Probabilistic Horn abduction and Bayesian networks

David Poole*
Department of Computer Science,
University of British Columbia,
Vancouver, B.C., Canada V6T 1Z4
poole@cs.ubc.ca
telephone: (604) 822 6254
fax: (604) 822 5485

September 15, 1993

Abstract

This paper presents a simple framework for Horn-clause abduction, with probabilities associated with hypotheses. The framework incorporates assumptions about the rule base and independence assumptions amongst hypotheses. It is shown how any probabilistic knowledge representable in a discrete Bayesian belief network can be represented in this framework. The main contribution is in finding a relationship between logical and probabilistic notions of evidential reasoning. This provides a useful representation language in its own right, providing a compromise between heuristic and epistemic adequacy. It also shows how Bayesian networks can be extended beyond a propositional language. This paper also shows how a language with only (unconditionally) independent hypotheses can represent any probabilistic knowledge, and argues that it is better to invent new hypotheses to explain dependence rather than having to worry about dependence in the language.

*Scholar, Canadian Institute for Advanced Research.

1 Introduction

Probabilistic Horn Abduction [48, 47] is a framework for logic-based abduction that incorporates probabilities with assumptions. This is being used as a framework for diagnosis [48] that incorporates both pure Prolog [30] and Bayesian Networks [39] as special cases. This paper expands on [48, 47] and develops the formal underpinnings of probabilistic Horn abduction, shows the strong relationships to other formalisms and argues that it is a good representation language in its own right. It can be motivated in a number of different ways:

Determining what is in a system from observations (diagnosis and recognition) is an important part of AI. There have been many logic-based proposals as to what is a *diagnosis* [17, 57, 13, 45, 12]. One problem with all of these proposals is that for any diagnostic problem of a reasonable size there are far too many logical possibilities to handle. For example, when considering fault models [14, 45], there is almost always an exponential number of logical possibilities (e.g., each component could be in one of its normal states or in the unknown state). For practical problems, many of the logically possible diagnoses are so unlikely that it is not worth considering them. There is a problem, however, in removing the unlikely possibilities a priori (those with a low prior probability): it may happen that the unlikely occurrence is the actual truth in the world. Analysis of the combinatorial explosions would however tend to suggest that we need to take into account probabilities of the diagnoses [13, 40, 34], and not even generate the unlikely diagnoses (i.e., those with a low posterior probability).

In a different strand of research, Bayesian networks [39], have proven to be a good representation for the sort of probabilistic independence found in many domains. While the independence of Bayesian networks has been expressed in logic (e.g., [3]), there has not been a mapping between logical specifications of knowledge and Bayesian network representations, where the logic is not at the meta-level to the probabilistic knowledge. This paper describes what could be termed as a logic of discrete Bayesian Networks, where the logic expresses the object level knowledge and the independence of Bayesian networks is an emergent property of the representation. In the uncertainty community there has also been a need to extend Bayesian networks to beyond a propositional language [5, 6, 22]. Probabilistic Horn abduction is naturally non-propositional, and provides a natural extension of Bayesian networks to a non-propositional language.

The work presented in this paper should be contrasted with other attempts to combine logic and probability in very powerful languages (e.g., [3]). We are trying to find the simplest language that is useful for our purposes, rather than combine many different features onto one framework. Our goal in this research is to investigate a simple yet powerful logic.

The representation proposed in this paper is interesting in its own right as a compromise between epistemic and heuristic adequacy [33]. It extends pure Prolog in a simple way to include probabilities. While all of the hypotheses are independent, by inventing new hypotheses, we can represent any probabilistic dependency. This simplicity allows us to experiment with a minimalist representation and only extend it when we need to. It is interesting to see how far we can go with a very simple representation language, only adding to it when it fails to do what we it want to do.

1.1 A Motivating Example

Before we present the language and the assumptions behind the representation, we first give an example to show what sorts of things we can represent.

The example is based on the three cascaded inverters of [14]. Figure 1 shows the connections of the inverters.

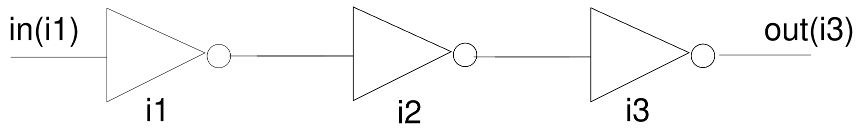

Figure 1: Three cascaded inverters.

The language is an extension of pure Prolog. We can write Prolog-like definite clauses to represent the general knowledge of the domain, and the specific instances known about the particular configuration:

$val(out(G), on, T) \leftarrow ok(G) \wedge val(in(G), \mathit{off}, T).$

$val(out(G), off, T) \leftarrow ok(G) \wedge val(in(G), on, T).$
$val(out(G), V, T) \leftarrow shorted(G) \wedge val(in(G), V, T).$
$val(out(G), off, T) \leftarrow blown(G).$
$val(in(G), V, T) \leftarrow conn(G1, G) \wedge val(out(G1), V, T).$
$conn(i1, i2).$
$conn(i2, i3).$

Here $val(P, V, T)$ means that port P has value V at time T; $in(G)$ is the input port of gate G and $out(G)$ is the output port of gate G. $ok(G)$ means G is working properly; $shorted(G)$ means that G is shorted and acts as a wire; and $blown(G)$ means that G always outputs the value *off*.

The language also has a "disjoint declaration" that defines a set of disjoint and covering hypotheses that have probabilities associated with them:

$disjoint([ok(G) : 0.95, shorted(G) : 0.03, blown(G) : 0.02]).$
$disjoint([val(in(i1), on, T) : 0.5, val(in(i1), off, T) : 0.5]).$

The first gives the prior probabilities of the states of gates. The second gives the prior probabilities of the inputs to the first gate. In our language, different instances of disjoint declarations are probabilistically (unconditionally) independent, and so we have stated that the gates break independently, and that the values of the input to gate $i1$ is independent of the states of the system.

The sorts of things that we can ask, and for which we have given enough information to compute include:

- What is the probability that gate $i2$ is ok given that the input to $i1$ is *off* and the output of $i3$ is *off* at time t_1?

$$P(ok(i2)|val(in(i1), off, t_1) \wedge val(out(i3), off, t_1))$$

  (The answer is 0.76).

- If the input of $i1$ were *on* what is the probability that the output of $i3$ will be *off*?

$$P(val(out(i3), off, t_1)|val(in(i1), on, t_1))$$

  (The answer is 0.899).

- What was the probability that the input to $i1$ was *on* at time t_2 given that the output to $i2$ was off at time t_2 and given that the output of $i3$ was off and the input of $i1$ was off at time t_1?

  $$P(val(in(i1), on, t_2) | val(out(i2), off, t_2) \wedge val(out(i3), off, t_1) \wedge val(in(i1), off, t_1))$$

  (the answer is 0.55).

Each of these answers is computed in terms of explanations, namely arguments with premises from the possible hypotheses. We assume independence amongst the hypotheses so that the prior probability of an explanation is obtained by multiplying the probabilities of the hypotheses in the explanation.

For example, the explanations of the observation $val(in(i1), off, t_1) \wedge val(out(i3), off, t_1)$, together with their prior probability are:

Explanation: $[val(in(i1), off, t_1), ok(i3), ok(i2), shorted(i1)]$
Prior = 0.01354

Explanation: $[val(in(i1), off, t_1), ok(i3), shorted(i2), ok(i1)]$
Prior = 0.01354

Explanation: $[val(in(i1), off, t_1), shorted(i3), ok(i2), ok(i1)]$
Prior = 0.01354

Explanation: $[val(in(i1), off, t_1), blown(i3)]$
Prior = 0.01000

Explanation: $[val(in(i1), off, t_1), ok(i3), ok(i2), blown(i1)]$
Prior = 0.009025

Explanation: $[val(in(i1), off, t_1), shorted(i3), blown(i2)]$
Prior = 0.0003000

Explanation: $[val(in(i1), off, t_1), shorted(i3), shorted(i2), shorted(i1)]$
Prior = 0.00001350

Explanation: $[val(in(i1), off, t_1), shorted(i3), shorted(i2), blown(i1)]$
Prior = 0.000009000

By the way the knowledge base was constructed, these explanations are disjoint and covering, and so we can compute the prior probability of

$$val(in(i1), \mathit{off}, t_1) \wedge val(out(i3), \mathit{off}, t_1))$$

by summing the probabilities of these explanations, which here is 0.05996.

2 Probabilistic Horn Abduction

In this section we develop the language, the assumptions behind it and a "semantics" of the language in terms of abduction. A more formal semantics is provided in Appendix A. The language is designed to be usable and does not allow us to state what cannot be computed in a straight forward manner.

The initial language is translated into an abductive framework with a number of assumptions about the knowledge base. The appendix gives a more formal model-theoretic semantics and demonstrates the equivalence between the two.

2.1 The Probabilistic Horn abduction language

Our language uses the Prolog conventions [64]:

Definition 2.1 A **term** is either a variable (starting with an upper case letter), a constant (starting with a lower case letter) or is of the form $f(t_1, \cdots, t_n)$ where f is a function symbol (starting with a lower case letter) and each t_i is a term. An **atomic symbol** (atom) is of the form p or $p(t_1, \cdots, t_n)$ where p is a predicate symbol (starting with a lower case letter) and each t_i is a term.

Definition 2.2 A **definite clause** is of the form: a. or $a \leftarrow a_1 \wedge ... \wedge a_n$. where a and each a_i are atomic symbols.

Definition 2.3 A **disjoint declaration** is of the form

$$disjoint([h_1 : p_1, \cdots, h_n : p_n]).$$

where the h_i are atoms, and the p_i are real numbers $0 \leq p_i \leq 1$ such that $p_1 + \cdots + p_n = 1$. Any variable appearing in one h_i must appear in all of the h_j (i.e., the h_i share the same variables). The h_i will be referred to as **hypotheses** or **assumables**.

Definition 2.4 A **probabilistic Horn abduction theory** (which will be referred to as a "theory") is a collection of definite clauses and disjoint declarations such that if a ground atom h is an instance of a hypothesis in one disjoint declaration, then it is not an instance of another hypothesis in any of the disjoint declarations.

Given theory T, we define

F_T the **facts**, is the set of definite clauses in T together with the clauses of the form

$$false \leftarrow h_i \wedge h_j$$

where h_i and h_j both appear in the same disjoint declaration in T, and $i \neq j$. Let F'_T be the set of ground instances of elements of F_T.

H_T the **hypotheses**, the set of h_i that appears in some disjoint declaration in T. Let H'_T be the set of ground instances of elements of H_T.

P_T is a function $H'_T \mapsto [0,1]$. $P(h'_i) = p_i$ where h'_i is a ground instance of hypothesis h_i, and $h_i : p_i$ is in a disjoint declaration in T. $P(h'_i)$ will be the prior probability of h'_i.

Where T is understood from context, we omit the subscript.

The disjoint declarations allow a very restricted form of integrity constraints [25]. It allow binary integrity constraints (the conjunction of two hypotheses is false) such that the ground instances of hypotheses form mutually exclusive and covering groupings that correspond to random variables.

A theory will define a set of **represented atoms** that are a subset of the atoms of T. The represented atoms will often be not listed explicitly, but will be left implicit (they will typically be instances of hypotheses and heads of clauses). The represented atoms are those about which the theory can answer questions. Questions about atoms not in the represented atoms will be beyond the scope of the theory. The theory is not expected to be able to answer queries outside of its scope.

2.2 Abduction

We first give the language an abductive characterisation, using the normal definition of the definite clauses. This is used to make explicit our assumptions and to build the theory in a natural manner. In Appendix A, we give a model theoretic characterisation that incorporates our assumptions, and show the equivalence of the formulations.

The formulation of abduction used is a simplified form [18] of Theorist [51, 43]. It is simplified in being restricted to Horn clauses. This can also be seen as a generalisation of an ATMS (with predefined nogoods) [59] to be non-propositional[1].

An abductive scheme is a pair $\langle F, H \rangle$ where

F is a set of Horn clauses. Variables in F are implicitly universally quantified. Let F' be the set of ground instances of elements of F.

H is a set of (possibly open) atoms, called the "assumables" or the "possible hypotheses". Let H' be the set of ground instances of elements of H.

Definition 2.5 [51, 42] If g is a closed formula, an **explanation** of g from $\langle F, H \rangle$ is a set D of elements of H' such that

- $F \cup D \models g$ and
- $F \cup D \not\models false$.

The first condition says that D is sufficient to imply g, and the second says that D is possible.

Definition 2.6 A **minimal explanation** of g is an explanation of g such that no strict subset is an explanation of g.

2.3 Assumptions about the rule base

In order to be able to simply interpret our rules probabilistically we make some assumptions about the rules and some probabilistic independence assumptions about the hypotheses.

The first assumption is syntactic, about the relationship between hypotheses and rules:

Assumption 2.7 There are no rules in F whose head unifies with a member of H.

[1] A main difference is in the philosophy of use. We assume that the Horn clauses are representing the object level knowledge, rather than, as in an ATMS, acting as a back end of a problem solver [11].

This does not seem to be a very severe restriction, in practice. It says that we do not want rules to imply a hypothesis. Presumably, if we had rules for h, the only reason that we would want to make h a hypothesis is if it was possible that h just happens to be true (without any other "cause"). In this case we can replace h in H by the hypothesis $h\_happens\_to\_be\_true$ and add the rule

$$h \leftarrow h\_happens\_to\_be\_true$$

Assumption 2.8 (acyclicity [1]) If F' is the set of ground instances of elements of F, it is possible to assign a natural number to every ground atom such that for every rule in F' the atoms in the body of the rule are strictly less than the atom in the head.

This assumption is described as natural by Apt and Bezem [1]. It is a generalisation of the hierarchical constraint of Clark [8]. It implies that there are no infinite chains when backchaining from any ground goal. This does not restrict recursion, but does mean that all recursion must be well founded.

These assumptions are made implicitly in [43], are explicit in [9] (who make the hierarchical constraint rather than the acyclic constraint), but are relaxed in [24].

When using abduction we often assume that the explanations are covering. This can be a valid assumption if we have anticipated all eventualities, and the observations are within the domain of the expected observations (usually if this assumption is violated there are no explanations). This is also supported by recent attempts at a completion semantics for abduction [43, 9, 24]. The results show how abduction can be considered as deduction on the "closure" of the knowledge base that includes statements that the given causes are the only causes. We make this assumption explicit here:

Assumption 2.9 The rules in F' for every ground non-hypothesis represented atom are covering.

That is, if the rules for a in F' are

$$\begin{array}{c} a \leftarrow B_1 \\ a \leftarrow B_2 \\ \vdots \\ a \leftarrow B_m \end{array}$$

if a is true, one of the B_i is true. The *completion* of a is

$$a \equiv B_1 \vee \cdots \vee B_n$$

Thus the *covering* assumption 2.9 says that Clark's completion [8] is valid for every non-assumable.

If the rules for a are not covering, we create a new hypothesis $a\_is\_true\_for\_some\_other\_reason$ and add the rule

$$a \leftarrow a\_is\_true\_for\_some\_other\_reason.$$

Lemma 2.10 [9] *Under assumptions 2.7, 2.8 and 2.9, if $expl(g,T)$ is the set of all minimal explanations of g from $\langle F_T, H_T\rangle$, and $comp(T)$ is F'_T augmented with the completion of every ground instance of every non-assumable (including Clark's equational theory [8]), then*

$$comp(T) \quad \models \quad \left(g \equiv \bigvee_{e_i \in expl(g,T)} e_i \right)$$

The next assumption has to do with the status of explanations

Assumption 2.11 The bodies of the rules in F' for an atom are mutually exclusive.

Given the above rules for a this means that $B_i \wedge B_j$ is always false for each $i \neq j$. To ensure that this assumption holds we can add extra conditions to the rules. See section 5.

Note that, whereas assumptions 2.7 and 2.8 are syntactic assumptions about the theory that can be automatically checked, assumptions 2.9 and 2.11 are statements about the world, and not about the knowledge base[2]. We do not require $F_T \models \neg(B_i \wedge B_j)$. The language is not powerful enough to state such constraints. For example, it may be the case that the value

[2] There is, however, a syntactic condition that can be used to check whether assumption 2.11 has been violated. This is when we can derive the bodies of two rules for an atom from a set of assumptions that are not inconsistent. For example, if $\{a, na\}$ and $\{b, nb\}$ are disjoint and covering sets of assumptions (and so a and b are independent by assumption 2.14), and we have rules $\{c \leftarrow a, c \leftarrow b\}$, then we know the *disjoint bodies* assumption 2.11 has been violated, as a and b cannot be both exclusive and independent.

on some wire is functional (the value cannot be both on and off at the same time). We cannot state this in our language. This is a deliberate design decision to make the language as simple as possible. See section 2.5 for the rationale behind the design decisions.

N.B. The assumptions here are not intended to be enforced by the system. It is up to the user (or some other system) to enforce these constraints. The status of these assumptions is that if you follow the constraints, then we make claims about the interpretability of the system. If these assumptions are violated then we make no guarantees.

Lemma 2.12 *Under assumptions 2.7 and 2.11, minimal explanations of atoms or of conjunctions of atoms are mutually exclusive (no two explanations can both be true).*

Lemma 2.12 does not hold for arbitrary formulae. In particular, the minimal explanations of a disjunction are not necessarily disjoint.

2.4 Probabilities

Associated with each possible hypothesis is a prior probability. We use this prior probability to compute arbitrary probabilities.

The following is a corollary of lemmata 2.10 and 2.12

Lemma 2.13 *Under assumptions 2.7, 2.8, 2.9 and 2.11, if $expl(g,T)$ is the set of minimal explanations of conjunction of atoms g from probabilistic Horn abduction theory T:*

$$\begin{aligned} P(g) &= P\Big(\bigvee_{e_i \in expl(g,T)} e_i\Big) \\ &= \sum_{e_i \in expl(g,T)} P(e_i) \end{aligned}$$

Thus to compute the prior probability of any g we sum the probabilities of the explanations of g.

To compute arbitrary conditional probabilities, we use the definition of conditional probability:

$$P(\alpha|\beta) = \frac{P(\alpha \wedge \beta)}{P(\beta)}$$

To find arbitrary conditional probabilities $P(\alpha|\beta)$, we find $P(\beta)$, which is the sum of the explanations of β. To compute the probability $P(\alpha \wedge \beta)$, we sum over the explanations of $\alpha \wedge \beta$. Note that the explanations of $\alpha \wedge \beta$ are also explanations of β. We can find the explanations of $\alpha \wedge \beta$ by explaining α from the explanations of β. Thus arbitrary conditional probabilities can be computed from summing the prior probabilities of explanations.

It remains only to compute the prior probability of an explanation D of g. We assume that logical dependencies impose the only statistical dependencies on the hypotheses. In particular we assume:

Assumption 2.14 Ground instances of hypotheses that are consistent (with F_T) are probabilistically independent.

N.B. We mean that the hypotheses are unconditionally independent. They may become dependent given observations (i.e., when conditioning on observations).

Example 2.15 If we have the disjoint declarations

$$disjoint([p(X) : 0.4, q(X) : 0.6]).$$
$$disjoint([r(Y) : 0.01, s(Y) : 0.99]).$$

then $p(t)$ is independent of $r(u)$ for all ground terms t and u. $p(t)$ is independent of $q(u)$ for all different ground terms u and t. $p(t)$ is independent of $p(u)$ for all different ground terms u and t.

Thus $p(a)$ is dependent on $q(a)$ (they are exclusive), but $p(a)$ is independent of $q(b)$. $p(a)$ is also independent of $p(b)$.

Under assumption 2.14, if $D = \{h_1, ..., h_n\}$ is a minimal explanation, then

$$P(h_1 \wedge ... \wedge h_n) \quad = \quad \prod_{1=1}^{n} P(h_i)$$

To compute the prior of the minimal explanation we multiply the priors of the hypotheses.

Assumption 2.14 implies the unique names assumption if there are parametrized hypotheses. If $h(X)$ is a parametrized hypothesis with each instance having probability p, and t_1 and t_2 are different ground terms, assumption 2.14

implies $P(h(t_1) \wedge h(t_2)) = p^2$. If $t_1 = t_2$, then $h(t_1) \wedge h(t_2) \equiv h(t_1)$, but $P(h(t_1)) = p$, a contradiction to the fact that probabilities are a measure over propositions, and that logically equivalent terms should have the same probability [39]. Thus, we are assuming that $t_1 \neq t_2$ for different terms t_1 and t_2. This assumption is the unique names assumption [56]. Note that it is for the probabilistic calculation that we are making the unique names assumption.

Appendix A gives the formal semantics for probabilistic Horn abduction, and justifies, in another way, the results of this section.

2.5 Rationale for the design choices

The language presented so far is quite weak in some respects. In this section we discuss why the language is as it is. The general theme is that the language is a simple extension to pure Prolog that lets us consistently interpret the numbers on hypotheses as probabilities. We have disallowed anything that will make this interpretation difficult. For example, we have not allowed the logic to be expressive enough to be able to prove that there is a dependency amongst the hypotheses beyond the disjointness of our random variables.

This simplicity makes the language semantically transparent, and allows for simple implementations. It is still powerful enough to express many of the causal and probabilistic interactions that we want to express. This work should be seen as an exercise in minimalist representations — we try to understand the limitations of very restricted languages and only add extra power if we can show we cannot do what we want with the tools available to us.

2.5.1 Language for specifying random variables

The first thing to notice is that we allow a very restricted and stylised form of integrity constraints to be specified by the use of the disjoint declaration. This is in contrast to earlier versions [48, 47] where we allowed arbitrary integrity constraints. The more expressive language in [48, 47] allows us to represent what the current version allows, however it lets us represent what we cannot interpret probabilistically, and makes the proof procedures more complicated without providing visible advantage.

For example, if $\{a_0, a_1, a_2\}$ and $\{b, nb\}$ each form a disjoint and covering sets of propositions (random variables), then we cannot treat these as independent if we can state $false \leftarrow a_0 \wedge b$. All we know here is that the variables are not independent — there are only ad hoc methods to allow us to provide the joint distribution. The current formulation does not give the power to state such constraints. The logical formulae provide no constraints on the hypotheses beyond the disjointedness of the values in the disjoint declaration.

A second, but related, problem with general integrity constraints[3] has to do with making implicit assumptions by the use of integrity constraints. For example, if we have $false \leftarrow a \wedge b$, when we are using a, we are implicitly assuming the negation of b and should pay the cost (in terms of making any explanation containing a less likely). This occurs at an extreme level when we have $\{b, nb\}$ disjoint hypotheses whose probability sums to 1, and have $false \leftarrow a \wedge b$ and $false \leftarrow a \wedge nb$. Here it should follow that a cannot occur (or at least with probability zero) and should be pruned from other explanations.

We also made sure that all atoms in disjoint declarations share the same variables. To see the problem with not requiring this, consider the (illegal) declaration

$$disjoint([p : 0.7, q(X) : 0.3]).$$

Given this declaration, p would be disjoint and covering with $q(a)$ and p disjoint and covering with $q(b)$. This would then place a dependence between $q(a)$ and $q(b)$. Given that the hypotheses are disjoint and covering, if $q(a)$ is true, then p is false, and so $q(b)$ is true. Similarly if $q(a)$ is false, p is true and $q(b)$ is false. Thus all instances of $q(X)$ would always have the same truth value. Then we may as well remove the variable (as the truth does not depend on the value of the variable). We have restricted the variables in disjoint assertions in order to avoid such tricky interactions that the user of the system may not be aware of, and so that an implementation does not need to look for them.

The language is also not powerful enough to state the constraints on the legal input. When we wanted the bodies of the rules to be disjoint, we did not require that we could prove the disjointedness of the bodies. We required that they be disjoint in the domain under consideration. Similarly, we require the rules to be covering, but cannot check this. This should not be seen as a

[3]This was pointed out to me by Mark Wallace of ECRC.

defect in this language — there will always be true things that the language is too weak to state (e.g., the finiteness of integers cannot be stated in the first order predicate calculus). This restriction was a conscious decision to allow us to build efficient implementations, and to avoid difficult to interpret statements as described above.

2.5.2 Assumptions concerning the knowledge base

Assumption 2.7 that says that hypotheses cannot form the head of rules is important to ensure that we can treat the hypotheses as independent. If we could prove some hypothesis (based on other assumptions) it would not be consistent that the hypotheses are independent. It is also important to ensure minimal explanations are disjoint. If we have a and b as hypotheses, and have $b \leftarrow a$ as well as $c \leftarrow b$. There are two minimal explanations of c, namely $\{a\}$ and $\{b\}$. These cannot be disjoint as one implies the other (and are equivalent under the Covering Assumption 2.9).

To see the importance of the acyclicity assumption 2.8 consider the facts $F = \{a \leftarrow d \wedge b, a \leftarrow c, d \leftarrow a\}$, and possible hypotheses $H = \{b, c\}$. Assumption 2.8 is violated in this example. Here there is one explanation for a, namely c, but under assumption 2.9, we can prove $a \Rightarrow b \vee c$, and not $a \Rightarrow c$. This violates the conclusion of Lemma 2.10.

Assumptions 2.9 and 2.11 are needed so that we can have disjoint and covering hypotheses. This means that we just sum the probabilities of the hypotheses.

2.5.3 Negation

The language provided does not have explicit negation. It does, however, give us an implicit negation.

For example, suppose we have the theory

$$\begin{array}{l} a \leftarrow h_1. \\ b \leftarrow h_2. \\ disjoint([h_1 : p_1, h_2 : 1 - p_1]). \end{array}$$

Under the *covering* assumption 2.9, we have $a \equiv h_1$ and $b \equiv h_2$. The disjoint declaration essentially tells us that $h_1 \equiv \neg h_2$. Thus we have $a \equiv \neg b$.

Thus, although we cannot state negation, we can interpret one atom as being the negation of another. The reason that we don't want to have explicit negation is that this would allow the logic to imply a dependence amongst variables that violates the *independence* assumption 2.14. Once this occurs, it is more difficult to interpret the probabilities. For example, $a \leftarrow h_1$ and $\neg a \leftarrow h_3$ places a dependence on h_1 and h_3.

For each atom we can create its negation. For each atom a, $\overline{a}$ is another atom which we interpret as the negation of a. Syntactically, $\overline{a}$ is just another atom.

If we have a disjoint declaration

$$disjoint([h_1 : p_1, h_2 : p_2, \cdots, h_n : p_n]).$$

we can create the negation of any hypothesis, say h_1, by using

$$\begin{array}{rcl} \overline{h_1} & \leftarrow & h_2 \\ & \vdots & \\ \overline{h_1} & \leftarrow & h_n \end{array}$$

The other hypotheses can be negated analogously.

If we have rules for a

$$\begin{array}{rcl} a & \leftarrow & b_{11} \wedge \cdots \wedge b_{1n_1}. \\ & \vdots & \\ a & \leftarrow & b_{k1} \wedge \cdots \wedge b_{kn_k}. \end{array}$$

we can define atom r_i to correspond to the i-th body, and have rules $a \leftarrow r_1$ through $a \leftarrow r_k$. There are also k rules of the form $r_i \leftarrow b_{i1} \wedge \cdots \wedge b_{in_i}$.

We can define the atom $\overline{a}$ that is the negative of a as

$$\overline{a} \leftarrow \overline{r_1} \wedge \cdots \wedge \overline{r_k}$$

$\overline{r_i}$ is defined as

$$\begin{array}{rcl} \overline{r_i} & \leftarrow & \overline{b_{i1}}. \\ \overline{r_i} & \leftarrow & b_{i1} \wedge \overline{b_{i2}}. \\ & \vdots & \\ \overline{r_i} & \leftarrow & b_{i1} \wedge \cdots \wedge b_{i(n_i-1)} \wedge \overline{b_{in_i}}. \end{array}$$

These definitions will be well grounded by the acyclicity of the rule base. We have added b_{ij} to all of the rules after the j-th to ensure the rules are disjoint (see Section 5.1).

Note that just because some atom a is represented, it does not mean that the negation of a need be represented. There are however, some cases for which it is useful to create the negation of atoms (see Section 5.1).

This negation is an extension to a simple form of the negation of Barbuti et. al. [4], which is used for negation as failure.

3 Representing Bayesian networks

In this section we give the relationship between Bayesian networks and our probabilistic Horn abduction. We show how any probabilistic knowledge that can be represented in a discrete (and finite) Bayesian network, can be represented in our formalism. We also demonstrate the alternate, namely that any propositional probabilistic Horn abduction theory is equivalent to a Bayesian network.

A *Bayesian network* [39] is a directed acyclic network where the nodes represent random variables, and the arcs represent a directly influencing relation. We will use the term "RV" to mean random variable so as to avoid confusion with the Prolog-style variable. If there is arc from RV b to RV a then b is said to be a parent of a.

Suppose we have a discrete[4] Bayesian network with random variables $a_1, ..., a_n$, such that random variable a_i can have values $v_{i,1}, ..., v_{i,r_i}$. We represent random variable a_i having value $v_{i,j}$ as the proposition $a_i(v_{i,j})$.

Suppose RV a_i has parents $\Pi_{a_i} = \{a_{i_1}, \cdots, a_{i_{n_i}}\}$ in a Bayesian network. The independence assumption embedded in a Bayesian Network [39] is that a RV is independent of its non-descendents given its parents. That is,

$$P(a_i|\Pi_{a_i} \wedge v) = P(a_i|\Pi_{a_i})$$

where v is a RV (or conjunction of RVs) such that a_i is not an ancestor of v (or any conjunct in v).

[4] I.e., all the random variables have a discrete and finite set of values. Probabilistic Horn abduction cannot handle random variables with infinite domains. General Bayesian networks have no such restriction.

The formal definition of a Bayesian network is often given in terms of the joint distribution of all of the RVs:

$$P(a_1, \cdots, a_n) = \prod_{i=1}^{n} P(a_i|\Pi_{a_i})$$

A discrete Bayesian network is represented by Probabilistic Horn abduction rules that relates a RV with its parents:

$$a_i(V) \leftarrow a_{i_1}(V_1) \wedge ... \wedge a_{i_{n_i}}(V_{n_i}) \wedge c\_a_i(V, V_1, ..., V_{n_i})$$

The intended interpretation of $c\_a_i(V, V_1, ..., V_{n_i})$ is that a_i has value V because a_{i_1} has value V_1,..., and $a_{i_{n_i}}$ has value V_{n_i}.

Associated with the Bayesian network is a conditional probability table which gives the conditional probabilities of the values of a depending on the values of $\Pi_{a_i} = \{a_{i_1}, \cdots, a_{i_{n_i}}\}$. This will consist of probabilities of the form

$$P(a_i = v_{i,j}|a_{i_1} = v_1, \cdots, a_{i_{n_i}} = v_{n_i}) = p_j$$

such that

$$\forall v_1, \cdots, v_{n_i} \left(\sum_{j=1}^{r_i} P(a_i = v_{i,j}|a_{i_1} = v_1, \cdots, a_{i_{n_i}} = v_{n_i}) \right) = 1$$

where $v_{i,1}, ..., v_{i,r_i}$ are the possible values for RV a_i This is translated into assertions

$$disjoint([c\_a(v_{i,1}, v_1, v_2, ..., v_{n_i}) : p_1, \cdots, c\_a(v_{i,r_i}, v_1, v_2, ..., v_{n_i}) : p_{r_i}])$$

Example 3.1 Consider a representation of the Bayesian network of Figure 3.1, with the following conditional probability distributions:

$$\begin{aligned}
P(fire) &= 0.01 \\
P(smoke|fire) &= 0.9 \\
P(smoke|\neg fire) &= 0.01 \\
P(tampering) &= 0.02 \\
P(alarm|fire \wedge tampering) &= 0.5
\end{aligned}$$

$$\begin{aligned}
P(alarm|fire \wedge \neg tampering) &= 0.99\\
P(alarm|\neg fire \wedge tampering) &= 0.85\\
P(alarm|\neg fire \wedge \neg tampering) &= 0.0001\\
P(leaving|alarm) &= 0.88\\
P(leaving|\neg alarm) &= 0.001\\
P(\text{report}|leaving) &= 0.75\\
P(\text{report}|\neg leaving) &= 0.01
\end{aligned}$$

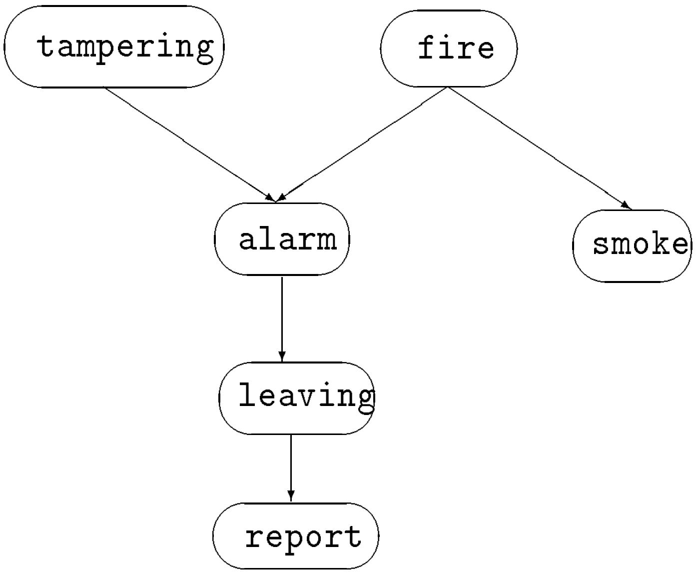

Figure 2: A Bayesian network for a smoking alarm.

The following is a probabilistic Horn abduction representation of this Bayesian network:

$disjoint([fire(yes) : 0.01, fire(no) : 0.99]).$
$smoke(Sm) \leftarrow fire(Fi) \wedge c\_smoke(Sm, Fi).$
$disjoint([c\_smoke(yes, yes) : 0.9, c\_smoke(no, yes) : 0.1]).$
$disjoint([c\_smoke(yes, no) : 0.01, c\_smoke(no, no) : 0.99]).$

$disjoint([tampering(yes) : 0.02, tampering(no) : 0.98]).$
$alarm(Al) \leftarrow fire(Fi) \wedge tampering(Ta) \wedge c\_alarm(Al, Fi, Ta).$
$disjoint([c\_alarm(yes, yes, yes) : 0.50, c\_alarm(no, yes, yes) : 0.50]).$
$disjoint([c\_alarm(yes, yes, no) : 0.99, c\_alarm(no, yes, no) : 0.01]).$
$disjoint([c\_alarm(yes, no, yes) : 0.85, c\_alarm(no, no, yes) : 0.15]).$
$disjoint([c\_alarm(yes, no, no) : 0.0001, c\_alarm(no, no, no) : 0.9999]).$
$leaving(Le) \leftarrow alarm(Al) \wedge c\_leaving(Le, Al).$
$disjoint([c\_leaving(yes, yes) : 0.88, c\_leaving(no, yes) : 0.12]).$
$disjoint([c\_leaving(yes, no) : 0.001, c\_leaving(no, no) : 0.999]).$
$report(Le) \leftarrow leaving(Al) \wedge c\_report(Le, Al).$
$disjoint([c\_report(yes, yes) : 0.75, c\_report(no, yes) : 0.25]).$
$disjoint([c\_report(yes, no) : 0.01, c\_report(no, no) : 0.99]).$

Note that here instead of creating c_fire and making it equivalent to $fire$, we just made $fire$ a hypothesis.

3.1 Equivalence Results

The basic equivalence result is the equivalence between joint distributions and explanations of the RVs having particular values:

Let P be the probability function sanctioned by a Bayesian network. Let T be the corresponding probabilistic Horn abduction theory. Let P_T be the probability function defined by the translation of a Bayesian network into the probabilistic Horn abduction framework. The aim is to show that P and P_T are the same.

The definition of the Bayesian network distribution is using the joint probability for all values in the network. This corresponds to a conjunction of values for all RVs.

Lemma 3.2 Suppose $a_1, \cdots, a_n$ are all of the RVs in a Bayesian network, with T as the corresponding probabilistic Horn abduction theory, then

$$P(a_1 = v_1 \wedge \cdots \wedge a_n = v_n) \quad = \quad P_T(a_1(v_1) \wedge \cdots \wedge a_n(v_n))$$

Proof: By definition of a Bayesian network:

$$P(a_1 = v_1 \wedge \cdots \wedge a_n = v_n) = \prod_{i=1}^{n} P(a_i = v_i | a_{i_1} = v_{i_1}, \cdots, a_{i_{n_i}} = v_{i_{n_i}})$$

By definition of c_a_i in the translation

$$P(a_i = v_i | a_{i_1} = v_{i_1}, \cdots, a_{i_{n_i}} = v_{i_{n_i}}) = P_T(c\_a_i(v_i, v_{i_1}, \cdots, v_{i_{n_i}}))$$

There is only one explanation of $a_1(v_1) \wedge \cdots \wedge a_n(v_n)$, namely $\{c\_a_i(v_i, v_{i_1}, \cdots, v_{i_{n_i}})) : i = 1..n\}$. Thus we have

$$\begin{aligned} P(a_1 = v_1 \wedge \cdots \wedge a_n = v_n) &= \prod P(a_i = v_i | a_{i_1} = v_{i_1}, \cdots, a_{i_{n_i}} = v_{i_{n_i}}) \\ &= \prod P_T(c\_a_i(v_i, v_{i_1}, \cdots, v_{i_{n_i}})) \\ &= P_T(a_1(v_1) \wedge \cdots \wedge a_n(v_n)) \end{aligned}$$

□

The equivalence between P and P_T now follows directly. These two measures agree on all of the combinations of values for all of the RVs, they both obey the probability axioms (Theorem A.13 shows P_T obeys the probability axioms), and so they agree on all formulae. Thus we have the following theorem:

Theorem 3.3 *If H is a set of assignments to random variables in a Bayesian Network, and H^* is the analogous propositions to H in the corresponding probabilistic Horn abduction theory T, then*

$$P(H) = P_T(H^*).$$

This should not be too surprising as the set of explanations of any formula correspond to an assignment of values for the ancestors in the Bayesian network. We can compute the values of any hypothesis by summing over the values of the ancestors of the hypothesis.

3.2 Propositional Abduction in Bayesian Networks

The preceding section showed how any Bayesian network can be represented directly in terms of (propositional) probabilistic Horn abduction.

The opposite is also true. Every propositional probabilistic Horn abduction theory corresponds directly to a Bayesian network. Here we give the mapping.

Each disjoint declaration maps to a random variable. These form roots (they have no ancestors) of the Bayesian network.

Every atom defined by rules also corresponds to a random variable.

If we have rules for a

$$\begin{array}{rcl} a & \leftarrow & b_{11} \wedge \cdots b_{1n_1}. \\ & \vdots & \\ a & \leftarrow & b_{k1} \wedge \cdots b_{kn_k}. \end{array}$$

we can define atom r_i to correspond to the i-th body, and have rules $a \leftarrow r_1$. through $a \leftarrow r_k$ as well as k rules of the form $r_i \leftarrow b_{i1} \wedge \cdots b_{in_i}$.

We also create nodes for the r_i (we don't have to but it makes the conditional probability tables simpler).

We make arcs between going from the b_{ij} to r_i, and give the conditional probability table for a conjunction. We make arcs going from the r_i to a, giving the conditional probability table for a disjunction.

When we do this mapping and then use the translation of the previous section to get back from the Bayesian network to a probabilistic Horn abduction theory, we get essentially the same probabilistic Horn abduction theory. This new theory has many new atoms (corresponding to negations, which may not have existed in the original theory, but must exist in the Bayesian network) which can be ignored. There also will be many more disjoint declarations, but these will correspond to extreme distributions (one value has probability 1), and can also be ignored (or partially evaluated away).

4 Discussion

4.1 Independence and dependence

It may seem at first that only allowing independent hypotheses places a restriction on what we can represent. People claim that there are dependencies amongst hypotheses in the world. The claim in this paper is that the world can be represented so that all of the hypotheses are independent. This apparent conflict can be resolved by noticing that the world does not determine what the hypotheses are. Just as, when defining n-dimensional Euclidean spaces, we can define the space with non-orthogonal axes, we can also define the space with orthogonal axes. When we do so everything becomes simpler. Just because we can axiomatise a world using dependent hypotheses does not mean that we cannot define the world using independent hypotheses.

The justification for this claim is based on noticing that Bayesian networks (that can represent arbitrary probabilistic interaction) can be represented in our framework that uses only independent hypotheses. Note however that, as there can be an exponential number of independent values given n probabilistic hypotheses, we may have to create an exponential number of independent hypotheses in the probabilistic Horn abduction framework. We are not claiming that we are getting something for nothing.

Note that others have also noticed the universality of just having independent hypotheses. For example, consider Reichenbach's *principle of the common cause*:

> "If coincidences of two events A and B occur more frequently than their independent occurrence, ... then there exists a common cause for these events ..." [55, p. 163].

When there is a dependency amongst random variables, we invent a hypothesis to explain that dependence. Thus the assumption of independence, while it gives a restriction on the knowledge bases that are legal, really gives no restriction on the domains that can be represented.

While we have the ability invent hypotheses, we don't need to consider non-independent hypotheses. I would argue that it is much simpler and more natural to invent new hypotheses to explain dependence rather than having to worry about dependence in the language.

4.2 Abduction and Prediction

When computing the prior probability of a hypothesis, we find the explanations for that hypothesis. This corresponds to the use of "abduction" [44].

If we consider conditional probability, as we normally do, we have

$$\begin{aligned} P(\alpha|\beta) &= \frac{P(\alpha \wedge \beta)}{P(\beta)} \\ &= \frac{\sum_{e_i \in expl(\alpha\wedge\beta,T)} P(e_i)}{\sum_{e_i \in expl(\beta,T)} P(e_i)} \end{aligned}$$

We can generate $expl(\alpha \wedge \beta, T)$ by explaining α from the elements of $expl(\beta, T)$. This corresponds to the combination of abducing to causes and default reasoning to predictions from these causes [44, 46, 62]. The results of this paper, give extra evidence that this forms the "right" characterisation of causal reasoning. Abducing the causes and then assumption-based reasoning from causes to predicting what should follow, is the common feature of both Bayesian networks (see also, for example, Shachter and Heckerman [61]) and recent assumption-based logical schemes [44, 46, 62].

There is another close similarity between the abductive approaches and the network propagation scheme of Pearl [39]. Finding the explanations of some g conceptually corresponds to working "up" the Bayesian network from g. Given evidence β, we first find $expl(\beta, T)$. This conceptually involves searching up (finding ancestors) the tree from β. The next step involves finding $expl(\alpha \wedge \beta, T)$. This can be obtained from explaining α from the explanations of β. If we want to compute this for all α, we can do this by working down (finding descendents) the tree from the explanations of β. This 2-phase approach is analogous to Pearl's network propagation scheme, with the initial moving up the tree corresponding to λ messages, and the second phase of moving down the tree from the explanations corresponds to the π messages of Pearl [39]. Given this analogy, it is also easy to see why the upward λ messages result in both λ and π messages (as we need to carry out both phases of the computation of the conditional probability), while the π messages only result in other π messages (we are in the second phase of computing the conditional probability, namely finding the explanations from those explanations of the observations). It is not clear, however, how far this analogy can be pushed.

4.3 Causation

There have been problems associated with logical formulations of causation [38]. There have been claims that Bayesian networks provide the right independencies for causation [39]. This paper provides evidence that abducing to causes and making assumptions as to what to predict from those assumptions [44, 46] is the right logical analogue of the independence in Bayesian networks (as described in section 4.2).

One of the problems in causal reasoning that Bayesian networks overcome [39] is in preventing reasoning such as "if c_1 is a cause for a and c_2 is a cause for $\neg a$, then from c_1 we can infer $\neg c_2$". This is the problem that occurs, for example, in the Yale shooting problem [19]. Our embedding says that this does not occur in Bayesian networks as c_1 and c_2 must already be known to be disjoint.

Figure 3 gives a Bayesian network for the Yale shooting problem[5]

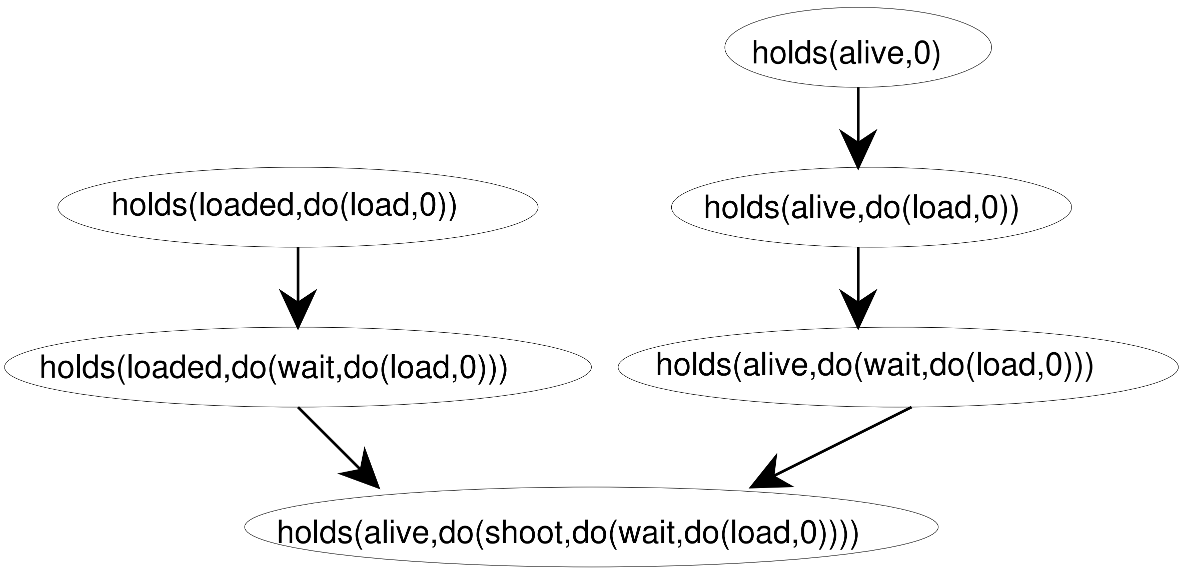

Figure 3: Bayesian network for the Yale shooting problem.

The following is translation of the above diagram into a probabilistic Horn

[5]Pearl [39] has a similar graphical representation, although it was not explicitly a Bayesian network. Here we have used a situation calculus type representation.

abduction theory[6]:

$$
\begin{aligned}
&holds(alive, do(shoot, S)) \leftarrow holds(alive, S) \wedge holds(\overline{loaded}, S).\\
&holds(\overline{alive}, do(shoot, S)) \leftarrow holds(\overline{alive}, S).\\
&holds(\overline{alive}, do(shoot, S)) \leftarrow holds(alive, S), holds(loaded, S).\\
&holds(X, do(wait, S)) \leftarrow holds(X, S).\\
&holds(loaded, do(load, S)).\\
&holds(alive, do(load, S)) \leftarrow holds(alive, S).\\
&holds(\overline{alive}, do(load, S)) \leftarrow holds(\overline{alive}, S).\\
&holds(alive, 0).\\
&holds(\overline{loaded}, 0).
\end{aligned}
$$

The following formula can be proved:

$$holds(\overline{alive}, do(shoot, do(wait, do(load, 0)))).$$

The solution that is derived from the Bayesian network representation is very similar to logic programming solutions to the frame problem (see e.g., [62, 1]) using the equivalence of our negation to negation as failure for acyclic theories [4] and, considering the completion of the program, to the recent deductive solutions to the frame problem proposed by Reiter [58] and Elkan [16].

The probabilistic Horn abduction representation allows us to consistently add probability to the temporal representations. For example, we could add that there is a 0.01 probability of the person dying, and a 0.03 probability of the gun becoming unloaded during a wait operation, and a 0.002 probability of the gun becoming spontaneously loaded while waiting (this is why it is dangerous to play with guns):

$$
\begin{aligned}
&holds(alive, do(wait, S)) \leftarrow holds(alive, S) \wedge alive\_persists(S).\\
&holds(\overline{alive}, do(wait, S)) \leftarrow holds(\overline{alive}, S).\\
&holds(alive, 0).\\
&holds(\overline{alive}, do(wait, S)) \leftarrow holds(alive, S) \wedge dies(S).
\end{aligned}
$$

[6] The atom $holds(\overline{\alpha}, \sigma)$ represents the negation of $holds(\alpha, \sigma)$. Here we have explicitly added the rules for the negations of the atoms (see Section 2.5.3). The rules have been generalised where appropriate.

$$disjoint([alive\_persists(S):0.99, dies(S):0.01])$$
$$holds(loaded, do(wait, S)) \leftarrow holds(loaded, S) \wedge loaded\_persists(S).$$
$$holds(loaded, do(wait, S)) \leftarrow holds(\overline{loaded}, S) \wedge spontaneously\_loads(S).$$
$$holds(\overline{loaded}, do(wait, S)) \leftarrow holds(loaded, S) \wedge becomes\_unloaded(S).$$
$$holds(\overline{loaded}, do(wait, S)) \leftarrow holds(\overline{loaded}, S) \wedge unloaded\_persists(S).$$
$$disjoint([loaded\_persists(S):0.97, becomes\_unloaded(S):0.03]).$$
$$disjoint([unloaded\_persists(S):0.998, spontaneously\_loads(S):0.002]).$$

This formulation gets other persistence problems right. For example, if we happened to observe that the person is alive after the shooting, then the gun must have become unloaded. If there were a number of wait operations, the unloading could have occurred at any of them.

5 Representational Methodology

Once we have a tool, it is important to know how to use it. The problem of a representational methodology [46] is an important and much overlooked part of automated reasoning research.

It may seem that the assumptions used in designing probabilistic Horn abduction were so restrictive that the system would be useless for real problems. In this section, I argue that this is not the case.

The general idea is to use definite clauses to write a simulation (in the "causal" direction [61]) based on different possible hypotheses. This axiomatisation must follow the assumptions about the rule base and about the independence of hypotheses, but we argue in this section that this is not too difficult.

Indeed it is arguable, that rather than stifling the imagination of the axiomatiser to write "what is true" in their domain, placing restrictions on the representation language provides guidance to how to go about thinking about the domain. One of the aims of restricting the language is to make it easier to write and understand an axiomatisation of the world. Whether this is true in practice, however, remains to be seen.

5.1 Disjoint and Covering Explanations

For our probabilistic analysis (section 2.4), we assumed that the explanations were disjoint and covering. If we want our probabilities to be correct, we must ensure that the rules for an atom are disjoint and covering.

If the rules for an atom a are not covering, we can invent another cause for the goal representing "all the other possible causes" of the atom [14, 45], and add

$$a \leftarrow a\_true\_for\_some\_other\_reason.$$

and make $a\_true\_for\_some\_other\_reason$ into a hypothesis.

Although disjointedness of rules places a restriction on the knowledge base, it does not place a restriction on the sorts of knowledge that we can represent. In general, suppose we have rules:

$$\begin{array}{rcl} a & \leftarrow & b_1. \\ & \vdots & \\ a & \leftarrow & b_n. \end{array}$$

Create $\overline{b_i}$ as the proposition that is the negation of b_i (see Section 2.5.3), we can make sure the rules are disjoint by transforming them into

$$\begin{array}{rcl} a & \leftarrow & b_1. \\ a & \leftarrow & \overline{b_1} \wedge b_2. \\ a & \leftarrow & \overline{b_1} \wedge \overline{b_2} \wedge b_3. \\ & \vdots & \\ a & \leftarrow & \overline{b_1} \wedge \cdots \wedge \overline{b_{n-1}} \wedge b_n. \end{array}$$

Thus we make the rules disjoint, by ordering the rules and making sure that the bodies of rules are false if the bodies of preceding rules are true.

Syntactically, this seems to increase the complexity of n rules to have $\frac{n(n+1)}{2}$ atoms in the body. While this is true, there are only $2n-1$ different atoms that need to be explained. Thus, in practice, the complexity need only increase linearly not as a square.

Example 5.1 Suppose we want to represent an "and-gate" that should have value 0 if either of the inputs are zero. Suppose we represent the proposition that port G has output V at time T as $val(G, V, T)$. We can ensure that the explanations are disjoint locally by ensuring that only one body can ever be true:

$$\begin{aligned}
val(out(G), off, T) &\leftarrow and_gate(G) \wedge ok(G) \\
&\quad\ \wedge val(input(1, G), off, T). \\
val(out(G), off, T) &\leftarrow and_gate(G) \wedge ok(G) \\
&\quad\ \wedge val(input(1, G), on, T) \\
&\quad\ \wedge val(input(2, G), off, T). \\
val(out(G), on, T) &\leftarrow and_gate(G) \wedge ok(G) \\
&\quad\ \wedge val(input(on, G), 1, T) \\
&\quad\ \wedge val(input(2, G), on, T).
\end{aligned}$$

Note that the third conjunct in body of the second rule (the "$val(input(1, G), on, T)$") is there to ensure that the bodies are disjoint.

This has repercussions in biasing the most likely explanation to the first rule, which is more general than the others. This problem of the most likely diagnosis depending on the representation seems endemic to approaches that try to find the diagnosis (either explanation or interpretation) that is "most likely" [39, 52]. We avoid this problem by not placing importance in the most likely explanations, but only in how they contribute to the probability of propositions.

5.2 Causation Events

When representing knowledge for abduction [45, 46], we have to be able to make sure that we can imply the observations. In general a fault or disease doesn't *imply* a particular observation. For example, having a cold does not imply sneezing, but could cause sneezing. A gate being in an unknown state does not imply any particular value for the output of the gate. To solve this problem we introduce another hypothesis that the cold caused the sneezing. In the other example, we have to hypothesise that the gate is producing a particular value. This idea is analogous to the notion of a "causation event" of Peng and Reggia [40].

The cold causing sneezing could be written as

$$sneeze \leftarrow cold \wedge cold\_caused\_sneeze$$

Following Peng and Reggia [40], one way to implement the causation events, is to use the relations $has\_disease(D)$ to mean that the patient has disease D; $actually\_causes(D, M)$ to mean that disease D "actually caused" manifestation M; and $has\_manifestation(M)$ to mean that the patient has manifestation M.

We can say that a manifestation is caused by the disease that actually causes it by:

$$\begin{array}{rcl} has\_manifestation(M) & \leftarrow & has\_disease(D) \\ & & \wedge actually\_causes(D, M). \end{array}$$

The conjunction

$$has\_disease(D) \wedge actually\_causes(D, M)$$

corresponds to Peng and Reggia's [40] causation event $M : D$.

We have the disjoint declarations for each i, j:

$$disjoint([actually\_causes(d_i, m_j) : p_{ij}, didnt\_actually\_cause(d_i, m_j) : q_{ij}])$$

where p_{ij} corresponds to the the "conditional causal probability" ("causal strength") of [40], and $q_{ij} = 1 - p_{ij}$. p_{ij} can be seen as the fraction of the cases where d_i is true that d_i "actually causes" m_j.

We also have the possible hypotheses

$$disjoint([has\_disease(d_i) : p_i, doesnt\_have\_disease(d_i) : q_i])$$

where p_i is the prior probability of the disease d_i, and $q_i = 1 - p_i$.

To implement this we still have to worry about making the rules disjoint. This is done in the same way as in section 5.1. If manifestation m has possible causes $d_1, \cdots, d_k$, we write:

$$\begin{array}{rcl} has\_manifestation(m) & \leftarrow & has\_disease(d_1) \\ & & \wedge actually\_causes(d_1, m). \\ has\_manifestation(m) & \leftarrow & has\_disease(d_2) \wedge doesnt\_have\_disease(d_1) \\ & & \wedge actually\_causes(d_2, m). \\ & \vdots & \end{array}$$

$$\begin{aligned} has\_manifestation(m) \leftarrow\ & has\_disease(d_k) \\ & \wedge doesnt\_have\_disease(d_1) \wedge \cdots \\ & \wedge doesnt\_have\_disease(d_{k-1}) \\ & \wedge actually\_causes(d_k, m). \end{aligned}$$

The advantage of making these disjoint is that we are now able to interpret "actually causing" observationally for the cases where there are two possible causes. Here we arbitrarily assign the "actual cause" to the first disease. It is now easy to interpret the notion of "actually causing", as there is no ambiguity in any data. This makes the concept of "actual cause" into an observational notion for which we can collect statistics and do not need a theory of causation that is deeper than the theory we want to represent.

5.2.1 Hypotheses with indeterminate output

There is one case where we have to be concerned about causation events as well as the problem of parametrizing possible hypotheses and the interaction with the independence assumption. I have argued elsewhere [45, 46] that there is much power obtainable and subtlety involved in parametrizing hypotheses appropriately. In this section we expand on previous analysis [46], and show how probabilities affect parametrization considerations when using causation events by considering some case studies.

As an example, suppose we have a gate G that takes two values as input, and outputs a value that can be in the range 1 to n. Suppose we want to represent the gate being in an unknown state[7] (this is applicable whether or not we have fault models [14, 45]). Suppose we represent the proposition that gate G has output V at time T as $val(G, V, T)$.

We cannot represent the hypothesis that the gate is in the unknown state by using the hypothesis $u(G)$ and the fact

$$val(out(G), V, T) \leftarrow u(G).$$

The problem is that the above fact states that a gate in the unknown state produces *all* values of output, rather than saying that it produces some out-

[7] The unknown state is a state that we do not know anything about. This state is different to the *ok* state and other fault states. It does not mean that the gate is in some state, but we do not know what state it is in.

put. Knowing a gate is in an unknown state does not imply any value for the output.

When there are no probabilities involved [46, 45] we parametrize the hypothesis by the values on which it depends. This could be done by having the hypothesis $produces(G, V, T)$ (interpreted as "gate G is faulty and produces value V at time T") and the rule

$$val(out(G), V, T) \leftarrow produces(G, V, T).$$

We would say that a port has only one value at a time by having the disjoint declaration[8]:

$$disjoint([ok(P) : p_0, produces(P, v_1, T) : p_1, \cdots, produces(P, v_r, T) : p_r]).$$

Suppose we know that gate g_1 has probability ϵ of being in the unknown state. Then $p_0 = 1 - \epsilon$. If we assume that each possible output value has equal chance, and that there are r possible output values, then p_i, the prior probability that it produces output value v_i is ϵ/r for $1 \leq i \leq r$.

When we have more than one observation, there is another problem. For the probabilities we assumed that the hypotheses were independent. We would not expect that

$$P(produces(g_1, 1, t_2) | produces(g_1, 1, t_1)) = P(produces(g_1, 1, t_2))$$

Once we know that the gate is in an unknown state at time t_1 it should not be so unlikely that it is in an unknown state at time t_2. The fact that the gate is in an unknown state is independent of the time. We would not expect that the gate has probability ϵ^k of being in the unknown state for k periods of time, rather we would expect that the gate has probability ϵ of being in an unknown state[9].

To work in general, we need a mixture of the above two ideas. Suppose a gate G has probability of ϵ of being in the unknown state, and that there are r possible output values, each of which has an equal prior chance of being produced by a gate in the unknown state. This can be represented as the

[8] Here we have assumed that there are no fault states other than the unknown state. These could be added to this declaration without changing the point of the discussion.

[9] Of course, probabilistic Horn abduction can represent either alternative.

hypotheses[10]

$$disjoint([produces(P,v_1,T):\tfrac{1}{r},\cdots,produces(P,v_r,T):\tfrac{1}{r}]).$$
$$disjoint([ok(G):p_0,\cdots,u(G):\epsilon]).$$

and the rule

$$val(out(G),V,T)\leftarrow u(G)\wedge produces(G,V,T).$$

$u(G)$ means G is in the unknown state, and $produces(G,V,T)$ means that given gate G is broken, it produces value V at time T. The system assumes once that the gate is broken, and then makes other assumptions of what values it is producing at different times.

The atom $produces(G,V,T)$ can be seen as a "causation event", that we invent because being in an unknown fault state does not imply any particular value.

5.2.2 Intermittent versus non-intermittent faults

Because of the way we parametrized the hypotheses, the above representation of faults says that the output is a function of only the time. The hypothesis $produces(G,V,T)$ and the above rules places no constraints on the values of the outputs at different times. This is a way to represent the fact that the gate can have an intermittent fault (it depends only on the time of observation). There is no constraint that says the gate produces the same output when given the same inputs at different times.

We can give the non-intermittency assumption by saying that the fault depends only on the input and not on the time. This can be done instead by having the hypothesis $prod(G,V,I_1,I_2)$ (meaning gate G produces output V when given I_1 and I_2 as input) and a rule

$$\begin{aligned} val(out(G),V,T) \;\leftarrow\; & u(G)\wedge prod(G,V,I_1,I_2)\\ & \wedge val(input(1,G),I_1,T)\\ & \wedge val(input(2,G),I_2,T). \end{aligned}$$

[10] Here we assume a uniform distribution of values. Any other distribution could be given.

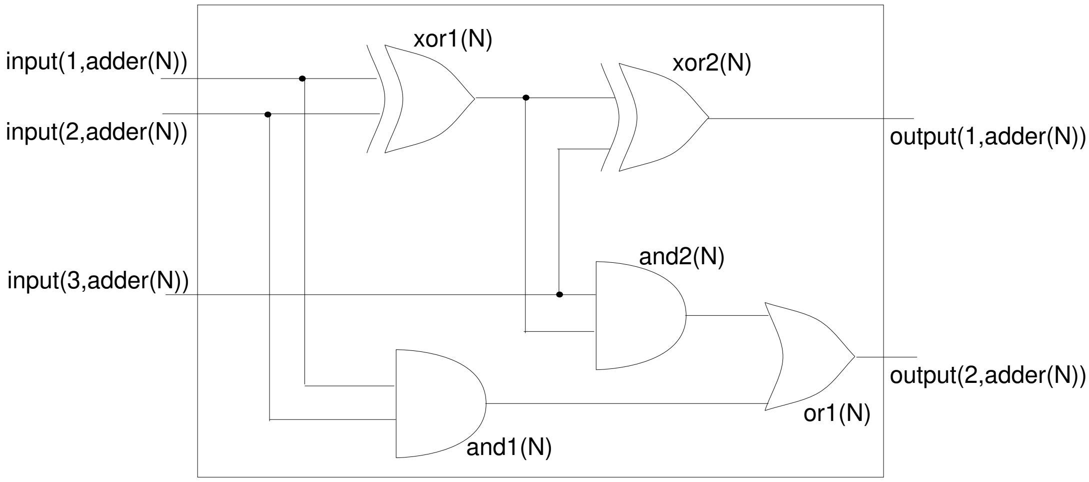

Figure 4: One bit adder, $adder(N)$.

5.3 Two Examples

In this section we show the complete theories for two non-trivial examples.

Example 5.2 This first example is an implementation of cascaded one bit adders (Figures 4 and 5), to form a ripple adder.

The axiomatisation is adapted from the consistency-based axiomatisation of Genesereth [17].

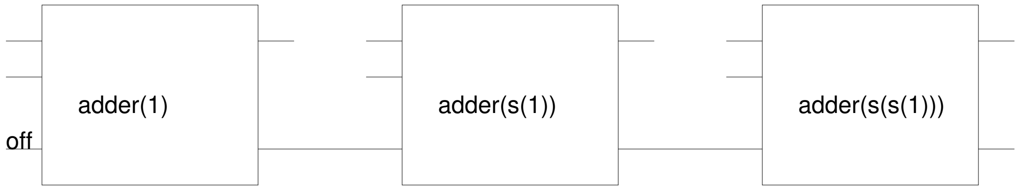

Figure 5: Three cascaded adders

$val(P, V, T)$ means that port P has value V at time T. We use as simple a representation of time as is needed. In this case we need to be able to have different observations at different times, and use constants to denote different times.

We first axiomatise how gates work. We must axiomatise how normal gates as well as faulty gates work [43, 45].

Each of the gates can be in one of four states (ok, stuck on, stuck off or unknown). When the gate is in the unknown u state it can produce either value with equal probability. We also used the intermittency assumption.

$$
\begin{aligned}
&val(output(G), off, T) \leftarrow \\
&\qquad gate(G, and) \wedge ok(G) \wedge val(input(1, G), off, T). \\
&val(output(G), off, T) \leftarrow \\
&\qquad gate(G, and), ok(G) \wedge val(input(1, G), on, T) \wedge \\
&\qquad val(input(2, G), off, T). \\
&val(output(G), on, T) \leftarrow \\
&\qquad gate(G, and) \wedge ok(G) \wedge val(input(1, G), on, T) \wedge \\
&\qquad val(input(2, G), on, T).
\end{aligned}
$$

$$
\begin{aligned}
&val(output(G), on, T) \leftarrow \\
&\qquad gate(G, or) \wedge ok(G) \wedge val(input(1, G), on, T). \\
&val(output(G), off, T) \leftarrow \\
&\qquad gate(G, or) \wedge ok(G) \wedge val(input(1, G), off, T) \wedge \\
&\qquad val(input(2, G), off, T). \\
&val(output(G), on, T) \leftarrow \\
&\qquad gate(G, or) \wedge ok(G) \wedge val(input(1, G), off, T) \wedge \\
&\qquad val(input(2, G), on, T).
\end{aligned}
$$

$$
\begin{aligned}
&val(output(G), off, T) \leftarrow \\
&\qquad gate(G, xor) \wedge ok(G) \wedge val(input(1, G), off, T) \wedge \\
&\qquad val(input(2, G), off, T). \\
&val(output(G), on, T) \leftarrow \\
&\qquad gate(G, xor) \wedge ok(G) \wedge val(input(1, G), off, T) \wedge \\
&\qquad val(input(2, G), on, T). \\
&val(output(G), on, T) \leftarrow \\
&\qquad gate(G, xor) \wedge ok(G) \wedge val(input(1, G), on, T) \wedge \\
&\qquad val(input(2, G), off, T).
\end{aligned}
$$

$$
\begin{aligned}
&val(output(G), off, T) \leftarrow \\
&\qquad gate(G, xor) \wedge ok(G) \wedge val(input(1, G), on, T) \wedge \\
&\qquad val(input(2, G), on, T).
\end{aligned}
$$

$$
\begin{aligned}
&val(output(G), on, T) \leftarrow stuck1(G). \\
&val(output(G), off, T) \leftarrow stuck0(G). \\
&val(output(G), V, T) \leftarrow u(G) \wedge produced(G, V, T).
\end{aligned}
$$

$$val(P, V, T) \leftarrow conn(Q, P) \wedge val(Q, V, T).$$

Note how we have made the rules disjoint, by adding extra conditions. For example, the second rule includes the condition $val(input(1, G), on, T)$ that is there just to make sure the rules are disjoint. Treated as a definite clause in isolation this rule is true without this second condition.

We also specify the random variables as outlined in the preceding section. Note that this implies that the gates fail independently.

$$
\begin{aligned}
&disjoint([ok(X) : 0.999, u(X) : 0.0000001, stuck1(X) : 0.0004999, stuck0(X) : 0.0005]). \\
&disjoint([produced(X, on, T) : 0.5, produced(X, off, T) : 0.5]).
\end{aligned}
$$

We axiomatise how the gates in an adder are connected, and what gates there are in an adder.

$$
\begin{aligned}
&conn(input(1, adder(N)), input(1, xor1(N))). \\
&conn(input(1, adder(N)), input(1, and1(N))). \\
&conn(input(2, adder(N)), input(2, xor1(N))). \\
&conn(input(2, adder(N)), input(2, and1(N))). \\
&conn(input(3, adder(N)), input(2, xor2(N))). \\
&conn(input(3, adder(N)), input(1, and2(N))). \\
&conn(output(xor1(N)), input(1, xor2(N))). \\
&conn(output(xor1(N)), input(2, and2(N))). \\
&conn(output(and1(N)), input(2, or1(N))). \\
&conn(output(and2(N)), input(1, or1(N))). \\
&conn(output(xor2(N)), output(1, adder(N))). \\
&conn(output(or1(N)), output(2, adder(N))).
\end{aligned}
$$

$conn(output(2, adder(N)), input(3, adder(N1))) \leftarrow succ(N, N1).$
$val(input(3, adder(1)), off, T).$

$gate(xor1(N), xor).$
$gate(xor2(N), xor).$
$gate(and1(N), and).$
$gate(and2(N), and).$
$gate(or1(N), or).$

$succ(N, s(N)).$

The relation $succ(N, N1)$ is used to state when one gate is next to another. This allows us to observe arbitrarily large cascaded adders.

In order for us to be able to observe inputs and to be able to predict expected values from unknown inputs we can make the inputs to the gates to be random variables. (The alternative is to write as facts what the inputs to the gates are [45]).

$disjoint([val(input(1, adder(N)), on, T) : 0.5, val(input(1, adder(N)), off, T) : 0.5]).$
$disjoint([val(input(2, adder(N)), on, T) : 0.5, val(input(2, adder(N)), off, T) : 0.5]).$

We can specify an observation such as that $10 + 11$ gave 001, as

$$\begin{array}{ll} & val(input(1, adder(1)), off, t1) \\ \wedge & val(input(1, adder(s(1))), on, t1) \\ \wedge & val(input(2, adder(1)), on, t1) \\ \wedge & val(input(2, adder(s(1))), on, t1) \\ \wedge & val(output(1, adder(1)), on, t1), \\ \wedge & val(output(1, adder(s(1))), off, t1) \\ \wedge & val(output(2, adder(s(1))), off, t1) \end{array}$$

Example 5.3 The second example is of the framework for depiction and image interpretation of Reiter and Mackworth [60]. Here we interpret simple line drawings of a map. These consist of lines and areas that depict roads, rivers, shores, lakes and land. Our axiomatisation is based on the abductive representation of [46].

scene	image
$linear(\sigma(X), road)$ $linear(\sigma(X), river)$ $linear(\sigma(X), shore)$	$chain(X)$
$area(\sigma(X), land)$ $area(\sigma(X), water)$	region(X)
$joins(\sigma(X), \sigma(Y), E)$ $flowsto(\sigma(X), \sigma(Y))$	$tee(X, Y, E)$
$docross(\sigma(X), \sigma(Y))$	$chi(X, Y)$
$source(\sigma(X), N)$ $petersout(\sigma(X), N)$	$open(X, N)$
$linear(\sigma(X), shore)$ $roadloop(\sigma(X))$	closed(X)
$beside(\sigma(X), \sigma(Y))$	$bounds(X, Y)$
$inside(\sigma(X), \sigma(Y)) \wedge outside(\sigma(X), \sigma(Z))$	$encloses(Y, X, Z)$

Figure 6: Image–scene predicates.

We axiomatise how scene objects could have produced image objects. Given an image we conjecture scene elements that could have produced that image.

The main difference between this axiomatisation and those of [60, 46] is that we have to make constructive derivations of the image. Rather than starting with all interpretations, and use consistency to prune those that are impossible, we make sure that we can only generate possible explanations. This follows the methodology given earlier in this section.

Following Reiter and Mackworth [60], for each image object I we assume a scene object $\sigma(I)$ which it depicts.

Figure 6 gives the correspondences between image and scene predicates.

We first allow one to write the building blocks of explanations. $area(S, land)$ means that the scene object S is land, given that it is a region. The probabilities reflect that in our (made-up) domain, 70% of the areas are water and 30% are land.

$region(I) \leftarrow area(\sigma(I), T).$

$disjoint([area(S, land) : 0.3, area(S, water) : 0.7]).$

Similarly $linear(S, T)$ means that linear scene object S is of type T, where T is one of *road*, *river* or *shore*. Linear scene objects are depicted as chains in the image.

$chain(I) \leftarrow linear(\sigma(I), T).$
$disjoint([linear(S, road) : 0.2, linear(S, river) : 0.5, linear(S, shore) : 0.3]).$

We now have axioms that describe how structured image objects (joins of chains and boundaries between chains and regions) could be produced in terms of scene objects.

$tee(X, Y, E)$ means that end E of chain X ends at chain Y. This can either be because X depicts a road that joins Y, or X depicts a river that starts at Y, or X depicts a river that flows into (river or shore) Y. We arbitrarily number ends of chains with a 0 or a 1, and one end of a river needs to be a mouth and one a source of the river.

$tee(X, Y, E) \leftarrow joins(\sigma(X), \sigma(Y), E) \wedge linear(\sigma(X), road).$
$tee(X, Y, E) \leftarrow joins(\sigma(X), \sigma(Y), E) \wedge linear(\sigma(X), river) \wedge$
$\quad linear(\sigma(Y), road) \wedge source(\sigma(X), E).$
$tee(X, Y, E) \leftarrow linear(\sigma(X), river) \wedge canflowto(\sigma(Y)) \wedge$
$\quad flowsto(\sigma(X), Y) \wedge mouth(\sigma(X), E).$
$canflowto(S) \leftarrow linear(S, river).$
$canflowto(S) \leftarrow linear(S, shore).$
$disjoint([joins(S, T, E) : 0.05, notjoins(S, T, E) : 0.95]).$
$disjoint([mouth(S, 0) : 0.5, mouth(S, 1) : 0.5]).$
$disjoint([flowsto(R, S) : 0.1, notflowsto(R, S) : 0.9]).$
$disjoint([source(R, 1) : 0.5, source(R, 0) : 0.5]).$

Similarly we can handle two chains crossing (a "chi" — χ) in the image. Here again we just use the notion of a causal event to make the implication always true. Here we arbitrarily decided that all types of crossing are equally likely.

$chi(X, Y) \leftarrow crossable(\sigma(X), \sigma(Y)) \wedge docross(\sigma(X), \sigma(Y)).$

$crossable(X, Y) \leftarrow linear(X, XT) \wedge linear(Y, YT) \wedge crosstype(XT, YT).$
$crosstype(road, road).$
$crosstype(road, river).$
$crosstype(river, road).$
$crosstype(road, shore).$
$crosstype(shore, road).$
$disjoint([docross(X, Y) : 0.2, dontcross(X, Y) : 0.8]).$

We can also have a chain being open or closed. Note here that the probabilities tell us that all shores form closed loops, and some, but very few roads form loops. It is also more likely that a river ends nowhere than a road does (to have a road ending we have to hypothesise that the road peters out, whereas for a river we have to hypothesis that the end is a source of the river).

$open(X, N) \leftarrow linear(\sigma(X), river) \wedge source(\sigma(X), N).$
$open(X, N) \leftarrow linear(\sigma(X), road) \wedge petersout(\sigma(X), N).$
$disjoint([petersout(X, E) : 0.1, doesntpeterout(X, E) : 0.9]).$
$closed(X) \leftarrow linear(\sigma(X), shore).$
$closed(X) \leftarrow linear(\sigma(X), road) \wedge roadloop(\sigma(X)).$
$disjoint([roadloop(X) : 0.01, notloop(X) : 0.99]).$

We can also have a chain bounding an area, and reason about what areas can be inside or outside loops. Image predicate $encloses(Y, X, Z)$ means that region Y is interior to chain X, and region Z is exterior to chain X.

$bounds(X, Y) \leftarrow linear(\sigma(X), XT) \wedge area(\sigma(Y), YT) \wedge$
$\quad beside(\sigma(X), \sigma(Y)) \wedge possbeside(XT, YT).$
$possbeside(road, land).$
$possbeside(river, land).$
$possbeside(shore, land).$
$possbeside(shore, water).$
$disjoint([beside(X, Y) : 0.1, notbeside(X, Y) : 0.9]).$
$disjoint([inside(X, Y) : 0.1, outside(X, Y) : 0.1, noside(X, Y) : 0.8]).$
$encloses(Y, X, Z) \leftarrow outside(\sigma(X), \sigma(Z)) \wedge inside(\sigma(X), \sigma(Y)) \wedge linear(\sigma(X), XT) \wedge$
$\quad area(\sigma(Y), YT) \wedge area(\sigma(Z), ZT) \wedge possreg(YT, XT, ZT).$
$possreg(land, road, land).$
$possreg(land, shore, water).$
$possreg(water, shore, land).$

An image becomes an observation that we condition on. For example, the image of figure 7 is represented as the observation

$chain(c1) \wedge chain(c2) \wedge chain(c3) \wedge region(r1) \wedge region(r2) \wedge tee(c2, c1, 1) \wedge bounds(c2, r2) \wedge bounds(c1, r1) \wedge bounds(c1, r2) \wedge encloses(r1, c1, r2), open(c2, 0) \wedge closed(c1) \wedge open(c3, 0) \wedge tee(c3, c2, 1) \wedge bounds(c3, r2)$

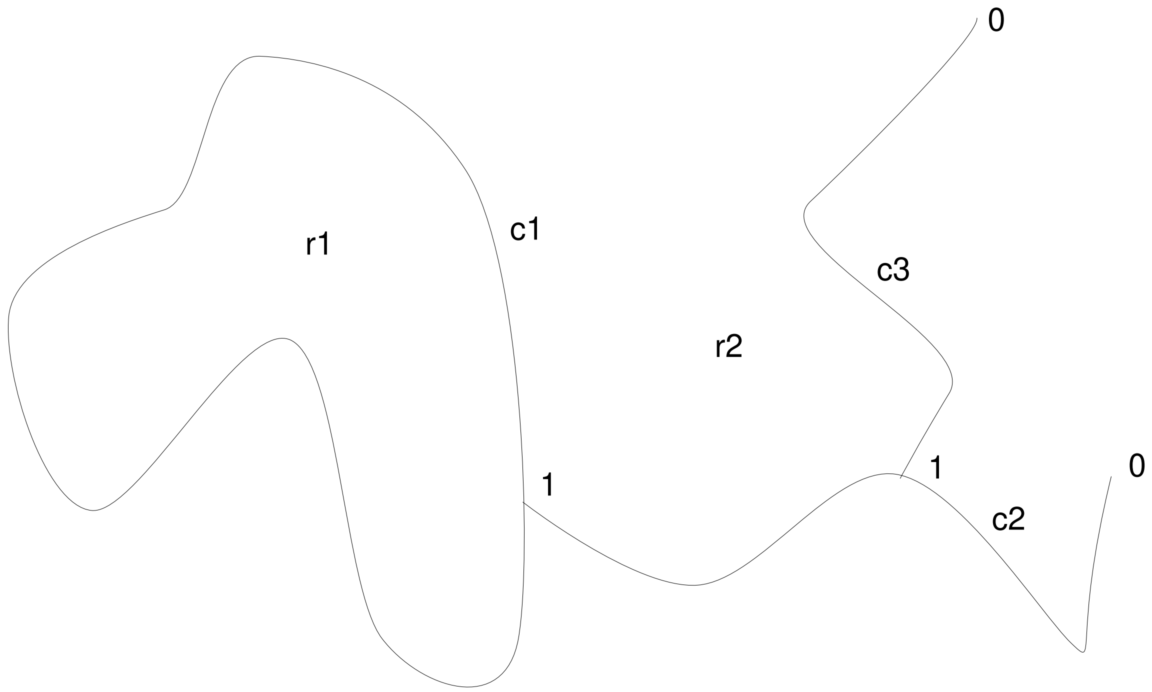

Figure 7: A simple image.

There are four explanations of this observation. These are given here with their corresponding prior probability:

Explanation: $\{linear(\sigma(c1), shore), linear(\sigma(c2), river), linear(\sigma(c3), river), area(\sigma(r1), water), area(\sigma(r2), land), flowsto(\sigma(c2), \sigma(c1)), mouth(\sigma(c2), 1),$

$beside(\sigma(c2), \sigma(r2)), beside(\sigma(c1), \sigma(r1)), beside(\sigma(c1), \sigma(r2)), outside(\sigma(c1), \sigma(r2)), inside(\sigma(c1), \sigma(r1)), source(\sigma(c2), 0), source(\sigma(c3), 0), flowsto(\sigma(c3), \sigma(c2)), mouth(\sigma(c3), 1), beside(\sigma(c3), \sigma(r2))\}$
Prior $= 9.8438 \times 10^{-12}$

Explanation: $\{linear(\sigma(c1), shore), linear(\sigma(c2), river), linear(\sigma(c3), road), area(\sigma(r1), water), area(\sigma(r2), land), flowsto(\sigma(c2), \sigma(c1)), mouth(\sigma(c2), 1), beside(\sigma(c2), \sigma(r2)), beside(\sigma(c1), \sigma(r1)), beside(\sigma(c1), \sigma(r2)), outside(\sigma(c1), \sigma(r2)), inside(\sigma(c1), \sigma(r1)), source(\sigma(c2), 0), petersout(\sigma(c3), 0), joins(\sigma(c3), \sigma(c2), 1), beside(\sigma(c3), \sigma(r2))\}$
Prior $= 7.875 \times 10^{-13}$

Explanation: $\{linear(\sigma(c1), shore), linear(\sigma(c2), road), linear(\sigma(c3), road), area(\sigma(r1), water), area(\sigma(r2), land), joins(\sigma(c2), \sigma(c1), 1), beside(\sigma(c2), \sigma(r2)), beside(\sigma(c1), \sigma(r1)), beside(\sigma(c1), \sigma(r2)), outside(\sigma(c1), \sigma(r2)), inside(\sigma(c1), \sigma(r1)), petersout(\sigma(c2), 0), petersout(\sigma(c3), 0), joins(\sigma(c3), \sigma(c2), 1), beside(\sigma(c3), \sigma(r2))\}$
Prior $= 6.3 \times 10^{-14}$

Explanation: $\{linear(\sigma(c1), road), linear(\sigma(c2), road), linear(\sigma(c3), road), area(\sigma(r1), land), area(\sigma(r2), land), joins(\sigma(c2), \sigma(c1), 1), beside(\sigma(c2), \sigma(r2)), beside(\sigma(c1), \sigma(r1)), beside(\sigma(c1), \sigma(r2)), outside(\sigma(c1), \sigma(r2)), inside(\sigma(c1), \sigma(r1)), petersout(\sigma(c2), 0), roadloop(\sigma(c1)), petersout(\sigma(c3), 0), joins(\sigma(c3), \sigma(c2), 1), beside(\sigma(c3), \sigma(r2))\}$
Prior $= 1.8 \times 10^{-16}$

The prior probability of the image is the sum of the prior probabilities of these four explanations, namely 1.0694×10^{-11}. We can use these explanations to compute arbitrary conditional probabilities. For example,

$$P(linear(\sigma(c2), river)|image) = \frac{9.8438 \times 10^{-12} + 7.875 \times 10^{-13}}{1.0694 \times 10^{-11}} = 0.99419$$

5.4 Arbitrary Individuals

One of the problems considered in [22], is that of when there can be arbitrary individuals that affect a value, and the individuals present can only be determined at run time.

As an example, consider the problem of a fire alarm going off, where it goes off if it was set off by one of the individuals present. All of the individuals can independently set off the alarm. We cannot write rules such as

$$alarm(sounds) \leftarrow present(P) \wedge set\_off\_alarm(P).$$

as the disjoint rules condition is violated. This is related to the problem in first order logic of being able to count the number of people present given just a database of relations of the form $present(P)$. We cannot prove there are 5 people present just because we have $present(P)$ true for 5 instances of P, unless we have stated that everyone else is not present. Like first-order logic, we cannot represent such knowledge.

We can however encode the problem so that we can count the number of people present. This is by forcing us to write $present(L)$ where L is a list of the people present.

We can now represent the problem by[11]:

$$\begin{array}{rcl}
alarm(sounds) & \leftarrow & present(L) \wedge one_set_off_alarm(L). \\
alarm(quiet) & \leftarrow & present(L) \wedge none_set_off_alarm(L). \\
one_set_off_alarm([H|T]) & \leftarrow & set_off_alarm(H). \\
one_set_off_alarm([H|T]) & \leftarrow & didnt_set_off_alarm(H) \\
 & & \wedge one_set_off_alarm(T). \\
none_set_off_alarm([H|T]) & \leftarrow & didnt_set_off_alarm(H) \\
 & & \wedge none_set_off_alarm(T). \\
none_set_off_alarm([]). & &
\end{array}$$

$$disjoint([set\_off\_alarm(P) : p_1, didnt\_set\_off\_alarm(P) : p_2]).$$

Here p_1 is the probability that a person would have set off an alarm given that no one before them had already set off the alarm. $p_2 = 1 - p_1$.

6 Comparison with Other Systems

Before comparing the work in this paper with other proposals, we should note that in this paper we have described a representation, and not a way

[11] Here we use the Prolog syntactic sugar for lists [64]. $[\cdot|\cdot]$ is just a binary function symbol, $[]$ is a constant. We use the notational convention that $[\alpha|[\beta]]$ is written as $[\alpha, \beta]$ for any sequence of symbols α and β.

to compute posterior probabilities. There are many ways that could be used to compute posterior probabilities, we could use some form of stochastic simulation (e.g., [20]), search (e.g., [49, 50]), or even using the mapping to Bayesian networks to translate ground instances of the theory into some secondary structure for propagating evidence (e.g., [27, 23]). We thus only compare the representation to other proposals, and not any implementations.

6.1 Other logic-based abductive schemes

There are many other proposals for logic-based abduction schemes (e.g., [53, 10, 18]). These however consider that we have to find all of the diagnoses. In practice there are prohibitively many of these. It is also not clear what to do with all of the explanations; there are too many to give to a user, and the costs of tests to determine which of the diagnoses is the "real" diagnosis is usually not outweighed by the advantages of finding the real diagnosis (see Ledley and Lusted [28] for an early description of the importance of probabilistic and value information in diagnosis). We provide an answer to the problem of what to do with the explanations: we use them to compute posterior probabilities that can be used for making decisions.

The closest version of abduction to that presented here is that of Goebel et. al. [18], where there is also the simple abductive scheme where we do not need to do any chaining in order to determine inconsistency. Essentially we have added probabilities to that scheme under certain assumptions about the knowledge base and independence.

6.2 Probability and diagnosis

de Kleer and Williams [13, 14] and Peng and Reggia [40] both incorporate probabilistic knowledge to find the most likely diagnoses, but do not provide as flexible and simple a representation language as the one here.

de Kleer and Williams [13, 14] have explored the idea of using probabilistic information in consistency-based diagnosis (see [43, 45] for comparisons between abductive and consistency-based diagnoses).

The major differences between their approach and the one presented in this paper is that they differ in what they want to find the probability of. de Kleer and Williams find the most likely interpretations (assignment of values to all hypotheses). This is the same as the diagnoses of Peng and Reggia

[40] and the composite beliefs of Pearl [37], but is different from the diagnoses of de Kleer, Mackworth and Reiter [12]. We find the probabilities of explanations; we remain agnostic about the value of "irrelevant" hypotheses. de Kleer and Williams cannot distinguish between diagnoses that differ in substantial ways from those that differ only in varying values that are irrelevant to the diagnosis. In our system, hypotheses that are not part of an explanation are irrelevant and are ignored. We do not place such an importance on the explanations, but rather on using the explanations to compute probabilities (see [52] for some of the issues involved in considering what we want to compute the probability of).

Peng and Reggia [40] also consider an abductive definition of diagnosis and incorporate probabilities, and best-first search. One difference is that they are trying to find probabilities of interpretations, but we are using explanations to find the probabilities of atoms. The main difference is in the underlying language. They use the notion of "hyper-bipartite" graphs made up of causation relations on sets of manifestations (can be observed), disorders (can be hypothesised), and pathological states.

One way to look at what they are doing is to consider it as a restriction of the system presented here where the language is propositional and allows only one element in the body of a clause. It is, however, not expressed in a logical language.

Hobbs et. al. [21] have devised a "cost-based abduction" for interpretation of natural language. Their scheme is similar to the one presented here, but they use costs associated with assumptions rather than probabilities. These costs can be seen as -log probabilities [7]. One can view the current work as extending Hobbs et. al.'s to derive posterior probabilities in a consistent manner.

6.3 Logic and Bayesian networks

The representation of Bayesian networks is related to the work by Charniak and Shimony [7, 63]. Instead of considering abduction, they consider models that consist of an assignment of values to each random variable. The *label* of [63] plays an analogous role to our hypotheses. They however, do not use their system for computing posterior probabilities. It is also not so obvious how to extend their formalism to more powerful logics.

Horsch and Poole [22], Breese [5] have defined systems that incorporate

Prolog style rules and Bayesian networks. These were designed to allow for dynamic construction of Bayesian networks. The rules of [22] cannot be interpreted logically but are macros that map into Bayesian network structure. The rules of Breese [5] are Prolog-style rules that are used to build a network. The output of the Prolog is a Bayesian network (or more generally is an influence diagram) that can be passed to a Bayesian network solver. Ours differs in that the Prolog like rules form the Bayesian network themselves. We would need another, meta-level, program to transform our Prolog rules into Bayesian networks for a traditional Bayesian network interpreter to solve.

6.4 Horn abduction and Dempster-Shafer

This work is also closely related to recent embeddings of Dempster-Shafer theory in ATMS [26, 54]. One difference between our embedding of Bayesian networks and Dempster-Shafer is in the independence assumptions used. Dempster-Shafer theory assumes that different rules are independent. We assume they are exclusive. Another difference is that these embeddings do not do evidential reasoning (by doing abduction), determining probability of hypotheses given evidence, but rather only determine the "belief" of propositions from forward chaining.

6.5 Argument systems

Doyle [15] and Loui [32] have argued that decisions can be best seen in terms of arguments for and against some propositions. Other have viewed nonmonotonic reasoning in terms of arguments [42, 29, 31, 41]. The explanations that we use can be seen as premises for logical arguments. We determine the probability of some proposition by coming up with arguments for the proposition. Rather than treating argument-based systems as an alternative to probabilistic reasoning, we treat the argument-based system as a representation for probabilistic reasoning.

6.6 Logic programming and uncertainty

There have been other attempts to incorporate uncertainty into logic programming. These have typically not considered probability, but other uncertainty calculi such as certainty factors [65]. Ng and Subrahmanian [35] have combined logic programming and probability by allowing us to axiomatise probability in logic, and then use Prolog-style rules to give the probabilistic information. The Prolog rules reason about the probability, at much more of a meta-level than that proposed here. They can write down any independence assumptions and any conditional probability statements with the system giving no guidance as to what to write. In other work, Ng and Subrahmanian [36] have considered how to incorporate statistical probability [3] into logic programming. This should be seen as complementary to the work in this paper.

7 Conclusion

This paper has presented a pragmatically-motivated simple logic formulation that includes definite clauses and probabilities over hypotheses. This was designed to be a compromise between representational adequacy, ease to interpret semantically what the knowledge means, and ease of implementation. It is suggested that this simple tool provides a good representation for many evidential reasoning tasks.

This is supported by the demonstration that probabilistic Horn abduction forms a logic of discrete Bayesian networks. There is a direct mapping between the knowledge in a Bayesian network and the knowledge in a probabilistic Horn abduction theory.

This is also interesting because it provides a link to earlier work on the use of assumption-based reasoning for default reasoning [42, 44, 46]. One of the ways of viewing default reasoning is where an adversary chooses the assumptions. One way of viewing the probabilistic Horn abduction is as an instance of assumption-based reasoning, but where nature chooses the assumptions. This paper also demonstrates the correspondence between the observations that need to be explained in abduction [45], and what is conditioned on in Bayesian probability.

A Formal Semantics

In this section we give the formal semantics for our language. As the language is very simple, the semantics will be correspondingly simple. The semantics will basically be that of Bacchus [3], restricted to our language, and incorporating our assumptions.

The logical statements will restrict the possible worlds, and the probabilities will provide measures over the possible worlds. The language is specially designed so that the logical facts neither implies a hypothesis nor implies the negation of a hypothesis. This means that we can treat the logic part and the probabilistic part of our semantics independently.

The tricky thing we have to worry about is that there are potentially infinitely many independent hypotheses, with non-extreme (i.e., not equal to 0 or 1) probabilities. If we have one function symbol and one parametrized hypothesis $h(X)$ with non extreme probability, we have infinitely many independent hypotheses of the form $h(t)$ for each ground term t. Once we have this, the probability of each possible world will be zero. Thus, we cannot just sum over the possible worlds to determine the probability of a proposition (as is done in [3]). We instead provide a measure over sentences that can be described in our language (as in [2]). As we have made sure that the logic provides no constraints on the probabilities, we only need to consider sentences made of hypotheses in order to define the probability space.

As discussed in section 2.4, we make the unique names assumption. This is specified formally by making the domain we consider be the set of ground terms in the language (similar the Herbrand Universe [30]).

Definition A.1 A **semantic structure** is a tuple $\langle W, D, \phi, \pi, P^* \rangle$, where

W is a non-empty set. Elements of W are called possible worlds.

D is a non-empty set (of individuals). D here is the set of ground terms in the language.

ϕ is a function that maps each n-ary function symbol to an element of $D^n \mapsto D$ (in particular ϕ maps each constant to an element of D). We can extend ϕ to ground terms, by the use of the recursive scheme $\phi(f(t_1, \cdots, t_n)) = \phi(f)(\phi(t_1), \cdots, \phi(t_n))$. In particular ϕ is the identity function so that $\phi(t) = t$ for any term t. Because the mapping does not depend on the world, these form "rigid designators".

π is a 1-1 function such that $\pi(w)$ maps each n-ary predicate symbol into a subset of D^n.

P^* is a function from $H \mapsto [0,1]$.

Definition A.2 We define the semantic relation $\models_{\langle W,D,\phi,\pi,P^*\rangle}$ between possible worlds and formulae:

The first case is for atoms

$$w \models_{\langle W,D,\phi,\pi,P^*\rangle} p(t_1,\cdots,t_n) \text{ if } \langle\phi(t_1),\cdots,\phi(t_n)\rangle \in \pi(w)(p).$$

The second is for conjunctions between atoms in the bodies of rules, as well as for conjunctions of definitions in our theory:

$$w \models_{\langle W,D,\phi,\pi,P^*\rangle} b_1 \wedge b_2 \text{ iff } w \models_{\langle W,D,\phi,\pi,P^*\rangle} b_1 \text{ and } w \models_{\langle W,D,\phi,\pi,P^*\rangle} b_2$$

The third rule defines the truth of definitions, and incorporates the assumptions about disjointedness and coveringness:

If $b_1,\cdots,b_n$ are all the bodies defining a

$w \models_{\langle W,D,\phi,\pi,P^*\rangle} (a \leftarrow b_1) \wedge \cdots \wedge (a \leftarrow b_n)$

if $\Big(w \models_{\langle W,D,\phi,\pi,P^*\rangle} a$ and $\exists i\ (w \models_{\langle W,D,\phi,\pi,P^*\rangle} b_i)$ and $\forall j \neq i\ (w \not\models_{\langle W,D,\phi,\pi,P^*\rangle} b_j)\Big)$

or $\Big(w \not\models_{\langle W,D,\phi,\pi,P^*\rangle} a$ and $\forall j\ (w \not\models_{\langle W,D,\phi,\pi,P^*\rangle} b_j)\Big)$

The fourth rule defines the "disjoint" assertion.

$$\begin{aligned}&w \models_{\langle W,D,\phi,\pi,P^*\rangle} disjoint([h_1:p_1,\cdots,h_m:p_m])\\ &\quad \text{if } \exists i\ (w \models_{\langle W,D,\phi,\pi,P^*\rangle} h_i) \text{ and } \forall j \neq i\ (w \not\models_{\langle W,D,\phi,\pi,P^*\rangle} h_j)\\ &\quad \text{and } \forall i\ P^*(h_i) = p_i\end{aligned}$$

Definition A.3 The set of possible worlds given theory T, denoted W_T is defined as

$$W_T = \{w \in W : w \models_{\langle W,D,\phi,\pi,P^*\rangle} T'\}$$

where T' is the set of ground instances of elements of T.

In order to interpret the probabilities of possible worlds, we create the algebra of subsets of W_T that can be described by finite formulae of instances of hypotheses.

Definition A.4

$$\Omega_{\langle W,D,\phi,\pi,P^*\rangle} = \{\omega \subseteq W_T : \exists \text{formula } f_\omega, \forall w \in \omega \; w \models_{\langle W,D,\phi,\pi,P^*\rangle} f_\omega\}$$

By formula we mean a finite formula make up of conjunctions and disjunctions of elements of H'.

The elements of $\Omega_{\langle W,D,\phi,\pi,P^*\rangle}$ form a sample space [2]. Elements of $\Omega_{\langle W,D,\phi,\pi,P^*\rangle}$ are closed under finite union, and complementation (given that we can complement any hypothesis by using the disjunct of the remaining hypotheses). Like Bacchus [2], we do not require sigma-additivity of our sample space. Because our language is so weak, we do not need countable unions.

Definition A.5 An **interpretable DNF formula** of hypotheses is a formula of the form

$$\bigvee_{i=1}^{n} \bigwedge_{j=1}^{k_i} h_{ij}$$

where h_{ij} is a ground instance of a hypothesis such that:

1. for each i, there is no $j_1 \neq j_2$ such that $\{h_{ij_1}, h_{ij_2}\}$ is a subset of an instance of a disjoint declaration in T (i.e., $\{h_{ij_1}, h_{ij_2}\}$ does not form part of an integrity constraint).
2. for each $i_1 \neq i_2$ there exists j_1, j_2 such that $\{h_{i_1 j_1}, h_{i_2 j_2}\}$ are in an instance of a disjoint declaration in T (i.e., the disjuncts are disjoint).
3. For no $i_1 \neq i_2$ is it the case that $\{h_{i_1 1}, \cdots, h_{i_1 k_{i_1}}\} \subseteq \{h_{i_2 1}, \cdots, h_{i_2 k_{i_2}}\}$.

Lemma A.6 Every finite formula make up of conjunctions and disjunctions of elements of H' is equivalent, given T (i.e. they describe the same subset of W_T) to an interpretable DNF formula.

> **Proof:** To satisfy the first and third conditions we can remove inconsistent conjuncts and any supersets of other formulae. The resulting formula is equivalent to the original.
>
> Suppose C_1 and C_2 are conjuncts that do not satisfy the second condition. Suppose, without loss of generality that $C_1 = \bigwedge_{j=1}^{k_1} h_{1j}$. Each h_{1j} is in a disjoint declaration. Let D_j be the disjunct of

the hypotheses in the instance of a disjoint declaration in which h_{1j} appears. Each D_j is true in all elements of W_T. Thus, $C_2 \wedge \bigwedge_{j=1}^{k_1} D_j$ is equivalent to C_2 given T. Distribute $C_2 \wedge \bigwedge_{j=1}^{k_1} D_j$ into DNF, remove the conjunct that is a superset of C_1, replace C_2 by this disjunct. We now have removed a violation of the second condition and have an equivalent formula. This can be done for all violations of the second formula. $\Box$

We can define a measure over the syntactic formulae as follows:

Definition A.7 If $\bigvee_{i=1}^{n} \bigwedge_{j=1}^{k_i} h_{ij}$ is an interpretable DNF formula, $h_{ij} \in H$ and P^* is a function from $H \mapsto [0,1]$, then define the function μ_{P^*} by

$$\mu_{P^*}\left(\bigvee_{i=1}^{n} \bigwedge_{j=1}^{k_i} h_{ij}\right) = \sum_i \prod_j P^*(h_{ij})$$

The following lemma shows that equivalent formulae will always have the same measure.

Lemma A.8 If f_1 and f_2 are interpretable DNF formulae such that $f_1 \equiv f_2$ then $\mu_{P^*}(f_1) = \mu_{P^*}(f_2)$.

Proof: Consider the set of all instances of hypotheses that appear in either f_1 or f_2. This subset of H' is finite. Each hypotheses in this set appears in a ground instance of a disjoint declaration. Let D be the set of all such disjoint declarations. Each disjoint declaration corresponds to a disjunct of the hypotheses of the disjunct declaration. Each conjunct in f_1 and f_2 has elements from a subset D. We can extend each conjunct to cover D, by conjoining to each hypothesis the disjunct of each disjoint declaration that is not represented in the conjunction. We then distribute to DNF. This procedure does not change the probability of any conjunct (as the probabilities of the disjunct of each disjoint declaration sum to one, and we can distribute multiplication over addition).

Once this procedure is carried to the two equivalent formulae, they will be syntactically identical (up to commutativity and associativity), as any difference can be extended into a possible world in which they have a different value. This cannot happen as they are equivalent.

Thus they must have the same measure. □

We can use this measure over formulae to induce a measure over the elements of $\Omega_{\langle W,D,\phi,\pi,P^*\rangle}$:

Definition A.9 Suppose $\omega \in \Omega_{\langle W,D,\phi,\pi,P^*\rangle}$ and ω can be described by formula f_ω. Suppose f_ω is equivalent to interpretable DNF formula $\bigvee_i \bigwedge_j h_{ij}$, then define

$$\mu_{\langle W,D,\phi,\pi,P^*\rangle}(\omega) = \sum_i \prod_j P^*(h_{ij})$$

The following lemma can be easily proven given Lemma A.8 showed that that above function is well defined.

Lemma A.10 $\mu_{\langle W,D,\phi,\pi,P^*\rangle}$ is a well-defined probability function, obeying the axioms of probability:

1. $\mu_{\langle W,D,\phi,\pi,P^*\rangle}(\omega) \geq 0$ for all $\omega \in \Omega_{\langle W,D,\phi,\pi,P^*\rangle}$
2. $\mu_{\langle W,D,\phi,\pi,P^*\rangle}(W_T) = 1$.
3. If $\omega_1 \cap \omega_2 = \{\}$ then $\mu_{\langle W,D,\phi,\pi,P^*\rangle}(\omega_1 \cup \omega_2) = \mu_{\langle W,D,\phi,\pi,P^*\rangle}(\omega_1) + \mu_{\langle W,D,\phi,\pi,P^*\rangle}(\omega_2)$, for all $\omega_1, \omega_2 \in \Omega_{\langle W,D,\phi,\pi,P^*\rangle}$

Definition A.11 If T is a probabilistic Horn abduction theory, and a is a formula, the probability of a given T, written $P_T(a)$ is defined as:

$$P_T(a) = \mu_{\langle W,D,\phi,\pi,P^*\rangle}(\{w \in W_T : w \models_{\langle W,D,\phi,\pi,P\rangle} a\})$$

The following lemma shows that P_T does indeed coincide with P^*.

Lemma A.12 *If h is a ground hypothesis defined in T then $P_T(h) = P^*(h)$.*

Proof:

$$P_T(h) = \mu_{\langle W,D,\phi,\pi,P^*\rangle}(\{w \in W_T : w \models_{\langle W,D,\phi,\pi,P\rangle} h\})$$

$\{w \in W_T : w \models_{\langle W,D,\phi,\pi,P\rangle} h\}$ is described by h, and so, by definition of $\mu_{\langle W,D,\phi,\pi,P^*\rangle}$, $P_T(h) = P^*(h)$. □

Theorem A.13 *If A is a ground atom or conjunction of ground atoms,*

$$P_T(A) = \sum_{e_i \in expl(A,T)} \prod_{h_{ij} \in e_i} P(h_{ij})$$

where $expl(A,T)$ is the set of minimal explanations of A from theory T.

In other words, the semantics justifies our use of summing the explanations to find the prior probability of a proposition.

> **Proof:** For this proof we treat A as a set of atoms as well as the conjunction of these elements. What is meant should be clear from context.
>
> Based on the acyclicity of T, we define a well founded ordering over sets of occurrences of atoms in T'. Because T' is acyclic (assumption 2.8), there is an assignment of natural number to occurrences atoms in T' such that the elements of the body of a rule are less than the head of the rule. Call this number the *depth* of the rule. Because there are no rules with a hypothesis as head (assumption 2.7), we can consistently assume that all hypotheses have depth zero.
>
> The induction ordering is based on the lexicographic ordering of pairs $\langle d, n \rangle$ where d is the depth of the element of the set with maximal depth, and n is the number of elements of this depth. Each time through the recursion either d is reduced or d is kept the same and n is reduced. This is well founded as both d and n are non-negative integers.
>
> For the base case, where $d = 0$, A is a conjunction of hypotheses. If A is inconsistent then there are no explanations, and the models of W_T can be described by $false$, the empty disjunct, and so $P_T(A) = 0$. If A is consistent, then there is one minimal explanation for A, namely A itself, so
>
> $$\sum_{e_i \in expl(A,T)} \prod_{h_{ij} \in e_i} P(h_{ij}) = \prod_{h \in A} P(A)$$
>
> Because the space defined by A has A as its interpretable DNF formula, we have
>
> $$P_T(A) = \prod_{h \in A} P(A).$$

For the inductive case, suppose $d > 0$ and a is a proposition in A with greatest depth. Let $R = A \backslash \{a\}$. As $d > 0$, a is not a hypothesis. Suppose there are m rules defining a:

$$\begin{array}{c} a \leftarrow B_1 \\ a \leftarrow B_2 \\ \vdots \\ a \leftarrow B_m \end{array}$$

Under the ordering above $B_i \cup R < d$, and so we can inductively assume the lemma for $B_i \cup R$. Thus

$$P_T(B_i \cup R) = \sum_{e_i \in expl(B_i \cup R, T)} \prod_{h_{ij} \in e_i} P(h_{ij})$$

Now the minimal explanations of a are obtained from the minimal explanations of the B_i; in particular

$$expl(\{a\} \cup R, T) = \bigcup_{i=1}^{m} expl(B_i \cup R, T)$$

Also, as far as the semantics are concerned $a \equiv \bigvee_i B_i$. Thus, by additivity (lemma A.10), and the fact that the B_i are disjoint

$$\begin{aligned} P_T(\{a\} \cup R) &= \sum_{i=1}^{m} P_T(B_i \cup R) \\ &= \sum_{i=1}^{m} \sum_{e_i \in expl(B_i \cup R, T)} \prod_{h_{ij} \in e_i} P(h_{ij}) \\ &= \sum_{e_i \in \bigcup_{i=1}^{m} expl(B_i \cup R, T)} \prod_{h_{ij} \in e_i} P(h_{ij}) \\ &= \sum_{e_i \in expl(\{a\} \cup R, T)} \prod_{h_{ij} \in e_i} P(h_{ij}) \end{aligned}$$

As $A = \{a\} \cup R$ the theorem is proved. $\Box$

Acknowledgements

Thanks to Alan Mackworth, Andrew Csinger, Michael Horsch, Keiji Kanazawa, George Tsiknis and the anonymous referees for valuable comments on this paper. Thanks to Mark Wallace for showing me that the language in [48] was too strong. Thanks to Fahiem Bacchus for help with the semantics. This research was supported under NSERC grant OGPOO44121, and under Project B5 of the Institute for Robotics and Intelligent Systems.

References

[1] K. R. Apt and M. Bezem. Acyclic programs. *New Generation Computing*, 9(3-4):335–363, 1991.

[2] F. Bacchus. Lp, a logic for representing and reasoning with statistical knowledge. *Computational Intelligence*, 6(4):209–231, November 1990.

[3] F. Bacchus. *Representing and Reasoning with Uncertain Knowledge.* MIT Press, Cambridge, Massachusetts, 1990.

[4] R. Barbuti, P. Mancarella, D. Pedreschi, and F. Turini. A transformational approach to negation in logic programming. *Journal of Logic Programming*, 8:201–228, 1990.

[5] J. S. Breese. Construction of belief and decision networks. Technical Memorandum 30, Rockwell Palo Alto Lab, Palo Alto, Cal., 1989.

[6] E. Charniak and R. Goldman. A semantics for probabilistic quantifier-free first-order languages, with particular application to story understanding. In *Proc. 11th International Joint Conf. on Artificial Intelligence*, pages 1074–1079, Detroit, Mich., August 1989.

[7] E. Charniak and S. E. Shimony. Probabilistic semantics for cost based abduction. In *Proc. 8th National Conference on Artificial Intelligence*, pages 106–111, Boston, July 1990.

[8] K. L. Clark. Negation as failure. In H. Gallaire and J. Minker, editors, *Logic and Databases*, pages 293–322. Plenum Press, New York, 1978.

[9] L. Console, D. Theseider Dupre, and P. Torasso. On the relationship between abduction and deduction. *Journal of Logic and Computation*, 1(5):661–690, 1991.

[10] P. T. Cox and T. Pietrzykowski. General diagnosis by abductive inference. Technical Report CS8701, Computer Science, Technical University of Nove Scotia, Halifax, April 1987.

[11] J. de Kleer. An assumption-based TMS. *Artificial Intelligence*, 28(2):127–162, March 1986.

[12] J. de Kleer, A. K. Mackworth, and R. Reiter. Characterizing diagnoses. In *Proc. 8th National Conference on Artificial Intelligence*, pages 324–330, Boston, July 1990.

[13] J. de Kleer and B. C. Williams. Diagnosing multiple faults. *Artificial Intelligence*, 32(1):97–130, April 1987.

[14] J. de Kleer and B. C. Williams. Diagnosis with behavioral modes. In *Proc. 11th International Joint Conf. on Artificial Intelligence*, pages 1324–1330, Detroit, August 1989.

[15] J. Doyle. Methodological simplicity in expert system construction: the case for judgements and reasoned assumptions. *The AI Magazine*, pages 39–43, Summer 1983.

[16] C. Elkan. Reasoning about action in first-order logic. In *Proc. 9th Can. Soc. Computational Studies of Intelligence Conf.*, pages 221–227, Vancouver, B.C., May 1992.

[17] M. R. Genesereth. The use of design descriptions in automated diagnosis. *Artificial Intelligence*, 24(1-3):411–436, December 1984.

[18] R. Goebel, K. Furukawa, and D. Poole. Using definite clauses and integrity constraints as the basis for a theory formation approach to diagnostic reasoning. In E. Shapiro, editor, *Proc. Third International Conference on Logic Programming*, pages 211–222, London, July 1986.

[19] S. Hanks and D. V. McDermott. Nonmonotonic logic and temporal projection. *Artificial Intelligence*, 33:379–412, 1987.

[20] M. Henrion. An introduction to algorithms for inference in belief nets. In M. Henrion, et al., editor, *Uncertainty in Artificial Intelligence 5*, pages 129–138. North Holland, 1990.

[21] J. R. Hobbs, M. E. Stickel, P. Martin, and D. Edwards. Interpretation as abduction. In *Proc. 26th Annual Meeting of the Association for Computational Linguistics*, pages 95–103, Buffalo, June 1988.

[22] M. Horsch and D. Poole. A dynamic approach to probabilistic inference using Bayesian networks. In *Proc. Sixth Conference on Uncertainty in AI*, pages 155–161, Boston, July 1990.

[23] F. V. Jensen, S. L. Lauritzen, and K. G. Olesen. Bayesian updating in causal probabilistic networks by local computations. *Computational Statistics Quaterly*, 4:269–282, 1990.

[24] K. Konolige. Abduction versus closure in causal theories. *Artificial Intelligence*, 53(2-3):255–272, February 1992.

[25] R. Kowalski. *Logic for Problem Solving*. Artificial Intelligence Series. North Holland, New York, 1979.

[26] K. B. Laskey and P. E. Lehner. Assumptions, beliefs and probabilities. *Artificial Intelligence*, 41(1):65–77, 1989.

[27] S. L. Lauritzen and D. J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society, Series B*, 50(2):157–224, 1988.

[28] R. S. Ledley and L. B. Lusted. Reasoning foundations of medical diagnosis. *Science*, 130(3366):9–21, July 1959.

[29] F. Lin and Y. Shoham. Argument systems: a uniform basis for nonmonotonic reasoning. In *Proc. First International Conf. on Principles of Knowledge Representation and Reasoning*, pages 245–255, Toronto, May 1989.

[30] J. W. Lloyd. *Foundations of Logic Programming*. Symbolic Computation Series. Springer-Verlag, Berlin, second edition, 1987.

[31] R. P. Loui. Defeat among arguments: a system for defeasible inference. *Computational Intelligence*, 3(2):100–106, May 1987.

[32] R. P. Loui. Defeasible decisions: what the proposal is and isn't. In M. Henrion et al., editor, *Uncertainty in Artificial Intelligence 5*, pages 99–116. North Holland, 1990.

[33] J. McCarthy and P. J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. In M. Meltzer and D. Michie, editors, *Machine Intelligence 4*, pages 463–502. Edinburgh University Press, 1969.

[34] E. M. Neufeld and D. Poole. Towards solving the multiple extension problem: combining defaults and probabilities. In *Proc. Third Workshop on Reasoning with Uncertainty*, pages 305–312, Seattle, July 1987.

[35] R. T. Ng and V. S. Subrahmanian. Non-monotonic negation in probabilistic deductive databases. In *Proc. Seventh Conf. on Uncertainty in Artificial Intelligence*, pages 249–256, Los Angeles, Cal., July 1991.

[36] R. T. Ng and V. S. Subrahmanian. Empirical probabilities in monadic deductive databases. In *Proc. Eighth Conf. on Uncertainty in Artificial Intelligence*, pages 215–222, Stanford, Cal., July 1992.

[37] J. Pearl. Distributed revision of composite beliefs. *Artificial Intelligence*, 33(2):173–215, October 1987.

[38] J. Pearl. Embracing causation in default reasoning. *Artificial Intelligence*, 35(2):259–271, 1988.

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

[40] Y. Peng and J. A. Reggia. *Abductive Inference Models for Diagnostic Problem-Solving.* Symbolic Computation – AI Series. Springer-Verlag, New York, 1990.

[41] J. L. Pollock. Defeasible reasoning. *Cognitive Science*, 11:481–518, 1987.

[42] D. Poole. A logical framework for default reasoning. *Artificial Intelligence*, 36(1):27–47, 1988.

[43] D. Poole. Representing knowledge for logic-based diagnosis. In *International Conference on Fifth Generation Computing Systems*, pages 1282–1290, Tokyo, Japan, November 1988.

[44] D. Poole. Explanation and prediction: an architecture for default and abductive reasoning. *Computational Intelligence*, 5(2):97–110, 1989.

[45] D. Poole. Normality and faults in logic-based diagnosis. In *Proc. 11th International Joint Conf. on Artificial Intelligence*, pages 1304–1310, Detroit, August 1989.

[46] D. Poole. A methodology for using a default and abductive reasoning system. *International Journal of Intelligent Systems*, 5(5):521–548, December 1990.

[47] D. Poole. Representing Bayesian networks within probabilistic Horn abduction. In *Proc. Seventh Conf. on Uncertainty in Artificial Intelligence*, pages 271–278, Los Angeles, July 1991.

[48] D. Poole. Representing diagnostic knowledge for probabilistic Horn abduction. In *Proc. 12th International Joint Conf. on Artificial Intelligence*, pages 1129–1135, Sydney, August 1991.

[49] D. Poole. Logic programming, abduction and probability. In *International Conference on Fifth Generation Computer Systems (FGCS-92)*, pages 530–538, Tokyo, June 1992.

[50] D. Poole. Search for computing posterior probabilities in Bayesian networks. Technical Report 92-24, Department of Computer Science, University of British Columbia, September 1992.

[51] D. Poole, R. Goebel, and R. Aleliunas. Theorist: A logical reasoning system for defaults and diagnosis. In N. Cercone and G. McCalla, editors, *The Knowledge Frontier: Essays in the Representation of Knowledge*, pages 331–352. Springer-Verlag, New York, NY, 1987.

[52] D. Poole and G. Provan. What is the most likely diagnosis? In P. P. Bonissone, M. Henrion, L. N. Kanal and J. F. Lemmer, editor, *Uncertainty in Artificial Intelligence 6*, pages 89–105. Elsevier Science Publishers B. V., 1991.

[53] H. E. Pople, Jr. On the mechanization of abductive logic. In *Proc. 3rd International Joint Conf. on Artificial Intelligence*, pages 147–152, Stanford, August 1973.

[54] G. Provan. An analysis of ATMS-based techniques for computing Dempster-Shafer belief functions. In *Proc. 11th International Joint Conf. on Artificial Intelligence*, pages 1115–1120, Detroit, August 1989.

[55] H. Reichenbach. *The Direction of Time*. University of California Press, Berkeley and Los Angeles, 1956.

[56] R. Reiter. Equality and domain closure in first order data bases. *J. ACM*, 27:235–249, 1980.

[57] R. Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32(1):57–95, April 1987.

[58] R. Reiter. The frame problem in the situation calculus: A simple solution (sometimes) and a completeness result for goal regression. In V. Lifschitz, editor, *Artificial Intelligence and the Mathematical Theory of Computation: Papers in Honor of John McCarthy*, pages 359–380. Academic Press, San Diego, California, 1991.

[59] R. Reiter and J. de Kleer. Foundations of assumption-based truth maintenance systems: preliminary report. In *Proc. 6th National Conference on Artificial Intelligence*, pages 183–188, Seattle, July 1987.

[60] R. Reiter and A. K. Mackworth. A logical framework for depiction and image interpretation. *Artificial Intelligence*, 41(2):125–155, 1989.

[61] R. D. Shachter and D. Heckerman. Thinking backwards for knowledge acquisition. *AI Magazine*, 8:55–62, 1987.

[62] M. Shanahan. Prediction is deduction, but explanation is abduction. In *Proc. 11th International Joint Conf. on Artificial Intelligence*, pages 1055–1060, Detroit, Mich., August 1989.

[63] S. E. Shimony and E. Charniak. A new algorithm for finding MAP assignments to belief networks. In *Proc. Sixth Conf. on Uncertainty in Artificial Intelligence*, pages 98–103, Cambridge, Mass., July 1990.

[64] L. Sterling and E. Shapiro. *The Art of Prolog.* MIT Press, Cambridge, MA, 1986.

[65] M. H. van Emden. Quantitative deduction and its fixpoint theory. *Journal of Logic Programming*, 4(1):37–53, 1986.