Logical and Probabilistic Reasoning to Support Information Analysis in Uncertain Domains

Marco Valtorta*
Department of Computer Science and Engineering
University of South Carolina
Columbia, SC 29208 USA

John Byrnes
HNC Fair Isaac
San Diego, CA 92130 USA

Michael Huhns
Department of Computer Science and Engineering
University of South Carolina
Columbia, SC 29208 USA

July 7, 2007

Abstract

Formal logical tools are able to provide some amount of reasoning support for information analysis, but are unable to represent uncertainty. Bayesian network tools represent probabilistic and causal information, but in the worst case scale as poorly as some formal logical systems and require specialized expertise to use effectively. We describe a framework for systems that incorporate the advantages of both Bayesian and logical systems. We define a formalism for the conversion of automatically generated natural deduction proof trees into Bayesian networks. We then demonstrate that the merging of such networks with domain-specific causal models forms a consistent Bayesian network with correct values for the formulas derived in the proof. In particular, we show that hard evidential updates in which the premises of a proof are found to be true force the conclusions of the proof to be true with probability one, regardless of any dependencies and prior probability values assumed for the causal model. We provide several examples that demonstrate the generality of the natural deduction system by using inference schemas not supportable in Prolog.

*Corresponding Author, mgv@cse.sc.edu

Keywords: Reasoning; Uncertainty; Probabilistic Reasoning; Bayesian Networks; Natural Deduction Proofs, Logic for Knowledge Representation

1 Background and Motivation

Support systems for information analysis must be able to quantify and track uncertainty in evidence findings, in data used by inferential processes, in the imperfect theories that emerge from the individual and collective experience of information analysts, and from other sources. Although they enjoy certain advantages in versatility and computational complexity, logical knowledge bases are ill-suited to represent uncertainty and then reason about it correctly, because knowledge representation languages based on classical logic do not provide facilities for representing and reasoning about uncertainty expressed in a probabilistic form. Bayesian probability theory defines the unique paradox-free method for reasoning with uncertainty. Recent research work shows that, in principle, facilities for representing and reasoning about uncertain information can be provided by extending the logical framework to support such representations as multiple-entity Bayesian networks and probabilistic relational models, but the scalability of such approaches is questionable.

We have been working on overcoming this problem in three ways. First, to simplify the construction and application of probabilistic models of situations of interest to an information analyst, we have implemented a simple version of Laskey and Mahoney's Bayesian network (BN) fragments approach [1]. A key feature of BN fragments is the distinction between nodes, which are in one-to-one correspondence with the nodes of a Bayesian network, and attributes of the nodes, which are used in the matching and composition process, as we described in previously published work [2, 3]. Second, we have developed an ontology of concepts that the designer of the decision support system can use in describing the nodes and attributes of Bayesian network fragments, which enables disciplined reuse and sharing of BN fragments. Third, as we report herein, we are developing approximate methods that automatically convert logical proofs into Bayesian networks. A proof is derived (in our work, using a natural-deduction format) from the application of a logical knowledge base to a particular situation. The Bayesian network can then be used to reason about the uncertainty of data sources, the uncertainty associated with expert judgment, conflicting data, and conflicting judgments. Conflicting data will be a major issue as larger knowledge bases are used, and particularly as more of their content is extracted automatically from text, because most logic engines fail catastrophically upon encountering a contradiction. Our major scientific hypothesis is that this kind of integration of proofs and Bayesian networks will provide the main advantages of a full integration of logical knowledge bases with Bayesian networks, while keeping computational complexity sufficiently low for practical use.

2 Logical and Probabilistic Models

Our objective is to produce models of systems and situations that will be sufficiently accurate that they can be used—where appropriate—to predict future states, to understand operations, to illuminate the factors relevant to decisions, and to control behaviors. We have realized that some knowledge is more easily and naturally represented in the form of statements in a logic language and some is more naturally represented in a Bayesian network formalism. We would like to take advantage of the strengths of each formalism while combining them into a single coherent system. However, there are tradeoffs in how the two are combined. The tradeoffs are as follows:

- First, we can extend a logic formalism (in this case a natural-deduction proof system) to include causality. This can be done by using special statements with associated conditional probabilities, for example, "coffee keeps me awake"

$$\text{coffee} \mapsto \text{awake}, \text{ where } P(\text{awake} \mid \text{coffee}) = 0.8 \tag{1}$$

  The problem is that if there are several statements about the causes for the same concept (e.g., tea also keeps me awake), then the representation may mislead a modeler into assuming that it is possible to specify the whole conditional probability of the effect given the causes by providing only marginal conditional probabilities, without requiring assumptions such as independence of causal influence. In other words, it is difficult to get the probabilities correct, because each parameter in the special formalism just described (with the $\mapsto$ symbol) represents only a partial (marginal) specification of a large conditional distribution, which is not specified, and for which the number of independent parameters is (approximately) the same as the number of configurations of the possible causes.

- Second, we can try to include logic statements directly within a Bayesian network. This is problematic in the case of a large theory, even in the propositional case, because it requires the modeler to reconstruct proofs, which are best carried out by an automated theorem prover. It is especially confusing for a probabilistic modeler to deal with proofs that go beyond what can be represented by simple rules (definite Horn clauses). A probabilistic modeler knows well that $P(A \rightarrow B) = m$ (which is equivalent to $P(\neg A \vee B) = m$) is *not* equivalent to $P(B = \mathsf{true} | A = \mathsf{true}) = m$, but might need help (from an automated system or a logical modeler) to carry through complicated proofs.

For the above reasons, we choose to pursue an integrated approach in which models are constructed from logical and probabilistic specifications, rather than by adding features to one of the two approaches.

3 Alternative Approaches

We have surveyed several proposed software systems (see the probabilistic-logical model repository at http://www.informatik.uni-freiburg.de/~kersting/plmr/ for references to most of them):

- Primula (Relational Bayesian Networks)
- SamIAM, Hugin, Netica, JavaBayes
- CILOG-2
- MEBN
- PRM
- DAPER
- OOBNs (Hugin)
- Prolog (CILOG-1)
- Relational Markov networks (RMNs)
- Markov Logic Networks (MLNs)

While many criteria can be used to partition the space of probabilistic-logical systems in different ways, we concentrate on an apparent dichotomy of goals in the merger of logic and probability in existing and prospective reasoning systems. The first goal is to allow uncertain knowledge to be brought to bear. This knowledge may be in the form of axioms with exceptions that can be properly summarized probabilistically, evidence that comes from unreliable observations ("virtual evidence"), or evidence that is intrinsically uncertain ("soft evidence") [4, 5, 6]. A typical system that emphasizes this goal is David Poole's CILOG-2, an extension to Horn Prolog that allows the association of probabilities to axioms [7, 8]. We categorize systems that address this goal as class A systems. The second goal is to leverage large amounts of information through statistical summarization. This approach makes it possible to describe a domain by separating a compact logical theory (set of axioms) from information represented in a relational database. A typical system that emphasizes this goal is the Probabilistic Relational Modeling part of the Primula system, which models separately the probabilistic and the relational specifications of a domain. We categorize systems that address this goal as class B systems. It is our contention that Class A systems are better suited to support the typical tasks of information analysis, such as the evaluation of hypotheses, the collection of data driven by value-of-information, and the analysis of robustness with respect to data and parametric assumptions.

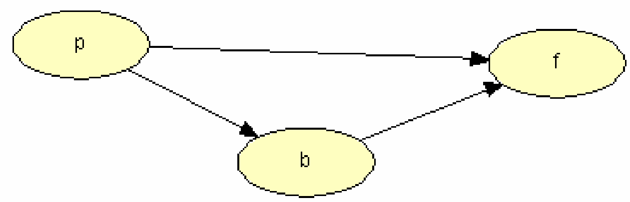

Figure 1: The classic "penguin triangle" Bayesian network

4 Expressing Knowledge Using Logical Theories and Probabilistic Networks

To clarify our approach, we begin with a simple example. Pearl [9] presents the following "probabilistic knowledge base" (PKB):

English Assertion	Numeric Confidence	Logical Representation	Probabilistic Formalization
All birds fly	m_1	$b \rightarrow f$	$P(f\|b) = m_1$
All penguins are birds	m_2	$p \rightarrow b$	$P(b\|p) = m_2$
Penguins do not fly	m_3	$p \rightarrow \neg f$	$P(f\|p) = 1 - m_3$

A user faced with a penguin and a pure logic tool that interprets "→" as material implication reaches a contradiction. The goal instead is to provide the user with a reasonable BN formalization of the problem; in this case one is presented in figure 1. Note that seven independent parameters are necessary to fully specify the joint probability distribution of the three binary variables p (penguin), b (bird), and f (flies). Typically, one would specify $P(p) = P(p = true)$ (the prior probability of being a penguin), which is irrelevant when the triangle is used to determine whether a penguin flies, $P(b \mid p) = m_2$, but also $P(b \mid \neg p)$ and the sensitivity and selectivity of flying with respect to being both a bird and a penguin.

$$\dfrac{\dfrac{p \to \neg f \qquad p}{\neg f} \qquad \dfrac{b \to f \qquad \dfrac{p \to b \qquad p}{b}}{f}}{\bot}$$

Figure 2: A natural deduction proof that the penguin triangle axioms are inconsistent

Pearl [9] describes a more logic-oriented approach to the penguin triangle as specifying "uncertain compatibility constraints." We make the approach explicit and refine it in the following way. First, we show how a contradiction is obtained by providing a natural deduction proof of the bottom symbol from the (non-logical) axioms, as in Figure 2. Second, we translate the axioms into a Bayesian network, resulting in the network of Figure 3, where the axioms become compatibility relations for the subformulas contained in them, thus constraining the satisfying interpretations of the propositional variables p (penguin), b (bird), and f (flies). The translation does not reflect the proof structure, such as the one described by Williamson in [10], but it has the advantage of encoding a correct independence structure, since each formula induces a dependency on its components, thus constraining the allowable interpretations of the axioms when they are known to hold.

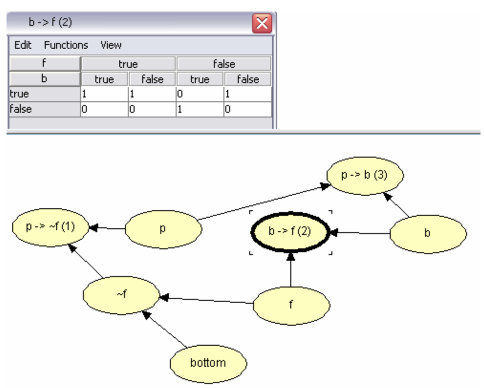

Figure 3: Bayesian network representing the penguin triangle proof

Pearl shows that the use of material implication instead of a conditional probability leads to erroneous results [9, Section 9.1]. We do not claim that Pearl's analysis is the end of the story. Consider again the Bayesian network in Figure 3. It is easy to show that, if penguins are always birds, then there is no possible configuration of b such that p is true, other than the one in which b is true. Moreover, if penguins are always birds, every time the rule "birds fly" is in force, we have an exception to the rule "penguins do not fly." So, the probability of a penguin flying is the same as the probability of a penguin-bird flying, thus contradicting the conclusion of Pearl's analysis. A numerical illustration, where the exceptions for the "birds fly" rule is 2%, is provided in Figure 4, where the update is carried out using BRUSE, a system under development and derived from BC-Hugin [4, 11]. The discrepancy is explained by recognizing that Pearl assumes that rules have independent exceptions, in the spirit of the Dempster-Shafer theory of evidence, while this is not the case in BRUSE.

Still, the pitfalls involved in using material implication in place of probabilistic ccnditionals, especially when dealing with causal information, are such that we take this as a good reason to provide a hybrid reasoning system in which causal models are properly modeled using Bayesian networks, rather than using material implications and the uncertain compatibility constraint model.

5 An Architecture for a Logical and Probabilistic Reasoning System

We intend that our framework for integrated logic and probabilistic reasoning be applicable to extremely large knowledge bases, even though probabilistic reasoning over complete such KBs will be infeasible. This is to be accomplished by using automated theorem provers to extract appropriate pieces,providing for the automated construction of significantly smaller Bayesian network fragments, and through the automated composition of such networks [3, 1].

Our initial reasoning architecture is described in Figure 5. We will illustrate the components of the architecture and their interaction on a simple example in the next section.

6 An Extended Example

We provide an extended example of using the integrated logical and probabilistic reasoning system. Since the propositional theory that formalizes the example includes at least one non-Horn clause, i.e., at least one clause that includes two non-negative literals, the theory cannot be handled correctly by Prolog or by forward chaining rule-based systems such as JESS or CLIPS. The example formalizes the following story: my cup contains either coffee (C) or

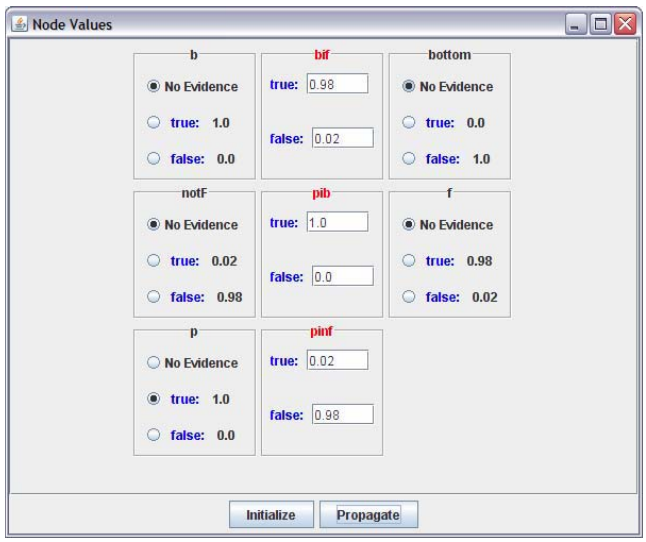

Figure 4: Soft evidential update for the penguin triangle

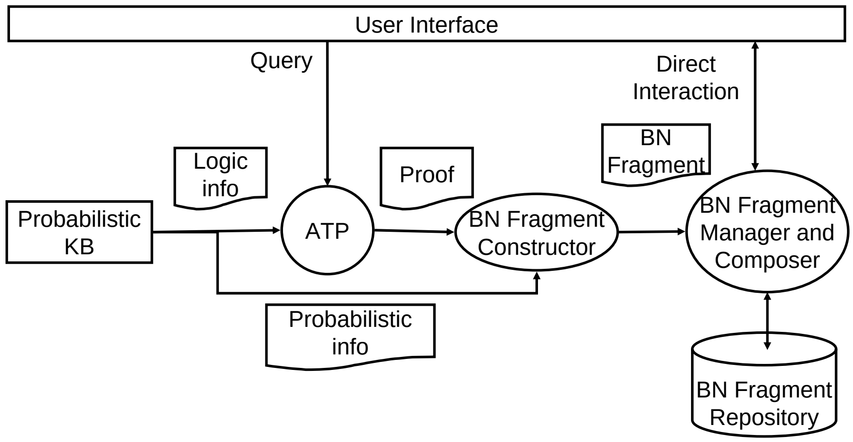

Figure 5: A basic architecture for integrated logic and probability reasoning

$$\dfrac{C \vee T \qquad \dfrac{C \rightarrow B \qquad C^{(1)}}{B} \qquad \dfrac{T \rightarrow B \qquad T^{(2)}}{B}}{B}\ (1,2)$$

Figure 6: A natural deduction proof for brown liquids

tea (T). Coffee is a brown liquid (B). Tea is a brown liquid. Thus it can be concluded that my cup contains a brown liquid.

The axioms in the knowledge base that formalizes the story are:

English Assertion	Logical Representation
My cup contains either coffee or tea	$C \vee T$
Coffee is a brown liquid	$C \rightarrow B$
Tea is a brown liquid	$T \rightarrow B$

We want to show B. Note that the theory allows for both tea and coffee to be in my cup at the same time. A natural deduction proof for B is given in Figure 6. The proof consists of three steps: two $\rightarrow$-elimination steps and one $\vee$-elimination step. The $\rightarrow$-elimination steps require one assumption each, namely C and T. The $\vee$-elimination step, which corresponds to a case analysis step, discharges the assumptions made in the $\rightarrow$-elimination steps.

An issue that had to be resolved is that of representing the proof in a con-

$$\dfrac{\Gamma \vdash C \vee T \qquad \dfrac{\Gamma \vdash C \to B \qquad \Gamma, C \vdash C}{\Gamma, C \vdash B}\,\to\text{-Elim} \qquad \dfrac{\Gamma \vdash T \to B \qquad \Gamma, T \vdash T}{\Gamma, T \vdash B}\,\to\text{-Elim}}{\Gamma \vdash B}\,\vee\text{-Elim}$$

Figure 7: This proof tree makes contexts explicit

venient machine-readable form. Due to the prevalence of XML, we decided to use a variation of the XML format used in the Vampire theorem prover [12]. Since Vampire is a resolution theorem prover, while we use natural deduction, we modified the schema by allowing for an explicit representation of the rule used and of the *context*, defined as the set of assumptions, used in a proof step.

Figure 7 presents the same proof as Figure 6, but in a way that emphasizes the contexts used. The proof in Figure 7, which includes the $\vdash$ symbol, will remind some readers of the sequent calculus. However, it is directly a natural deduction proof with the exact same structure as the proof in 6; it only uses a different syntax to denote active assumptions.

We also decided to use the IKL Lisp-like language to represent formulas within a proof [13, 14, 15]. The XML document representing the brown liquids proof contains a list of contexts and the proof steps, which are both numbered and refer to the contexts, as shown in Figure 8. Note that the IKL syntax requires what seems to be, in the propositional case, superfluous parentheses. For readability, we drop the parentheses in the following discussion and in the accompanying figures.

The natural deduction proof is converted to a Bayesian network in the following way. Each non-atomic formula used in the proof, is the child of its component subformulas, with a conditional probability table (CPT) that encodes the main connective introduced or eliminated. For example, in Figure 9, the (nodes corresponding to the) atomic formulas C and T are parents of the (node corresponding to the) formula $(\text{or } C\ T)$, and the CPT for the family of those three nodes, $P((\text{or } C\ T) \mid C,\ T)$ is an OR table. The Bayesian network also represents the nonempty contexts (sets of assumptions) used in the proof. For example, formula C is the context for the first step of the proof, namely the implication elimination with premises C and $(\text{if } C\ B)$ and conclusion B. Accordingly, the node corresponding to formula C is a parent of the node $Context1$ in the Bayesian network. In the CPT for a context node, the context is true if and only if all of its parents are true. As an illustration, consider the Bayesian network structure of Figure 9.

The construction algorithm just outlined ensures that any possible (i.e., non-zero probability) configuration (i.e., assignment of truth or false values) of the variables in the Bayesian network that correspond to formulas is a true interpretation (a model) of the formulas that appear in the steps of the proof and that no other assignments have positive probability, when the value true is entered as evidence for the (nodes corresponding to the) formulas of the theory. Figure 10 illustrates this, where it is shown that the only state of positive

```xml
<contexts>
        <context id='1'>
                <formula>
                        (C)
                </formula>
        </context>

        <context id='2'>
                <formula>
                        (T)
                </formula>
        </context>
</contexts>

<proofSteps>
        <proofStep id='1'>
                <rule>
                        if E
                </rule>

                <premises>
                        <formula contextId='1'>
                                (C)
                        </formula>
                        <formula>
                                (if (C) (B))
                        </formula>
                </premises>
                <conclusion>
                        <formula>
                                (B)
                        </formula>
                </conclusion>
        </proofStep>
...
```

Figure 8: A natural deduction proof in XML format

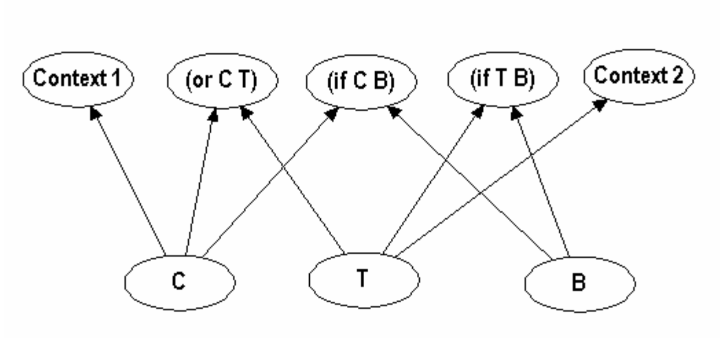

Figure 9: The Bayesian network representation of the brown liquids proof

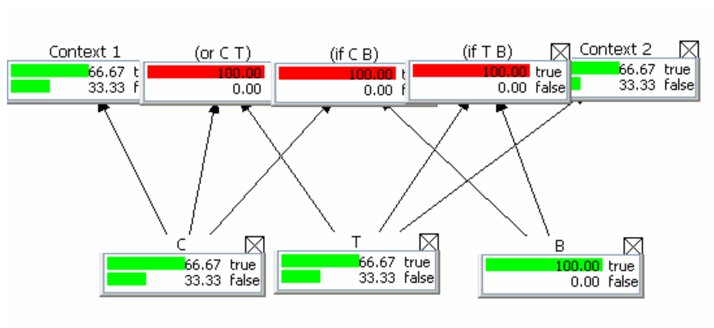

Figure 10: B logically follows from the axioms in the brown liquids domain

probability of the B variable is the one in which B is true, when evidence is entered for (if T B), (if C B), and (or C T). (Evidence entered is indicated by red bars in a color version of the figure.) Moreover, for a particular set of contexts, the possible configurations are models of the assumptions in the contexts and of the formulas.

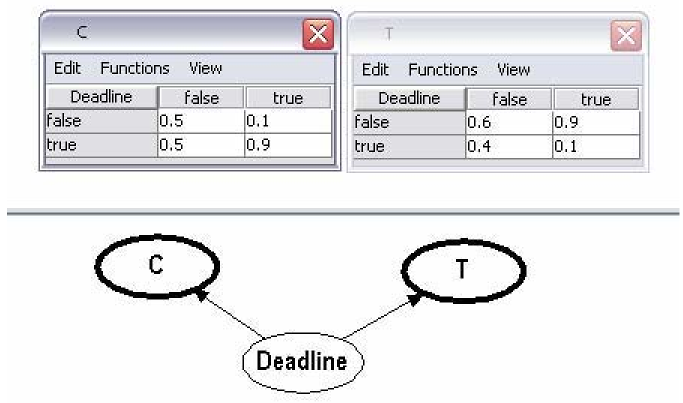

Figure 11: A probabilistic causal model that relates work deadlines to coffee and tea in my cup

Now, imagine that we have probabilistic information relating some of the variables in our domain of interest. In particular, following our example, imagine a probabilistic causal model is available that relates the presence of tea or coffee in my cup to the amount of work I need to get done before the end of the workday, as described in Figure 11. We can now compose the logically derived model of Figure 9 and the probabilistic causal model of Figure 11 into a single model using the Bayesian network fragment composition algorithm described in [3] and obtain the combined model of Figure 12.

The combined model is a Bayesian network and can be subjected to processing as any such network. The most important kind of processing is to compute the posterior probability of each variable in the network given a set of findings (i.e., evidence). For example, we may be interested in the probability of a

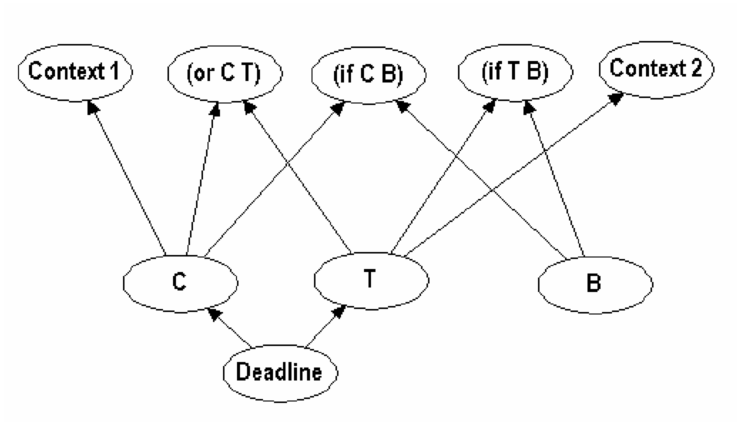

Figure 12: A model composed from logical and probabilistic components

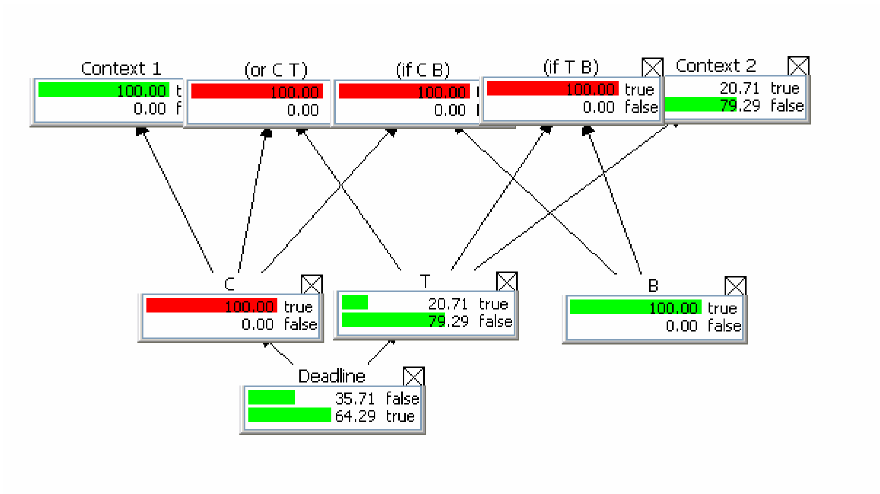

Figure 13: Probability update in the model of the previous figure

deadline given that we observe coffee in my cup and that all axioms hold (to meet deadlines we work late and consume coffee to stay awake). The posterior probabilities, computed using the commercial Bayesian network shell Hugin (`www.hugin.com`), are shown in Figure 13, where we observe a roughly 64% probability of my working on a deadline, which happens to be quite a bit higher than the baseline in the model.

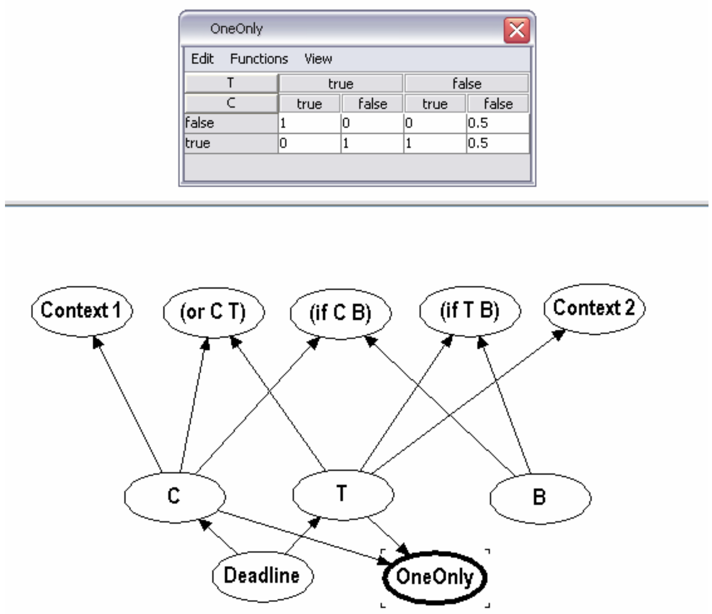

Figure 14: Coffee and tea may not be together in the cup

We also want to allow probability update in the presence of information about the probability of the formulas in the network. For this purpose, we use BRUSE, a refinement of BC-Hugin [11], a shell that allows the specification of evidence in the form of a set of findings, where each finding is a marginal probability on a variable in the network. In this way, one can specify the probability of a formula holding in the network. BRUSE computes rather efficiently the posterior distribution of the variables in the network with the following properties: the distribution is the closest one (according to cross-entropy) to the original one for which (1) all findings hold, and (2) all d-separation conditions hold [16, 4]. Suppose that, in our example, we add the information that my cup may not contain both coffee and tea at the same time. For simplicity, rather than expressing this constraint as a logical axiom ($\neg(C \wedge T)$) and con-

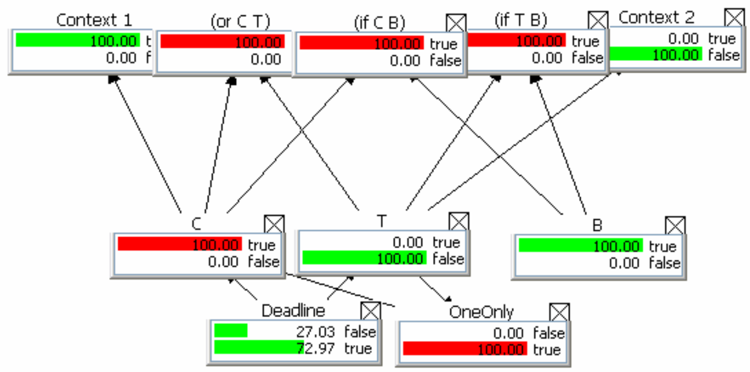

Figure 15: Probability update in the model of the previous figure

verting it into a Bayesian network, we encode this information directly in the Bayesian network, as shown in Figure 14. Comparing Figure 15 with Figure 13 shows that that the probability of working under a deadline given that there is coffee in my cup and all axioms hold is about 10% higher than before adding the constraint. Moreover, assume that there are exceptions to this rule. Figure 16 shows the result of running BRUSE on the network, with the exceptions to the rule quantified at 10%.

7 A Second Example

The second example formalizes the following story: I have a swimming pool (A). If I have a swimming pool and it does not rain (D), I will go swimming (B). If I go swimming, I will get wet (C). It can thus be concluded that I will get wet. This example is due to Loveland and Stickel [17], who use it to show that goal trees are incomplete. They also use the example to motivate the use of ancestor contradiction checks, which they show to be complete. The propositional theory that formalizes the example include at least one non-Horn clause, the theory cannot be handled correctly by Prolog or by forward chaining rule-based systems such as JESS or CLIPS.

The axioms in the knowledge base that formalizes the story are:

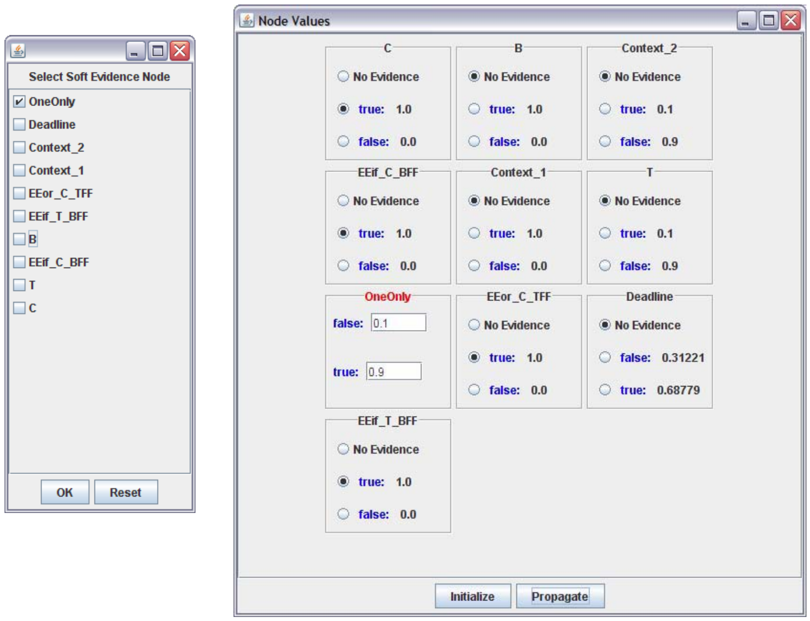

Figure 16: Soft evidential update with a 10% exception rate for the constraint that only one drink may be present in the cup

$$\dfrac{D \vee \neg D \qquad \dfrac{B \to C \qquad \dfrac{(A \wedge \neg D) \to B \qquad \dfrac{A \qquad \neg D^{(1)}}{A \wedge \neg D}}{B}}{C} \qquad \dfrac{D \to C \qquad D^{(2)}}{C}}{C}\ (1,2)$$

Figure 17: A natural deduction proof for Loveland and Stickel's swimming pool example

English Assertion	Logical Representation
I have a swimming pool	A
If I have a swimming pool and it does not rain, I will go swimming	$(A \wedge \neg D) \to B$
If I go swimming, I will get wet	$B \to C$
If it rains, I will get wet	$D \to C$

We want to show C.

The second axiom in the propositional theory that formalizes the example has a negation in its antecendent, and it is therefore not a Horn clause. Therefore, the theory cannot be handled correctly by Prolog or by forward chaining rule-based systems, such as JESS or CLIPS.

A natural deduction proof for C is given in Figure 17, where we omitted the proof of $D \vee \neg D$. This proof consists of five steps: one $\wedge$-introduction step, three $\to$-elimination steps, and one $\vee$-elimination step. The $\wedge$-introduction step and the last $\to$-elimination step require one assumption each, namely $\neg D$ and D. The $\vee$-elimination step, which corresponds to a case analysis step, discharges both assumptions.

The Bayesian network representation of the proof is given in Figure 18. We adopt the usual convention that $\neg D$ is a shorthand for $D \to \bot$. For simplicity, we omitted the node representing $\bot$ and the edge from it to $\neg D$. Figure 19 shows the configuration of the Bayesian network variables when the evidence indicating that the four axioms are true is entered; note that C is true when the axioms are true.

8 Correctness of the Logical Model

We want to demonstrate that models constructed as described above behave according to our logical intuitions. We consider an arbitrary "mixed" model of the sort in Figure 12.

Note that the *roots* (nodes without parents) of the proof BN need not be atomic formulas, and not all subformulas of a logical node formula need to be present in the proof BN. For example, we might have proved $R \wedge Q$ from $(R \wedge Q) \wedge S$ without introducing the atomic formulas R and Q. In this case, S does not appear in the proof BN. Although it is not necessary to do so, it

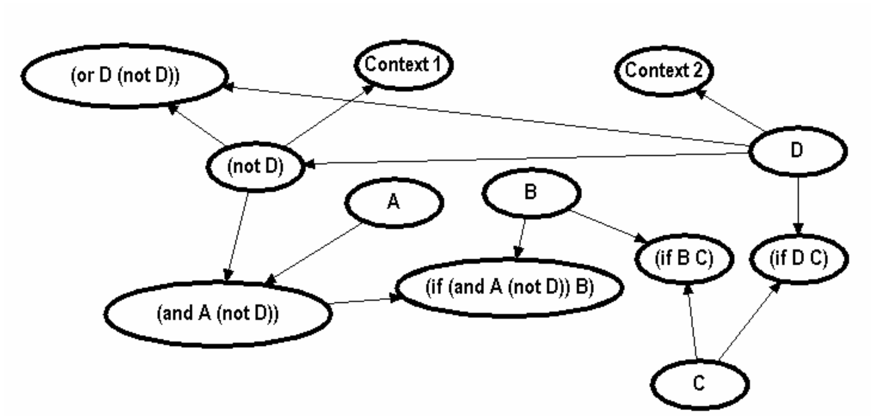

Figure 18: The Bayesian network representation of the swimming pool proof

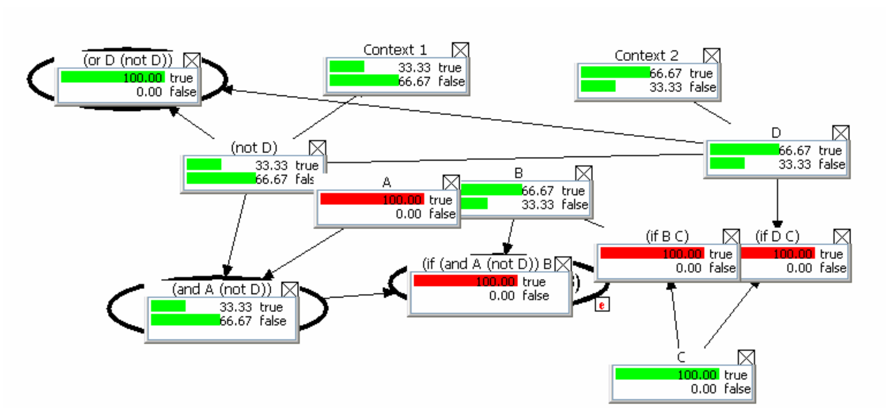

Figure 19: C logically follows from the axioms in the swimming pool domain

is convenient for this presentation to extend the proof BN so that all complex formulas that are not roots have exactly two parents. Also, we define $\neg A$ to be an abbreviation for $A \rightarrow \bot$.

We permit the causal BN to introduce arbitrary dependencies between the root nodes of the logical BN. We make the restriction that nodes of the causal BN that are not shared with the proof BN may not be part of the language of the proof BN. The proof BN just described, for example, could not be joined with a causal BN that refers to R, because R is in the proof BN but would not be properly linked. This can always be overcome by extending the proof BN so that it does contain R before joining with the causal BN.

As described in section 6, the general claim that we wish to prove is that for an arbitrary probability distribution P over the language of the proof and for an arbitrary formula G occurring in the proof dependent on assumptions $A_1, \ldots, A_n$, $P(G = \mathsf{true} \mid A_1 = \mathsf{true}, \ldots, A_n = \mathsf{true}) = 1$. For simplicity, we will use $P(a_i)$ to abbreviate $P(A_i = \mathsf{true})$ and $P(\bar{a_i})$ to abbreviate $P(A_i = \mathsf{false})$. Let $\mathcal{L}$ represent the set of all nodes of the causal BN and all roots of the proof BN. We let P be an arbitrary probability distribution over $\mathcal{L}$ satisfying the independence relations induced by the DAG, with the restriction that $P(\bot) = 0$. Our first step will be to extend P to all logical formulas over $\mathcal{L}$.

Let $\mathbf{v}$ be an arbitrary function from $\mathcal{L}$ to $\{\mathsf{true}, \mathsf{false}\}$. For an arbitrary complex formula A, we define $\mathbf{v}[A]$ recursively on the structure of A by case depending on the main connective of A. If $A \equiv B \wedge C$ then we define $\mathbf{v}[A] = \mathsf{true}$ if $\mathbf{v}[B] = \mathsf{true}$ and $\mathbf{v}[C] = \mathsf{true}$; $\mathbf{v}[A] = \mathsf{false}$ otherwise. Similarly, $\mathbf{v}[B \vee C] = \mathsf{true}$ iff either of $\mathbf{v}[B]$ or $\mathbf{v}[C]$ are true, $\mathbf{v}[B \rightarrow C] = \mathsf{true}$ iff $\mathbf{v}[B] = \mathsf{false}$ or $\mathbf{v}[C] = \mathsf{true}$, and $\mathbf{v}[\bot] = \mathsf{false}$. When $\mathbf{v}[A] = \mathsf{true}$ we write $\mathbf{v} \models A$ and otherwise we write $\mathbf{v} \not\models A$. $\mathbf{v}$ is called a *valuation* over $\mathcal{L}$.

Using the set of all valuations over $\mathcal{L}$ as a sample space with all possible subsets as events, we can define a probability distribution P' such that, for any $A \in \mathcal{L}$, $P'(\{\mathbf{v} \mid \mathbf{v} \models A\}) = P(a)$. Since P is given for all $A \in \mathcal{L}$, P uniquely determines P' over any subset of valuations. We now use P to denote P', since this usage is unambiguous. Note in particular that $P(\bot) = P(\{\mathbf{v} \mid \mathbf{v} \models \bot\}) = P(\emptyset) = 0$ as required.

Since the composite DAG is, in fact, a BN, and since P is given over the parents of all logical nodes, the full distribution $\hat{P}$ over the entire DAG is uniquely determined. Since $\hat{P}$ extends P, we again use P to denote it. We next show that for every node A in the BN, $P(a) = P(\{\mathbf{v} \mid \mathbf{v} \models A\})$, by induction over the complexity of A. Complexity here means the depth of the parse tree of the formula relative to $\mathcal{L}$ (for example, the complexity of $B \wedge C$ is one greater than the maximum complexity of B or C), where all formulas in $\mathcal{L}$ are given complexity 1 regardless of their structure.

We have established the claim already for $A \in \mathcal{L}$, satisfying the base case, in which the complexity of A is 1. For the inductive step we suppose that A is of higher complexity. Suppose $A \equiv B \wedge C$. A must have parents in the proof BN in order that $A \notin \mathcal{L}$, so B and C are both nodes in the proof BN. By the inductive hypothesis, $P(B) = P(\{\mathbf{v} \mid \mathbf{v} \models B\})$ and $P(C) = P(\{\mathbf{v} \mid \mathbf{v} \models C\})$.

By the conditional probability table for $\wedge$,

$$\begin{aligned}
P(B \wedge C = \textsf{true}) &= P(b, c) \\
&= P(\{\mathbf{v} \mid \mathbf{v} \models B\}, \{\mathbf{v} \mid \mathbf{v} \models C\}) \\
&= P(\{\mathbf{v} \mid \mathbf{v} \models B\} \cap \{\mathbf{v} \mid \mathbf{v} \models C\}) \\
&= P(\{\mathbf{v} \mid \mathbf{v} \models B \text{ and } \mathbf{v} \models C\}) \\
&= P(\{\mathbf{v} \mid \mathbf{v} \models B \wedge C\})
\end{aligned}$$

If $A \equiv B \vee C$, then by the conditional probability table for $\vee$,

$$\begin{aligned}
P(B \vee C) &= P(b, c) + P(b, \bar{c}) + P(\bar{b}, c) \\
&= P(\{\mathbf{v} \mid \mathbf{v} \models B \text{ and } \mathbf{v} \models C\}) + P(\{\mathbf{v} \mid \mathbf{v} \models B \text{ and } \mathbf{v} \not\models C\}) \\
&\quad + P(\{\mathbf{v} \mid \mathbf{v} \not\models B \text{ and } \mathbf{v} \models C\}) \\
&= P(\{\mathbf{v} \mid \mathbf{v} \models B \cup \mathbf{v} \models C\}) \\
&= P(\{\mathbf{v} \mid \mathbf{v} \models B \vee C\})
\end{aligned}$$

The case in which $A \equiv B \rightarrow C$ is similar.

We have thus established that for every node A in the composite network, $P(A) = P(\{\mathbf{v} \mid \mathbf{v} \models A\})$, indicating that the semantics of our network is what we would expect. Next consider any formula occurrence G in the proof and let $\Gamma = \{A_1, \ldots, A_n\}$ be the set of all assumptions on which G depends. Our original goal was to show that $P(g \mid \Gamma) = 1$. Since $\Gamma \vdash G$, the soundness of the logical rules gives us that for every $\mathbf{v}$ such that $\mathbf{v} \models \Gamma$, $\mathbf{v} \models G$. Thus $\{\mathbf{v} \mid \mathbf{v} \models \Gamma, G\} = \{\mathbf{v} \mid \mathbf{v} \models \Gamma\}$. This allows us to reason:

$$\begin{aligned}
P(g \mid \Gamma) &= \frac{P(g, \Gamma)}{P(\Gamma)} \\
&= \frac{P(\{\mathbf{v} \mid \mathbf{v} \models \Gamma, G\})}{P(\{\mathbf{v} \mid \mathbf{v} \models \Gamma\})} \\
&= 1.0
\end{aligned}$$

The correctness proof demonstrates the desirable charactersitic that the conclusion of a proof is in state "true" with probability 1.0, conditional on the premises of the proof being the state "true". More generally, it demonstrates that any formula in the proof is true with probability 1.0, conditional on those premises on which it depends in the proof. One possible use of this is to observe that a given context node of the network is true; then every formula of the proof within that context will have value true as well.

Although we leave the "background context" Γ implicit when translating the proof in Figure 9, we could have introduced a node for Γ as well, with all premises of the proof pointing to that node. Making a hard evidential observation on that node (which is only true when *all* of its parents are) then forces the truth of all conclusions that occur within that context, just as in the other contexts. Making a soft evidential observation on that node can provide a way to quantify that we have a certain degree of trust for the information that comes from a particular knowledge base. If we are reasoning over knowledge from many sources, we can assign a different context node for each source and quantify the trustworthiness of each source in this way. Note that this cannot be accomplished by assigning the appropriate degree of belief to each premise individually, because doing this indicates that the premises succeed or fail independently rather than together. A conclusion that depends on a very large number of highly likely premises will mistakenly be given a lower probability than it would be given when the context is used to indicate the single degree of certainty for the set of premises.

Another property of the networks that is made apparent by the correctness proof is the desirable semantic interpretation that the probability for a given node is the total probability of the set of models of that node. This is similar to the technique presented in [18] and allows one to define arbitrary distributions over logical models, while maintaining consistency with the Bayesian networks derived from proofs.

In some cases it might be desirable to force all atomic formulas to be explicitly represented in proofs. This can be done straightforwardly by converting the natural deduction proof to its so-called *long $\beta\eta$-normal form*. Certain search strategies for natural deduction proving will automatically generate such proofs without the need for conversion [19].

9 Conclusions

In this paper, we have argued for an integrated approach to logical and probabilistic modeling, where natural deduction proofs obtained from an automated theorem prover are converted into Bayesian network fragments that are composed with other Bayesian network fragments that encode probabilistic causal models, to achieve truly integrated probabilistic-logical models. We describe briefly how to convert proofs into Bayesian network fragments and provide a proof of the correctness of the conversion. The modules described in the paper are still under development. In particular, the program that converts proofs to Bayesian networks, while not restricted to Horn clauses, supports only propositional theories, and extending it to first-order logic remains a major focus of our effort.

Acknowledgments

This work was funded in part by the Disruptive Technology Office Collaboration and Analyst System Effectiveness (CASE) Program, contract FA8750-06-C-0194 issued by Air Force Research Laboratory (AFRL). The views and conclusions are those of the authors, not of the US Government or its agencies. The contributions of Scott Langevin, Laura Zavala, Jingsong Wong, Jingshan Huang, and Dylan Kane are appreciated.

References

[1] K. B. Laskey and S. M. Mahoney, "Network fragments: Representing knowledge for constructing probabilistic models," in *Proceedings of the Thirteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI-97)*, (Providence, RH), pp. 334–341, August 1997.

[2] M. Valtorta, J. Dang, H. Goradia, J. Huang, and M. Huhns, "Extending Heuer's analysis of competing hypotheses method to support complex decision analysis," in *Proceedings of the 2005 International Conference on Intelligence Analysis (IA-05) (CD-ROM)*, 2005. Extended version available at http://www.cse.sc.edu/~mgv/reports/IA-05.pdf.

[3] J. Cheng, R. Emami, L. Kerschberg, J. E. Santos, Q. Zhao, H. Nguyen, H. Wang, M. Huhns, M. Valtorta, J. Dang, H. Goradia, J. Huang, and S. Xi, "Omniseer: A cognitive framework for user modeling, reuse of prior and tacit knowledge, and collaborative knowledge services," in *Proceedings of the 38th Hawaii International Conference on System Sciences (HICSS38)*, (Big Island, HI), January 2005.

[4] M. Valtorta, Y.-G. Kim, and J. Vomlel, "Soft evidential update for probabilistic multiagent systems," *International Journal of Approximate Reasoning*, vol. 29, pp. 71–106, January 2002.

[5] H. Chan and A. Darwiche, "On the revision of probabilistic beliefs using uncertain evidence," in *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, (Acapulco, Mexico), pp. 99–105, 2003.

[6] J. Vomlel, "Probabilistic reasoning with uncertain evidence," *Neural Network World, International Journal on Neural and Mass-Parallel Computing and Information Systems*, vol. 14, no. 5, pp. 453–456, 2004.

[7] D. Poole, *Computational Intelligence: A Logical Approach*. Oxford: Oxford University Press, 1998.

[8] D. Poole, "First-order probabilistic inference," in *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, (Acapulco, Mexico), p. to appear, 2003.

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

[10] J. Williamson, "Bayesian networks for logical reasoning," in *Proceedings of the AAAI Fall Symposium on Using Uncertainty in Computation*, pp. 136–164, AAAI-Press, October 2001.

[11] Y.-G. Kim, M. Valtorta, and J. Vomlel, "A prototypical system for soft evidential update," *Applied Intelligence*, vol. 21, July–August 2004.

[12] A. Riazanov and A. Voronkov, "The design and implementation of Vampire," *AI Communications*, vol. 15, pp. 91–110, 2002.

[13] P. J. Hayes, "IKL guide," tech. rep., Florida Institute for Human and Machine Cognition, 2006. Unpublished Memorandum available at http://www.ihmc.us/users/phayes/IKL/GUIDE/GUIDE.html.

[14] P. J. Hayes and C. Menzel, "IKL specification document," tech. rep., Florida Institute for Human and Machine Cognition, 2006. Unpublished Memorandum available at http://www.ihmc.us/users/phayes/IKL/GUIDE/GUIDE.htmIKL/SPEC/SPEC.html.

[15] Harry Delugach, Editor, "Common logic—a framework for a family of logic-based languages," tech. rep., International Standards Organization, 2005. ISO/IEC JTC 1/SC 32N1377, International Standards Organization Final Committee Draft, 2005-12-13, available at http://cl.tamu.edu/docs/cl/32N1377T-FCD24707.pdf.

[16] J. Vomlel, *Methods of Probabilistic Knowledge Integration*. PhD thesis, Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University, December 1999.

[17] D. W. Loveland and M. Stickel, "A hole in goal trees: Some guidance from resolution theory," *IEEE Transactions on Computers*, vol. 25, pp. 335–341, April 1976.

[18] J. Paris, *The Uncertain Reasoner's Companion: A mathematical perspective. Cambridge Tracts in Theoretical Computer Science 39*. Cambridge, England: Cambridge University Press, 1994.

[19] J. Byrnes, *Proof Search and Normal Forms in Natural Deduction*. PhD thesis, Department of Philosophy, Carnegie Mellon University, 1999.