T^A) = \infty$.

2. For each LHS of a rule R, construct and AND node v with $c(v, T^A) = \infty$, make it a parent of the node constructed for the literal on the RHS of R (in step 1), and make it a child of all the nodes constructed for the literals in the LHS of R.

3. Construct an AND node s, with parent nodes corresponding to the facts to be proved.

The cost of assuming nodes false is immaterial, so we can just set $c(v, F) = 0$ for every node v.

Example: Given the rule instances in table 3.1, used for word-sense disambiguation in natural language, with rb=river-bank(bank1), sb=savings-bank(bank1), w=water(water5), p=plant(plant7), we want to explain the evidence: say(bank1)$\wedge$say(water5).

Using the above construction, we get the WAODAG in figure 3.1, with best partial model (total cost 5) shown.

Definition 5 *A root-only assignment for a WAODAG is an assignment where only root nodes may be assumed (i.e. have values in $\{F^A, T^A\}$).*

It is possible to force root-only assignments for a WAODAG to be globally minimal by setting the cost of all non-root nodes to infinity (in practice, it suffices to set a cost greater than the sum of the root costs). We now show that given an WAODAG D, we can create a corresponding WAODAG D', such that minimal cost models of D correspond to correspond to root-only minimal cost models of D', and where the only minimal cost models of D' are root-only assignments:

Proof: by construction, as follows ($D' = D$ initially):

[1]We assume that literals with the same name in different rules are the same literal.

1. For each AND node v with cost $c(v, T^A) < \infty$ in D', construct an OR node w in D', and a new root node u, where $c(u, T^A) = c(v, T^A)$, and make both $v \searrow w$ and $u \searrow w$. Transfer all the children[2] of v to w.

2. For each OR node v with cost $c(v, T^A) < \infty$, create a new root-node u, with $c(u, T^A) = c(v, T^A)$. Make $u \searrow v$.

3. For all non-root nodes v in D', set $c(v, T^A) = \infty$.

It is clear that each time a node is selected in a minimal cost model of D to be assumed true, the node constructed from it in D' will be assumed true in some root-only minimal cost model for D', Q.E.D.

We now make the distinction between partial and complete assignments. An assignment is complete if every variable is assigned a non-U value. Formally:

Definition 6 *An assignment (or model) $\mathcal{A}$ is* complete *iff* $\forall v \in V,\ \mathcal{A}(v) \neq U$.

If an assignment assigns non-U values to some subset S of V, we say that $\mathcal{A}$ is complete w.r.t. S. An assignment is partial if some (possibly empty) subset of V is either unassigned, or assigned the value U. An assignment of U to a variable v is thus a convenient notational alternative to specifying that a variable is unassigned.

A variant of the Best Selection Problem is one of selecting a minimal cost *complete* model. Clearly, if the cost of assuming a node false is 0 for all nodes, then a minimal cost complete satisfying model will be equal in cost to the (least cost) satisfying partial model, and thus constitute a least-cost solution to the original Best Selection Problem. Unfortunately, having a cost of 0 for assuming things false will be incompatible with our probabilistic semantics. We intend to treat assumability costs as negative logarithms of probabilities (so that summing costs is akin to multiplying probabilities), and we want $P(\mathcal{A}(v) = F^A) = 1 - P(\mathcal{A}(v) = T^A)$ to hold for all root nodes, in order to comply with the axioms of probability theory. Thus, the cost of assuming a node v false is:

$$c(v,\ F^A) = -log(1 - e^{-c(v,\ T^A)}) \tag{3.2}$$

which is non-0 unless the cost of assuming v true is infinite.

3.2 Probabilistic Semantics

We now provide a probabilistic semantics for the cost based abduction system. We construct a boolean belief network out of the weighted AND/OR DAG, and show the correspondence between the solution to the Best Selection Problem and finding the MAP explanation for a given fact (or set of facts).

We assume that the rule based system is in the WAODAG format with root-only assignment. We now construct a belief network B_D from the given WAODAG D (B_D is the belief network induced by D) and show that a minimal cost satisfying complete model for D corresponds to a maximum-probability assignment of root-nodes given the evidence in the belief network B_D (where the evidence is exactly the set of facts to be proved using the rule system).

Definition 7 *The belief network B_D that is induced by D is the following network:*

1. *B_D has exactly the nodes and arcs of D. Thus, we use the same name for a node of B and the corresponding node of D. Nodes retain their labels*[3].

[2] If the AND node is s, create a new sink AND node, s', and make $w \searrow s'$.

[3] A belief network AND node has a 1 in its conditional distribution array for the case of all parents being true, and 0 elsewhere. An OR node is defined analogously.

2. *For each root node v in B_D, set $P(v = T) = e^{-c(v,\ T^A)}$.*

3. *The node s is the "evidence node". That is, our evidence $\mathcal{E}$ is the assignment $\{s = T\}$.*

Defining an assignment for the network analogously with the WAODAG assignment, we assume, without loss of generality, that we are only interested in assignment to the set of root nodes[4]. We want to find the "best" satisfying model $\mathcal{A}$, which assigns values from $\{T^A,\ F^A,\ U\}$ to the set of all root nodes, i.e. the assignment that maximizes $P(\mathcal{A} \mid \mathcal{E})$. An assignment of U to a root node means that it is omitted from the calculation of joint probabilities, as $P(v_i = U) = 1$. Intuitively, we are searching for the most probable explanation for the given evidence. This can be done by running a Bayesian network algorithm for finding MAP on the root nodes, as defined in [Pearl, 1988]. We now show the following result:

Theorem 3 *$\mathcal{A}$ is a satisfying complete model that maximizes $P(\mathcal{A} \mid \mathcal{E})$ w.r.t. belief network B_D, iff $\mathcal{A}$ is a minimal cost satisfying complete model for D.*

Proof: In a belief network, all root nodes (given no evidence) are mutually independent. Thus, for any assignment of values to root nodes, $\mathcal{A} = (a_1,\ a_2,\ ...,\ a_n)$, where $a_i = (v_i,\ q_i)$ and $q_i \in \{F^A,\ T^A,\ T,\ F\}$[5].

$$P(a_1,\ a_2, ...,\ a_n) = P(a_1)\ P(a_2)\ ...\ P(a_n) \tag{3.3}$$

However, we also have (by definition of conditional probabilities):

$$P(\mathcal{A} \mid \mathcal{E}) = \frac{P(\mathcal{E} \mid \mathcal{A})P(\mathcal{A})}{P(\mathcal{E})} \tag{3.4}$$

But as $P(\mathcal{E} \mid \mathcal{A}) = 1$ when the assignment is a satisfying model (because all nodes are strict OR and AND nodes), and 0 otherwise, and $P(\mathcal{E})$ is a constant, we can eliminate everything but $P(\mathcal{A})$ from the maximization. Also, we have:

$$P(\mathcal{A}) = \prod_{i=1}^{n} P(a_i) = \prod_{i=1}^{n} e^{-c(a_i)} = e^{-\sum_{i=1}^{n} c(a_i)} \tag{3.5}$$

Since e^x is monotonically increasing in x, we see that maximizing $P(\mathcal{A} \mid \mathcal{E})$ is equivalent to minimizing the cost of the assignment, Q.E.D.

We now generalize the DAG so that nodes can have any gating function[6]. The definition of a model is extended in the obvious way.

Theorem 4 *Given a gate-only belief network, with a single evidence node, the problem of finding the most probable complete satisfying model given the evidence is equivalent to finding a minimal cost complete model for the weighted gated DAG.*

Proof outline: The theorem follows in a manner similar to the proof theorem 3, because:

1. Root nodes are still statistically independent, because we have a belief network.
2. The probability of the evidence given a satisfying model is 1, but is 0 given any non-satisfying complete model.
3. The same cost function semantics (costs are negative logarithms of probabilities) is used.

If we want to find the best *partial* model, and are only interested in satisfying models, the above theorem still holds. It is no longer true, however, that finding the minimum cost model is equivalent to finding the MAP over the entire belief net.

[4]Maximizing the probability over assignments to root nodes is equivalent to finding the MAP, when we allow only *complete* models, because a complete assignment for the root nodes induces a unique model for all other nodes.

[5]Additionally, we use the a_i's to denote the event of node v_i having value q_i.

[6]Gate nodes are any probabilistic nodes which have only entries of 1 and 0 in their conditional distribution arrays.

3.3 Semantics for Partial Assignments

We have provided adequate semantics only for *complete* models, and showed that finding such a least-cost proof is equivalent to finding the MAP for the induced belief network. The issue of semantics for *incomplete* models remained open, however. We now provide a semantics for the partial models.

We recall that the cost based abduction system has rules of the form:

$$R: \quad (p_1 \wedge p_2 \wedge \ \ldots \ \wedge p_k)^{c_R} \ \rightarrow q$$

where q can be proved true with this rule for a cost of c_R if all of p_i are true. If some of the p_i are not known to be true (or false), they may be *assumed true* for a cost c_i. In our semantics, this c_i is the negative log probability of p_i being true in the world, if p_i does not appear as a consequent of any rule. If it does appear as a consequent, then the cost is negative log probability of p_i being true given that all its antecedents are false. This works correctly for *complete* assignments, i.e. the minimal cost complete assignment to all literals will in fact be an MAP when the costs are negative log probabilities as above.

In the semantics for *partial* models, we suggest the same probabilities as for the complete model case, except for the case where p_i appears both as an antecedent and as a consequent (of different rules). In the latter case, we just use the relevant prior probability, instead of the conditional probability. For example, if the only rule that has q on the right-hand side is $p \rightarrow q$, and there is a rule $q \rightarrow r$, the cost of assuming q will be $-log(P(q))$, not $-log(P(q|\neg p))$ as in the complete model case.

This scheme works if the rules have no negation (i.e. the resulting graph is pure AND/OR), and all the evidence is in the form of positive literals. This is true because in our construction, the only internal nodes which may be assumed true are OR nodes. The resulting best partial model will not have any nodes assigned false, since such an assignment would not contribute to an explanation (it may have many unassigned nodes).

In order to represent the vacation-planning example and other interesting explanation problems, however, we need to generalize cost-based abduction so as to allow negation and multi-valued variables. Instead of doing that, we will just note the kind of independence assumptions that cost-based abduction makes, and let them carry over to arbitrary belief networks.

We note that in the case of AND/OR networks, cost-based abduction behaves as follows. Nodes that are assigned U values are treated as if not in the graph, i.e. sets of nodes N_1 and N_2 which are not connected after removing nodes assigned U are assumed (perhaps incorrectly) independent, and the probability of the assignment will contain the product of their distribution, $P(N_1)\ P(N_2)$, instead of $P(N_1, N_2)$. Thus, when evaluating the cost (or probability) of an assignment, we essentially compute the product:

$$P'(\mathcal{A}) \ = \ \prod_{x \in nodes(\mathcal{A})} P(\mathcal{A}_{\{x\}} | \mathcal{A}_{\uparrow(x)}) \tag{3.6}$$

where $nodes(\mathcal{A})$ is the set of all the nodes assigned by $\mathcal{A}$ and $\uparrow(x)$ is the set of all the direct predecessors of x. Note that in the limiting case of complete assignments we are making no unwarranted independence assumptions, because the above product reduces to exactly the joint probability of the network, and thus we get the equivalence to finding most probable a-posteriori models. Evaluating these partial conditional probabilities in a belief network may be non-trivial (exponential time), but we are not concerned with efficiency issues at the moment.

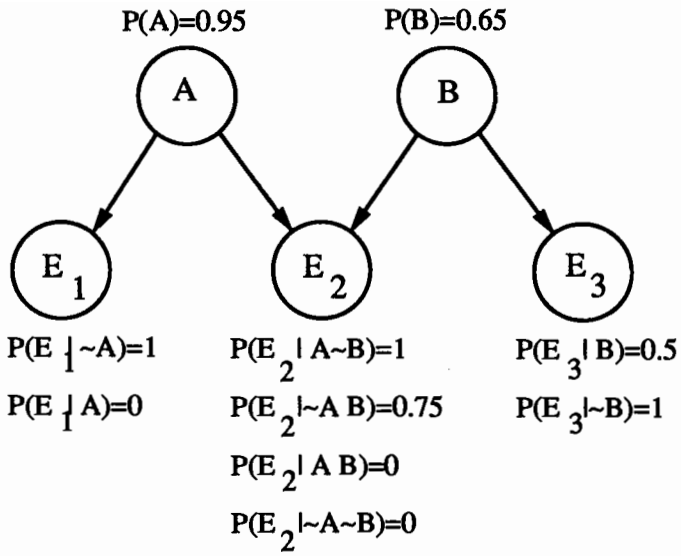

Figure 3.2: Counterexample for Cost-Based Abduction

3.4 Inadequacy of Cost-Based Abduction

Partial-model cost-based abduction will not run into the overspecification problem on our vacation-planning example[7], but the unwarranted independence assumptions will cause problems, such as selecting clearly incorrect answers in certain cases, as will be evident from the following example. Consider the following binary valued belief network of figure 3.4, with nodes A, B, E_1, E_2, E_3, and where the evidence is that all the E_i are true.

Clearly, partial model cost-based abduction will prefer the wrong choice here ($\{A = U,\ B = F\}$) over the correct choices ($\{A = F,\ B = T\}$ or $\{A = F,\ B = U\}$). The assignment $\{A = U,\ B = F\}$ is wrong because it contains no models consistent with the evidence, i.e. both models $\{A = F,\ B = F\}$ and $\{A = T,\ B = F\}$ have probability 0 given the evidence.

In terms of our wish list, cost-based abduction does well on likelihood, simplicity, predictiveness, and relevance. It does poorly on internal consistency in the presence of negation. In fact, as we showed in the example, it actually may select completely inconsistent explanations. It is moderately good on degree of cover and completeness.

[7]It will prefer {Healthy, Alive} with Vacation Spot unassigned to {Unhealthy, not Alive, No vacation}, but will also prefer not to assign Healthy at all!

Chapter 4

Irrelevance-based Explanation

In chapter 2, we showed that the overspecification problem could not be solved purely by the method of "evidential support" as Pearl suggested. In chapter 3, we showed that cost-based abduction may mishandle negation by failing to assign certain variables, because of unreasonable independence assumptions. We are still interested in an explanation scheme that, in the vacation-planning example, would leave the "Vacation Spot" variable unassigned, because our intuition would suggest that it is "irrelevant" to "Alive" in the context of being healthy.

This chapter begins with the definition of irrelevance-based assignments and explanation, and then proceeds to define an instance of these, independence-based assignments and explanations. Independence-based assignments are shown to have interesting properties, which facilitate design of an effective algorithm for computing them. We present a best-first algorithm for computing independence-based explanations, as a modification of an algorithm for finding MAPs. We then define the algorithm formally, and prove its correctness. The chapter concludes by examining how well independence-based explanations behave, and prepare the ground for improved instances of irrelevance-based explanations.

4.1 Defining Irrelevance-based Explanation

We define our notion of best probabilistic explanation for the observed facts as the most probable partial assignment (model) that ignores irrelevant variables. The criteria under which we decide which variables are irrelevant is defined formally in the following sections. For the moment, we leave that part of the definition open-ended and rely on the intuitive understanding of irrelevance. Suffice it to say that our definitions of irrelevance attempt to capture the intuitive meaning of the term.

Definition 8 *For a set of variables V, an assignment[1] $\mathcal{A}_S$ (where $S \subseteq V$), is an* irrelevance-based assignment *if the nodes $V - S$ are irrelevant to the assignment.*

In the vacation planning example, we would say that the vacation-location is irrelevant to the assignment $\{Alive, Healthy\}$.

Definition 9 *For a distribution over the set of variables V with evidence $\mathcal{E}$, an assignment $\mathcal{A}_S$ is an* irrelevance-based MAP *w.r.t. evidence $\mathcal{E}$ if it is the most probable irrelevance-based assignment that is complete with respect to the evidence nodes (i.e. all the evidence nodes are assigned by $\mathcal{A}_S$), such that $\mathcal{A}_S$ is consistent with $\mathcal{E}$.*

[1] $\mathcal{A}$ denotes assignments. The subscript denotes the set of assigned nodes. Thus, $\mathcal{A}_S$ denotes an assignment that is *complete* w.r.t. S, i.e. assigns values to *all* the nodes in S.

We use different definitions of irrelevance-based assignments to generate different versions of irrelevance-based MAPs. With the "intuitive" definition, in our vacation-planning example, the irrelevance-based MAP is $\{Alive, Healthy\}$, which is the desired scenario.

We say that the irrelevance-based MAP with respect to the evidence $\mathcal{E}$ is the best explanation for $\mathcal{E}$. Note that the definition above is not restricted to belief networks. Our formal definitions of irrelevance, however, are restricted to belief networks, and rely on the directionality of the networks, the "cause and effect" directionality. In belief networks, an arc from u to v states that u is a possible cause for v. Thus, the only possible causes of a node v are its ancestors, and thus (as in Pearl's evidential support), all nodes that are not ancestors of evidence nodes are unassigned. Additionally, we do not assign (i.e. are not "interested" in) nodes that are irrelevant to the evidence given the causes. The ancestors are only *potentially* relevant, because some other criterion may cause us to decide that they are *still* irrelevant, as shown in the next section.

4.2 Independence-Based MAPs

Probabilistic irrelevance is traditionally viewed as statistical independence, or even independence given that we know the value of certain variables. In [Pearl, 1988], a notion of independence of one set of variables from a second set of variables, given a third set of variables (all disjoint) is used. The notation used there is $I(X, Y, Z)$, to mean that variable set X is independent of variable set Z given variable set Y. If I obeys a certain set of axioms (called the "semi-graphoid" axioms), then there exists probability distribution that obeys any set of independencies. A belief network is one way to represent the distribution in an efficient form. In the belief network representation, a path-based criterion called d-separation is used to decide independence (see appendix A for details). However, neither d-separation nor independence as defined by the I notation suffice as a criterion for deciding which nodes are irrelevant. In our example, the "vacation spot" and "alive" nodes are clearly not d-separated by the "healthy" node, nor are they independent given that we know the value of the "healthy" node, as we require.

As a starting point for our notion of probabilistic irrelevance, we use Subramanian's strong irrelevance ([Subramanian, 1989]). In that paper, $SI(f, g, M)$ is used to signify that f is irrelevant to g in theory M if f is not necessary to prove g in M and vice versa (see [Subramanian, 1989] for the precise definition). We use the syntax of that form of irrelevance, but change the semantics. That is because we are interested in irrelevance of f to g even if g is not true. We define probabilistic irrelevance relative to sets of models, rather than theories (as in [Subramanian, 1989]). This is necessary because the general probabilistic representation does not have implications, just conditional probabilities[2].

Partial assignments induce a set of models. For example, for the set of variables $\{x, y, z\}$, each with a binary domain, the assignment $\{x = T, y = F\}$ with z unassigned induces the set of models $\{(x = T, y = F, z = F), (x = T, y = F, z = T)\}$. We will limit ourselves to the sets of models induced by partial assignments, and use the terms interchangeably. We say that $In(f, g|\mathcal{A})$ if f is independent of g given $\mathcal{A}$ (where $\mathcal{A}$ is a partial assignment), i.e. if $P(f|g, \mathcal{A}) = P(f|\mathcal{A})$. We allow f and g to be either sets of variables or assignments (either partial or complete) to sets of variables. If the distribution is not strictly positive, it is possible for $P(f|g, \mathcal{A})$ to be undefined, because it is possible that $P(g, \mathcal{A}) = 0$. In such cases, we choose to allow that independence does, in fact, hold. The difference between our notion of independence and that of Pearl's is that $I(X, Y, Z)$ does not require a certain assignment to Y, just that the assignment be known; whereas our notion does require it. For any disjoint sets of variables X, Y, Z, we have that $I(X, Y, Z)$ implies $In(X, Z|\mathcal{A}_Y)$, but *not* vice-versa.

[2]Belief networks can be represented in terms of implications with weighted assumption costs, but the number of implications may be exponential in the number of nodes, in the general case.

We treat assignments as sets of pairs (v, V) where v is a node and V is the value assigned to it. We define a function, *span*, which gives us the set of nodes mentioned in an assignment. Formally:

Definition 10 *Given an assignment $\mathcal{A}$ for a set of variables (or nodes), we define the* span *of $\mathcal{A}$, as:*

$$span(\mathcal{A}) = \{v | \exists d \ (v, d) \in \mathcal{A}\} \tag{4.1}$$

By definition, $span(\mathcal{A}_S) = S$. An assignment is complete with respect to node-set S if $span(\mathcal{A}) \subseteq S$.

We now define our first notion of an irrelevance-based assignment formally (we call it *independence-based assignment*, or *IB assignment* for short):

Definition 11 (IB condition) *We say that the* independence-based *condition (*IB-condition *for short) holds at node v w.r.t. assignment $\mathcal{A}_S$ iff $\mathcal{A}_{\{v\}}$ is independent of $\uparrow(v) - S$ (the set of all the ancestors of v that are* not *in S), given $\mathcal{A}_{S \cap \uparrow(v)}$.* [3]

The idea behind this definition is that the unassigned ancestors of each assigned node v should remain unassigned if they cannot affect v (and thus cannot be used to explain v). Nodes that are not ancestors of v are never used as an explanation of v anyway, because they are not potential causes of v.

Definition 12 (IB assignment) *An assignment $\mathcal{A}_S$ is an independence-based assignment iff for every node $v \in S$, the IB-condition holds at v w.r.t. $\mathcal{A}_S$.*

We define independence-based MAP as an irrelevance-based MAP where independence-based assignments are substituted for irrelevance-based assignments:

Definition 13 (IB MAP) *For a distribution over some set of variables B with evidence $\mathcal{E}$, an assignment $\mathcal{A}_S$ is an* independence-based MAP *if it is the most probable independence-based assignment that is complete with respect to the evidence nodes, such that $\mathcal{A}_S$ is consistent with $\mathcal{E}$.*

Clearly, since $\mathcal{A}^S$ assigns all the nodes assigned by $\mathcal{E}$, and is consistent with $\mathcal{E}$, then $P(\mathcal{E}|\mathcal{A}^S) = 1$, whenever $P(\mathcal{A}^S) \neq 0$.

In our example, using independence-based MAPs, we have a best scenario of (Alive, Healthy, vacation location undetermined) with a probability of 0.8 as desired. We can avoid assigning a value to vacation location because the only node v with unassigned ancestors is v=alive, and the conditional independence In(alive, vacation spot | Healthy) holds.

We say that assignment $\mathcal{A}$ subsumes assignment $\mathcal{B}$ iff $\mathcal{A} \subseteq \mathcal{B}$. This is equivalent to saying that the set of complete models satisfying $\mathcal{A}$ is a (non strict) superset of the set of complete models satisfying $\mathcal{B}$. Together with the axioms of probability theory, this implies that over any probability distribution, $P(\mathcal{A}) \geq P(\mathcal{B})$. Assignment $\mathcal{A}$ strictly subsumes assignment $\mathcal{B}$ iff $\mathcal{A}$ subsumes $\mathcal{B}$ it and $P(\mathcal{A}) > P(\mathcal{B})$. Assignment $\mathcal{A}$ properly subsumes assignment $\mathcal{B}$ iff $\mathcal{A}$ subsumes $\mathcal{B}$ and $\mathcal{A} \neq \mathcal{B}$. When looking for most probable IB assignments, we prefer assignments that are maximal w.r.t. subsumption. If the distribution is strictly positive, proper subsumption implies strict subsumption, and only a maximal IB assignment can be an IB-MAP. We take the space at this point to define evidential support, for later use. We say that an assignment is evidentially supported if all the nodes in the assignment are ancestors of some evidence node. Formally:

Definition 14 (evidential support) *An assignment $\mathcal{A}_S$ to a belief network is* evidentially supported *w.r.t. evidence $\mathcal{E}$ iff $\mathcal{E} \subseteq \mathcal{A}_S$ (thus $E \subseteq S$), and every node $v \in S$ is either in $span(\mathcal{E})$ or is in $\uparrow^+(e)$ for some $e \in span(\mathcal{E})$.* [4]

[3] $\uparrow$ is shorthand for "immediate predecessors of". Thus, $\uparrow(v)$ is the set of the direct predecessors (parents) of v.

[4] $\uparrow^+$ is the non-reflexive, transitive closure of $\uparrow$. Thus, $\uparrow^+(e)$ is the set of all the ancestors of e.

Definition 15 (proper evidential support) *An assignment $\mathcal{A}_S$ to a belief network, is* properly evidentially supported *w.r.t. evidence $\mathcal{E}$ iff it is evidentially supported w.r.t. the evidence, and for every node $v \in S$, there exists a directed path to some node in E that traverses only nodes in S.*

If an IB assignment is evidentially supported but is not properly evidentially supported, then we can get a properly evidentially supported IB assignment that subsumes it by deleting all the nodes v for which there is no path from v to E passing entirely through nodes in the assignment. In fact, as we will show in the following sections, every IB MAP is subsumed by some IB MAP that is properly evidentially supported. This implies that, for strictly positive distributions, all IB MAPs are properly evidentially supported. They are also maximal w.r.t. subsumption, as for positive distributions, subsumption implies a higher probability as well (strict subsumption). Since for positive distribution, all IB MAPs are maximal and properly evidentially supported, we need only search for IB MAPs among IB assignments that are *maximal* and properly evidentially supported. The IB MAP is a maximal probability assignment with these properties that is subsumed by the evidence. For non-strictly positive distributions the above argument does not always hold, but we still believe that it makes sense to look for the maximal (w.r.t. subsumption) IB MAPs, as that allows for a simpler explanation (fewer nodes assigned).

4.3 Properties of Independence-based Assignments

The independence constraints in the definition of independence-based assignments lead to several interesting properties, that are desirable from a computational point of view.

We make the following observation: if, for each assigned variable v, v is independent of all of its unassigned *parents* given the assignment to the rest of its parents, then the independence-based condition holds at v, i.e. v is independent of *all* its unassigned *indirect ancestors* as well as its unassigned parents. Formally:

Theorem 5 *For strictly positive distributions, if $\mathcal{A}_S$ is a complete assignment w.r.t. node set S, then for any node $v \in S$, the IB condition holds at v w.r.t. $\mathcal{A}_S$ iff $In(\mathcal{A}_{\{v\}}, \uparrow(v) - S | \mathcal{A}_{S\uparrow(v)})$.*

For a proof, see appendix B.

If the independence condition holds at every node, then the assignment is independence based, thus:

Theorem 6 *In a belief network with a strictly positive distribution, $In(\mathcal{A}_{\{v\}}, \uparrow(v) - S | \mathcal{A}_{S\cap\uparrow(v)})$ for every $v \in S$ iff $\mathcal{A}_S$ is an independence-based partial assignment.*

Proof: Immediate from theorem 5 and definition 12.

Thus, to test whether an assignment is independence-based, we only need to test the relation between each node and its parents, and can ignore all the other ancestors. Theorem 6 allows us to test whether an assignment is independence-based in time linear in the size of the network, and is thus an important theorem to use when we are considering the development of an algorithm to compute independence-based MAPs. The following theorem allows for efficient computation of $P(\mathcal{A}_S)$:

Theorem 7 *If $In(v, \uparrow(v) - S \mid \mathcal{A}_{S\cap\uparrow(v)})$ for every node $v \in S$, then the probability of $\mathcal{A}_S$ is:*

$$P(\mathcal{A}_S) = \prod_{v \in S} P(\mathcal{A}_{\{v\}} | \mathcal{A}_{S\cap\uparrow(v)}) \tag{4.2}$$

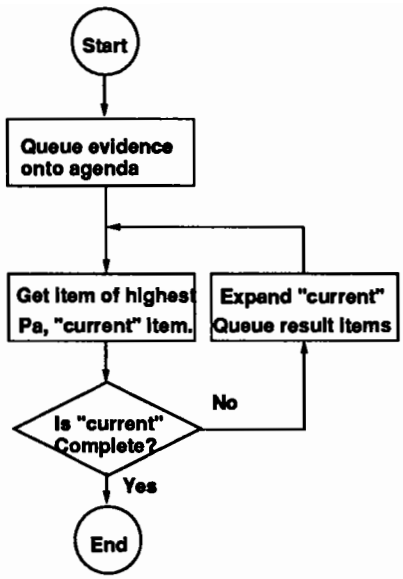

Figure 4.1: Top Level of Algorithm for Finding MAPs

For a proof, see appendix B. The theorem allows us to find $P(\mathcal{A}_S)$ in linear time for independence-based assignments, as the terms of the product are simply conditional probabilities that can be read off from the conditional distribution array (or other representation) of nodes given their parents.

Finally, we observe that for AND/OR DAGs where conditional probabilities are restricted to be either 0 or 1 (except, perhaps, at root nodes), IB MAPs are equivalent to partial model cost-based abduction, if we do not allow non-root nodes to be assumed. That is because in cost-based abduction, if a node is assigned true, then it is independent of all of its unassigned parents, i.e. the IB condition holds at that node (actually, at the belief-network image of the node). This equivalence holds even if we allow negation, as long as we do not allow non-root assumptions and all the conditional probabilities are either 0 or 1.

4.4 Independence-Based MAP Algorithm

Our independence-based MAP algorithm is based on a variant of our complete MAP algorithm, which we outlined in [Shimony, 1990]. In this section, we shall review that algorithm, and show what modifications are needed to convert it to an IB-MAP computation algorithm. Finally, we shall present the algorithm more formally, and prove its correctness.

4.4.1 Review of MAP Algorithm

An agenda of states is kept, sorted by the evaluation function P_a (estimated current probability), which is a product of all conditional probabilities seen in the current state. A state is essentially an assignment of values to some set of nodes, S. The operation of the algorithm is shown in figure 4.1.

Expansion consists of selecting a fringe node of S (i.e. a node that has neighbors not in S) and creating a new agenda item for each of the possible assignments to neighboring nodes. This is equivalent to the original description of the algorithm, presented in [Shimony and Charniak, 1990].

The heuristic evaluation function P_a for an agenda item, which is an assignment $\mathcal{A}_S$ to the set of nodes S, is the following product is computed as the evaluation function:

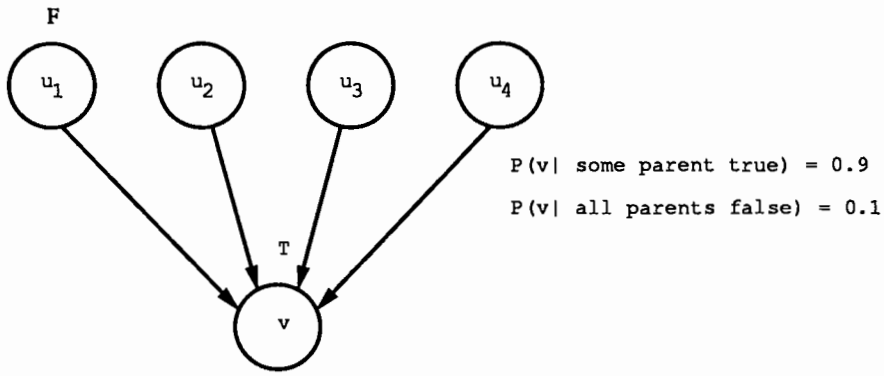

Figure 4.2: Expanding a Node

$$P_a = \prod_{v \in G(S)} P(\mathcal{A}_{\{v\}} | \mathcal{A}_{\uparrow(v)}) \tag{4.3}$$

where $G(S) = \{v | v \in S \wedge \forall w \in \uparrow(v),\ w \in S\}$, i.e. the product is over all assigned nodes which have all their parents assigned as well. The evaluation function is optimistic, and is precise for complete assignments, as the product reduces to exactly the joint distribution of the network in that case. Thus, P_a is an *admissible* heuristic evaluation function w.r.t. a best-first search algorithm.

The advantage of this best-first algorithm is that it can be easily modified to produce the next-best complete assignments in order of decreasing probability. This is done in the following manner (refer to figure 4.1): instead of ending with the first complete assignment, output it, and simply continue to loop (getting the next agenda item).

4.4.2 Algorithm Modifications

The algorithm modifications needed to compute the independence-based partial MAP are in checking whether an agenda item is completed, and in the expansion of an agenda item. The former holds for an agenda item iff it is an independence-based partial assignment. The other conditions are guaranteed because the evidence nodes are assigned initially. Checking whether an assignment is independence-based is easy, due to theorem 6.

The second modification is required because, when extending a node, some of the parents may not be assigned, as we will show presently. Also, only nodes with unassigned *parents* are considered fringe nodes, since we do not need to assign nodes with no evidence nodes below them. Completeness in the modified algorithm is different in that an agenda item may be complete even if not all variables are assigned.

To take advantage of theorem 6, we precompute for each node v a set of all the cases where conditional independence occurs. These are *independence-based hypercubes*, which are sub-spaces of the conditional distribution array (of v given its parents) with equal conditional probability entries. For example, in the case of the "dirty" OR node of figure 4.2, $P(v = T | u_i = T) = 0.9$ (for $1 \leq i \leq 4$) is independent of u_j, $j \neq i$. This defines four 3-dimensional hypercubes of "don't-care" values. We also have the one-dimensional hypercube where all $u_i = F$. When the algorithm expands v, it only assigns values to parents of which v is not independent (given the assignment to its other parents), i.e. it generates one agenda item for each such hypercube.

Naturally, since a belief net is not always a tree, some parent nodes may already be assigned. Consider, for example, figure 4.2. We are at the noisy OR node v, with parents u_1, u_2, u_3, u_4, where v has the value T, and u_1 has already been assigned F. We now have to expand all the "interesting" states of the parents of v, i.e. the states of the nodes u_i.

In the complete MAP case, we add the following 8 assignments for the nodes (u_2, u_3, u_4):

$$\{(F,F,F),(F,F,T),(F,T,F),(F,T,T),$$

$$(T,F,F),(T,F,T),(T,T,F),(T,T,T)\}$$

That is, all possible complete assignments to these three variables. When we need to find the partial MAP, however, only the following 4 assignments are added:

$$\{(T,U,U),\ (U,T,U,),\ (U,U,T),\ (F,F,F)\}$$

If a hypercube is ruled out by a prior assignment to a parent node (as is the case with the hypercube $u_1 = T$ here), it is ignored. Otherwise, the hypercubes are unified with the prior assignment, as in this case, the 3-dimensional hypercubes are reduced to 2-dimensional hypercubes by the prior assignment of $u_1 = F$. All the other assignments are "uninteresting"[5], and are not used, because they would assign values to variables that cannot change the probability of v, that is, they are subsumed by the 4 assignments listed above.

Finally, to compute next-best partial assignments in decreasing order, we perform the same simple modification as for the complete MAP algorithm: simply continue to run, producing independence based partial assignments. A useful termination condition is now a probability threshold, i.e. stop producing assignments once the probability of an assignment is below some fraction of that of the first partial MAP produced.

4.4.3 Formal Presentation of the Algorithm

We shall formalize the algorithm in terms of an input assignment $\mathcal{E}$, the evidence, and and output IB assignment. We shall define an expansion operator τ, and a termination condition, and show that the algorithm terminates with an IB-MAP.

We assume a total ordering $\mathcal{O}$ on B, the nodes of the network, such that no node comes before its (possibly indirect) descendants. With respect to that, a fringe node w is minimal in an assignment if it is the first node w.r.t. the ordering that has unassigned parents. If w is a fringe node in an assignment, such that the independence-based assignment condition holds at w w.r.t. the assignment, then it is an independence-based inactive (or just inactive, for short) fringe node. If the latter does not hold, then it is an active fringe node. If w is the first active node in the assignment, it is called a minimal active fringe node. Given an assignment and an ordering, the minimal active fringe node is unique. Unless otherwise specified, we shall assume an implicit ordering $\mathcal{O}$ is present, and define the function $index : B \to \mathcal{N}$, the index of a node w.r.t. $\mathcal{O}$.

An assignment $\mathcal{A}$ to a node w and a subset of its parents is called a *hypercube* based on w. If $\mathcal{A}$ is complete w.r.t. w and a subset S of $V = \uparrow(w)$, and $P(\mathcal{A}_{\{w\}}|\mathcal{A}_S)$ is independent of the nodes $V - S$, that is:

$$\exists p \ \ \forall \mathcal{B} \in C_{V-S} \quad P(\mathcal{A}_{\{w\}}|\mathcal{A}_S, \mathcal{B}) = p \tag{4.4}$$

(where C_{V-S} is the set of all complete assignments to $V - S$) then $\mathcal{A}$ is an independence-based hypercube (acronym IB-H), and p is the conditional probability of the hypercube.

Definition 16 *An independence-based hypercube $\mathcal{A}$ based on w is maximal if there does not exist a different independence-based hypercube $\mathcal{B}$ based on w that subsumes it (i.e. it is maximal with respect to subsumption).*

[5] An "uninteresting" assignment is one subsumed by some other assignment. For example, $A_1 = \{T, U, U\}$ subsumes $A_2 = \{T, F, U\}$ because A_1 contains all the models of A_2 and $P(v = T|u_2 = T) = P(v = T|u_2 = T, u_3 = F)$. A formal definition of subsumption appears in the next subsection.

Generally, there are several maximal IB hypercubes based on w. Note also that a maximal IB-H has the *setwise smallest* set of nodes assigned. We currently assume, for computation of hypercubes, that the distribution is positive. See appendix A as to how problems can arise when the distribution is not strictly positive, and how we can handle them.

We say that two hypercubes (or other assignments) $\mathcal{A}$ and $\mathcal{B}$ are consistent if and only if they agree on all the variables they refer to. Formally:

Definition 17 *Hypercubes $\mathcal{A}$ and $\mathcal{B}$ are consistent iff* $\forall v \in span(\mathcal{D})\ \exists! V\ (v, V) \in \mathcal{D}$, *where* $\mathcal{D} = \mathcal{A} \cup \mathcal{B}$.

If two assignments (hypercubes) are not consistent then we say that they are inconsistent.

The two following theorems explain why we are only interested in properly evidentially supported assignments. The reason is that all other assignments are subsumed by properly evidentially supported assignments:

Theorem 8 *If independence-based assignment $\mathcal{A}_S$ is subsumed by the evidence $\mathcal{E}$, but is not evidentially supported w.r.t. $\mathcal{E}$, then there exists an independence-based assignment $\mathcal{A}_{S'}$ that subsumes $\mathcal{A}_S$ and is evidentially supported w.r.t. $\mathcal{E}$.*

Proof: By construction. Since the belief network structure is a DAG, then so is any subgraph. Order nodes of S that are not ancestors of some node in E (nodes in E are considered to be ancestors here) in a list such that no node precedes its descendants (this can be done because we have a DAG). Now, proceed to eliminate nodes from the list (and from the assignment). As each node is eliminated, the assignment remains independence-based, as only nodes with no children are eliminated, and the independence-based assignment criterion for each node depends only on ancestor nodes. We can thus eliminate the entire list, and remain with an assignment that is evidentially supported, is still subsumed by $\mathcal{E}$, and is independence-based. Q.E.D.

Theorem 9 *If $\mathcal{A}_S$ is an independence-based assignment that is subsumed by $\mathcal{E}$, then there exists an independence-based assignment $\mathcal{A}_{S'}$ that subsumes $\mathcal{A}_S$ and is properly evidentially supported w.r.t. $\mathcal{E}$.*

Proof: By construction. Remove from the assignment $\mathcal{A}_S$ all nodes that are not ancestors of E as in the proof of theorem 8. Then, remove all the nodes T that have no path to a node in E that lies entirely in S, in a similar manner: sort the nodes of T into a list such that no node precedes its descendants. Removing the nodes of T will achieve a properly evidentially supported assignment, if we preserve the independence-based assignment condition. But removing the nodes of T in sequence will always preserve the criterion, because no node v is removed if it has children in the resulting assignment (if it did, then the node v would not have been in T, as there would be a path from v to a node in E). Q.E.D.

In order to define the IB MAP algorithm, we need to define an expansion operator. Let $\mathcal{P}$ be the set of all possible (either partial or complete) assignments. We define our expansion operator $\tau : \mathcal{P} \cup 2^{\mathcal{P}} \rightarrow 2^{\mathcal{P}}$, as follows:

Definition 18 *$\tau(\mathcal{F})$ consists of exactly the assignments $\mathcal{A}_S$ that obey the following conditions:*

- *If $\mathcal{F} \in \mathcal{P}$, then $\mathcal{F}$ subsumes $\mathcal{A}_S$ and there exists a minimal fringe node $w \in span(\mathcal{F})$ and a maximal IB-H $\mathcal{B}$ (based on w, with $span(\mathcal{B}) = \{w\} \cup X$), such that both the following conditions hold:*
  1. $S = span(\mathcal{F}) \cup X$
  2. $\mathcal{A}_S = \mathcal{F} \cup \mathcal{B}_{\{w\} \cup X}$

- *If $\mathcal{F} \in 2^{\mathcal{P}}$, then exists an assignment $\mathcal{A}_{S'} \in \mathcal{F}$ that subsumes $\mathcal{A}_S$, such that there exist a minimal fringe node $w \in span(\mathcal{A}_{S'})$ and a maximal IB-H $\mathcal{B}$ (based on w, with $span(\mathcal{B}) = \{w\} \cup X$), such that both the following conditions hold:*

  1. $S = S' \cup X$
  2. $\mathcal{A}_S = \mathcal{A}_{S'} \cup B_{\{w\} \cup X}$

We define τ over both assignments and sets of assignments so that we can apply τ recursively, for a simpler proof of correctness. In the actual algorithm, τ is only applied to a single assignment at a time.

Theorem 10 *If assignment $\mathcal{A}_S$ is τ-reachable from $\mathcal{E}$ then [6] it is properly evidentially supported by $\mathcal{E}$.*

Proof: By induction on the number of applications of the τ operator. The theorem clearly holds for 0 applications, as the only assignment in that case is $\{\mathcal{E}\}$, which is clearly properly evidentially supported. Now, assuming that the theorem holds for n applications of τ, then another application of τ can only assign values to nodes that are in some IB-hypercube based on a node w already assigned. IB-hypercubes assign only direct parents of w, and w is either in E or there exists a path from w to E passing only through assigned nodes, by the induction assumption. Hence, there will always by a path from the nodes assigned in the $n+1$ application of τ to E. The theorem follows by induction, Q.E.D.

An assignment $\mathcal{A}_S$ is IB-*terminated* when each assigned node $w \in S$ either has no parents, or the independence-based assignment condition holds at w. The latter is true iff the assignment for every $w \in S$, $A_{\{w\} \cup S\uparrow(v)}$ is subsumed by some IB-H based on w.

We now show that every "interesting" assignment is reachable from the evidence, using only the τ operator:

Theorem 11 *Every maximal (w.r.t. subsumption) independence-based assignment $\mathcal{A}_S$ that is properly evidentially supported w.r.t. $\mathcal{E}$ is τ-reachable from $\mathcal{E}$.*

Proof: see appendix B.

Using an agenda $\mathcal{F}$ (a set of states, or assignments), evaluation function P_a, evidence $\mathcal{E}$ and expansion operator τ, the algorithm is defined formally as follows:

Algorithm 1 (IB MAP algorithm)

1. *Set the agenda, $\mathcal{F} = \{\mathcal{E}\}$.*
2. *Set $\mathcal{A}_S$ to be a member of $\mathcal{F}$ of maximum $P_a(\mathcal{A}_S)$, and remove it from $\mathcal{F}$.*
3. *If $\mathcal{A}_S$ is IB-terminated, halt ($\mathcal{A}_S$ is an IB-MAP).*
4. *Set $\mathcal{F} = \mathcal{F} \cup \tau(\mathcal{A}_S)$, and go to step 2.*

The evaluation function P_a is exactly the same as the one for the complete MAP algorithm. It is obviously optimistic, and because of theorem 7, it is exact for IB assignments (the goal states). In implementation, P_a is actually computed before adding an assignment to the agenda, and the agenda is always kept sorted (e.g. using a heap). We now show that the algorithm is correct.

Theorem 12 *The IB-MAP algorithm terminates, and when it halts it does so with $\mathcal{A}_S$ being a most-probable properly evidentially supported IB assignment.*

[6] An assignment $\mathcal{A}$ is τ-reachable from $\mathcal{E}$ if it is in $\tau^*(\mathcal{E})$, where the asterisk stands for reflexive, transitive closure.

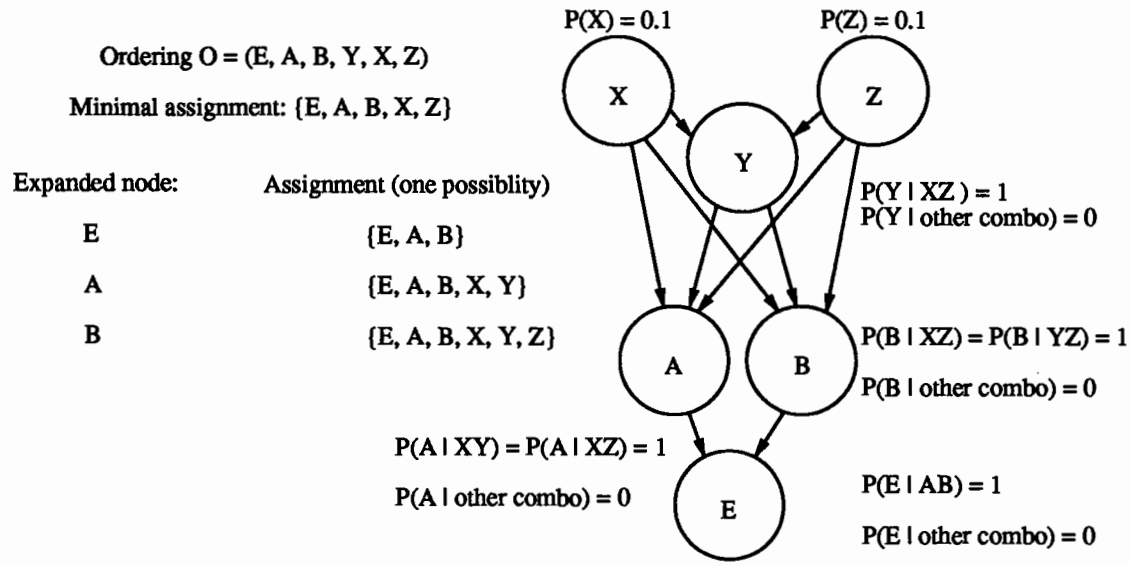

Figure 4.3: How can a non-maximal assignment occur?

Proof: The algorithm terminates, because the number of states added to the agenda in step 3 is finite, and since it always adds nodes to each assignment $\mathcal{A}_S$, it will eventually assign all the nodes above $span(\mathcal{E})$, in which case the IB condition is vacuously true. Naturally, the runtime may be exponential. The assignment found when the algorithm terminates is IB (that is the termination condition). It is properly evidentially supported (from theorem 10) and the fact that all assignments generated are τ accessible from $\mathcal{E}$. The evaluation function is admissible, and all possible maximal properly evidentially supported IB assignments are τ-accessible. The theorem follows from the latter two properties, and from the correctness condition of heuristic search w.r.t. evaluation functions, Q.E.D.

Continuing to run the algorithm after finding a first assignment will find next-best IB-assignments, in decreasing order of probability. Note that theorem 12 does not guarantee a *maximal* IB-MAP. In fact, figure 4.3 shows a simple counterexample, where all the nodes are binary, E is the evidence node, and is known to be true. Given the set of agenda states shown, the non-maximal assignment, $\{E, A, B, X, Y, Z\}$ is reached. The assignment where $\{E, A, B, X, Z\}$ subsumes the latter resulting assignment, and is both IB and properly evidentially supported.

However, for positive distributions, subsumption also implies a higher probability, which guarantees that the IB-MAP found is indeed maximal. For other distributions, to find the maximal IB-MAPs, we need to compare all IB-MAPs with equal probability, which is not hard in most cases. We are assured that the maximal IB-MAP will indeed appear if we continue to run the algorithm, because of theorem 11. We do not bother to do that, as we equate equal probability assignments as being of equal "interest" as explanations.

4.5 Evaluating Independence-Based Explanation

In terms of our wish list, irrelevance based explanation does well on likelihood, predictiveness, consistency, degree of cover and completeness. That is because likelihood, predictiveness and consistency are maximized directly, and every relevant node that is an ancestor of an evidence node is assigned. It does only moderately well on relevance, because slightly changing conditional probabilities may cause assignment to variables that are still intuitively irrelevant, which may in turn cause the wrong explanation to be preferred. The latter problem manifests if we slightly modify our vacation-planning problem, as shown in figure 4.4 and described in the following paragraph.

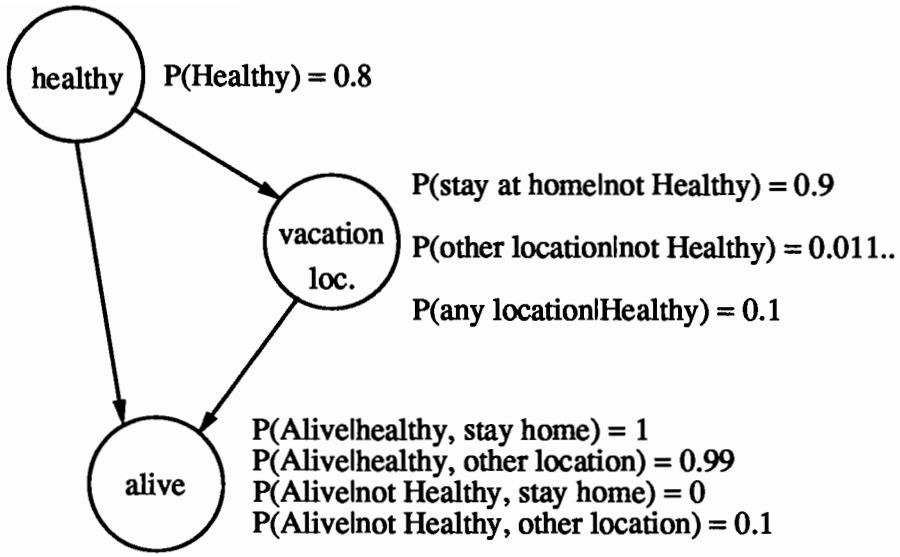

Figure 4.4: Belief network for the modified vacation-planning problem

Change the probability of being alive given the location so that staying at home (when healthy) still gives a probability of 1 of being alive, but going anywhere else only gives a probability of 0.99 of being alive (say an accident is possible during travel). We no longer have independence, and thus are forced into the bad case of finding the "not alive" scenario as the best explanation. This is counter-intuitive, and we need to find a scheme that can handle "almost" independent cases.

The instability problem shown above becomes particularly acute if the belief network is constructed using probabilities calculated from real statistical experiments. That can be done either by first constructing the topology of the network and experimenting to fill in the conditional probabilities, or by using a method such as in [Cooper and Edward, 1991] or as in [Pearl and Verma, 1991] to get the topology as well as the conditional probabilities directly from the experiments. In either case, even if exact independence exists in the real world, the conditional probabilities computed based on experiments are very unlikely to be *exactly* equal.

In chapter 5, we shall address the issue of "almost" independent cases, and how to relax the IB criterion in order to allow that. We also need to handle the problem that the IB-MAP algorithm is potentially exponential. Nothing can be done about it in the general case, because the problem is NP-hard. We can, however, see whether it executes in reasonable time in practice, and what can be done to improve practical running time, such as improving the evaluation function, or using a different algorithm altogether. We shall refer to such practical issues in chapter 6.

Chapter 5

Improved Explanation Schemes

In previous chapters, we discussed various schemes for explanation, and pointed out their merits and demerits. We now summarize the results in table 5.1.

It seems that our proposed solution, independence-based MAPs, works for the greatest number of desirables, as seen in the table. We will thus use that as a point from which to proceed, trying to improve the relevance filtering of independence-based explanation. In order to do that, we will attempt to relax the independence constraint that stands at the heart of the scheme. This will allow us to leave a larger set of variables unassigned, hopefully ones that, although not independent, are still intuitively irrelevant. We need to achieve that without jeopardizing the performance with respect to all the other desirables.

In the following sections, we suggest two ways to relax the independence criterion: δ-independence and quasi-independence. The former relaxes the independence requirement by requiring independence within a tolerance of $1 - \delta$, and considers a parent node relevant only if the statistical dependence is greater than the tolerance. Quasi-independence considers a parent node relevant only if its contribution to the probability of the node it explains is greater than its own prior probability. We consider the latter a better method theoretically, as we will show, but the former is better from a computational point of view.

5.1 δ-Independence-Based Explanation

We will preserve the performance on likelihood and consistency by keeping the maximum probability partial model selection criterion around, and changing only the criterion that decides which nodes should be assigned. IB-MAP uses an independence criterion for deciding not to assign variables,

	Posteriors	MAPs	Cost-based abduction	IBMAPs
Likelihood	Good	Good	Good	Good
Simplicity	Moderate	Bad	Good	Good
Predictiveness	Moderate	Moderate	Good	Good
Relevance	Bad	Bad	Good	Moderate
Consistency	Bad	Good	Bad	Good
Degree of Cover	Moderate	Good	Moderate	Good
Completeness	Good	Good	Moderate	Moderate

Table 5.1: Evaluation of Explanation Schemes

namely it does not assign any more variables if, for every node v in S:

$$P(\mathcal{A}_{\{v\}}|\mathcal{A}_{S\cap\pi(v)}) = P(\mathcal{A}_{\{v\}}|\mathcal{A}_{\pi(v)}) \tag{5.1}$$

There is more than one way to relax the constraint. One is to allow for approximate equality in the above equation. We call such a relaxed constraint δ-independence. With δ-independence, we require that the equality hold only within a factor of $1-\delta$, for some small δ, and then define δ-IB assignments and δ-IB MAPs based on that. In this section, we begin by formally defining δ-IB explanations. We then explore the properties of δ-IB assignments, and lastly, discuss how to adapt the IB-MAP algorithm to compute δ-IB MAPs.

5.1.1 δ-Independence Definitions

We start off by defining our notion of δ-independence:

Definition 19 *We say that A is δ-independent of B given $\mathcal{A}_S$ (written $In_\delta(A, B|\mathcal{A}_S)$ for short), where A, B and S are mutually disjoint sets of variables, iff for every assignment $\mathcal{D} \in \mathcal{C}_A$,*

$$\min_{\mathcal{B}\in\mathcal{P}_B} P(\mathcal{D}|\mathcal{A}_S, \mathcal{B}) \geq (1-\delta) \max_{\mathcal{B}\in\mathcal{P}_B} P(\mathcal{D}|\mathcal{A}_S, \mathcal{B}) \tag{5.2}$$

Note that both the minimization and maximization terms are over all *partial* assignments (as well as all the complete assignments) to B.

We expand the definition to include the case of A being a (possibly partial) *assignment* rather than a set of variables, by substituting A for $\mathcal{D}$ in the above equation. Likewise for the case of B being an assignment. This definition is parametric, i.e. δ can vary between 0 and 1.

If δ is 0, definition 19 is essentially equivalent to exact independence, as shown in the following theorem:

Theorem 13 *If the probability distribution is strictly positive and $\delta = 0$, then $In_\delta(A, B|\mathcal{A}_S)$ if and only if $In(A, B|\mathcal{A}_S)$.*

If the distribution is not strictly positive, some conditional probabilities may be undefined, as we state in appendix A, and as a result the following proof fails "on a technicality". The theorem still holds if $P(\mathcal{A}_S, \mathcal{B}) \neq 0$ for every $\mathcal{B} \in \mathcal{C}_B$.

Proof: If $\delta = 0$, the δ-independence definition becomes:

$$\min_{\mathcal{B}\in\mathcal{C}_B} P(A|\mathcal{A}_S, \mathcal{B}) = \max_{\mathcal{B}\in\mathcal{C}_B} P(A|\mathcal{A}_S, \mathcal{B}) \tag{5.3}$$

That is because the $1-\delta$ term drops, and the minimum over a set is never strictly greater than the maximum over the same set.

But the above equation holds iff all the terms: $P(A|\mathcal{A}_S, \mathcal{B})$ are equal (over the set of all $\mathcal{B} \in \mathcal{C}_B$).

We recall that, for positive distributions, all conditional probabilities are defined, and, from the axioms of probability theory:

$$P(A|\mathcal{A}_S) = \sum_{\mathcal{B}\in\mathcal{C}_B} P(A|\mathcal{B}, \mathcal{A}_S)P(\mathcal{B}|\mathcal{A}_S) \tag{5.4}$$

($\rightarrow$): If we have 0-independence then all the terms on the left hand side of the product in the summation above are equal, and can be moved outside the summation, thus

$$P(A|\mathcal{A}_S) = P(A|\mathcal{B}, \mathcal{A}_S) \sum_{\mathcal{B}\in\mathcal{C}_B} P(\mathcal{B}|\mathcal{A}_S) \tag{5.5}$$

But the summation is now just the sum of probabilities of a sample space, and is equal to 1. Thus, we have $\mathrm{In}(A, B|\mathcal{A}_S)$.

($\leftarrow$): If we have $In(A, B|\mathcal{A}_S)$, then $P(A|\mathcal{B}, \mathcal{A}_S) = P(A|\mathcal{A}_S)$ for all $\mathcal{B} \in \mathcal{C}_B$. Thus, all the $P(A|\mathcal{B}, \mathcal{A}_S)$ terms are equal to each other, in which case their minimum is equal to their maximum, and we have 0-independence according to equation 5.3, Q.E.D.

We we say that the δ-independence based condition (δ-IB condition) holds at a node v if v is δ-independent of its unassigned parents given the assignment to its assigned parents. Formally:

Definition 20 *The δ-independence based condition holds at node $v \in S$ w.r.t. assignment $\mathcal{A}_S$ iff $In_\delta(\mathcal{A}_{\{v\}}, \uparrow^+(v) - S|\mathcal{A}_{S\cap\uparrow(v)})$.*

We define a δ-independent based assignment as an assignment where the δ-IB condition holds at every node, i.e. each node is δ-independent of its unassigned ancestors given its assigned parents. Formally:

Definition 21 *An assignment $\mathcal{A}_S$ is δ-independence based iff for every $v \in S$, the δ-IB condition holds at v w.r.t. $\mathcal{A}_S$.*

Finally, we define δ-IB MAPs (or explanations) by substituting δ-IB assignments for IB assignments in the definition of IB-MAPs:

Definition 22 *A δ-IB assignment $\mathcal{A}_S$ is a δ-independence based MAP (δ-IB MAP) w.r.t. to evidence $\mathcal{E}$ iff $\mathcal{A}_S$ is evidentially supported and subsumed by $\mathcal{E}$, and there is no other δ-IB assignment assignment evidentially supported and subsumed by $\mathcal{E}$ of greater probability given the evidence.*

As in the case of IB-MAPs, $P(\mathcal{E}|\mathcal{A}_S) = 1$ whenever $P(\mathcal{A}_S) \neq 0$, because of the subsumption requirement in the definition. Thus, it is still sufficient to maximize $P(\mathcal{A}_S)$, the prior probability.

5.1.2 Properties of δ-IB Assignments

The locality theorem holds for δ-IB assignments, as it does for IB-assignments:

Theorem 14 *If, for a node v in span$(\mathcal{A}_S)$, $In_\delta(\mathcal{A}_{\{v\}}, \uparrow(v) - S|\mathcal{A}_{S\cap\uparrow(v)})$ holds, then $In_\delta(\mathcal{A}_{\{v\}}, \uparrow^+(v) - S|\mathcal{A}_{S\cap\uparrow(v)})$ holds.*

Theorem 15 *If, for every $v \in S$, $\delta - In(\mathcal{A}_{\{v\}}, \uparrow(v) - S|\mathcal{A}_{S\cap\uparrow(v)})$, then $\mathcal{A}_S$ is a δ-independence based partial assignment.*

The proofs of these two theorems can be found in appendix B.

In contrast to IB-assignments, computing the exact probability of a δ-independence based assignment seems to be hard (since we cannot use theorem 7, and would need to find posterior probabilities of non-root nodes). Fortunately, the following easily computable bound inequalities are always true:

$$P(\mathcal{A}_S) \leq \prod_{v\in S} \max_{B\in\mathcal{C}_{\uparrow(v)-S}} P(\mathcal{A}_{\{v\}}|\mathcal{A}_S, \mathcal{B}) \tag{5.6}$$

$$P(\mathcal{A}_S) \geq \prod_{v\in S} \min_{B\in\mathcal{C}_{\uparrow(v)-S}} P(\mathcal{A}_{\{v\}}|\mathcal{A}_S, \mathcal{B})$$

These bounds get better as δ approaches 0, as their ratio is at least $(1-\delta)^{|S|}$. Although these bounds are not very good from a theoretical point of view, in practice they almost always suffice to distinguish the most probable assignment from the second most probable assignment.

The upper bound is useful as an optimistic heuristic evaluation function, and we use it in our algorithm. The lower bound (together with the upper bound) is useful for comparing the probability of several δ-IB assignments, as we will see in the next subsection.

5.1.3 δ-IB MAP Algorithm

An algorithm very similar to the IB-MAP best-first search algorithm will also compute δ-IB MAPs. There are only three minor changes required in the algorithm:

1. The hypercubes we use for expansion are no longer IB hypercubes, but δ-IB hypercubes. The difference will be made evident in the following paragraphs.
2. The evaluation function changes: we now use the upper bound of equation 5.6 as an evaluation function.
3. The completion of the algorithm is different. First, we test for an assignment being δ-IB, rather than IB. Second, when we have an assignment, we have to make sure that it is indeed of highest probability. The latter was guaranteed for the IB MAP algorithm because the evaluation function was exact for IB assignments. We no longer have this property w.r.t. δ-IB assignments.

δ-IB hypercubes are similar to IB hypercubes, except that we require that a δ-independence condition hold, rather than exact independence. This implies, naturally, that every IB hypercube is also a δ-IB hypercube. Formally:

If assignment $\mathcal{A}$ is complete w.r.t. w and a set $S \subseteq \uparrow(w)$, and $P(\mathcal{A}_{\{w\}}|\mathcal{A}_S)$ is δ-independent of the nodes $\uparrow(w) - S$. Formally, if:

$$\forall \mathcal{B} \in \mathcal{C}_{\uparrow(w)-S} \quad p_{max} \geq P(\mathcal{A}_{\{w\}}|\mathcal{A}_S, \mathcal{B}) \geq p_{min} \tag{5.7}$$

where p_{min}, the lower probability bound of the hypercube, and p_{max}, the upper probability bound of the hypercube are defined as follows:

$$\begin{array}{rcl} p_{min} & = & \min_{\mathcal{B} \in \mathcal{C}_{\uparrow(w)-S}} P(\mathcal{A}_{\{w\}}|\mathcal{A}_S, \mathcal{B}) \\ p_{max} & = & \max_{\mathcal{B} \in \mathcal{C}_{\uparrow(w)-S}} P(\mathcal{A}_{\{w\}}|\mathcal{A}_S, \mathcal{B}) \end{array} \tag{5.8}$$

then $\mathcal{A}$ is a δ-IB hypercube.

For node expansion we are interested in *maximal* hypercubes:

Definition 23 *A δ-IB hypercube $\mathcal{A}$ based on w is maximal if there does not exist a different δ-IB hypercube $\mathcal{B}$ based on w that subsumes $\mathcal{A}$ (i.e. $\mathcal{A}$ is maximal with respect to subsumption).*

The evaluation function for the algorithm, H_δ, is the following product:

$$H_\delta(\mathcal{A}_S) = \prod_{v \in G} \max_{\mathcal{B} \in \mathcal{C}_{\uparrow(v)-S}} P(\mathcal{A}_{\{v\}}|\mathcal{A}_S, \mathcal{B}) \tag{5.9}$$

where $G \in S$ is the set of nodes that have already been expanded, or have no unassigned parents.

The completion of the algorithm is rather more involved. Testing whether an assignment is δ-IB is simple, we just test the δ-IB condition at each node (if the nodes are ordered, we only need to test the nodes above the minimal fringe node). But upon finding a δ-IB assignment, we have to continue running, collecting successive δ-IB assignments into a set $\mathcal{F}$. We continue to do that until the highest minimal bound of all the assignments in $\mathcal{F}$ is greater than the maximal bound of all assignments not in $\mathcal{F}$. That guarantees that the highest probability δ-IB assignment is in $\mathcal{F}$.

After $\mathcal{F}$ is stabilized, we need to find the best assignment in $\mathcal{F}$. To do that, we need to find the probability of each assignment. We do that by attaching special AND nodes to the belief network, one for each assignment. An AND node for assignment $\mathcal{A}_S$ is constructed such that it is a direct descendent of all the nodes in S, and is true iff $\mathcal{A}_S$ holds. The network is then evaluated once,

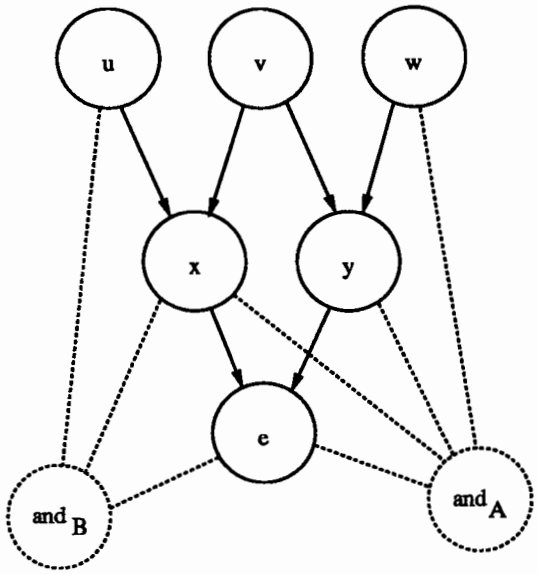

Figure 5.1: Postprocessing for δ-IB MAPs

and the exact probabilities can then be retrieved as the probability of these extra nodes, for all the assignments in $\mathcal{F}$.[1]

For example, consider the network of figure 5.1, ignoring the parts in dotted lines for the moment. Suppose that our evidence is $\{e = T\}$, and that there are two assignments in $\mathcal{F}$:

$$\begin{aligned} \mathcal{A} &= \{(e,T),(x,F),(y,T),(w,t)\} \\ \mathcal{B} &= \{(e,T),(x,T),(u,T)\} \end{aligned}$$

These two assignments could be in $\mathcal{F}$ because, $p_{max}(\mathcal{A}) > p_{max}(\mathcal{B}) > p_{min}(\mathcal{A})$, and both the p_{min} values are greater than the p_{max} of any other assignment. The evaluation of the exact probabilities of the assignments, $P(\mathcal{A})$ and $P(\mathcal{B})$ is done in parallel by adding the nodes and_A and and_B to the diagram, as shown in 5.1 (dotted lines and circles). and_A is a binary node a conditional probability function as follows:

$$P(and_A = T) = \begin{cases} 1 & \text{if } \{e = T, x = F, y = T, w = t\} \\ 0 & \text{otherwise} \end{cases}$$

Clearly, and_A is true if and only if the event $\mathcal{A}$ occurs, and thus $P(and_A = T) = P(\mathcal{A})$. A similar construction for and_B assures that $P(and_B = T) = P(\mathcal{B})$. Evaluating the network computes the probabilities $P(and_A = T)$ and $P(and_B = T)$, as desired (even though, as we stated before, this may take exponential time).

5.1.4 δ-IB Explanation: Evaluation

As can be clearly seen, δ-IB MAP explanation handles cases where nodes are irrelevant, even if the assignments are *almost* statistically independent of these nodes. In our modified vacation-planning example, the δ-IB MAP is {Alive, Healthy}, with a probability of approximately 0.2, for any $\delta \geq 0.01$. That is because Alive is δ-Independent of the vacation location given Healthy, permitting {Alive, Healthy} to be a δ-IB assignment. Despite this advantage, a major drawback

[1]The post processing step may take exponential time, unfortunately, as this problem is NP-hard. In many cases (whenever one explanation stands out as much more likely than the next best explanations), it will be the case that $|\mathcal{F}| = 1$ obviating the need for the post processing step.

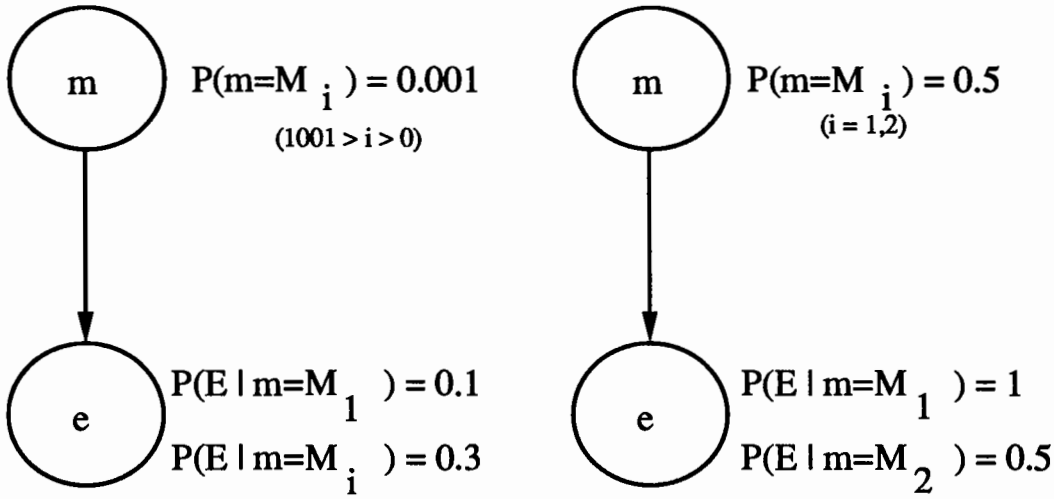

Figure 5.2: Where δ-independence Fails

with δ-independence is that δ has to be specified in a manner external to the network, and we would rather avoid that if at all possible. Also, we might like to be able to use a variable δ, for the following reason.

Consider a network consisting of only two nodes, as in figure 5.2. The evidence node is binary, and is known to be true. Node m has 1000 equally likely (a-priori) states, (as in figure 5.2a). The conditional probability of the evidence given that $m = M_i$ is 0.3 for all $i > 1$ and 0.1 for $i = 1$. Do we want to conclude that the explanation for the evidence is that $m = m_i$, for some $i \geq 2$? Probably not, because the model where $m = m_i$ is a very low probability model, given the evidence. In order to leave m unassigned when using δ-independence, we will need $\delta \geq \frac{2}{3}$. Suppose, then, that we set $\delta = \frac{2}{3}$.

Let us now look at a slightly modified diagram, shown in figure 5.2b. Now m has only 2 states, both equally likely, where $P(E|m = M_1) = 0.5$ and $P(E|m = M_2) = 1$. But δ-independence, with $\delta = \frac{2}{3}$) will *still* leave m unassigned, whereas intuitively we would wish to assign $m = M_2$. Our next explanation scheme will deal with this problem.

5.2 Quasi-Independence-Based MAPs

We now discuss a method of relaxing the independence criterion that is *not* based on some arbitrary imposed measure, as is δ-independence. The method is suggested by the following properties of cost-based abduction:

1. Cost-based abduction prefers not to assign a set of variables O whenever for every $\mathcal{B} \in \mathcal{P}_O$, $cost(\mathcal{A}_S) \leq cost(\mathcal{A}_S, \mathcal{B})$ holds.

2. In cost-based abduction, costs are negative logarithms of probabilities.

The equivalent criterion to cost-based abduction in terms of probabilities does not assign the O nodes if:

$$P'(\mathcal{A}_S) \geq \max_{\mathcal{B} \in \mathcal{P}_O} P(\mathcal{A}_S, \mathcal{B}) \tag{5.10}$$

In evaluating the "probability" $P'(\mathcal{A}_S)$, probability is calculated assuming that the O nodes are not in the diagram (hence the "primed" probability function P'), a bad independence assumption.

But it is precisely this independence assumption that allows us to do "cost sharing" and assign *some* nodes that are not in the evidence.

It seems more appropriate to replace this ad-hoc independence assumption by a different probabilistic criterion. Such a criterion is achieved by making the cost-based abduction criterion for leaving nodes unassigned more restrictive, in such a way that it is still weaker than the independence-based condition. We obey both these constraints by substituting $\min_{\mathcal{B}\in\mathcal{P}_O} P(\mathcal{A}_S|\mathcal{B})$ for $P(\mathcal{A}_S)$ (the first term is never greater than the second term) in equation 5.10.

This new criterion, besides having the required properties with respect to to cost-based abduction and the independence-based condition, makes sense because the minimization term is the lowest probability model (given $\mathcal{B}$) that we could get by adversely assigning the O nodes. The most probable assignment achievable by assigning the O variables is $\max_{\mathcal{B}\in\mathcal{P}_O} P(\mathcal{A}_S, \mathcal{B})$, as we saw earlier. Note also that $P(\mathcal{A}_S, \mathcal{B}) = P(\mathcal{A}_S|\mathcal{B})P(\mathcal{B})$, where $P(\mathcal{A}_S|\mathcal{B})$ evaluates how much $\mathcal{B}$ predicts $\mathcal{A}_S$, and $P(\mathcal{B})$ is the prior probability of $\mathcal{B}$, which means that we essentially scale the contribution of $\mathcal{B}$ by its prior probability. We need to do that because $\mathcal{B}$ is a good explanation to $\mathcal{A}_S$ only if it is itself sufficiently probable (unless the $\mathcal{A}_S$ cannot be reasonably explained in any other way).

Restructuring our criterion, so as to get a definition more akin to irrelevance-based assignments, we get the following criterion:

Definition 24 *The* quasi-independence *condition holds at a node v w.r.t. assignment $\mathcal{A}_S$ iff:*

$$\min_{\mathcal{B}\in\mathcal{P}_{\uparrow^+(v)-S}} P(\mathcal{A}_{\{v\}}|\mathcal{A}_{S\cap\uparrow(v)}, \mathcal{B}) \geq \max_{\mathcal{B}\in\mathcal{P}_{\uparrow^+(v)-S}} P(\mathcal{A}_{\{v\}}, \mathcal{B}|\mathcal{A}_{S\cap\uparrow(v)}) \quad (5.11)$$

Note that this is a weakening of the IB condition. That is because if A_S is an irrelevance-based assignment, then the left hand side of the above inequality is equal to $P(\mathcal{A}_{\{v\}}|\mathcal{A}_{S\uparrow(v)})$ for every $\mathcal{A}_{\uparrow(v)-S}$, and that is greater or equal to the right-hand-side of the inequality (because $P(x) \geq P(x \wedge y)$ is a tautology).

We proceed to define quasi-independence based assignments akin to other irrelevance-based assignments:

Definition 25 *An assignment $\mathcal{A}_S$ is quasi-independence based iff the quasi-independence condition holds at every $v \in S$.*

Lastly, we define δ-independence based partial MAPs and quasi-independence based MAPs, as follows:

Definition 26 *An assignment $\mathcal{A}$ is a δ-independence (or quasi-independence) based partial MAP if it is a δ-independence (or quasi-independence, respectively) based assignment which subsumes the evidence, such that for any other assignment $\mathcal{A}'$ that obeys these conditions, $P(\mathcal{A}) \geq P(\mathcal{A}')$.*

Evaluating the left-hand side of equation 5.11 is easy. Unfortunately, evaluating the right-hand side seems to be hard in the general case (although it is easy for polytrees). Neither does a locality theorem (such as theorem 5) hold for the quasi-IB condition. For example, consider figure 5.3, where the evidence is $\{e = T\}$ ($\{E\}$ in shorthand). Now, we have that $P(E|X)$ and $P(E|\neg X)$ are both greater than $P(E, X)$ and $P(E, \neg X)$. This means that the "local" quasi-IB condition holds (i.e. where assignments to the the non-immediate ancestor, y, are not considered). But this does not imply that equation 5.11 holds, because we have that $P(E|Y) < P(E, \neg Y)$.

Because of the above properties, we suspect that even recognizing a quasi-independence based assignment is hard, which makes it unlikely that we can construct an effective algorithm for computing quasi-independence based MAPs.

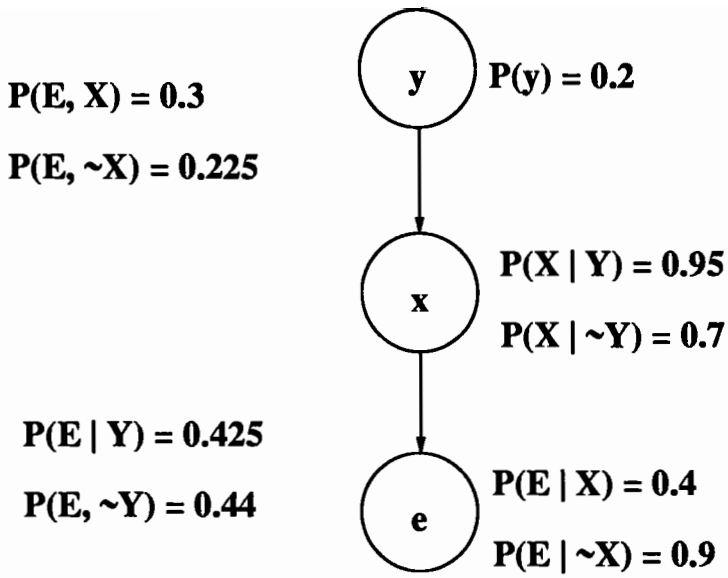

Figure 5.3: Why Locality Fails for Quasi-Independence

5.3 Evaluation of Relaxed Independence

Both δ-independence and the quasi-independence schemes solve the modified vacation-planning problem. Theoretically, quasi-independence is superior to δ-independence. Consider again the network of figure 5.2, repeated for convenience as figure 5.4. We have already seen how δ-independence fails if we have to use the same value of δ in the networks of figure 5.4. Quasi-independence, however, will (correctly) not assign the m node in figure 5.4a, and correctly assign $m = M_2$ in 5.4b, as desired.

The quasi-IB condition is neither stronger nor weaker than the δ-IB condition. It may be of interest to note, however, that the quasi-IB condition behaves in a manner similar to the δ-IB condition, if we made δ dependent on the prior probability of a node's parents. For example, in the two node case of figure 5.4b, as long as $P(E|m = M_1) \geq P(E|m = M_2)$, the quasi-IB condition is equivalent to the δ-IB condition if we let $\delta = P(m = M_2)$. As $P(E|m = M_1) \geq P(E|m = M_2)$ is true in figure 5.4b, making $\delta = P(m = M_2) = 0.5$ makes the δ-IB condition equivalent to the quasi-IB condition for the diagram. The equivalence holds even if we changed the probabilities $P(E|m = M_1)$ and $P(E|m = M_2)$, as long as $P(E|m = M_1) \geq P(E|m = M_2)$ still holds.

One may now ask: why did we introduce δ-independence in the first place, given that it is inferior to quasi-independence? The answer is that:

1. δ-independence is still a better criterion than the exact independence-based criterion.
2. δ-independence may still be useful in practice, and is better computationally.
3. In the next section, we generalize δ-independence, by allowing disjunctive assignments. This generalization alleviates the problem illustrated in figure 5.4.

5.4 Specificity in Explanations

In this section, we generalize irrelevance-based explanations by allowing disjunctive assignments. We show that, in essence, disjunctive assignments have been proposed before, in order to deal with the specificity of an explanation with respect to a is-a hierarchy. It turns out that we can deal with these aspects of explanation specificity as well as find a criterion that is better than δ-independence,

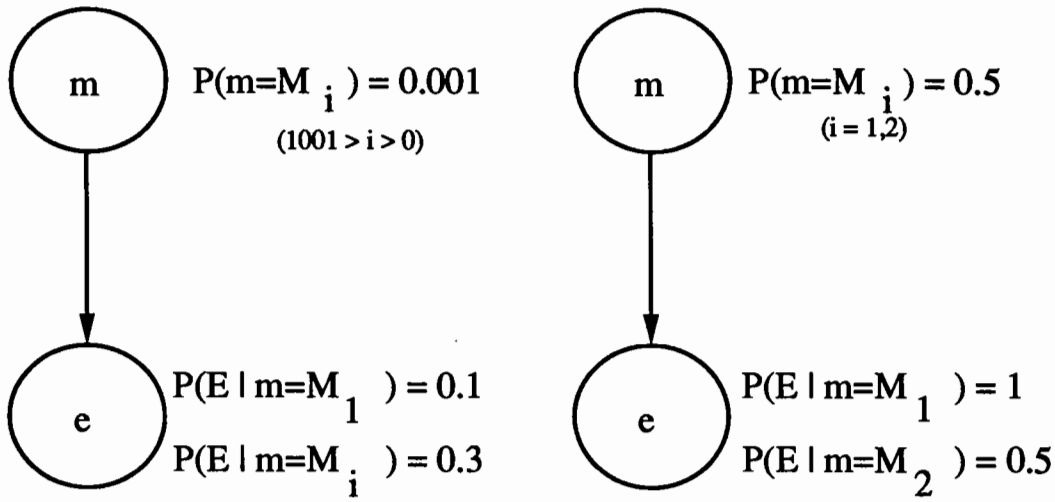

Figure 5.4: δ-independence versus Quasi-independence

by allowing disjunctive assignments. In addition, the new criterion proposed in this section has better computational complexity properties than quasi-independence.

5.4.1 Value Aggregation for Specificity

A method proposed by Goldman and Charniak for the new version of WIMP allows aggregation of node values into a single value. This makes sense only if a node has more than 2 states. The kind of specificity that this scheme handles is specificity of event description w.r.t. some hierarchical knowledge base of events, such as the one in WIMP. For example, suppose that one event type is "shopping", and that there are events lower down in the hierarchy, "supermarket shopping", "liquor-store shopping", etc. that are subtypes of "shopping". In the belief network representation, a multiple valued node consisting of all possible events is used. Posterior probabilities are computed. If the probabilities of the individual subtypes of shopping events is low, one may still aggregate all these into a single value that corresponds to "shopping", and if that has a high probability, a decision on the shopping explanation can be made. In this example, the system selects a less specific explanation (less specific, at least, than a particular subtype of shopping), in order to get a high probability explanation.

Of course, this scheme works only if the taxonomic hierarchy is a strict hierarchy, i.e. each object has only one parent and there are no "negative" links. We will assume that this is indeed the case, as is done in WIMP. This means that the is-a hierarchy does not have multiple inheritance. The implication of this is that the number of possible aggregated values for a node with n possible values is at most $2n$.

We can see how allowing aggregation of node values can help us alleviate the overspecification problem. In the vacation-planning example, we can aggregate all the vacation locations (not including staying at home) into one value, "Going-away-on-vacation". This new value is a less specific instance of the vacation locations. Doing that immediately solves the overspecification problem in this case, as the most probable complete model will be the desired one (i.e. Healthy, Going-away-on-vacation, Alive).

Rather than actually aggregating values into a single value, we can generalize the meaning of assignments. We can allow an assignment to assign a disjunction of values to a node or variable. The result will be the same as when aggregating node values into a single value. We do not want

to allow any old disjunction to be assigned, however. We want to limit ourselves to assigning disjunctions that correspond to concepts, or to different events in our hierarchy of event types. The most general event is the "anything happens" event, which corresponds to the disjunction of all the values of a node. Assigning the "anything happens" disjunction to a node, is exactly equivalent to leaving it unassigned. Thus, we see that allowing the assignment of disjunctions to nodes in explanations is a generalization of irrelevance-based explanations.

We remain with the question: when do we allow a particular disjunction to be assigned to a node in a proposed explanation? The answer to this question is not at all obvious. For example, if we allowed any disjunction corresponding to a concept to be used every time, then all explanations will assign the most general disjunction (a disjunction of all the node's values) to each node. Essentially, this is equivalent to leaving all non-evidence nodes unassigned, which gives us the highest probability assignment. This result is, however, somewhat suboptimal as far as usefulness goes.

Instead, we propose the following criteria: first, the disjunction has to correspond to a pre-existing concept. The reason for this assumption is that we want an explanation to consist only of natural events and concepts. This is equivalent to assuming that a set of allowable disjunctions is provided to the system. Second, we only assign a disjunction if the probability of the descendent nodes is roughly statistically independent (by which we mean δ-independent, for δ some constant between 0 and 1) of which value (from the disjunction) we condition on.

To get a picture of where this is leading us, consider the special case where the only higher level concept is the "any event" concept. In this case, allowing the assigning of disjunctions under the above constraints is exactly equivalent to δ-independence based assignments; or in the case of $\delta = 0$, to independence-based assignments. That is because the only allowed disjunctions are those with a single value, or those with all the values of a node. The second constraint forces us to assign the disjunction only if (δ) independence occurs, exactly as in the case of (δ) independence-based assignments.

We will ignore in this thesis the representation issue, and just assume that for each (multi-valued) node, a set of all permissible disjunctive assignment is given, in some form. Thus, for each node in the belief network, with a domain D_v, the set of permissible disjunctions M_v is given, where $M_v \subseteq 2^{D_v}$, as well as the set of all conditional probabilities of each permissible disjunctive assignment to v given the parents of v. In what follows, we will usually omit referring to M_v, assuming its presence implicitly.

One may argue that we do not need to introduce the first constraint and M_v at all. We could allow any disjunction, as long as the second constraint, that conditional independence hold, is obeyed. In fact, this seems equivalent to an argument of the following form: we (as intelligent agents) construct our concepts from empirical data. Therefore, if (conditional) independence occurs, i.e. it does not matter which of a set of values is assigned, we are justified in creating a new concept that corresponds to that set of values. This argument seems reasonable, but we would rather leave this point undecided. Suffice it to say that our definitions require the existence of the set of allowable disjunctions M_v, but if we decide that it is not needed, we can just set $M_v = 2^{D_v}$ for every variable in the network, thereby voiding the first constraint.

5.4.2 Defining Generalized δ-IB Explanations

We call an assignment that includes disjunctions a *generalized* assignment. Generalized assignments are sets of assignments, comprising the set of assignments (logically) consistent with it. For example, if generalized assignment $A = \{v = 1 \vee 2, x = 1\}$, then it is equivalent to the set of assignments $\{\{v = 1, x = 1\}, \{v = 2, x = 1\}\}$. A generalized assignment is also a sample space event, the union of the events comprising its member assignments.

We say that the general δ independence based (Gδ-IB for short) condition holds at a node v w.r.t. a generalized assignment $\mathcal{A}$ if the probability of v given any assignment $\mathcal{B}$ in $\mathcal{A}$ is within

some bounds (determined by δ). Formally:

Definition 27 *The generalized δ IB condition holds at $v \in S$ w.r.t. generalized assignment $\mathcal{A}_S$ iff the following inequality holds:*

$$\min_{B \in \mathcal{A}_{\uparrow(v)}} P(\mathcal{A}_{\{v\}}|\mathcal{B}_{\uparrow(v)}) \geq (1-\delta) \max_{B \in \mathcal{A}_{\uparrow(v)}} P(\mathcal{A}_{\{v\}}|\mathcal{B}_{\uparrow(v)}) \tag{5.12}$$

We define the G-IB condition as the Gδ-IB condition with $\delta = 0$. As we did in previous chapters, we proceed to define Gδ-IB assignments as assignments where the Gδ-IB condition holds at every node. Formally:

Definition 28 *A generalized assignment $\mathcal{A}_S$ is Gδ-IB iff for every node $v \in S$, the Gδ-IB condition holds.*

Finally, we define a Gδ-IB MAP as the most probable Gδ-IB assignment where the evidence nodes are assigned correctly. Formally:

Definition 29 *A generalized assignment $\mathcal{A}_S$ is a Gδ-IB MAP w.r.t. evidence $\mathcal{E}$ iff it is a maximum probability Gδ-IB assignment such that all the evidence nodes are assigned and $P(\mathcal{E}|\mathcal{A}_S) = 1$.*

As before, we only interested in generalized assignments that are maximal w.r.t. subsumption and are properly evidentially supported. If the distribution is positive, we get the "maximal w.r.t. subsumption" for free. Note that the definitions here are local (i.e. refer only to a node and its direct predecessors), not as in previous definitions of irrelevance-based assignments. We do that for the sake of simplicity, as for strictly positive distributions, the G-IB condition holds if and only if $P(\mathcal{A}_{\{v\}})$ is independent of all the ancestors of v given $\mathcal{A}_{\uparrow(v)}$. This allows for local testing of whether an assignment is G-IB. We believe that this equivalence holds for the Gδ-IB condition with $\delta \neq 0$, but cannot at present prove (or disprove) that.

As Gδ-IB MAP explanations are a generalization of δ-IB MAP explanations, they are at least as good with respect to the explanation problems they handle. In addition, allowing the correct sets of disjunctions, we can also handle the example where δ-IB explanations failed (and where quasi-IB explanations succeeded). Consider again figure 5.4a. For any $\delta > 0$, if we allow the disjunction $M_2 \vee ... \vee M_{1000}$ as a permissible disjunction, then we get a G-IB MAP of $\{E, m = M_2 \vee ... \vee M_{1000}\}$. In figure 5.4b we also get the right answer, as long as $\delta < 0.5$. In the analysis at the end of this chapter, we will discuss a case where almost all explanation systems fail, but Gδ-IB MAP explanations do well.

What remains to be done is to show that there exists some effective algorithm for computing Gδ-IB MAP explanations. We will do that for the case of $\delta = 0$, in essence computing G-IB explanations only. The extension of the algorithm to any δ will be left for future research.

5.4.3 Generalized δ-IB MAP algorithm

We mean to design an algorithm that uses best-first search, and is essentially a generalization of algorithm 1. We do so by generalizing the concept of hypercubes, to allow assignment of disjunctions. Generalized hypercubes are generalized assignments that assign permissible disjunctions to a node and its parents.

Definition 30 *A generalized assignment $\mathcal{A}$ is a generalized hypercube (G-hypercube) based on node v iff $span(\mathcal{A}) = \{v\} \cup \uparrow(v)$, and if $w \in span(\mathcal{A})$ then $\mathcal{A}(w) \in M_w$.*

We define maximal generalized IB hypercubes, in a manner similar to chapter 4.

Definition 31 *A G-hypercube $\mathcal{A}$ based on v is an IB G-hypercube (based on v) iff the generalized IB condition holds at v w.r.t. $\mathcal{A}$.*

Definition 32 *An IB G-hypercube $\mathcal{A}$ is* maximal *if it is maximal w.r.t. subsumption, i.e. there is no IB G-hypercube $\mathcal{B}$ not equal to $\mathcal{A}$ that subsumes $\mathcal{A}$.*

With these changes in the definition of hypercubes, we can see that for a G-IB MAP algorithm we can use algorithm 1, except that maximal G-IB hypercubes are used in place of maximal IB hypercubes. The termination condition is that the G-IB condition hold at every node (it is a weaker condition than the IB-condition). Computing the probability of a G-IB assignment can be done efficiently, in a manner similar to that of IB assignments.

5.4.4 Evaluation of G-IB MAP Explanations

We have shown that generalizing irrelevance-base explanations to allow a limited assignment of disjunctions further alleviates the overspecification problem. We get the added bonus that the disjunction allows us to choose a less specific event (in a particular node) as long as it is irrelevant which even subtype occurred. It also alleviates some of the problems encountered by δ-IB explanations, while it does not have the bad computational properties of quasi-independence, as we argued in the previous section.

Another problem that can occur with any of the proposed explanation schemes, as well as cost-based abduction and even Hobbs and Stickel's least cost proofs, is the following. Suppose that we want to explain John's being at the train tracks, and that there are 100 different ways for him to get to the train tracks of his own volition. Suppose, also, that the agent will select one of them at random when it *intends* to go to the tracks. There is also an entirely different explanation for John's being at the tracks, he could have been abducted. The latter explanation is less likely than his getting there of his own volition. A reasonable belief network representation of the facts is shown in figure 5.5. We give the kidnapping a prior probability of $2 * 10^{-10}$, and the intention to go to the tracks a prior probability of 10^{-8}, a much higher probability.

It can be easily shown that all the explanation systems that we mentioned above will prefer the kidnapping explanation, because they will have to assign a value to the method node, which will cause the probability of any of the "own volition" explanations to be low. Even if we use one the weighted abduction systems, the problem still remains. The only seeming way out of the problem is to allow assigning the disjunction of "all methods" to the method node (as in generalized IB explanations). This will allow us to get the correct "own volition" explanation. In fact, the method of posterior node probabilities also happens to give the right answer, but we have already shown that the posterior node probabilities scheme is undesirable for other reasons (possible inconsistencies and irrelevant explanations).

5.5 Summary

In this chapter, we examined the weaknesses of independence-based MAPs, and considered several ways to overcome them. We proposed methods that are more liberal in allowing nodes to remain unassigned, as well as generalizing irrelevance-based explanations by allowing disjunctive assignments. The methods proposed in this chapter alleviated the instability problem that was a shortcoming of IB MAPs. We then evaluated the performance of the proposed schemes, both in terms of generality, and in terms of computational complexity.

There remain two theoretical problems, however; both problems appear in all the systems we have surveyed. The first problem is that while irrelevance-based MAPs do not assign *irrelevant* nodes, it is possible that nodes be assigned values with *negative* impact on the evidence. This becomes possible if the prior probability of such an assignment is high. For example, consider

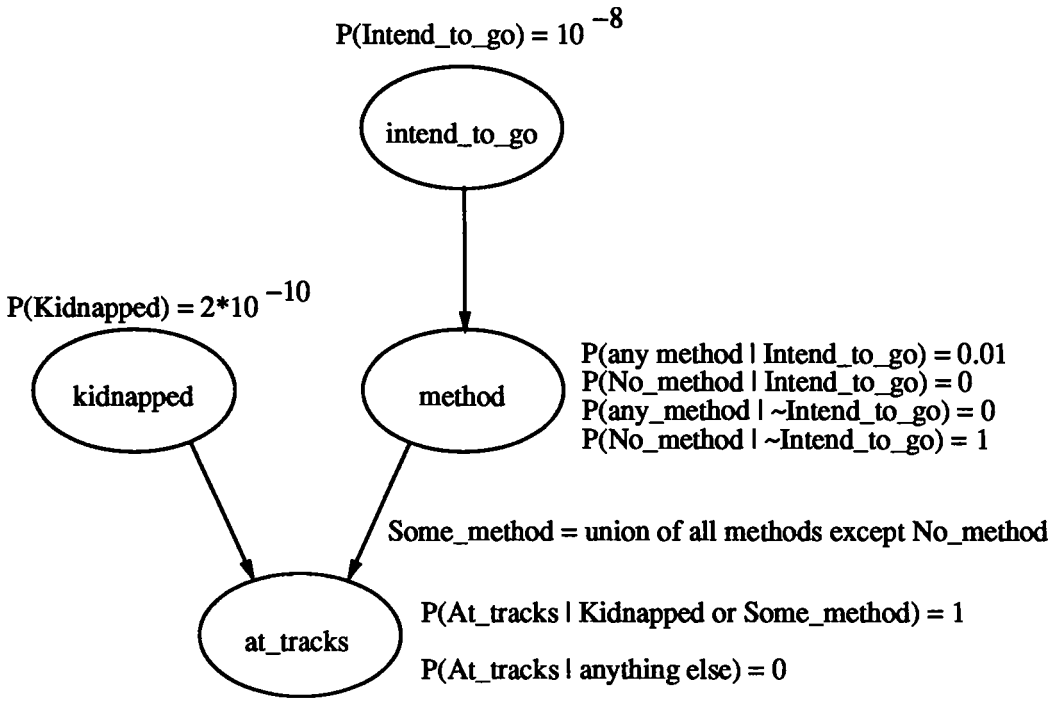

Figure 5.5: Train Tracks Example Network

figure 5.4b. Suppose that we modify the prior probability of m, such that $P(m = M_1) = 0.3$ and $P(m = M_2) = 0.7$. If we do that, then m is not irrelevant to e, whichever criterion we use (with δ-independence, assume some $\delta < 0.5$). As a result, the most probable model given E is $\{E, m = M_2\}$. This seems reasonable, except when we note that this means the we are selecting $m = M_2$ as an *explanation* for E. But since assigning $m = M_2$, actually *lowers* the probability of E, it is hard to accept $m = M_2$ as an explanation of E. It may be possible to define a new scheme which leaves m unassigned in these cases, or perhaps selects the best assignment as in the generalized δ-IB MAPs, and then unassigns m in a second pass. It is not clear, however, whether this kind of solution would work well in the general case.

The second problem is similar to the one presented in figure 5.5. We handled the problem by allowing disjunctive assignments. Suppose, however, that we deliberately muddle the representation by breaking the "method" node into 100 binary nodes, one for each method of travel. Obviously, disjunctive assignments of values to nodes will not solve the latter variant of the problem. Allowing a larger class of assignments may be useful, but it is unclear how we would characterize that class in a useful way. These problems are not trivial, but seem to occur rarely in practice. The solution of these problems is left for future research.

Chapter 6

Experimental Results and Alternate Algorithms

We have shown that our schemes are theoretically reasonable by arguing that δ-independence and quasi-independence are superior to all the other existing criteria for deciding which variables should remain unassigned in an explanation. We have even generalized irrelevance-based explanations to allow assignment of disjunctions to nodes. We still need to show that, in practice, our schemes generate reasonable explanations. We do that by using the theory to produce explanations for a toy domain.

We do not test our theory on a larger domain at this juncture, because that requires availability of a system that can generate sufficiently general belief networks. In particular, a system that uses many instances of multi-valued nodes and negation is required so that the added power of irrelevance-based explanation is utilized. Currently available and accessible systems are insufficient for that purpose. For example, networks generated by the WIMP story understanding program are mostly AND/OR trees with binary-valued nodes. For such networks, there is no need to employ irrelevance-based explanation, because cost-based abduction works just as well in this case. That is because irrelevance-based explanation is equivalent to cost-based abduction for AND/OR networks (when only root nodes may be assumed), as we argued in chapter 4.

Our explanation schemes are useful for finding explanations only if we can construct effective algorithms for them. We have designed algorithms for irrelevance-based explanations in previous chapters. In this chapter, we present and analyze timing experiments to show that the algorithms are effective. We test our algorithms on our toy domain, which is a medium-size example, and on randomly generated belief networks, some of which are much larger than the toy domain.

We will conclude the chapter with a (successful) attempt to construct an entirely different kind of algorithm for IB-MAPs: via reduction to linear systems of inequalities. We will show that this reduction is a "natural" reduction, in that we will have a constant number of equations for each maximal IB hypercube. Performance evaluation of such an algorithm are left for future research. We also discuss how other partial MAP schemes, such as δ-independence based MAPs, can use the reduction to inequalities.

6.1 Commonsense Explanations: a Toy Domain

We present a toy domain, for performing explanation experiments. The full set of variables for our domain is given in table 6.1. The domain is that of "things we see around our house", and explanations for them. These are commonly observed things, such as our front lawn, the road, weather conditions, and the neighbors' dog. We also have several other things we can observe, such as listen to the weather forecast. There are variables that we usually cannot observe, such

as weather conditions 100 miles upwind from us yesterday. There are also some variables that are sometimes observed and sometimes are not.

For example, if we only look at the road, we may see that it is wet, but cannot see that it is raining directly. We may *infer* that as an explanation, however. Sometimes, it *is* possible to observe the rain directly, such as by going outside. The fact that some variable assignments (instantiations) are sometimes available as evidence, and sometimes are causes, is one of the reasons that we could not accept the restriction of a hard partitioning of nodes into evidence nodes and hypothesis nodes. Spatial and temporal reasoning is made degenerate, a simplification of [Dean and Kanazawa, 1991], where the variables are duplicated for every interesting discrete spatial and temporal value. We did that in the case of weather and weather prediction.

We also need to specify the causal connections and probabilities for our domain. These are represented as a belief network in a format defined by the IDEAL program, [Srinivas and Breese, 1989]. The topology of the belief network is shown in figure 6.1.

To get a feeling for the kinds of distributions we have in our domain, some of the smaller conditional probability arrays are shown in table 6.2. As we can see in the table, there are very few strict (probability 1) implications, ANDs or ORs. This is a domain with a lot of uncertainty. We omit showing all the conditional distribution arrays, as they consist of large, uninteresting, sets of numbers.

Since the causal relationships and probabilities are cooked up to be reasonable, but are not actual real-world distributions, we have to evaluate the goodness of the explanations based on our intuition. If some of the explanations appear a bit odd, that is because our world model is somewhat simplistic, and because our distributions do not always capture real-life frequencies. For example, feel-temps depends only on inside-temps, while other relevant things, such as state of health or air currents, are not in the model. Table 6.3 shows explanations generated by the IB MAP algorithm, for several sets of evidence. The explanation columns exclude the evidence itself, for brevity. In many of the examples, the IB explanations were the same as, or similar to, the δ-IB explanations. In the examples of table 6.3, differences between IB explanations and δ-IB explanations are marked by asterisks, and explained later in the text.

Note the use of some of the evidence to set prior constraints (assignments to root nodes), rather than as actual evidence to be explained. In our examples, this manifests as assignment to the season, landlord and weather-yesterday nodes, as shown in table 6.3.

Analyzing the first two examples, we see that if the lawn is wet, and we heard that it was going to rain, the best explanation is that it is raining (together with some other stuff explaining the rain and the weather report). But if the lawn is wet, and the weather report predicted no rain, the best explanation is that the sprinkler is on and that we paid our water bill. The weather node is not assigned, because Sprinkler-on and Water-bill-paid are sufficient to explain the observation that the lawn is wet.

In the third example, Birds-chirping explains Hear-birds-chirp, and inside-temps=warm (as opposed inside-temps=hot) explains feel-temps=cold (the distributions were set up so that there is a reasonably high probability of feeling cold when it is warm inside). The seemingly better explanation of inside-temps=cold is downgraded because it is summer, making it unlikely that it is cold inside. Other terms in the explanation tell us why it is warm inside, as opposed to hot.

In the fourth example, rain and Squirrels-in-roof explain Knocks-on-roof (presumably they feel like running around inside the roof when it rains). The inside-temps=warm, outside-temps=cold and Hole-in-roof explain why the squirrels are in the roof. Birds-chirping explains Hear-birds-chirp. Squirrels-on-roof has some effect on Knocks-on, but is unlikely (especially so with the rain) and thus is set to false. In fact, our intuition would dictate that we leave this variable unassigned. That cannot be done with IB MAPs, because statistical independence does not occur here (but δ-IB MAPs finesse this problem in this case). The other items are needed to explain the rain and the inside and outside temperatures.

Variable name	Values	Notes
weather	Rain, Hail, Snow, No-precipitation	here and now
weather-yesterday	Rain, Hail, Snow, No-precipitation	here
weather-upwind	Rain, Hail, Snow, No-precipitation	for prediction
weather-report	Rain, Hail, Snow, No-precipitation	
cloudy	Partial, Fully, None	
season	Summer, Winter, Spring, Autumn	
road-condition	Dry, Wet, Icy, Snow	
lawn-condition	Dry, Wet, Icy, Snow	
garbage-cans	Standing, Overturned, Not-visible	
garbage-truck	Passed, Visible, Not-yet	comes on Monday
day-of-week	days of the week	affects garbage
hear-bark	binary	
water-bill-paid	binary	
sprinkler-on	binary	
hydrant-open	binary	
heating-on	binary	
cooling-on	binary	
outside-temps	Cold, Warm, Hot	
inside-temps	Cold, Warm, Hot	
feel-temps	Cold, Warm, Hot	how it feels
my-radio-on	binary	
other-radio-on	binary	neighbors' radio
dog-out	binary	neighbors' dog
knocks-on-roof	binary	
squirrels-in-roof	binary	
squirrels-on-roof	binary	
state-of-repair	Hole-in-roof, No-hole	
landlord	Good, Bad, Indifferent	affects state-of-repair
hear-bird-chirp	binary	
birds-chirping	binary	
hear-music	binary	

Table 6.1: Variables for Our Toy Domain

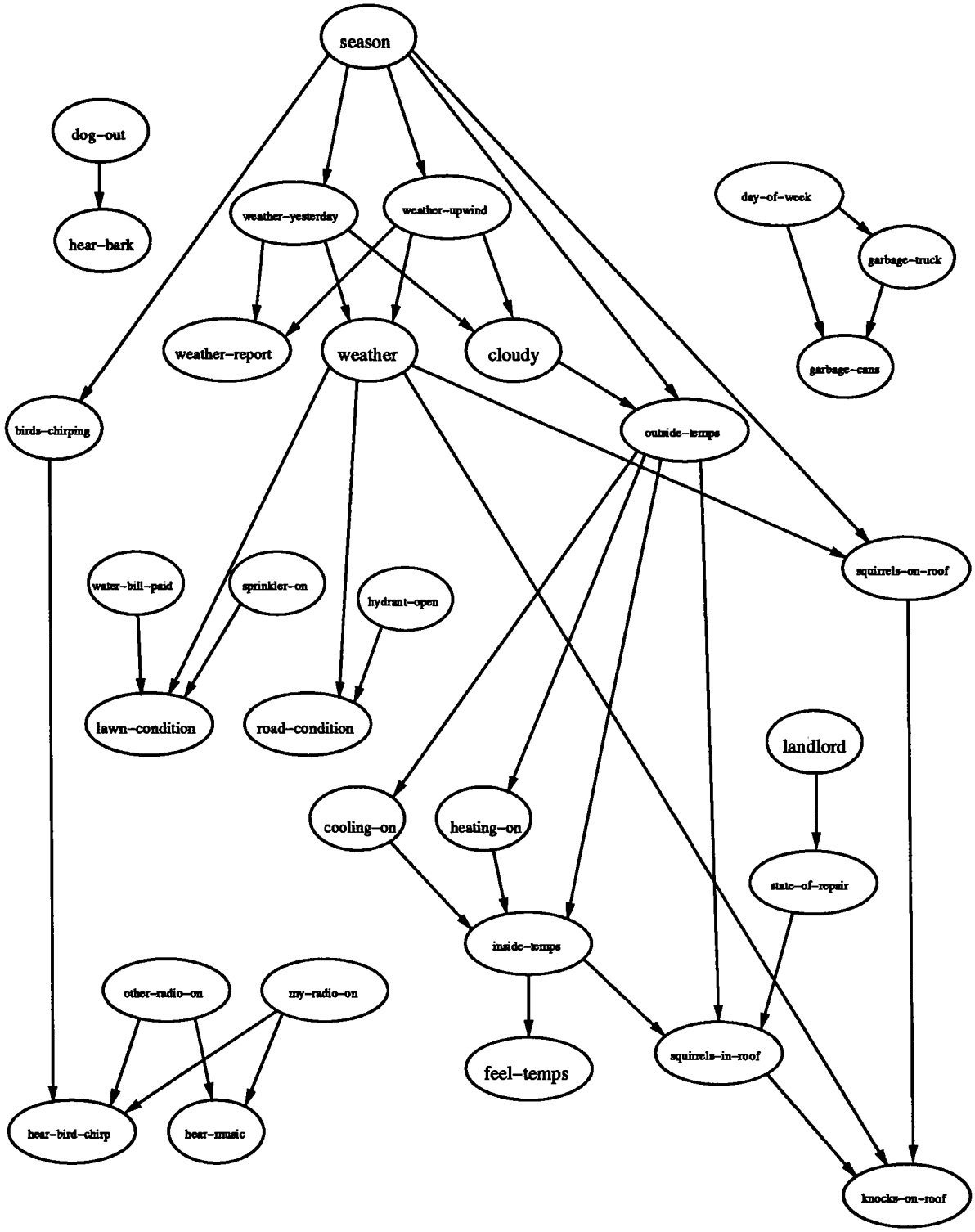

Figure 6.1: Belief Network for Toy Domain

Node value	Given	Probability
Hydrant-open		0.01
Birds-chirping	season = summer	0.7
Birds-chirping	season = winter	0.1
Birds-chirping	season = spring	1.0
Birds-chirping	season = autumn	0.7
Squirrels-on-roof	weather = rain, season = summer	0.2
Squirrels-on-roof	weather = hail, season = summer	0
Squirrels-on-roof	weather = snow, season = summer	0
Squirrels-on-roof	weather = no-precipitation, season = summer	0.3
Squirrels-on-roof	weather = rain, season = winter	0.1
Squirrels-on-roof	weather = hail, season = winter	0
Squirrels-on-roof	weather = snow, season = winter	0.01
Squirrels-on-roof	weather = no-precipitation, season = winter	0.2
Squirrels-on-roof	weather = rain, season = spring	0.2
Squirrels-on-roof	weather = hail, season = spring	0.1
Squirrels-on-roof	weather = snow, season = spring	0.1
Squirrels-on-roof	weather = no-precipitation, season = spring	0.3
Squirrels-on-roof	weather = rain, season = autumn	0.1
Squirrels-on-roof	weather = hail, season = autumn	0
Squirrels-on-roof	weather = snow, season = autumn	0.1
Squirrels-on-roof	weather = no-precipitation, season = autumn	0.2

Table 6.2: Some Distributions for the Toy Domain

Evidence	First-best explanation	Second-best explanation
season=summer lawn-condition=wet weather-report=rain *	weather=rain weather-upwind=rain weather-yesterday=no	Same, except weather-yesterday=rain
season=summer lawn-condition=wet weather-report=no **	Sprinkler-on Water-bill-paid weather-yesterday=no weather-upwind=no	Same, except weather-yesterday=rain
My-radio-on Hear-birds-chirp feel-temps=cold landlord=indifferent season=summer weather-yesterday=rain	cloudy=partial Cooling-on ¬Heating-on outside-temps=hot inside-temps=warm Birds-chirping	Same, except ¬Cooling-on and outside-temps=warm
Knocks-on-roof Hear-bird-chirp feel-temps=cold landlord=indifferent season=spring weather-yesterday=no ***	weather=rain weather-upwind=rain cloudy=fully Heating-on ¬Cooling-on outside-temps=cold inside-temps=warm Squirrels-in-roof ¬Squirrels-on-roof Birds-chirping Hole-in-roof	same, except ¬Heating-on outside-temps=warm

Table 6.3: Example of Commonsense Explanations

The δ-IB explanations for the same evidence differ in the following manner (refer to table 6.3 for the comparison):

* In the best explanation, weather-yesterday does not get assigned, and the second-best explanation is that the sprinkler is on.

** In the best explanation, weather-yesterday is not assigned, and in the second-best explanation, it is raining (despite the weather prediction), and sprinkler-on is not assigned.

*** Does not assign the squirrels-on-roof node.

As we can see, the explanations generated by the system appear reasonable, as well as the choice of what nodes to omit from the explanation. δ-IB explanations are frequently essentially the same as IB explanations, except that several more irrelevant nodes are left unassigned by the δ-IB assignments.

6.2 Best-first IB-MAP Algorithm results

We present timing experiment results for our IB MAP algorithm and for our δ-IB MAP algorithm. Performing experiments on purely random *networks* does not appear useful, as there would not be many occurrences of conditional independence in that case. Instead, we elected to experiment on networks where the IB hypercubes are generated randomly, as we discuss later on. The timing results are for an implementation in LISP (compiled using LUCID LISP), on a SPARC workstation. The workstation had 24MB of main memory, and that is relevant, as system time (such as garbage collection and page fault processing) is included.

6.2.1 Timing Experiments on Toy Example

We performed timing experiment results for 10 instances of evidence for our toy domain. Among these 10 instances, are the ones for which explanations were found in the previous section. Results are presented both for finding the first-best explanation, and for finding the first and second best explanations. In either case, we timed both the IB MAP algorithm and the δ-IB MAP algorithm (with $\delta = 0.1$).

	IB, one sol.	IB, two sol.	δ-IB, one sol.	δ-IB, two sol.
Average (time)	7	9	35	40.8
Average (states)	27.3	33	93	101
Best (time)	0.5	0.7	0.9	0.8
Best (states)	3	5	3	4
Worst (time)	23	37.3	126	161
Worst (states)	91	120	311	342

Table 6.4: Timing Results for Commonsense Explanations

Results are shown in 6.4, and are presented in terms of CPU seconds and in terms of number of states popped from the agenda. In the table, best timing is the time for the problem that took the least time. Likewise, worst timing is the timing for the problem that took the greatest execution time. Time for initializing the hypercubes for the algorithm was approximately 3.5 seconds, and is not included.

As we see, on typical instances of evidence in our domain, the algorithm terminates in reasonable time. The table lists number of states that were *expanded*, the number of states generated is much

larger than that. Out of these 10 cases, we encountered 2 where the post-processing step was necessary for δ-IB MAPs.

A previous implementation of the algorithm did very badly when a larger number of states where expanded, much worse than linear in the number of states. Attempts to locate where that implementation of the algorithm spent most of its time revealed (via internal timing analysis) that 80%-95% for the time was spent on checking whether a new state has already been visited, before pushing it into the agenda. The implementation of that part of the algorithm is was inefficient, and was replaced by a test of state equality upon *retrieval* from the agenda. The new duplicate state remover is very efficient, as its total run-time is negligible compared to other parts of the algorithm. There is no appreciable impact on the rest of the algorithm, thus a major speedup was accomplished.

6.2.2 Timing Experiments on Random Networks

When we set out to perform an experiment on random networks, we need to specify the distribution. We cannot just specify "uniform distribution", as there is no such thing for belief networks. Note that the purpose of using randomly generated belief networks is just to show that the algorithm performs reasonably well in many cases on different belief networks of small to medium size, *not* to get a run-time asymptotic bound or a real performance estimate.

We base our experiment partially on an algorithm provided by IDEAL to generate a random topology. IDEAL's algorithm takes a number of nodes and returns a DAG based on a random subset selection of parents for each node, such that no node has more than some maximum allowed in-degree. IDEAL's random network generation algorithm then proceeds to generate a random conditional distribution by selecting a uniformly distributed value in $[0, 1]$ for each conditional case. The algorithm then normalizes the numbers so that the sum of probabilities is 1.

Other input parameters are to the algorithm are an optional maximum in-degree, which we leave at the default 3, and an optional maximum node state count, which we let revert to the default 4. We keep these numbers relatively low, as otherwise the time for *generating* a network, as well as the storage requirements, becomes intractable (large distribution arrays). The complexity of our IB MAP algorithms is only dependent on the number of *hypercubes*, assuming a favorable representation of the distributions[1]. This complexity is as low as $O(k)$ for a node with k parents (for AND or OR nodes), as opposed to the time and space required for the conditional distribution array, which are exponential in k.

We borrow IDEAL's network topology generator for our experiment, but reject the distribution generator. The reason for that is that networks that correspond to knowledge bases are likely to contain many conditional independencies in the conditional distribution arrays. That is because many nodes tend to look like AND or OR nodes, especially in networks generated by rules-based systems. Therefore, we need another way to generate the distributions. What comes to mind immediately is to allow nodes to be dirty or noisy ANDs and ORs, with some randomly selected noise. This seems reasonable for binary nodes, but our diagrams are *not* limited to such nodes. We thus propose a method based on randomly generated IB hypercubes.

To generate the hypercubes, we consider, for each node v, all the conditional cases of v given its parents. For each possible state of v, except for one state (explanation for this seeming oddity appears later in the text), selected at random with uniform distribution, we do the following:

1. Push the hypercube consisting of all possible assignments to the parents of v onto the stack.

2. While the stack is not empty, do the following with each hypercube popped off the stack:

[1] One obvious favorable representation is one that specifies the IB hypercubes explicitly, but there may be other favorable representations.

(a) If the hypercube is 0 dimensional, pick a probability for it, otherwise do the following steps:

(b) Generate a random number uniformly in the range $[0, 1]$.

(c) If the number is less than some given p, then pick a probability for the current hypercube. Otherwise do the following steps:

(d) Select a parent of v that is not assigned by the hypercube at random. Let w be that parent.

(e) For each possible value $d \in D_w$, generate a hypercube that is the union of the current hypercube and the assignment $\{w = d\}$. Push the resulting hypercubes onto the stack (this is a partition of the current hypercube).

There are two remaining issues: how do we pick p, and what does it mean to select a probability for the hypercube. Note that the greater p is, the larger (and fewer) the resulting hypercubes are likely to be. For example, if $p = 1$, the hypercubes will never be partitioned. We elect to use $p = 0.5$ (for no particular reason) except for the first iteration, where we use $p = 0$. The latter choice is because we do not want v to be independent of *all* of its parents, we always want this (largest) hypercube to be partitioned. As for picking a probability for a hypercube, we would like to do that with uniform probability from the range $[0, 1]$. Unfortunately, we cannot do that, as then the probabilities will not sum to 1, except for the case where v is a binary node, where this problem goes away. Neither can we renormalize the probabilities, as that will destroy the IB property of the hypercubes that we have just generated. Instead, we do the following: when we pick a probability for a hypercube, we check to see the total probability picked before, and generate the probability for the current hypercube with uniform distribution in the remaining range. The probabilities for the conditional cases for the last state of v (for which we did not generate hypercubes), are just set so that probabilities sum to 1 exactly.

We then generate random evidence, consisting of random assignments to 1-3 nodes, and test our IB MAP and δ-IB MAP algorithms on the resulting network and evidence. We let the number of nodes vary from 10 to 400 (we crashed on memory space problems with larger diagrams), and average out over 10 problem instances for each diagram size.

Results shown in figure 6.2. These results are so noisy that the graph is non-monotonically increasing with the size of the diagram. When obtaining each data point, we also computed the standard deviation over the 10 instances. That standard deviation in run-time was in many cases much greater than the average run-time, showing that for each data point the bulk of the cumulative runtime was probably in a single problem instance. The latter fact makes the results unreliable as an indicator of typical running time for a particular size of DAG.

We then proceeded to try the experiment on fewer diagram sizes, but with approximately 100 instances for each problem size, to decrease the noise level, as seen in figure 6.2. Some of the larger problem instances crashed LUCID LISP's garbage collector (probably because of excessive memory requirements), and were replaced by problem instances which did not cause that behavior. These problem instances would have required a large running time. We believe that the oddly placed 400 node data point is caused by the fact that we could not get the timing for about 10 problem instances, which we believe would have had a very large running time had they ran to completion.

The resulting graph is still too noisy to try to obtain a curve fit of any sort, and we do not attempt to do that. We can still conclude from these results, however, that the IB MAP algorithm terminates in reasonable time on small to medium networks, but performs badly on the larger networks, as expected.

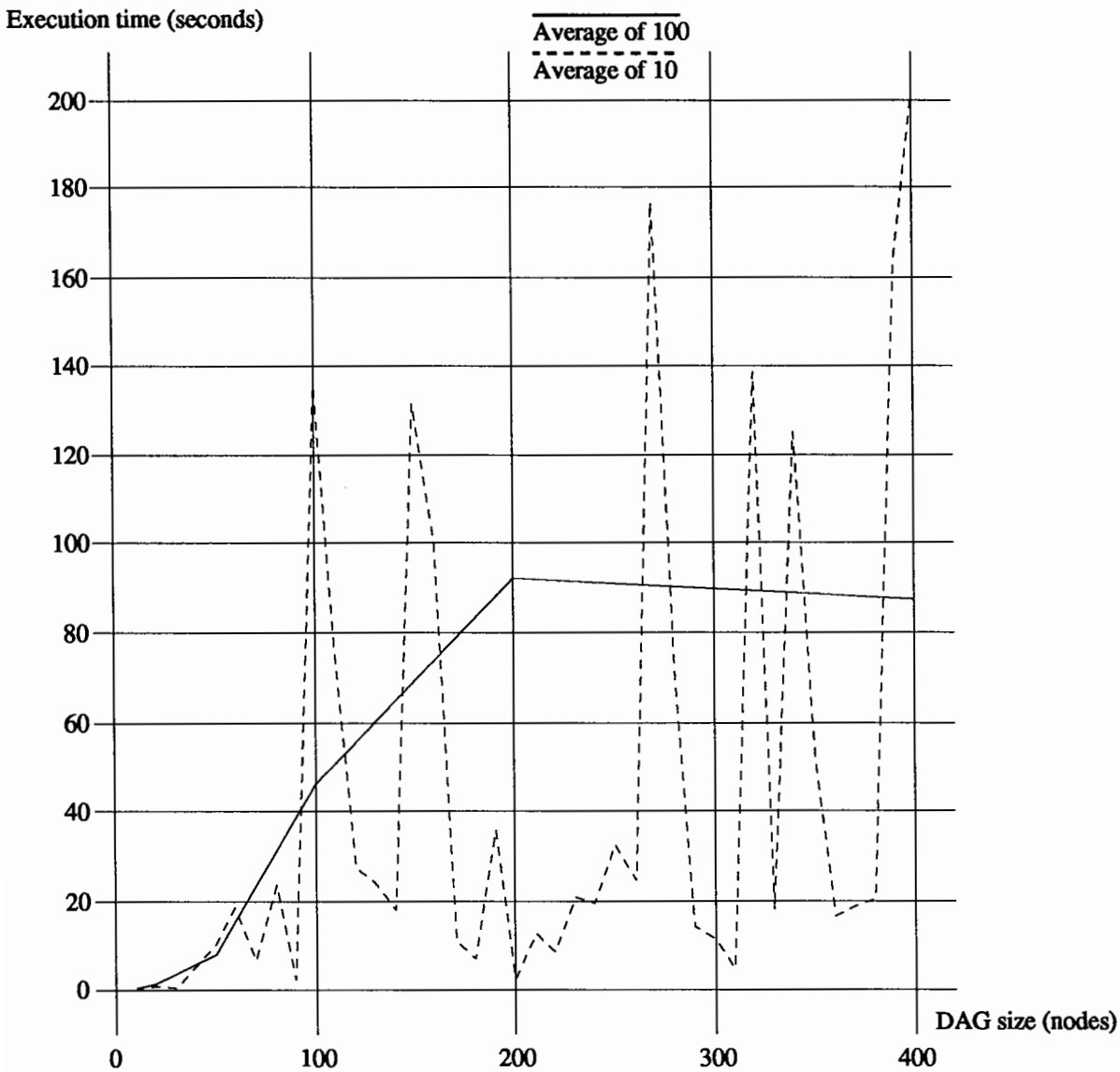

Figure 6.2: Results of IB MAP algorithm for Random Networks

6.3 IB-MAPs and Linear Inequalities

In [Santos Jr., 1991a], [Santos Jr., 1991d], [Santos Jr., 1991b], and [Santos Jr., 1991c], a method of converting the complete MAP problem to a linear inequality system was shown. The main problem (from the point of view of this thesis) with using systems of inequalities for finding IB-MAPs, is that the in the scheme of [Santos Jr., 1991c], the result is always a *complete* assignment, whereas IB-MAPs are typically *partial* assignments. This rules out the use of a simple mapping of belief-network nodes to inequality system variables. However, the method proposed in [Santos Jr., 1991d] does not do that either. Instead, it has variables for every *conditional case* in the belief network, i.e. for every distribution array entry of a node given its parents. We will review the basic idea of [Santos Jr., 1991d] here, and show how it can be modified to deal with IB-MAPs.

6.3.1 MAPs and Linear Inequalities

This subsection is a modified copy (with permission) of work done in [Santos Jr., 1991d]. We need to apply the linear constraint satisfaction approach to Bayesian networks. This entails constructing a constraint system which is computationally equivalent to the Bayesian network. Although this could be done by first transforming the Bayesian network into a cost-based abduction graph ([Charniak and Shimony, 1990b]) and then transforming the graph into a constraint system ([Santos Jr., 1991a], [Santos Jr., 1991c]), a more natural and straightforward method will be given below. We will show how to directly transform a Bayesian network into an equivalent constraint system.

We first observe that a Bayesian network can be completely described by a finite collection of random variables and a finite set of conditional cases based on the random variables (see appendix A). Given a Bayesian network, we can construct a pair $(V, \mathcal{D})$ where V is the set of nodes in the network and $\mathcal{D}$ (the distribution) is a pair (C, P). C is a a set of conditional cases associated with the network. Formally, $\{v = d | x_1 = d_1, ..., x_k = d_k\} \in C$ iff $\{x_1, \ldots, x_k\} = \uparrow(v)$ and $d_i \in D_{x_i}$ for every $k \geq i \geq 1$. P is a function from C to $[0, 1]$, the conditional probability of the conditional case, $P(v = d | x_1 = d_1, ..., x_k = d_k)$. We can clearly see that $(V, \mathcal{D})$ completely describes the Bayesian network.

Our basic approach in constructing a constraint system from a given Bayesian network is to represent and enforce the constraints that exist between any two or more random variables, and the constraint introduced by the evidence.

Given a Bayesian network $B = (V, \mathcal{D})$ and evidence $\mathcal{E}$, we construct a constraint system $L(B, \mathcal{E}) = (\Gamma, I, c)$ as follows:

1. Γ is a set indexed by the set of all conditioning cases in C. We will use the same names for variables in Γ and for the conditioning cases in C.

2. c is a function from $\Gamma \times \{T, F\}$ to the positive reals, called the *cost* function, and is defined as follows:

$$\begin{aligned} c(q, T) &= -log(P(q)) \\ c(q, F) &= 0 \end{aligned}$$

3. I consists of the following constraints:

   (a) Let Q_v be the set of all conditioning cases where variable $v \in V$ appears on the left-hand side of the condition. For each v, the following equation is in I:

$$\sum_{q \in Q_v} q = 1 \tag{6.1}$$

(b) Let $R_v^{w^d}$ be the set of all conditioning cases where $w = d$ is in the condition and v is in the left-hand side of the conditioning case. Let S_{w^d} be the set of all conditioning cases where $w = d$ is in the left-hand side of the conditioning case. For every w, v, d such that $R_v^{w^d}$ and S_{w^d} are not empty, the following equation is in I:

$$\sum_{q \in S_{w^d}} q - \sum_{q \in R_v^{w^d}} q = 0 \tag{6.2}$$

(c) Let E_v be the set of all conditioning cases where v appears on the left-hand side of the condition, whenever there is some value d such that $(v, d) \in \mathcal{E}$. For each such v, the following equation is in I:

$$\sum_{q \in E_v} q = 1 \tag{6.3}$$

Note the similarity to the cost function in chapter 3. The equality constraints of type a assure the selection of exactly one conditional case per variable. Equalities of type b assure consistency of the conditional cases, and Equalities of type c introduce the evidence. The linear system constructed above is actually composed of equalities, rather than inequalities, but the solution method will be the same.

We say that the system $L(B)$ constructed above is the constraint system *induced* by B. The construction is straightforward and is done in time linear in the size of the Bayesian network. Every complete assignment $\mathcal{A}$ for B induces a 0-1 solution $s[\mathcal{A}]$ on $L(B)$, and every 0-1 solution s for $L(B)$ induces a complete consistent assignment $\mathcal{A}[s]$ for B. Both induced assignments can be computed in linear time given their counterpart assignment.

A function over assignments to the system of inequalities, called the *objective function*, is defined as follows:

$$\Theta_L(s) = \sum_{q \in \Gamma} \{s(q)c(q, T) + (1 - s(q))c(q, F)\} \tag{6.4}$$

If s is a 0-1 solution to $L(B, \mathcal{E})$ that minimizes $\Theta_L(s)$, then $\mathcal{A}[s]$ is an MAP assignment for B with evidence $\mathcal{E}$. Thus, to find the MAP for a given network and evidence, construct and find an optimal 0-1 solution for the induced system of equalities. This induces an easily computable MAP assignment on the belief network.

6.3.2 Reduction of IB-MAPs

The basic idea for representing belief networks and enforcing the IB condition is to use the same general scheme as for complete MAPs, but instead of having a separate variable for each conditional case, we have a variable for each maximal IB hypercube (see chapter 4 for the definition of hypercubes).

Using the notation of appendix A, our belief network is $B = (G, \mathcal{D})$, where G is the underlying graph and $\mathcal{D}$ the distribution. We usually omit reference to $\mathcal{D}$ and assume that all discussion is with respect to the same arbitrary distribution. For each node v and each domain value D_v, there is a set of k_{v^d} maximal IB hypercubes based on v (where $d \in D_v$). We denote that set by $\mathcal{H}^{v^d}$, and assume some indexing on the set. Member j of $\mathcal{H}^{v^d}$ is denoted $\mathcal{H}_j^{v^d}$, with $k_{v^d} \geq j \geq 1$.

A system of inequalities L is a triple (V, I, c), as discussed in the previous subsection. We construct a system of inequalities from the belief network and the evidence, as follows:

Definition 33 *From the belief network B and the evidence $\mathcal{E}$, we construct a system of inequalities $L_{IB}(B, \mathcal{E})$ as follows:*

1. *V is a set of variables $h_i^{v^d}$, indexed by the set of all evidentially supported maximal hypercubes $H_\mathcal{E}$ (the set of hypercubes H such that if H is based on w, then w is evidentially supported). Thus, $V = \{h_i^{v^d} | H_i^{v^d} \in H_\mathcal{E}\}$.*[2],

2. *$c(h_i^{v^d}, T) = -log(P(H_i^{v^d}))$, and $c(h_i^{v^d}, F) = 0$.*

3. *I is the following collection of inequalities:*

   (a) *For each set of two inconsistent hypercubes $h_i^{v^d}, h_j^{w^{d'}} \in \mathcal{H}_\mathcal{E}$, such that $w \neq v$:*

$$h_i^{v^d} + h_j^{w^{d'}} \leq 1$$

   (b) *For each evidentially supported node v:*

$$\sum_{d \in D_v} \sum_{i=1}^{k_{v^d}} h_i^{v^d} \leq 1$$

   (c) *For each pair of nodes w, v such that $v \in \uparrow(w)$, and for each value $d \in D_v$:*

$$\sum_{i=1}^{k_{v^d}} h_i^{v^d} - \sum_{d' \in D_w \wedge (v,d) \in H_j^{w^{d'}}} h_i^{w^{d'}} \geq 0$$

   (d) *For each $(v, d) \in \mathcal{E}$:*

$$\sum_{i=1}^{k_{v^d}} h_i^{v^d} \geq 1$$

The intuition behind these inequalities are as follows: inequalities of type **a** enforce consistency of the solution. Type **b** inequalities enforce selection of at most a single hypercube based on each node. Type **c** inequalities enforce the IB constraint, i.e. at least one hypercube based on v must be selected if v is assigned. Type **d** inequalities introduce the evidence.

Following the nomenclature of the previous section (and [Santos Jr., 1991a]), we define an assignment s for the variables of L as a function from V to $\mathcal{R}$. Furthermore:

1. If the range of s is in $\{0, 1\}$ then s is a 0-1 assignment.

2. If s satisfies all the inequalities of types **a**-**d** then s is a solution for L.

3. If solution s for L is a 0-1 assignment, then it is a 0-1 solution for L.

We continue by showing (theorem 16) that for every maximal evidentially supported IB assignment to B there exists at least one 0-1 solution to $L_{IB}(B, \mathcal{E})$. That means that all such IB assignments can be found by finding solutions the the system of inequalities. We will also show that for every 0-1 solution to the system of inequalities, there exists a unique evidentially supported IB assignment. This allows us to convert a solution to $L_{IB}(B, \mathcal{E})$ into an evidentially supported IB assignment to the belief network.

Theorem 16 *For maximal IB assignment $\mathcal{A}$ to B that is evidentially supported w.r.t. evidence $\mathcal{E}$, there exists at least one*[3] *0-1 solution to $L_{IB}(B, \mathcal{E})$.*

[2]The superscript v^d states that node v is assigned value d by the hypercube (which is based on v), and the subscript i states that this is the ith hypercube among the hypercubes based on v that assign the value d to v.

[3]Possibly more than one, because there may be more than one set of hypercubes that form the same IB MAP.

Proof: By construction. Run the IB MAP algorithm (algorithm 1, page 32) on the network, with evidence $\mathcal{E}$, until assignment $\mathcal{A}$ turns up. Collect all the (maximal) hypercubes that were picked to get $\mathcal{A}$. For each node v that is assigned, and no hypercube based on v was picked[4], select some maximal IB hypercube based on v that is consistent with $\mathcal{A}$, and does not assign any nodes not in *span*($\mathcal{A}$). We call the maximal IB hypercubes picked by the algorithm or selected in the latter stage the *selected* hypercubes.

Lemma 1 *After running the algorithm, for every assigned v there exists a maximal IB hypercube that is consistent with $\mathcal{A}$ and does not assign any new nodes. Furthermore, the union of all the selected hypercubes is consistent and is exactly $\mathcal{A}$.*

Proof: Every assignment $\mathcal{B}$ to v and some of its parents that assigns a value to v is subsumed by *some* maximal IB hypercube based on v. Thus, we just pick a maximal IB hypercube $\mathcal{H}$ that subsumes $\mathcal{B}$. The hypercube H is clearly consistent with $\mathcal{A}$, because $\mathcal{H}$ subsumes $\mathcal{A}$ (subsumption is transitive). Subsumption also implies that H does not assign any new nodes. The union of all the selected hypercubes is consistent because it subsumes (follows from set theory) $\mathcal{A}$, and $\mathcal{A}$ is consistent. The union of the selected hypercubes is also subsumed by $\mathcal{A}$, because each node in *span*($\mathcal{A}$) is assigned in *some* selected hypercube. Thus, $\mathcal{A}$ is exactly equal to the union of the selected maximal IB hypercubes, Q.E.D. (lemma 1).

Now, construct an assignment s to the variables V as follows: for each selected hypercube $H_i^{v^d}$, set $s(h_i^{v^d}) = 1$. For all other variables h, set $s(h) = 0$.

Lemma 2 *The assignment s is a 0-1 solution for $L_{IB}(B, \mathcal{E})$.*

Proof: By definition, s is a 0-1 assignment. We will show that s is a solution by showing that it obeys all of the inequalities of types a-d:

Type **a**. Suppose that some inequality of this type does not hold. This implies that some set of two variables h_1, and h_2, we have $s(h_1) = s(h_2) = 1$. This implies in turn that two inconsistent hypercubes were selected, which contradicts lemma 1.

Type **b**. Inequalities of this type can be violated iff there is some node v for which more than one maximal hypercube is selected. But this cannot happen, as the algorithm only expands nodes once (in each solution path), and in the construction a hypercube was added only for nodes which had no hypercubes based on them.

Type **c**. The first summand is equal to the number of selected hypercubes based on v that assigns d to node v. It is equal 1 if $(v, d) \in \mathcal{A}$, and 0 otherwise. The second summand is equal to the number of hypercubes based on w that assign the value d to v. It is always 0 if $(v, d) \notin \mathcal{A}$, and at most 1 otherwise. Thus, for each w , v, d, the inequality holds, as it holds whether $(v, d) \in \mathcal{A}$ (because $1 - x \geq 0$ when $x \leq 1$) or not (because $0 - 0 \geq 0$).

Type **d**. Immediate, because there is at least one maximal IB hypercube $H_i^{v^d}$ based on each evidence node such that $(v, d) \in \mathcal{E}$, and thus $s(h_i^{v^d}) = 1$. Q.E.D. (lemma 2).

Theorem 2 follows from the construction, and from lemmas 1, 2. Q.E.D.

Let us now define a mapping from 0-1 solutions to assignments to the belief network, as follows:

Definition 34 *Given a 0-1 solution s for $L = L_{IB}(B, \mathcal{E})$, we define its induced assignment $\mathcal{A}[s]$ to belief network B as:*

$$\mathcal{A}[s] = \bigcup_{s(h_i^{v^d})=1} \{(w, d') | (w, d') \in H_i^{v^d}\} \tag{6.5}$$

[4]It is possible that when it is v's turn to be expanded, the IB condition already holds at v, and thus v is never expanded. This may happen if some parents of v are assigned when some sibling of v is expanded.

We call the set of hypercubes $H_i^{v^d}$ for which the respective variables $s(h_i^{v^d}) = 1$ the hypercubes selected by s. We also say that the respective variables $s(h_i^{v^d})$ are selected by s. We proceed to show that this is indeed the desired construction:

Theorem 17 *If s is a 0-1 solution to $L_{IB}(B, \mathcal{E})$, then $\mathcal{A}[s]$ is a consistent, evidentially supported IB assignment to B that subsumes $\mathcal{E}$.*

Proof: The assignment $\mathcal{A}[s]$ is consistent, because all pairs of hypercubes selected by s are consistent. The latter holds for each pair of hypercubes h_1, h_2 because if they are based on different nodes, then some constraint of type **a** is violated; and if they are based on the same node v, then the inequality of type **b** for node v is violated.

Assignment $\mathcal{A}[s]$ subsumes $\mathcal{E}$, because of the inequalities of type **d**. For each node v, there must be at least one variable $h_i^{v^d}$ selected by s such that $(v, d) \in \mathcal{E}$, because the summand has to be at least 1. Thus, the union of the hypercubes selected by s assigns values to all the nodes $v \in nodes\mathcal{E}$, such that $(v, d) \in \mathcal{E}$, as required.

$\mathcal{A}[s]$ is evidentially supported, because a variable $h_i^{v^d} \in V$ only if v is evidentially supported (definition 33), and only nodes above v can be assigned by a maximal IB hypercube based on v.

Finally, the IB condition holds for every node $v \in \mathcal{A}[s]$: suppose that for some node $v \in span(\mathcal{A}[s]$, the IB condition does not hold. This would imply that the assignment to v and its parents, according to $\mathcal{A}[s]$, is not subsumed by any IB hypercube. But that is clearly false, because there exists some inequality of type **c** such that the second summand is 1, with d such that $(v, d) \in \mathcal{A}[s]$. That is because if v must be assigned the value d in *some*, selected hypercube based on *some* w. This implies that:

$$\sum_{i=1}^{k^{v^d}} h_i^{v^d} \geq 1 \tag{6.6}$$

and this implies, in turn, that some hypercube $H_i^{v^d}$ is selected by s. But maximal IB hypercube $H_i^{v^d}$ clearly subsumes $\mathcal{A}[s]$, a contradiction. Thus, the IB constraint holds for every node, and $\mathcal{A}[s]$ is an IB assignment. Q.E.D.

Now that we have defined the system of inequalities equivalent to the problem of finding evidentially supported IB assignments, we will provide an objective function such that a minimum cost solution of the system provides an IB-MAP for the belief network:

Definition 35 *We define the objective function for IB solution as follows:*

$$\Theta_{L_{IB}}(s) = \sum_{h_i^{v^d}} \{s(h_i^{v^d}) c(h_i^{v^d}, T) + (1 - s(h_i^{v^d})) c(h_i^{v^d}, F)\} \tag{6.7}$$

Clearly, since $c(h_i^{v^d}, F) = 0$, and $c(h_i^{v^d}, T)$ is the negative log probability of $H_i^{v^d}$, we can write:

$$\Theta_{L_{IB}}(s) = \sum_{h_i^{v^d}} s(h_i^{v^d}) c(h_i^{v^d}, T) = -\sum_{h_i^{v^d}} s(h_i^{v^d}) log(P(H_i^{v^d})) \tag{6.8}$$

For a 0-1 solution, the objective function is:

$$\Theta_{L_{IB}}(s) = -\sum_{s(h_i^{v^d})=1} log(P(H_i^{v^d})) \tag{6.9}$$

Definition 36 *An optimal 0-1 solution for the constraint system L is a 0-1 solution that minimizes the objective function $\Theta_{L_{IB}}(s)$.*

We now show that optimal 0-1 solutions map to IB-MAPs, if the distribution $\mathcal{D}$ is strictly positive:

Theorem 18 *If the distribution $\mathcal{D}$ of belief network B is strictly positive, then $\mathcal{A}[s]$ is a maximal (w.r.t. subsumption) IB-MAP for B w.r.t. evidence $\mathcal{E}$, where s is an optimal 0-1 solution for $L_{IB}(B, \mathcal{E})$.*

Proof: From theorem 17, every 0-1 solution s induces an evidentially supported IB assignment $\mathcal{A}[s]$ that subsumes $\mathcal{E}$. But we also know that s is optimal, i.e. that there is no other 0-1 solution s' with a lower valued objective function. Since for every maximal, evidentially supported IB assignment $\mathcal{A}'$ that subsumes $\mathcal{E}$, there exists a 0-1 solution s' such that $\mathcal{A}' = \mathcal{A}[s']$, then $\mathcal{A}[s]$ is the IB assignment that produces the optimal s, i.e. there is no other assignment $\mathcal{A}'$ with an s' better than s. All that remains to be shown is a) optimizing s also maximizes $\mathcal{A}[s]$, and b) if s is an optimal 0-1 solution, then $\mathcal{A}[s]$ is a maximal among all evidentially supported assignments IB that subsume $\mathcal{E}$.

Proof for a): the probability of $\mathcal{A}[s]$ is, according to theorem 7:

$$P(\mathcal{A}[s]) = \prod_{v \in span(\mathcal{A}[s]} P(\mathcal{A}[s]^{\{v\}} | \mathcal{A}[s]^{nodes\mathcal{A}[s]\uparrow(v)}) \tag{6.10}$$

But that is also equal to the product of the probabilities of the selected hypercubes, thus:

$$P(\mathcal{A}[s]) = \prod_{s(h_i^{v^d})=1} P(H_i^{v^d}) = \prod_{s(h_i^{v^d})=1} e^{log(P(h_i^{v^d}))} \tag{6.11}$$

Using the rules for exponents and moving the minus sign outside the resulting summation, we get:

$$P(\mathcal{A}[s]) = e^{\sum_{s(h_i^{v^d})=1} log(P(h_i^{v^d}))} = e^{-\Theta_{L_{IB}}(s)} \tag{6.12}$$

and since e^{-x} is strictly monotonically decreasing in x, optimizing s (minimizing the objective function) is equivalent to maximizing $P(\mathcal{A}[s])$.

Proof for b) $\mathcal{A}[s]$ is maximal, because if it strictly subsumes some other IB assignment with the requisite properties $\mathcal{A}'$, then it also has a strictly larger probability (because the distribution is strictly positive). So, if there exists some s' such that $\mathcal{A}' = \mathcal{A}[s']$, then it must have a higher cost. If there is no such s', then it cannot be found as a 0-1 solution. Q.E.D.

Having proved the correctness of our reduction of finding IB MAPs to minimizing an objective function over linear inequality constraints, it is our hope that experiments will show that in most cases the 0-1 solution for the system of inequalities can be found in reasonable time. The experiments performed in [Santos Jr., 1991c] are encouraging, because there seems to be no great difference between the systems of equations we generate, and the kinds of systems generated for AND/OR trees, on which the experiments were performed. There is no guarantee that this will work, as in general finding such 0-1 solutions to linear systems of inequalities is NP-hard. The conclusions as to the usefulness of the reduction should thus be made based on empirical studies, which are left for future research.

6.3.3 δ-IB MAPs and Inequalities

It is our belief that δ-IB MAPs can use a similar reduction to linear systems of inequalities. In fact, the reduction will be exactly the same, except for the following changes:

1. Instead of maximal IB hypercubes, use δ-IB hypercubes.

2. The cost of assigning 1 to a hypercube variable should be the negative log probability of the optimistic (upper) bound.

3. Once an optimal 0-1 solution is found, it is not guaranteed to be optimal. Several successively less optimal 0-1 solutions need to be found, and then a post-processing step, as that of the δ-IB MAP algorithm, needs to be performed.

We do not believe that we should work on the details of this reduction, however, until the reduction for IB MAPs proves empirically useful.

6.4 Simulation and IB-MAPs

Simulation algorithms are used both for finding posterior probabilities, and for finding MAPs for Markov networks. It is not immediately obvious how such techniques can be used for finding irrelevance-based explanations. The problem is that traditional simulation algorithms eventually assign all the nodes, and it is not at all clear how to adapt such algorithms to leave nodes unassigned in a controlled manner. We will thus not attempt to do so, beyond very general suggestions as to when this should not even be attempted, in our opinion.

For a simulation algorithm to work (assuming that the problem presented in the previous paragraph is solved), it is important that an irrelevance-based assignment be recognizable efficiently. Otherwise, even one simulation cycle will take an inordinate amount of time. In the case of IB assignments and δ-IB assignments, our locality theorems guarantee efficient, linear-time recognition. This is not the case, however, for quasi-independence based assignments. We thus believe that simulation techniques may prove useful for finding irrelevance-based MAPs for the IB and for the δ-IB schemes, but not for quasi independence.

6.5 Summary

In this chapter, we have shown by experiment and by example that irrelevance-based explanation is a useful notion. We have also shown that our algorithms, despite being exponential-time in principle, execute in reasonable time in practice, for a medium-size example.

Despite that, we are not really happy with the performance of the algorithms. It is likely that for really large networks, unacceptable performance will result. We proposed a reduction of finding IB MAPs to finding an optimal 0-1 solution to a linear system of inequalities. Even though the latter is a hard problem, experiments performed by other researchers are encouraging, in that good performance is obtained for systems resulting from reduction of the problem of finding minimal-cost proofs for AND/OR graphs, which is a problem very similar to finding IB MAPs.

Chapter 7

Summary

We have compiled a list of desirables for explanations. We then proceeded to examine existing systems, with a special focus on probabilistic systems, with respect to the list. Currently used explanation systems were found lacking in several respects, a shortcoming which we attempted to remedy by providing our idea of irrelevance-based explanations.

We made the connection between non-probabilistic schemes and probability theory by providing a probabilistic semantics for cost-based abduction. The semantics provides a common ground for evaluating non-probabilistic schemes in a probabilistic framework, which we have done for cost-based abduction.

Within the framework of irrelevance-based explanation, we examined the question of how to define irrelevance, in the context of a belief network. We examined definitions of irrelevance in the literature, and attempted to generalize them to fit our requirements. That we have done, with varying degrees of success, with our schemes of chapters 4 and 5. We then looked for ways of implementing these schemes of explanation, and have done so by designing a best-first algorithm that can easily provide alternate explanations, as well as the best explanations.

We then presented an extended example, over a toy domain, and performed timing experiments that showed that the algorithms execute in reasonable time over medium-sized networks. We also proposed alternate algorithms for computing irrelevance-based explanations.

In short, the contributions of this thesis are:

1. It provides a framework for evaluating explanation systems.
2. It provides probabilistic semantics for schemes of explanation that are originally non-probabilistic.
3. It defines the meta-scheme of irrelevance-based explanation, to solve problems inherent in MAP based explanation. We investigate four sub-classes of irrelevance-based explanation:
   (a) Irrelevance-based partial MAPs.
   (b) δ-independence based partial MAPs.
   (c) Quasi-independence based partial MAPs.
   (d) Generalized δ-independence based MAPs.
4. We prove the following important properties of partial assignments:
   (a) Independence-based partial assignments can be recognized in linear time.
   (b) The probability of an independence-based partial assignment is linear-time computable.
   (c) δ-independence based partial assignments can be recognized in linear time.
   (d) Reasonable bounds for the probability δ-independence based partial assignments can be computed in linear time.

5. Design an effective best-first algorithm for computing MAPs, that executes in linear time for poly-trees. The algorithm computes next-best explanations with a trivial modification, and can be easily adapted for computing various types of partial MAPs.

6. Design effective algorithms for implementing explanations in the proposed schemes, specifically independence-based MAPs and δ-independence-based MAPs. These algorithms are also best-first (a modification of our MAP algorithm), and can be easily extended to compute next-best explanations.

7. Suggest alternate algorithms for finding irrelevance-based explanations, through reduction to linear systems of inequalities, based on the ideas in [Santos Jr., 1991a], [Santos Jr., 1991d], [Santos Jr., 1991b], and [Santos Jr., 1991c].

The research done in the thesis opens up some interesting lines of future research, some of a theoretical nature, on the theory of explanation, and others on practical algorithmic issues:

1. Theoretical issues:

   (a) Solving the remaining problem that high prior probability nodes may be selected in an explanation when they are *relevant*, but have a *negative* support on the evidence, as discussed in the summary of chapter 5.

   (b) What to do when disjunctive assignments cannot capture a concept, as in the case where a multiple valued node is split into several binary nodes, as in the end of chapter 5?

2. Practical issues:

   (a) Improved heuristics for the best-first search MAP algorithms. In particular, it remains to be seen whether the heuristic proposed in [Charniak and Husain, 1991] can be made admissible for MAPs and partial MAPs, while remaining a useful heuristic.

   (b) Experiments on how well using the reduction of computation of partial MAPs to linear systems of inequalities works in practice should be performed. An empirical comparison of timing results for our best-first IB MAP algorithm to the performance of the linear systems method would be particularly interesting.

Appendix A

Probabilistic Networks

In this appendix, we shall review the traditional probabilistic network models: Markov networks, and Bayesian networks (belief networks), with examples of what can be done with these models. We start off with examples, interspersed with the formal background definitions and theorems that are used implicitly throughout the thesis.

A.1 Probability and Evidential Reasoning

When we use probability theory to represent uncertainty, we buy into all the standard properties of probability theory. We have a set of random variables V, each with its own domain ω_v of events[1], where $v \in V$. All the values in ω_v are assumed to be *disjoint* events, and they span all the possible outcomes of v. If that holds, then ω_v is a sample space (or probability space). We say that for each v, the domain ω_v is the set of *outcomes* of any set of *trials* for variable v.

For each variable, we can define a distribution over its sample space, i.e. a function $P : \omega_v \rightarrow [0,1]$, called the *probability*. If $d \in \omega_v$, $P(d)$ is the probability of d, the fraction of trials for v which comes out d, as the number of trials approaches infinity. The sum of all the probabilities over a sample space is 1.

Likewise, define a *joint sample space* over the set of variables V, as the Cartesian product of all the sample spaces:

$$\Omega_V = \prod_{v \in V} \omega_v \tag{A.1}$$

and a joint probability distribution $\mathcal{D}$ over that space. Again, the sum of probabilities over $\mathcal{D}$ is 1. We can also talk about the probability of any subset of the sample space. We write the probability of any such subspace A (also called an event) as $P_{\mathcal{D}}(A)$. Whenever unambiguous (i.e. there is only one distribution $\mathcal{D}$ we are considering) we omit the subscript and also the reference to Ω_V.

For convenience, we will review the axioms of of probability theory, in a form that takes into account our simplifying assumption that the joint sample space is finite:

$$\forall A \in 2^{\Omega_V} \quad P(A) \geq 0 \tag{A.2}$$

$$\sum_{A \in \Omega_V} P(A) = 1 \tag{A.3}$$

$$\forall A, B \in 2^{\Omega_V} \quad A \cap B = \phi \rightarrow P(A \cup B) = P(A) + P(B) \tag{A.4}$$

[1] Actually, random variables are functions defined over the probability space, but we choose to ignore that and just assume that it is always the identity function, thus that the sample space (the domain for each random variable v), is also the range of this function.

The above axioms state that the probability of an event is greater than or equal to 0, that the sum of probabilities over the entire sample space is 1, and that the probability of the union of two disjoint events is the sum of probabilities of the individual events.

We will use an example from the domain of simple commonsense explanation: our set of variables is {road-wet, rain}, each with the domain $\{T, F\}$ (standing for true and false respectively). We will assume that the variables are ordered, for convenience, i.e. we actually have the pair of variables (road wet, rain). Our sample space is: $\{(F,F),(F,T),(T,F),(T,T)\}$. We also use an assignment notation to refer to events, e.g. the event (T,T) can be written (road-wet=T, rain=T). We use assignments and events interchangeably throughout the thesis. One valid distribution for our sample space is:

$$P(F,F) = 0.3 \quad P(F,T) = 0.1 \quad P(T,F) = 0.2 \quad P(T,T) = 0.4$$

These numbers (together with the axioms of probability) are sufficient to allow us to determine the probability of any event. For example, the event rain=T is the union of the events {(rain=T,road-wet=F)} and {(rain=T,road-wet=T)}, and thus $P(\text{rain} = T) = 0.5$. The latter is known as a *marginal* probability, where we have just "marginalized" over all the possible values of road-wet.

Sometimes, only a subset of the sample space is used, by introducing using a *conditioning event*, and the probabilities used w.r.t. the new sample space are called *conditional probabilities*. Conditional probabilities are denoted $P(A|B)$ (probability of A given B), where B is the conditioning event, and are defined whenever the probability of the conditioning event B is not 0) as:

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \tag{A.5}$$

For example, the probability of Road-wet[2] given Rain is:

$$P(\text{Road-wet}|\text{Rain}) = \frac{P(\text{Road-wet} \cap \text{Rain})}{P(\text{Rain})} = \frac{0.4}{0.5} = 0.8$$

Suppose that we want to find the probability of Rain given Road-wet, and that we do not know the probability of Road-wet. In that case, we might wish to use Bayes' formula:

$$P(A|B) = \frac{P(A)P(B|A)}{P(A)P(B|A) + P(\neg A)P(B|\neg A)} \tag{A.6}$$

or its generalization, where we use the sum of any partition of the values of A in the denominator, instead of just adding two variables. In our example, in order to use Bayes' formula, we need to know the probability of Road-wet given no Rain (which we calculate out to be 0.4). Now we can write:

$$\begin{aligned} P(\text{Rain}|\text{Road-wet}) &= \frac{P(\text{Rain})P(\text{Road-wet}|\text{Rain})}{P(\text{Rain})P(\text{Road-wet}|\text{Rain}) + P(\neg\text{Rain})P(\text{Road-wet}|\neg\text{Rain})} \\ &= \frac{0.5 \times 0.8}{0.5 \times 0.8 \; + \; 0.5 \times 0.4} = \frac{2}{3} \end{aligned}$$

Using Bayes' rule in such a manner is a simple instance of *evidential reasoning*, where we get evidence (Road-wet) and need to find the probability of a cause (Rain) given the evidence. Evidential reasoning is frequently used within probabilistic systems of explanation.

The problem of using an explicit distribution, as in this section, is twofold:

[2]When a random variable has only the states true and false, we sometimes use the capitalized name of the variable to denote the event that the variable is true. Likewise, we use the capitalized name of the variable with a preceding negation symbol to denote that the variable is false.

1. The sample space is exponential in the number of random variables, which makes it hard to process and gather the statistics required for reasoning from the real world.

2. Human experts used for supplying the distributions directly (out of experience) usually find it much easier to supply conditional probabilities rather than joint probabilities

For these reasons, models where the number of probabilities used is smaller and where mostly conditional probabilities are to be specified are preferred. Two such models are belief networks and Markov networks.

A.2 Conceptual and Statistical Independence

In the previous section, we reviewed probability and discussed the problem caused by the size of the sample space. In this section we define independence and show how it can be used to alleviate that problem.

Many events in the real world are considered to be independent, for example a pair of dice tossed in my office coming up double-sixes is (usually) independent of the price of tea in China. We say that these events are conceptually independent. We also talk about *statistical* independence of events A and B, which is defined as follows: events A and B are statistically independent if and only if $P(A \cap B) = P(A)P(B)$. An alternate way to say it is that $P(A) = P(A|B)$, i.e. that the probability of A stays the same even if we condition on B.

Statistical independence (and its generalization to many variables, joint independence) is very useful, because if in a set of random variables V of size n (each with d possible values) all variables are jointly independent, then it is sufficient to specify nd probability values, instead of an unwieldy d^n probability values. Having *all* the variables jointly independent is just wishful thinking. However, it is reasonable to expect that there are a great many independencies in the system, which allows us to reduce the number of probabilities we need to specify. We also need models that take advantage of the independencies, which is what probabilistic networks do for us.

To conclude this section, we note that there is an intuitive correspondence between conceptually independent events and statistically independent events. In artificial intelligence, we want to be able to specify certain things in isolation whenever possible, essentially for the same reasons for outlined in the previous section with respect to specifying probability distributions. We can do that when whatever we are specifying is conceptually independent of everything else. That is why the AI community is interested in independence (or sometimes in the closely related *irrelevance*). This is also why formalisms that handle independencies are useful throughout AI, and not just for probabilistic reasoning.

A.3 Belief Networks

In previous sections we noted the necessity of having probabilistic models that handle independencies, and where simple conditional probabilities may be specified, instead of full joint probabilities. A very useful model is *belief networks*, also called *Bayesian networks*. This section reviews belief networks, which are a sub-concept of *influence diagrams*. We will not discuss the latter concept, but a discussion of influence diagrams (as well as belief networks) can be found in various published textbooks, such as [Pearl, 1988].

Belief networks are directed acyclic graphs (DAGs), where each node stands for a random variable, and each arc stands for a direct statistical dependency, or even direct causal relationship (where the tail of the arrow is the cause and its head the effect). For this reason, we use the terms *node* and *variable* interchangeably throughout, whenever unambiguous. For each node, the conditional probabilities of (each value of) the node given all possible assignments to its parents are given. These conditional probabilities fully determine the entire joint distribution of the network.

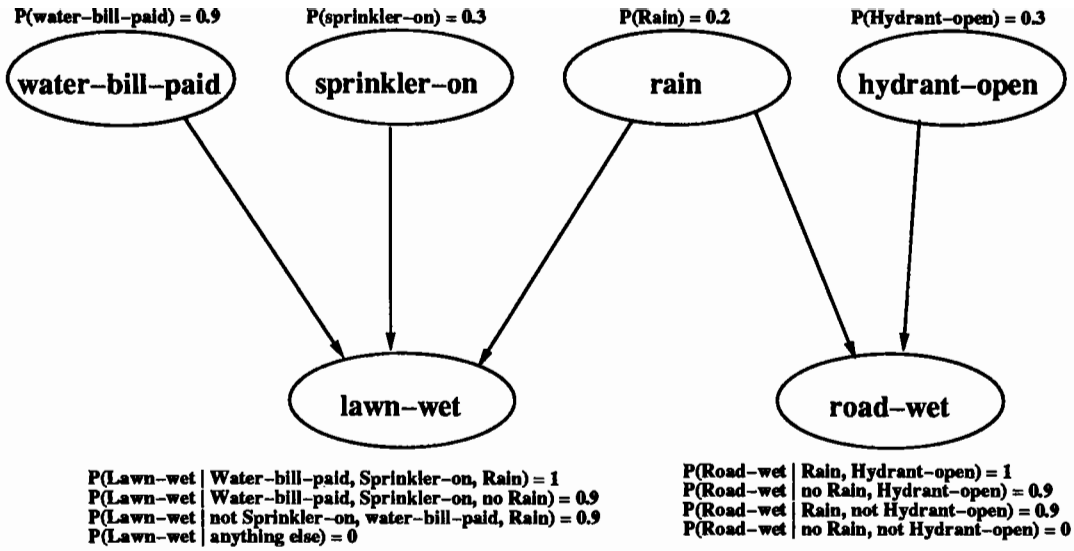

Figure A.1: Belief Network for Example

For a node v that has no parents (called a *root* node), prior probabilities are specified, i.e. for each value in the domain of v, the probability of that v getting that value must be provided.

Formally: let $\mathcal{A}$ is an assignment to the network (i.e. an arbitrary function that for each variable v gives a value in its domain, ω_v). Let $\mathcal{A}_S$ stand for the restriction of $\mathcal{A}$ to variable set S, i.e. $\mathcal{A}_S \in \mathcal{A}$, but the node set of $\mathcal{A}_S$ is S, and not all of V. The joint distribution of the network is:

$$P(\mathcal{A}) = \prod_{v \in V} P(\mathcal{A}_{\{v\}} | P(\mathcal{A}_{parents(v)}) \tag{A.7}$$

This equation is equivalent to the assumption that once the assignment to the parents of a node v is known, v is independent of any of its other ancestors.

Let us extend our example slightly, now that we have the tool of belief networks and can easily specify a network with more variables. Our set of variables is now: {rain, sprinkler-on, water-bill-paid, hydrant-open, road-wet, lawn-wet}. Suppose that we know that the first four variables are independent, and that the latter two variables are somehow caused by them (or, rather, by the events which these variables refer to). Suppose that we also know that the hydrant cannot affect the lawn, and that the state of the sprinkler (or our water bill) cannot affect the road. Using arcs to represent the direct dependencies, we get the graph of figure A.1.

What we have now is the DAG underlying the belief network. We still need to specify the conditional probabilities for all nodes given their parents, as well as the prior probabilities of the root nodes (rain, sprinkler-on, water-bill-paid, hydrant-open). We wrote down these probabilities next to the nodes in figure A.1. Note that since the nodes are all binary valued, we need to specify only one prior probability for each root node. That is because the axioms of probability theory dictate that, for example, $P(\neg\text{Rain}) = 1 - P(\text{Rain})$, and likewise for the other root nodes. Also, for non-root nodes (road-wet, lawn-wet) it is sufficient to supply the conditional probabilities for the positive assignment (Road-wet), and the probabilities for the negative assignment can be calculated using the axioms of probability theory. When a variable name appears in a probability equation, the meaning is that the equation holds for every possible state of the variable. Thus, the following equation (from figure A.1):

$$P(\text{Lawn-wet}|\neg\text{Sprinkler-on, water-bill-paid, Rain}) = 0.9$$

actually stands for the following two equations:

$$P(\text{Lawn-wet}|\neg\text{Sprinkler-on, Water-bill-paid, Rain}) = 0.9$$

$$P(\text{Lawn-wet}|\neg\text{Sprinkler-on, } \neg\text{Water-bill-paid, Rain}) = 0.9$$

The distribution of the entire network is given by the product of all the conditional distributions, as above:

$$\begin{aligned} P(\text{water-bill-paid, sprinkler-on, rain, hydrant-open, road-wet, lawn-wet}) = \\ P(\text{hydrant-open})P(\text{rain})P(\text{sprinkler-on})P(\text{water-bill-paid}) \\ P(\text{road-wet}|\text{hydrant-open, rain}) \\ P(\text{lawn-wet}|\text{rain, sprinkler-on, water-bill-paid}) \end{aligned}$$

We are now ready to perform evidential reasoning on our belief network. We allow conjunctive evidence, that is, events that are assignments of values to a subset of the nodes in the network. Suppose that our evidence is: {Road-wet, Lawn-wet}. We now wish to calculate the probability of each of the variables given our evidence. A popular scheme for doing that is *evidence propagation*, proposed in [Kim and Pearl, 1983]. That method works (in time linear in the size of the network) when the DAG is a poly-tree (i.e. a DAG such that the underlying undirected graph is a tree). In the case of our simple example, this condition holds, so we use the algorithm to find the probabilities given the evidence (also called *posterior probabilities*):

Sometimes, what we are after is an *explanation* of the evidence, i.e. the answer to the question: "Why are the road and lawn wet?". As we stated in the introduction, there is no agreed-upon, domain independent procedure for finding the answer. We review two commonly used methods here:

1. Threshold on posterior probabilities.

2. A Maximum A-Posteriori assignment (MAP).

To find the explanation for (Road-wet, Lawn-wet) using the first scheme, we find the posterior probabilities as above. We do that by running an evidence propagation algorithm (such as that used by IDEAL, [Srinivas and Breese, 1989]) on that network, with evidence {Road-wet, Lawn-wet}. The resulting posterior probabilities are shown in figure A.2. We then use a threshold value of probability (say 0.5) to decide whether the event is a part of the explanation. In our example, $P(\text{Rain}) = 0.766$ and $P(\text{Water-bill-paid}) = 0.926$ thus we say that Rain and Water-bill-paid is the explanation of observed evidence.

Using the MAP scheme, we find the assignment to all the variables that has maximum probability given the evidence, and call that assignment the explanation for the observed evidence. Actually, it is sufficient to maximize the prior probability of the assignment, with the extra constraint that the evidence nodes are assigned consistently with the evidence. Here, the maximum probability assignment is:

{Rain, Road-wet, Lawn-wet, Water-bill-paid, ¬Sprinkler-on, ¬Hydrant-open}

with a prior probability of approximately 0.0714, and a posterior probability of approximately 0.317.

We stated that nodes are dependent if there is an arc between them. In order to find whether nodes are *independent*, however, we need to use a different criterion, called d-separation. In a belief network, d-separation implies independence. [Pearl, 1988] uses the notation $I(X, Y, Z)$ to mean that node set X is independent of node set Y given node set Z. $I(X, Y, Z)$ holds whenever X and Y are d-separated given Z. If nodes are d-separated, (given some evidence, or even no evidence), then they are independent. Two (disjoint) nonempty sets of nodes X and Y are d-separated given node-set Z (where Z may be empty, and is disjoint from X and Y) if and only if the following conditions hold:

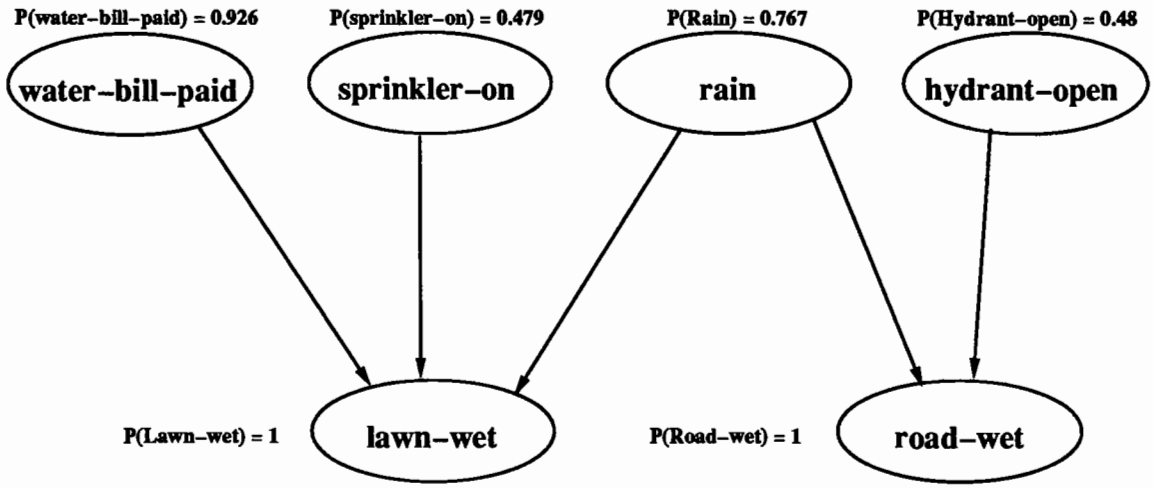

Figure A.2: Posterior Probabilities

1. Every directed path from X to Y or vice versa passes through Z.
2. If v is a node such that there is a path from X to v and a path from Y to v, then $v \notin Z$ and there is no path from v to Z.

The first condition is obvious. The second condition appears a bit odd, but it is based on the following (see figure A.1): we can see that rain is independent of hydrant-open with no evidence. But once we know that the road is wet, they become dependent. For example, if the road is wet, and we know that the hydrant is not open, the probability of rain goes up to 1.

Further discussion of probabilistic reasoning and explanation, as well as a discussion of the pros and cons of these schemes of explanation, appears in chapter 2.

Finally, we address the problem that can occur with distributions that are not strictly positive, and how to handle it. We said that the distribution of a belief network is a product of conditional probabilities of all the nodes given their parents. If, however, some probabilities can be zero, it is possible for conditional probabilities to be undefined. It is still true that the product of all conditional probabilities gives us the joint probability (because even the undefined conditional probabilities cannot be greater than 1, and they get multiplied by 0 in the product). When we want to make decisions about local independence, however, we are in trouble. We chose to say that independence does hold, whenever zero probability states like these occur. We do that by stating that the undefined conditional probabilities are "don't care" values. Thus, when we maximize hypercubes in chapter 4, we treat them the same way "don't care" values are treated in Karnaugh maps, i.e. the values are "wild-cards" that can be assumed to be equal to any value we want, and can be used for constructing each maximal IB hypercube (even if the IB hypercubes have different probabilities).

A.4 Markov Networks

Markov networks are undirected graphs, where the nodes are random variables, as in belief networks. As in belief networks, arcs stand for direct dependence of variables. A node v is independent of all the other nodes, given all its neighbors. The set of immediate neighbors of v forms a *Markov blanket* for v. This property is equivalent to stating that the Markov property holds w.r.t. v and its neighbors.

Stating the distribution of the network is not as simple as for belief networks. We cannot just give conditional probabilities, as undirected cycles may lead to contradictions. Instead, one must

specify a compatibility function for all the cliques of the graph, and from these the distribution can be computed. This fact is one of the reasons why we will not deal with Markov networks very much in the thesis, and why we do not go into the details of Markov networks here. The other reason is that since the graph is undirected, we have no way of enforcing the direction of causality.

Finally, we will note that most algorithms used for belief networks can usually be applied to Markov networks. In fact, some belief network algorithms, such as the algorithm for computing posterior probabilities presented in [Lauritzen and Speigelhalter, 1988], transform belief networks into Markov networks as a pre-processing step.

Appendix B

Proofs for Theorems

This appendix provides the lengthier proofs not provided in the text.

Theorem 1 *The given cost selection problem is NP-complete.*

Proof: We show that the problem is NP-hard, using reduction from the Vertex Cover Problem (VC), which is a known NP-complete problem [Garey and Johnson, 1979]. The problem is defined as follows: given a graph $G = \{V, E\}$, and an integer K, is there a subset $S \subseteq V$ such that $\mid S \mid \leq K$, and for all edges $e = (v_1, v_2)$ where $e \in E$, at least one of v_1, v_2 are in S ?

The reduction is as follows: given a graph $G = \{V, E\}$ and an integer K, construct a weighted AND/OR DAG, $T = \{G'$, c, f, $s\}$ as follows:

1. The nodes of G' are: an AND node s with cost $c(s, T) = \mid V \mid +1$, for every $v \in V$ a root node v' with cost $c(v', T) = 1$, and for every edge $e \in E$ an OR node v_e with cost $c(v_e, T) = \mid V \mid +1$. All other costs are 0.

2. The parents of s are exactly all the OR nodes constructed in step 1.

3. The parents of each OR node, v_e are exactly nodes v' where in G the edge e corresponding to v_e is incident on the node v corresponding to v' .

Clearly, the construction process is of complexity $O(\mid V \mid + \mid E \mid)$, and is thus of polynomial complexity.

Claim: The VC problem on G with cost K has an affirmative answer iff there exists a minimal cost satisfying truth assignment for G' with cost at most K.

Proof: We show that a satisfying truth assignment for G' with cost C induces a vertex cover for G with cost C, and vice versa.

- The cost of an assignment is exactly the total cost of the vertices assumed true. Obviously, only cost 1 vertices will be assumed true in the minimal cost assignment, because assuming all the root vertices true induces a satisfying truth assignment with cost $\mid V \mid$ which is less than the cost of assuming a non-root vertex true. But when we have a satisfying cost assignment where only root vertices are assumed, with a cost C, then there is a vertex cover of cost C for the VC problem, consisting exactly of the set of vertices corresponding to the set of assumed vertices, because each "edge" v_e is covered by at least one "vertex" v' .

- A subset S that solves the VC problem with cost C induces a satisfying truth assignment on G' , consisting of assuming true all root vertices corresponding to nodes in S, setting all other root vertices to F^A, and assigning all other vertices to T. Clearly, this assignment has cost C, and is consistent because each vertex corresponding to an edge has at least one parent assumed true (solution to VC), and the sink node has all parents with value T, hence it has value T.

The Given Cost Selection Problem with cost K is in NP, as the following non-deterministic polynomial time algorithm, solves it:

1. Propose an assignment for the WAODAG.
2. If the assignment is consistent, (takes $O(\mid V + E \mid)$ steps to check), continue.
3. If the assignment has cost at most K (takes $O(\mid V \mid)$ steps to check), halt with an affirmative answer.

The given cost selection problem is in NP, and is NP hard, and is thus NP-complete, Q.E.D.

Theorem 5 *If, for some belief network B with strictly positive distribution, $\mathcal{A}_S$ is a complete assignment w.r.t. node set S, then for any node $x \in S$, $In(\mathcal{A}_{\{x\}}, \uparrow^+(x) - S | \mathcal{A}_{S \cap \uparrow(x)})$ iff $In(\mathcal{A}_{\{x\}}, \uparrow(x) - S | \mathcal{A}_{S \cap \uparrow(x)})$.*

Proof ($\leftarrow$): Define U as the set of all the nodes of the belief network B that are not in S. The proof outline is as follows:

1. Construct a belief network B' that is the same as B, but has extra y nodes (as we will show), and an x' node that is true just in case $\mathcal{A}_{\{x\}}$ occurs in the original network.
2. Show that the B' has the same distribution as B when B' is marginalized to the original nodes in B.
3. Argue that the a-posteriori distribution of B given $\mathcal{A}_{S \cap \uparrow(x)}$ is equal to the a-posteriori distribution of B' given $\mathcal{A}_{S \cap \uparrow(x)}$ and the new y nodes.
4. Show that x' is d-separated from $\uparrow^+(x) - S$ given $\mathcal{A}_S$ and the y nodes.
5. The d-separation in B' implies independence in B', hence independence in B, as required.

We expand the arcs coming into each node x in S to reflect the conditional independencies in $P(x | \uparrow(x))$. For example, consider the subgraph of figure B.1a. s and x are nodes in S, u is an unassigned node. In this example, u and s are binary nodes (without loss of generality), and x is a (possibly) multiple-valued node, with k values, $x_1, ..., x_k$.

Suppose (without loss of generality) that we know that $P(x = x_1 | s = T)$ is independent of u ($s = T$ is the value assigned to node s in $\mathcal{A}_S$). We expand all possible assignments to s and u, and use y nodes for each such possible assignment. Because of the independence, the nodes for $\{s = T, u = F\}$ and $\{s = T, u = T\}$ are combined into one node, y_T. For the other cases, we have nodes y_{FF} and y_{FT} (see figure B.1b). The y nodes are binary, with probability 1 of being true just in case the state of their parents is equal to their subscripts, and 0 otherwise (for example, $P(y_{FT} | s = F, u = T) = 1$, and 0 given any other combination). The x' node is a binary node, that is set to be true iff the original x node is assigned x_1. Thus, $P(x' | y_T) = P(x = x_1 | s = T)$, and likewise for the other conditional probabilities.

Now, the new x node has exactly the distribution of the old x node, except that now $P(x = x_1 | x') = 1$ independent of s and u, and all the other conditional probabilities stay the same. We need this x' construct to handle the case where x is a multiple-valued node. It is sufficient to prove that x' is independent of any ancestors not in S to show that $P(x = x_1 | s = T)$ is independent of them, because these distributions are equal by construction. But the distribution with $s = T$ in the original graph is equivalent to the distribution where $s = T, y_T = T, y_{FT} = F, y_{FF} = F$ in the expanded graph.

Clearly, x' is d-separated from all its ancestors not in S (that are not y nodes), because all paths reaching x' from above are blocked by y nodes. All paths reaching x' from below either have

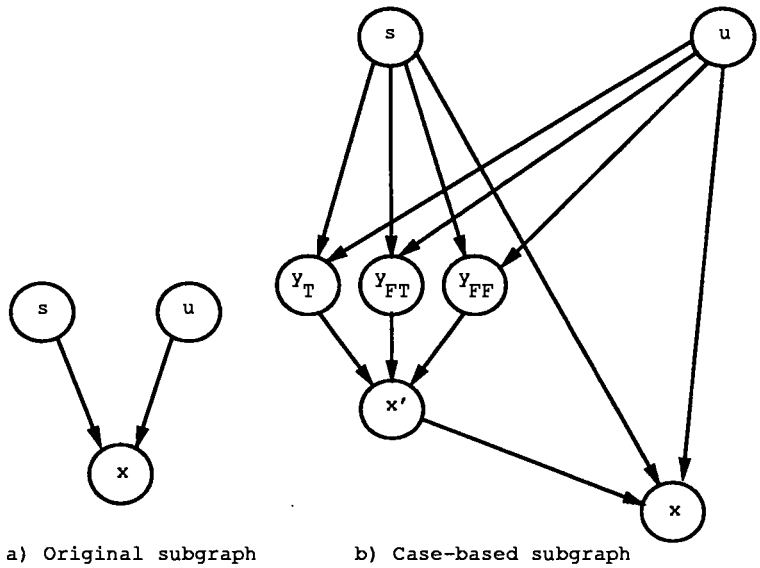

Figure B.1: Exploiting d-separation in proof

a converging node x, or some node below x. But there are, by definition, no nodes in $S \cap \uparrow(x)$ below x, and thus all paths reaching x' from below are blocked. Thus, x' is d-separated from all its ancestors not in S, and thus $P(x = x_1)$ is independent of these ancestors given $s = s_1$ and also given $\mathcal{A}_{S \cap \uparrow(x)}$.

Since this construction and d-separation argument generalizes for non-binary nodes s and u, and also to the case of more than 2 parents per node, then $In(\mathcal{A}_{\{x\}}, \uparrow^+(x) - S | \mathcal{A}_{S \cap \uparrow x})$.

Proof ($\rightarrow$): Follows immediately, because $In(\mathcal{A}_{\{x\}}, \uparrow^+(x) - S | \mathcal{A}_{S \uparrow(x)})$ translates into a set of equalities that is a superset of $In(\mathcal{A}_{\{x\}}, \uparrow(x) - S | \mathcal{A}_{S \cap \uparrow(x)})$, Q.E.D.

Theorem 7 *If for every node $v \in S$, $In(v, \uparrow(v) - S, \mathcal{A}_{S \cap \uparrow(v)})$, then we can compute the probability of the assignment as:*

$$P(\mathcal{A}_S) = \prod_{v \in S} P(\mathcal{A}_{\{v\}} | \mathcal{A}_{S \cap \uparrow(v)})$$

Proof: Let B stand for the set of nodes in the belief network, and let $O = B - S$. Let $\mathcal{B}$ stand for any arbitrary assignment to the variables in O, and $\mathcal{D} = \mathcal{A}_S \cup \mathcal{B}$. Now, the distribution of a belief network is the product of conditional probabilities. Thus, we can write:

$$P(\mathcal{D}) = \prod_{v \in B} P(\mathcal{D}_{\{v\}} | \mathcal{D}_{\uparrow(v)})$$

Since $\mathcal{A}_S$ is a partial assignment that is consistent with $\mathcal{D}$, its probability is just a marginalization of $\mathcal{D}$, that is, we need to sum over all assignments to the O variables (i.e. over $\mathcal{C}_O$).

$$P(\mathcal{A}_S) = \sum_{\mathcal{B} \in \mathcal{C}_O} P(\mathcal{A}_S, \mathcal{B})$$

We can now operate on the right hand side by using the equation for $P(\mathcal{D})$ and then partitioning the product according to the O and S sets:

$$P(\mathcal{A}_S) = \sum_{\mathcal{B} \in \mathcal{C}_O} \prod_{v \in B} P((\mathcal{A}_S \cup \mathcal{B})_{\{v\}} | (\mathcal{A}_S \cup \mathcal{B})_{\uparrow(v)})$$

$$P(\mathcal{A}_S) = \sum_{\mathcal{B} \in \mathcal{C}_O} \prod_{v \in S} P(\mathcal{A}_{\{v\}} | (\mathcal{A}_S \cup \mathcal{B})_{\uparrow(v)}) \prod_{v \in O} P(\mathcal{B}_{\{v\}} | (\mathcal{A}_S \cup \mathcal{B})_{\uparrow(v)})$$

But the product over S above contains only nodes that are dependent only on nodes in S. Thus, we can write:

$$P(\mathcal{A}_S) = \sum_{\mathcal{B}\in\mathcal{C}_O} \prod_{v\in S} P(\mathcal{A}_{\{v\}}|\mathcal{A}_{S\cap\uparrow(v)}) \prod_{v\in O} P(\mathcal{B}_{\{v\}}|(\mathcal{A}_S \cup \mathcal{B})_{\uparrow(v)})$$

$$P(\mathcal{A}_S) = \prod_{v\in S} P(\mathcal{A}_{\{v\}}|\mathcal{A}_{S\cap\uparrow(v)}) \sum_{\mathcal{B}\in\mathcal{C}_O} \prod_{v\in O} P(\mathcal{B}_{\{v\}}|(\mathcal{A}_S \cup \mathcal{B})_{\uparrow(v)}) \quad \text{(B.1)}$$

But the product over S is exactly what we need to be equal to $P(\mathcal{A}_S)$, so it is sufficient to prove that the sum on the right-hand side is equal to 1. This we can show by noting that we have a sum of all the states of a sample space, which have to sum to 1, as we show in the following paragraphs. We use two properties of belief networks:

1. Belief networks are directed *acyclic* graphs.
2. For every node $v \in B$, the probability of a node given its parents is a distribution, that is:

$$\forall \mathcal{B} \in \mathcal{C}_{\uparrow(v)}. \sum_{\mathcal{A}\in\mathcal{C}_{\{v\}}} P(\mathcal{A}|\mathcal{B}) = 1$$

We now number the k nodes of O in non-descending order, that is, we give them the labels $v_1, ..., v_k$ such that:

$$\forall i,j.\ i > j \ \Rightarrow\ v_j \notin \uparrow^+ (v_i)$$

This can be done because of property 1 of belief networks.

Now we can start manipulating our summation of equation B.1:

$$\sum_{\mathcal{B}\in\mathcal{C}_O} \prod_{v\in O} P(\mathcal{B}_{\{v\}}|(\mathcal{A}_S \cup \mathcal{B})_{\uparrow(v)}) = \sum_{\mathcal{B}\in\mathcal{C}_{O-\{v_1\}}} \sum_{\mathcal{B}_{\{v_1\}}} \prod_{i=1}^{k} P(\mathcal{B}_{\{v_i\}}|(\mathcal{A}_S \cup \mathcal{B})_{\uparrow(v_i)})$$

But in the product on the right-hand side above, terms for $i > 1$ contain no reference to v_1 and are thus independent of it. We can thus re-write the summation as:

$$\sum_{\mathcal{B}\in\mathcal{C}_{O-\{v_1\}}} \prod_{i=2}^{k} P(\mathcal{B}_{\{v_i\}}|(\mathcal{A} \cup \mathcal{B})_{\uparrow(v_i)}) \sum_{\mathcal{B}\in\mathcal{C}_{\{v_1\}}} P(\mathcal{B}_{\{v_1\}}|(\mathcal{A} \cup \mathcal{B})_{\uparrow(v_1)})$$

But the right-most summation is equal to 1 by property 2 of belief networks, and can thus be eliminated. And since for every $i > j$, term i does not refer to v_j, we can eliminate all the nodes in this manner, and we get:

$$\sum_{\mathcal{B}\in\mathcal{C}_O} \prod_{v\in O} P(\mathcal{B}_{\{v\}}|(\mathcal{A}_S \cup \mathcal{B})_{\uparrow(v)}) = 1$$

The theorem follows from the above equation and from equation B.1, Q.E.D.

Theorem 11 *Every maximal (w.r.t. subsumption) independence-based assignment $\mathcal{A}_S$ that is properly evidentially supported w.r.t. $\mathcal{E}$ is τ-reachable from $\mathcal{E}$.*

Proof: We show that for any τ based sequence of assignments beginning with $\{\mathcal{E}\}$, where each assignment $\mathcal{A}_{S'_k}$ where node v_k is about to be expanded, subsumes $\mathcal{A}_S$. We show that such an assignment exists for every k, and then that all the nodes v_i, $i \leq k$, are assigned exactly as in $\mathcal{A}_S$.

Lemma 3 *If $\mathcal{A}_{S'_k}$ is any assignment in $\tau^*(\{\mathcal{E}\})$ that subsumes $\mathcal{A}_S$, where the k subscript denotes that node v_k is the next node to be expanded (even if it cannot be expanded because it is unassigned), then for all nodes where $i \leq k$, we have that $V_i \in S \leftrightarrow V_i \in S'$.*

($V_i \in S \leftarrow V_i \in S'$): trivial, because $\mathcal{A}_{S'_k}$ subsumes $\mathcal{A}_S$.

($V_i \in S \rightarrow V_i \in S'$): by contradiction. Consider the smallest i for which $V_i \in S \wedge V_i \notin S'$. But, because all nodes with index less than i are assigned the same way as in $\mathcal{A}_S$, and V_i cannot influence the IB condition for any node with index greater than i, we can remove the assignment to V_i from S and all nodes in S that are properly evidentially supported only through V_i, and the resulting assignment is still properly evidentially supported and independence-based. Node V_i cannot affect the IB condition at nodes with smaller i, because the τ operator expansion of a node assures that the IB condition holds at that node, and all these nodes have already been expanded. Thus, since V_i (and perhaps several other nodes) can be removed from the assignment $\mathcal{A}_S$ while preserving its IB condition and proper evidential support, then $\mathcal{A}_S$ is not maximal with respect to subsumption, which contradicts the antecedent of this lemma, Q.E.D. (lemma 3).

Lemma 4 *For every k, there exists an assignment $\mathcal{A}_{S'_k} \in \tau^*(\{\mathcal{E}\})$ that subsumes $\mathcal{A}_S$.*

Proof: we need to show that $\mathcal{A}_{S'_k}$ that obeys the antecedent actually exists. This is true if at every node v there exists a hypercube that subsumes the assignment to the parents of v by $\mathcal{A}_S$. But the latter holds because τ can assign parents of v according to any *maximal* IB hypercube, i.e. for any partial assignment to the parent nodes, there exists (possibly more than 1) maximal IB hypercube that contains it. For nodes that are not expanded by τ, the condition holds vacuously. Thus, there exists, for every k, an assignment $\mathcal{A}_{S'_k} \in \tau^*(\{\mathcal{E}\})$ that subsumes $\mathcal{A}_S$, proving the lemma.

Now, since for every k there exists an assignment that subsumes $\mathcal{A}_S$, and where all the nodes with index below k are assigned exactly as in $\mathcal{A}_S$, that is τ-reachable from $\mathcal{E}$, all we need to do is take k to be the largest-index node of S, and the resulting assignment is exactly $\mathcal{A}_S$. Q.E.D.

The following theorem on δ-IB assignments does not appear in the text, but is useful for proving other theorems on δ-IB assignments. It states that to bound the probability of a node (given some parents), it is sufficient to bound the probabilities over all assignments to the *parents*, and there is no need to assign values to other ancestors.

Theorem 19 *For every assignment A_S to a belief network with variables B, the following conditions hold:*

$$\min_{\mathcal{B} \in \mathcal{C}_{U \cap \uparrow(v)}} P(\mathcal{A}_{\{v\}} | \mathcal{A}_{S \cap \uparrow(v)}, \mathcal{B}) \leq \min_{\mathcal{D} \in \mathcal{P}_{U \cap \uparrow^+(v)}} P(\mathcal{A}_{\{v\}} | \mathcal{A}_{S \cap \uparrow(v)}, \mathcal{D}) \tag{B.2}$$

$$\max_{\mathcal{B} \in \mathcal{C}_{U \cap \uparrow(v)}} P(\mathcal{A}_{\{v\}} | \mathcal{A}_{S \cap \uparrow(v)}, \mathcal{B}) \geq \max_{\mathcal{D} \in \mathcal{P}_{U \cap \uparrow^+(v)}} P(\mathcal{A}_{\{v\}} | \mathcal{A}_{S \cap \uparrow(v)}, \mathcal{D}) \tag{B.3}$$

Where $U = B - S$ and $v \in S$.

Proof: We will assume, without loss of generality, that $S \subset (\uparrow(v) \cup \{v\})$. We can do that because we always explicitly refer to nodes that are ancestors of v in the proof.

Define $S' = S - \{v\}$. Let us consider all assignments $\mathcal{D} \in \mathcal{C}_{S' \cup U'}$ (where $U' \subset U \cap \uparrow^+(v)$), that are complete w.r.t. $S' \cup U'$. Define $O = U \cap \uparrow^+(v) - U'$, the "other" nodes. For every such assignment, we can write (conditioning):

$$P(\mathcal{A}_{\{v\}} | \mathcal{A}_{S'}, \mathcal{D}_{U'}) = \sum_{\mathcal{B} \in \mathcal{C}_O} P(\mathcal{A}_{\{v\}} | \mathcal{A}_{S'}, \mathcal{D}_{U \cap \uparrow^+(v)}, \mathcal{B}) P(\mathcal{B})$$

But since the first term in the summation is conditioned on complete assignments, and since v is independent of all its non-immediate ancestors given *all* its immediate predecessors, and we are conditioning on *all* of the immediate predecessors of v, then

$$P(\mathcal{A}_{\{v\}} | \mathcal{A}_{S'}, \mathcal{D}_{U'}) = \sum_{\mathcal{B} \in \mathcal{C}_O} P(\mathcal{A}_{\{v\}} | \mathcal{A}_S, \mathcal{D}_{\uparrow(v)}, \mathcal{B}_{\uparrow(v)}) P(\mathcal{B}) \tag{B.4}$$

But since the set of assignments $\mathcal{B} \in \mathcal{C}_O$ constitute a sample space, then:

$$\sum_{\mathcal{B} \in \mathcal{C}_O} P(\mathcal{B}) = 1 \tag{B.5}$$

Using the convexity theorem, we get:

$$P(\mathcal{A}_{\{v\}} | \mathcal{D}_{S'}, \mathcal{A}_{U'}) \geq \min_{\mathcal{B} \in \mathcal{C}_O} P(\mathcal{A}_{\{v\}} | \mathcal{A}_{S \cap \uparrow(v)}, \mathcal{D}_{\uparrow(v)}, \mathcal{B}_{\uparrow(v)}) \tag{B.6}$$

In fact, the minimization term in equation B.6 need only be taken over assignments to immediate predecessors of v, because $P(\mathcal{A}_{\{v\}} | \mathcal{A}_{S \cap \uparrow(v)}, \mathcal{D}_{\uparrow(v)}, \mathcal{B}_{\uparrow(v)})$ is independent of the nodes in O that are not in $\uparrow(v)$. Also, for every possible O as defined above:

$$\min_{\mathcal{B} \in \mathcal{C}_{O \uparrow(v)}} P(\mathcal{A}_{\{v\}} | \mathcal{B}_{\uparrow(v)}) \geq \min_{\mathcal{B} \in \mathcal{C}_{U \cap \uparrow(v)}} P(\mathcal{A}_{\{v\}} | \mathcal{A}_{S \cap \uparrow(v)}, \mathcal{B})$$

That is because the right-hand side minimizes over at least all the assignments that the left-hand side does. The right-hand side of the above equation is equal to the left-hand side of equation B.2, and thus the other ancestors of v do not change the probability. And, as the left-hand side of equation B.6 is correct for any arbitrary $\mathcal{D} \in \mathcal{C}_{U'}$, and the set of all such assignments is exactly the set of all possible non-strictly partial assignments to $U \uparrow^+(v)$, then the minimum of $P(\mathcal{A}_{\{v\}} | \mathcal{D}_{S'}, \mathcal{A}_{U'})$ over that set is equal to the right-hand side of equation B.2. And since we have just shown that the term is greater or equal to the left-hand side, we have proved equation B.2.

As for equation B.2, we can use the same convexity arguments, with equations B.4 and B.5, to get

$$P(\mathcal{A}_{\{v\}} | \mathcal{A}_{S'}, \mathcal{D}_{U'}) \leq \max_{\mathcal{B} \in \mathcal{C}_O} P(\mathcal{A}_{\{v\}} | \mathcal{A}_{S \cap \uparrow(v)}, \mathcal{D}_{\uparrow(v)}, \mathcal{B}_{\uparrow(v)})$$

We then use arguments as above to prove equation B.3, Q.E.D.

Theorem 14 *For every node v in assignment A_S where $In_\delta(\mathcal{A}_{\{v\}}, \uparrow(v) - S | \mathcal{A}_{S \cap \uparrow(v)})$ holds, $In_\delta(\mathcal{A}_{\{v\}}, \uparrow^+(v) - S | \mathcal{A}_{S \cap \uparrow(v)})$ holds as well.*

Proof: Let us define the following variables:

$$MIN_c = \min_{\mathcal{B} \in \mathcal{C}_{\uparrow(v)-S}} P(\mathcal{A}_{\{v\}} | \mathcal{A}_{S \cap \uparrow(v)}, \mathcal{B})$$

$$MAX_c = \max_{\mathcal{B} \in \mathcal{C}_{\uparrow(v)-S}} P(\mathcal{A}_{\{v\}} | \mathcal{A}_{S \cap \uparrow(v)}, \mathcal{B})$$

$$MIN_p = \min_{\mathcal{B} \in \mathcal{P}_{\uparrow^+(v)-S}} P(\mathcal{A}_{\{v\}} | \mathcal{A}_{S \cap \uparrow(v)}, \mathcal{B})$$

$$MAX_p = \max_{\mathcal{B} \in \mathcal{P}_{\uparrow^+(v)-S}} P(\mathcal{A}_{\{v\}} | \mathcal{A}_{S \cap \uparrow(v)}, \mathcal{B})$$

The condition of the theorem implies that:

$$MIN_c \geq (1 - \delta) MAX_c$$

Now, theorem 19 states that it is always true that $MIN_c \leq MIN_p$ and $MAX_c \geq MAX_p$, thus we can substitute the p subscript term for the c subscript terms in the equation, and get:

$$MIN_p \geq (1 - \delta) MAX_p$$

This equation is equivalent to stating that $In_\delta(\mathcal{A}_{\{v\}}, \uparrow^+(v) - S | \mathcal{A}_{S \cap \uparrow(v)})$. Q.E.D.

Theorem 15 *If, for every $v \in S$, $In_\delta(\mathcal{A}_{\{v\}}, \uparrow(v) - S | \mathcal{A}_{S \cap \uparrow(v)})$, then $\mathcal{A}_S$ is a δ-independence based partial assignment.*

Proof: Since, for every $v \in S$, the condition of theorem 14 holds, then we have that (theorem 14):

$$\forall v \in S. \;\; In_\delta(\mathcal{A}_{\{v\}}, \uparrow^+(v) - S | \mathcal{A}_{S \cap \uparrow(v)})$$

and thus $\mathcal{A}_S$ is δ-independence based by definition 22, Q.E.D.

Appendix C

Notation

We denote variables (or nodes, which we use interchangeably with variables) either as lower case letter near the end of the alphabet (e.g. x, y), or long, all lower-case names (e.g. water-bill-paid). We denote sets of variables (or nodes) by upper-case letters near the end of the alphabet (e.g. V, X).

Script capitals near the beginning of the alphabet (e.g. $\mathcal{A}$, $\mathcal{B}$) denote assignments. Assignments are binary relations, (v, d) such that $v \in V$ and d is a value in the domain of v, (the domain is denoted D_v). Sometimes we refer to an assignment as the set of pairs in the relation, and perform set operations on it.

Traditionally assignments are functions, but that holds only for *consistent* assignments. Assignment $\mathcal{A}$ is consistent iff:

$$\forall (v_1, d_1), (v_2, d_2) \in \mathcal{A} \quad v_1 = v_2 \rightarrow d_1 = d_2 \tag{C.1}$$

that is, each variable is assigned a single value.

Assignment $\mathcal{B}$ is a restricted assignment if it is a subset of some other assignment, but maps a smaller set of nodes. Thus, if the variable-set of $\mathcal{A}$ is V, and the variable set of $\mathcal{B}$ is S, where $S \subseteq V$ and $\mathcal{B} \subseteq \mathcal{A}$, then $\mathcal{B}$ is a restriction of $\mathcal{A}$ to variable-set S. We use the set of variables we are restricting by as a subscript, thus $\mathcal{B} = \mathcal{A}_S$. Naturally, $\mathcal{B}_S$ is equal to $\mathcal{B}$.

An assignment to a binary valued variable is sometimes denoted by the variable's capitalized name to denote an assignment of true (T), and by the capitalized name preceded by a negation sign to denote an assignment of false (F). For example, Water-bill-paid denotes the assignment (water-bill-paid, T), and ¬Water-bill-paid denotes the assignment (water-bill-paid, F). We also denote assignments using an *equal sign*, e.g. $x = T$ is the same as (x, T).

We denote the set of all consistent assignments that are complete w.r.t. a set of variables S by $\mathcal{C}_S$. We denote the set of all consistent non-strictly partial assignments to the set of variables S by $\mathcal{P}_S$. By definition, $\mathcal{C}_S \subseteq \mathcal{P}_S$.

We define generalized assignments as assignments where more than one value may be assigned to a node (a "disjunctive" assignment). The range of such an assignment to a node v is a set $d \in 2^{D_v}$. We designate a set of allowable disjunctions for a node, $M_v \subseteq 2^{D_v}$.

To sum up, the following table provides notation examples and their denotations:

Notation	Denotes
x	Variable or node x
X	Set of variables or nodes
$\mathcal{A}$	An assignment (binary relation)
$\mathcal{A}_S$	Assignment $\mathcal{A}$ restricted to S
$span(\mathcal{A})$	Set of nodes assigned by $\mathcal{A}$
D_v	Domain of variable v
$\mathcal{E}$	The *evidence* assignment
water-bill-paid	A variable or node
Water-bill-paid	water-bill-paid= T
$\mathcal{P}_S$	All partial assignments over variable-set S
$\mathcal{C}_S$	All complete assignments over variable-set S
M_v	Set of allowable domain subsets(disjunctions) for v
$H_i^{v^d}$	Hypercube i assigning v value d
$h_i^{v^d}$	Variable indexed by above hypercube

Bibliography

[Appelt, 1990] Douglas E. Appelt. A theory of abduction based on model preference. In *Proceedings of the AAAI Symposium on Abduction*, 1990.

[Charniak and Goldman, 1988] Eugene Charniak and Robert Goldman. A logic for semantic interpretation. In *Proceedings of the ACL Conference*, 1988.

[Charniak and Husain, 1991] Eugene Charniak and Saadia Husain. A new heuristic for minimal cost proofs. In *Proceedings of AAAI Conference*, 1991.

[Charniak and Shimony, 1990a] Eugene Charniak and Solomon E. Shimony. Probabilistic semantics for cost-based abduction. In *Proceedings of the 8th National Conference on AI*, August 1990.

[Charniak and Shimony, 1990b] Eugene Charniak and Solomon E. Shimony. Probabilistic semantics for rule based systems. Technical Report CS-90-02, Computer Science Department, Brown University, February 1990.

[Cooper and Edward, 1991] Gregory F. Cooper and Herskovits. Edward. A bayesian method for the induction of probabilistic networks from data. Technical Report SMI-91-1, University of Pittsburgh, January 1991.

[Cooper, 1984] Gregory Floyd Cooper. *NESTOR: A Computer-Based Medical Diagnosis Aid that Integrates Causal and Probabilistic Knowledge.* PhD thesis, Stanford University, 1984.

[Cooper, 1990] Gregory F. Cooper. The computational complexity of probabilistic inference using bayesian belief networks. *Artificial Intelligence*, 42 (2-3):393–405, 1990.

[Dean and Kanazawa, 1991] Thomas Dean and Keiji Kanazawa. A model for reasoning about persistence and causation. *Computational Intelligence*, 5 (3):142–150, 1991.

[Dempster, 1968] A. P. Dempster. A generalization of bayesian inference. *Royal Statistical Society*, B 30:205–247, 1968.

[Derthick, 1988] Mark Derthick. *Mundane Reasoning by Parallel Constraint Satisfaction.* PhD thesis, Carnegie Mellon University, 1988. Technical report CMU-CS-88-182.

[Eric J. Horvitz, 1989] Gregory F. Cooper Eric J. Horvitz, H. Jacques Suermondt. Bounded conditioning: Flexible inference for decisions under scarce resources. In *5th Workshop on Uncertainty in AI*, August 1989.

[Feldman and Yakimovsky, 1974] Jerome A. Feldman and Yoram Yakimovsky. Decision theory and artificial intelligence: I. a semantics-based region analyzer. *Artificial Intelligence*, 5:349–371, 1974.

[Garey and Johnson, 1979] M. R. Garey and D. S. Johnson. *Computers and Intractability, A Guide to the Theory of NP-completeness*, page 190. W. H. Freeman and Co., 1979.

[Geeman and Geeman, 1984] Stuart Geeman and Donald Geeman. Stochastic relaxation, gibbs distributions and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.

[Genesereth, 1984] Michael R. Genesereth. The use of design descriptions in automated diagnosis. *Artificial Intelligence*, pages 411–436, 1984.

[Harsanyi, 1985] J. C. Harsanyi. Acceptance of empirical statements: A bayesian theory without cognitive utilities. *Theory and Decision*, 18:1–30, 1985.

[Henrion, 1986] Max Henrion. Propagating uncertainty by logic sampling in bayes' networks. Technical report, Department of Engineering and Public Policy, Carnegie Mellon University, 1986.

[Hobbs and Stickel, 1988] Jerry R. Hobbs and Mark Stickel. Interpretation as abduction. In *Proceedings of the 26th Conference of the ACL*, 1988.

[Kautz and Allen, 1986] Henry A. Kautz and James F. Allen. Generalized plan recognition. In *Proceedings of the Fifth Conference of AAAI*, August 1986.

[Kim and Pearl, 1983] Jin H. Kim and Judea Pearl. A computation model for causal and diagnostic raesoning in inference systems. In *Proceedings of the 6th International Joint Conference on AI*, 1983.

[Lauritzen and Speigelhalter, 1988] S.L. Lauritzen and David J. Speigelhalter. Local computations with probabilities on graphical structures and their applications to expert systems. *Journal of the Royal Statistical Society*, 50:157–224, 1988.

[McDermott, 1987] Drew V. McDermott. Critique of pure reason. *Computational Intelligence*, 3:151–60, November 1987.

[Modestino and Zhang, 1989] J. W. Modestino and J. Zhang. A markov field model-based approach to image interpretation. In *IEEE Computer Vision and Pattern Recognition*, 1989.

[Neapolitan, 1990] Richard E. Neapolitan. *Probabilistic Reasoning in Expert Systems*, chapter 8. John Wiley and Sons, 1990.

[Ng and Mooney, 1990] Hwee Tou Ng and Raymond J. Mooney. On the coherence in abductive explanation. In *Proceedings of the 8th National Conference on AI*, pages 337–342, August 1990.

[Norvig, 1991] Peter Norvig, January 1991. Personal communication.

[Pearl and Verma, 1991] Judea Pearl and T. S. Verma. A theory of inferred causation. In *Knowledge Representation and Reasoning: Proceedings of the second International Conference*, pages 441–452, April 1991.

[Pearl, 1988] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Mateo, CA, 1988.

[Peng and Reggia, 1987] Y. Peng and J. A. Reggia. A probabilistic causal model for diagnostic problem solving (parts 1 and 2). In *IEEE Transactions on Systems, Man and Cybernetics*, pages 146–162 and 395–406, 1987.

[Poole and Provan, 1990] David Poole and Gregory M. Provan. What is an optimal diagnosis? In *Proceedings of the 6th Conference on Uncertainty in AI*, pages 46–53, 1990.

[Santos Jr., 1991a] Eugene Santos Jr. Cost-based abduction and linear constraint satisfaction. Technical Report CS-91-13, Computer Science Department, Brown University, 1991.

[Santos Jr., 1991b] Eugene Santos Jr. Cost-based abduction, linear constraint satisfaction, and alternative explanations. In *Proceedings of the AAAI Workshop on Abduction*, 1991.

[Santos Jr., 1991c] Eugene Santos Jr. A linear constraint satisfaction approach to cost-based abduction. Technical report, Computer Science Department, Brown University, 1991.

[Santos Jr., 1991d] Eugene Santos Jr. On the generation of alternative explanations with implications for belief revision. In *Proceedings of the 7th Conference on Uncertainty in AI*, 1991.

[Selman and Kautz, 1989] Bart Selman and Henry Kautz. The complexity of model-preference default theories. In Reinfrank et. al., editor, *Non-Monotonic Reasoning*, pages 115–130. Springer Verlag, Berlin, 1989.

[Shachter, 1986] R. D. Shachter. Evaluating influence diagrams. *Operations Research*, 34 (6):871–882, 1986.

[Sher, 1990] David B. Sher. Towards a normative theory of scientific evidence - a maximum likelihood solution. In *Proceedings of the 6th Conference on Uncertainty in AI*, pages 509–515, 1990.

[Shimony and Charniak, 1990] Solomon E. Shimony and Eugene Charniak. A new algorithm for finding map assignments to belief networks. In *Proceedings of the 6th Conference on Uncertainty in AI*, 1990.

[Shimony, 1990] Solomon E. Shimony. On irrelevance and partial assignments to belief networks. Technical Report CS-90-14, Computer Science Department, Brown University, 1990.

[Srinivas and Breese, 1989] Sampath Srinivas and Jack Breese. Ideal: Influence diagram evaluation and analysis in lisp, May 1989. Documentation and Users Guide.

[Stickel, 1988] Mark E. Stickel. A prolog-like inference system for computing minimum-cost abductive explanations in natural-language interpretation. Technical Report 451, Artificial Intelligence Center, SRI, September 1988.

[Subramanian, 1989] Devika Subramanian. *A Theory of Justified Reformulations.* PhD thesis, Stanford University, 1989. Technical report STAN-CS-89-1260.