MEBN: A Logic for Open-World Probabilistic Reasoning

Kathryn Blackmond Laskey KLASKEY@GMU.EDU
Department of Systems Engineering and Operations Research
MS4A6
George Mason University
Fairfax, VA 22030, USA

Abstract

Uncertainty is a fundamental and irreducible aspect of our knowledge about the world. Probability is the most well-understood and widely applied logic for computational scientific reasoning under uncertainty. As theory and practice advance, general-purpose languages are beginning to emerge for which the fundamental logical basis is probability. However, such languages have lacked a logical foundation that fully integrates classical first-order logic with probability theory. This paper presents such an integrated logical foundation. A formal specification is presented for multi-entity Bayesian networks (MEBN), a knowledge representation language based on directed graphical probability models. A proof is given that a probability distribution over interpretations of any consistent, finitely axiomatizable first-order theory can be defined using MEBN. A semantics based on random variables provides a logically coherent foundation for open world reasoning and a means of analyzing tradeoffs between accuracy and computation cost. Furthermore, the underlying Bayesian logic is inherently open, having the ability to absorb new facts about the world, incorporate them into existing theories, and/or modify theories in the light of evidence. Bayesian inference provides both a proof theory for combining prior knowledge with observations, and a learning theory for refining a representation as evidence accrues. The results of this paper provide a logical foundation for the rapidly evolving literature on first-order Bayesian knowledge representation, and point the way toward Bayesian languages suitable for general-purpose knowledge representation and computing. Because first-order Bayesian logic contains classical first-order logic as a deterministic subset, it is a natural candidate as a universal representation for integrating domain ontologies expressed in languages based on classical first-order logic or subsets thereof.

Keywords: Bayesian networks, Bayesian learning, graphical probability models, knowledge representation, multi-entity Bayesian network, random variable, probabilistic ontology

1 Introduction

First-order logic is primary among logical systems from both a theoretical and a practical standpoint. It has been proposed as a unifying logical foundation for defining extended logics and interchanging knowledge among applications written in different languages. However, its applicability has been limited by the lack of a coherent semantics for plausible reasoning. A theory in first-order logic assigns definite truth-values only to sentences that have the same truth-value (either true or false) in all interpretations of the theory. The most that can be said about any other sentence is that its truth-value is indeterminate. A reasoner that requires logical proof before it can draw conclusions is inadequate for many practical applications. This problem has been addressed with a proliferation of plausible reasoning logics, but these have lacked firm theoretical grounding. The need for plausible reasoning is especially acute for the problem of knowledge interchange. Different applications have different ontologies, different semantics, and different knowledge and data stores. Legacy applications are usually only partially documented, and may rely on tacit usage conventions that even proficient users do not fully understand or appreciate. Even if these problems could be circumvented and a full formal specification for each application could be achieved in first-order logic, the alignment of different applications into a single unified ontology, semantics, and data store is an ill-specified problem with no unique solution. This is a consequence of the fundamental truth that axiom sets in first-order logic do not in general admit unique interpretations. Because knowledge interchange is fraught with irreducible uncertainty, it should be founded on a logic that supports plausible inference.

Among the many proposed logics for plausible inference, probability is the strongest contender as a universal representation for translating among different plausible reasoning logics. There are numerous arguments in favor of probability as a rationally justified calculus for plausible inference under uncertainty (e.g., de Finetti, 1934/1975; Howson and Urbach, 1993, Jaynes, 2003; Savage, 1954). Until recently, the development of a fully general probabilistic logic was hindered by the lack of modularity of probabilistic reasoning, the intractability of worst-case probabilistic inference, and the difficulty of ensuring that a set of probability assignments specified a unique and well-defined probability distribution. Probability is not truth-functional. That is, the probability of a compound expression cannot be expressed solely as a function of the probabilities of its constituent expressions. The number of probabilities required to express a fully general probability distribution over truth-values of a collection of assertions is exponential in the number of assertions, making a brute-force approach to specification and inference infeasible for all but the smallest problems. Typically, independence assumptions are used to decompose complex problems into manageable sub-problems. Recently developed graphical probability languages (e.g., Jensen, 2001; Neapolitan, 2003; Pearl, 1988) exploit independence relationships to achieve parsimonious representation and efficient inference. The introduction of graphs to represent conditional dependence relationships has sparked rapid evolution of increasingly powerful languages for computational probabilistic reasoning (e.g., Buntine, 1994; D'Ambrosio, et al, 2001; Getoor et al, 2000, 2001; Gilks et al, 1994; Glesner and Koller, 1995; Halpern, 1991; Koller and Pfeffer, 1997; Laskey, et al, 2001; Laskey and Mahoney, 1997; Ngo and Haddawy, 1997; Pfeffer, 2001; Sato, 1998; et al., 1996). Different communities appear to be converging around certain fundamental approaches to representing uncertain information about the attributes, behavior, and interrelationships of structured entities (cf., Heckerman, et al., 2004).

This paper presents a logical foundation for this emerging consensus. First-order Bayesian logic combines the expressive power of first-order logic with a sound and logically consistent treatment of uncertainty. Multi-entity Bayesian networks (MEBN)[1] is a language for expressing first-order Bayesian theories. MEBN semantics unifies the standard model-theoretic semantics for first-order logic with the theory of random variables as used in mathematical statistics. Although MEBN syntax is designed to highlight the relationship between a MEBN theory and its first-order logic counterpart, the main focus of the paper is the underlying logic and not the language itself. That is, MEBN syntax should be viewed not as a competitor to other syntactic conventions for expressing first-order probabilistic knowledge, but as a vehicle for expressing logical notions that cut across surface syntactic differences.

MEBN fragments (MFrags) use directed graphs to specify local dependencies among a collection of related hypotheses. MTheories, or collections of MFrags that satisfy global consistency constraints, implicitly specify joint probability distributions over unbounded and possibly infinite numbers of hypotheses. MTheories can be used to reason consistently about complex expressions involving nested function application, arbitrary logical formulas, and quantification. A set of built-in MFrags provides the full expressive power of first-order logic with functions and equality, the most commonly used variant of first-order logic.

MEBN semantics assigns probabilities to sets of models of an associated first-order logic (FOL) theory. The probability of a sentence is defined as the probability of the set of models in which it is true. An important feature of the MEBN formalism is the ability to specify nested

[1] MEBN is pronounced "MEE-ben."

sequences of theories, in which each theory incorporates new axioms that do not contradict previously asserted axioms. The probability calculus provides an inference and learning theory for MTheories. An inference algorithm called *situation-specific Bayesian network* (*SSBN*) *construction* is presented. In response to a probabilistic query, SSBN construction produces a sequence of Bayesian networks that approximates the probability distribution implicitly represented by the MTheory. If the associated FOL theory is inconsistent, SSBN construction discovers the inconsistency in finitely many steps. For queries about consistent, finitely axiomatizable FOL theories, SSBN construction may terminate with an exact answer or may converge to the correct answer in the infinite limit. Theories with infinitely many axioms are represented as nested sequences of MTheories. Such an infinite sequence may or may not converge to a globally consistent joint distribution over interpretations, depending on whether the axioms define a generative process capable of representing the statistical behavior of the sequence. No probabilistic logic can do better than this. A construction due to Oakes (1986) demonstrates that for *any* generative probabilistic theory, no matter how expressive and flexible, there exist infinite sequences of findings that falsify the probabilistic predictions of the theory.

The remainder of the paper is organized as follows. Section 2 provides an overview of first-order logic and introduces notational conventions that will be used throughout the paper. Section 3 provides an overview of ordinary Bayesian networks, the propositional knowledge representation formalism for which MEBN is a first-order extension. Section 4 defines the syntax and semantics of MEBN logic and relates MEBN logic to other work in probabilistic knowledge representation. Section 5 sketches how MEBN logic can be applied to representing different kinds of knowledge. Section 6 describes learning and theory refinement, and demonstrates that learning is an integral part of MEBN logic. The final section is a summary and discussion. Proofs and algorithms are given in the appendix.

2 First-Order Logic

Davis (1990) defines a logic as a schema for defining languages to describe and reason about entities in different domains of application. Certain key issues in representation and inference arise across a variety of application domains. A logic encodes particular approaches to these issues in a form that can be reused across domains. A logic has the following basic elements (cf., Sowa, 2000):

- The *vocabulary* consists of symbols that can be combined to form expressions to represent and reason about entities in a given domain of discourse. Symbols are of two kinds:
  a. *Logical symbols* (e.g., variables, connectives, punctuation) are common to any language based on the logic;
  b. *Non-logical symbols* (e.g., constant symbols, function symbols, relation symbols) vary from language to language, and provide vocabulary tailored to a particular domain of application.
- The *syntax* consists of rules for combining these symbols to form legal expressions. The *proof rules* specify ways in which new legal expressions can be derived from existing legal expressions. The proof rules provide the *operational semantics* for computer languages that implement the logic.
- The *semantics* characterizes the meaning of expressions. Semantics includes two aspects:
  a. The *theory of reference* specifies what the expressions denote in the domain of discourse. The theory of reference corresponds to the *denotational semantics* of a computer language implementing the logic.
  b. The *model theory* specifies domain-independent aspects of meaning that are purely logical consequences of collections of expressions. The model theory establishes an isomorphism, or one-to-one meaning-preserving mapping, between different formally

equivalent collections of expressions, regardless of the domain of discourse to which each collection refers or the objects to which the expressions refer. The model theory corresponds to the *axiomatic semantics* of a computer language implementing the logic.

A *theory* is a collection of sentences in a given language[2], called the *proper axioms* of the theory, together with all the consequences of those sentences as determined by the semantics of the logic. In a computational theory, expressions are encoded as data structures on a computer and the proof rules are implemented as computer programs. To be useful for practical problems, a computational theory must be able to represent task-relevant aspects of the domain well enough for the purpose, and must admit implementations that quickly and accurately map expressions representing user queries to the logical consequences of the axioms with respect to the query.

A logic with *propositional* expressive power can reason about particular individuals but cannot express generalizations. A logic with *first-order* expressive power can reason about general properties and relationships that apply to collections of individuals. *Higher-order* logics can generalize not just over particular individuals in the application domain, but also over functions, relations, sets, and/or sentences defined on the domain. *Modal* logics allow reasoning not just about the truth-values of expressions, but also about necessity, possibility, belief, desirability, permissibility, and other non truth-functional qualifiers of statements. The greater expressive power of higher-order and modal logics allows one to say complex things more compactly, but tends to complicate proof and model theories.

By far the most commonly used, studied, and implemented logical system is first-order logic (FOL), invented independently by Frege and Pierce in the late nineteenth century (Frege, 1879/1967; Pierce, 1898). The notational conventions of this paper are similar to those used in standard references (e.g., Davis, 1990; Genesereth and Nielsson, 1987; Russell and Norvig, 2002; Sowa, 2000). The basic syntax of first-order logic can be summarized as follows:

- The *logical symbols* consist of the logical connectives $\neg$ (not), $\wedge$ (and), $\vee$ (or), $\Rightarrow$ (implies), and $\Leftrightarrow$ (if and only if); the equality relation =; the universal and existential quantifiers $\forall$ and $\exists$;[3] the comma, the open and close parentheses, and a countably infinite collection of variable symbols. Variables are denoted as alphanumeric strings beginning with lowercase letters, e.g., *x*, *person*32, *something*.[4]
- The *nonlogical symbols* consist of constant symbols, function symbols, and predicate symbols. Constant symbols are written as alphanumeric strings beginning with either numbers or uppercase letters, e.g., 1978; *Marcus*, *Machine*37. Function and predicate symbols are denoted as alphanumeric strings beginning with uppercase letters, e.g., *Red*, *BrotherOf*, *StandardDeviation*. Each function and predicate symbol has an associated integer indicating the number of arguments it takes.
- A *term* is a constant symbol, a variable symbol, or a function symbol followed by a parenthesized list of terms separated by commas, e.g., *Machine*37, *m*, *RoomTemp*(*MachineLocation*(*m*)), *Manager*(*Maintenance*,2003). Terms are used to refer to entities in the domain. They serve as arguments to functions and predicates.
- An *atomic formula* is:
  - A predicate symbol followed by a parenthesized list of terms, e.g., *Warmer*(*MachineLocation*(*m*),30,*Celsius*); or

[2] Sentences are legal expressions that make assertions about the domain.

[3] A formal specification of first-order logic requires only two connectives and one quantifier (e.g., $\neg$, $\Rightarrow$, and $\exists$); the others can be defined from these.

[4] Although words are often used to convey intended meaning, the variable, function and predicate symbols are treated by the logic as meaningless tokens. A theory may contain axioms that enforce intended meanings, but there is nothing in the logic itself to prevent *person*32 from being used to refer to a frog or an asteroid.

  - A parenthesized expression consisting of a term followed by an equal sign followed by another term, e.g., (*Fernandez* = *Manager*(*Maintenance*,2003)).
- A formula is:
  - An atomic formula;
  - An expression of the form $\neg\alpha$, $(\alpha\wedge\beta)$; $(\alpha\vee\beta)$; $(\alpha\Rightarrow\beta)$, or $(\alpha\Leftrightarrow\beta)$, where α and β are formulas, e.g.,

    ((*Fernandez* = *Manager*(*Maintenance*,2003))

    $\vee$ (*Nguyen* = *Manager*(*Maintenance*,2003))); or
  - An expression of the form $\forall\mu\alpha$ or $\exists\mu\alpha$, where μ is a variable symbol and α is a formula, e.g. $\exists x$ (*Employee*(*x*) $\wedge$ (*x* = *Manager*(*department*,*year*))).
- An *open formula* is a formula in which some variables are *free*, or not within the scope of a quantifier, e.g., (*r*=*MachineLocation*(*m*)). A *closed formula*, or *sentence*, is a formula in which there are no free variables, e.g.,

  $\forall m$ (*Isa*(*Machine*,*m*) $\Rightarrow$ $\exists r$ (*Isa*(*MachineRoom*,*r*) $\wedge$ (*r*=*MachineLocation*(*m*)))).

Parentheses may be omitted in any of the above expressions if no confusion will result.

First-order logic is applied by defining a set of *axioms*, or sentences intended to assert relevant truths or assumptions about a domain. The axioms, together with the set of logical consequences of the axioms, comprise a *theory* of the domain. If the axioms are consistent, the set of consequences is a proper subset of all syntactically correct sentences. Because anything follows from a contradiction, if the axioms are inconsistent, the set of consequences consists of all sentences. Until referents for the symbols are specified, a theory is a syntactic structure devoid of meaning. An *interpretation* for a theory specifies a definition of each constant, predicate and function symbol in terms of the domain. An interpretation assigns each constant symbol to a specific individual entity, each predicate to a set containing the entities for which the predicate holds, and each function symbol to a function defined on the domain. The purely logical consequences of a set of axioms consist of the sentences that are true in all interpretations, also called the *valid* sentences. A logical system is *complete* if all valid sentences can be proven and *negation complete* if for every sentence, either the sentence or its negation can be proven. Kurt Gödel proved both that first-order logic is complete, and that no consistent logical system strong enough to axiomatize arithmetic can be negation complete (cf., Stoll, 1963; Enderton, 2001).

A number of proof systems have been defined for first-order logic. Resolution with Skolemization is a refutation-complete proof system[5] that is straightforward to specify, implement and control. Russell and Norvig (2002) present a detailed description of resolution with Skolemization and a proof of refutation-completeness. Natural deduction is a complete proof system that is more intuitive than resolution, but harder to implement. Davis (1990) presents a natural deduction proof system for first-order logic.

Special-purpose logics built on first-order logic give pre-defined meaning to reserved constant, function and/or predicate symbols. Such logics provide built-in constructs that are useful in many applications. For example, there are logics that provide constants, predicates, and functions for reasoning about types, space and time, parts and wholes, actions and plans, etc. When a logic is applied to reason about a particular domain, the modeler assigns meaning to additional domain-specific constant, predicate and function symbols. This is accomplished by specifying a set of proper axioms encoding knowledge about the domain. A domain ontology (Gruber, 1993; Sowa, 2000) expresses knowledge about the types of entities in a domain of application, the attributes and allowable behaviors of entities of a given type, allowable relationships among entities of different types, and (optionally) characteristics of particular

[5] That is, if a sentence is unsatisfiable, resolution will generate a proof of unsatisfiability in finitely many steps.

individual entities. Formal ontologies are usually expressed in languages based on first-order logic or one of its subsets.

MEBN logic extends first-order logic to provide a means to assign probabilities to sentences of FOL theories in a logically consistent manner. For any MEBN theory there is a corresponding FOL theory having the same purely logical consequences. A consistent, finitely axiomatizable FOL theory can be translated to an infinity of MEBN theories, all having the same purely logical consequences, that assign different probabilities to statements whose truth-values are not determined by the axioms of the FOL theory. MEBN logic extends the propositional logic of directed graphical probability models, or Bayesian networks. Before providing a formal specification for MEBN logic, the next section gives a brief overview of Bayesian networks.

3 Bayesian Networks

Graphical probability and decision models (Whittaker, 1990, Cowell, et al., 1999) have become increasingly popular both as a parsimonious language for representing knowledge about uncertain phenomena and as an architecture to support efficient algorithms for inference, search, optimization, and learning. A graphical probability model expresses a probability distribution over a collection of interrelated hypotheses as a graph and a collection of local probability distributions. The graph encodes dependencies among the hypotheses. The local probability distributions specify numerical probability information. Specification is tractable because each local distribution depends on only a small set of directly related hypotheses. Tractable exact or approximate inference is possible for complex tasks because independence relationships allow inference to be decomposed into local inference problems involving only small numbers of hypotheses.

A Bayesian network (e.g., Pearl, 1988; Jensen, 2001; Neapolitan, 2003) is a graphical probability model in which the dependency graph is an acyclic directed graph. Figure 1 shows a Bayesian network for a diagnosis task. The nodes in the graph denote random variables. In mathematical statistics, a random variable is defined as a function that maps elements of a set called the sample space to elements of another set called the outcome space.[6] Random variables in a Bayesian network map entities in a domain of application to attributes or features of the entities. For example, in the Bayesian network of Figure 1, the *EngineStatus* random variable maps a piece of equipment to a value in the set $\{Satisfactory, Overheated\}$, depending on whether its engine is operating normally or is overheated. Each random variable can take on one of a mutually exclusive and collectively exhaustive set of possible values. Given any state of information about the other random variables, each possible value for a random variable has a probability that ranges between zero and one. This probability represents the likelihood, given the available information, that the attribute in question takes on the indicated value.

Probabilities for the possible values of the random variables are specified by means of local distributions that together implicitly specify a joint distribution over all possible configurations of values for the random variables. The graph for a Bayesian network represents a set of conditional independence assertions satisfied by the implicitly encoded probability distribution (Cowell, et al., 1999; Jensen, 2001; Lauritzen, 1996; Pearl, 1988; Whittaker, 1990). The graph must contain no directed cycles, ensuring non-circularity in the specification of probabilities. The parents of a node in the graph denote the random variables whose values directly influence the probability of the node's random variable. The probability that a random variable takes on a given value is independent of the values of the random variable's non-descendants given the values of its parents. For example, in Figure 1, if the values of *BeltStatus* and *RoomTemp* are specified, the

[6] Additional technical conditions must be satisfied for a function to be a random variable: the sample space must be a probability space; the outcome space must be a measurable space; and the function must be measurable. The graph and local distributions of a Bayesian network implicitly specify a set of random variables satisfying these conditions.

probabilities for the values of *EngineStatus* do not depend on the value of *MaintenancePractice* or *TempSensor*. That is, although the organization's maintenance practices and the temperature sensor reading are relevant to whether the engine is functioning properly, the influence operates via the condition of the belt and temperature of the room. Once the condition of the belt and the temperature of the room are given, there is no remaining influence from other ancestors of *EngineStatus*.

The local distribution for a root node consists of a single probability distribution. For non-root nodes, a probability distribution is specified for each combination of possible values of the node's parents. In Figure 1, for example, only one probability distribution needs to be specified for *MaintenancePractice*. For *EngineStatus*, a probability distribution must be specified for each combination of values of its parents. If the possible values of *BeltStatus* and *RoomTemp* are {*OK*, *Broken*} and {*Normal*, *High*}, respectively, then four probability distributions must be specified – one for each member of the set {(*OK*, *Normal*), (*OK*, *High*), (*Broken*, *Normal*), (*Broken*, *High*)}.

Some authors assume that random variables in a Bayesian network have finitely many possible values. Some require only that each random variable have an associated function mapping values of its parents to probability distributions on its set of possible values. In an unconstrained local distribution on finite-cardinality random variables, a separate probability is specified for each value of a random variable given each combination of values of its parents. Because the complexity of specifying local distributions is exponential in the number of parents, constrained families of local distributions are often used to simplify specification and inference. In distributions exhibiting context-specific independence (Geiger and Heckerman, 1991; Boutilier, et al., 1996; Mahoney and Laskey, 1999; Mahoney, 1999), the parent configurations are partitioned into subsets having a common distribution for the child random variable. Independence of causal influence (ICI) refers to a class of local distributions in which each parent random variable makes an independent contribution to the probability distribution of the child random variable. The most common ICI models are the "noisy or" and other noisy functional dependence models (Jensen, 2001; Pearl, 1988). Local expression languages (D'Ambrosio, 1991) can be used to specify arbitrary functional relationships between states of the parent random variables and probabilities of the child random variable. When a random variable and/or its parents have infinitely many possible values, local distributions cannot be listed explicitly, but can be specified as parameterized functions. When a random variable has an uncountable set of possible values, then the local distributions specify probability density functions with respect to a measure on the set of possible outcomes (cf., DeGroot and Schervish, 2002; Robert, 2001).

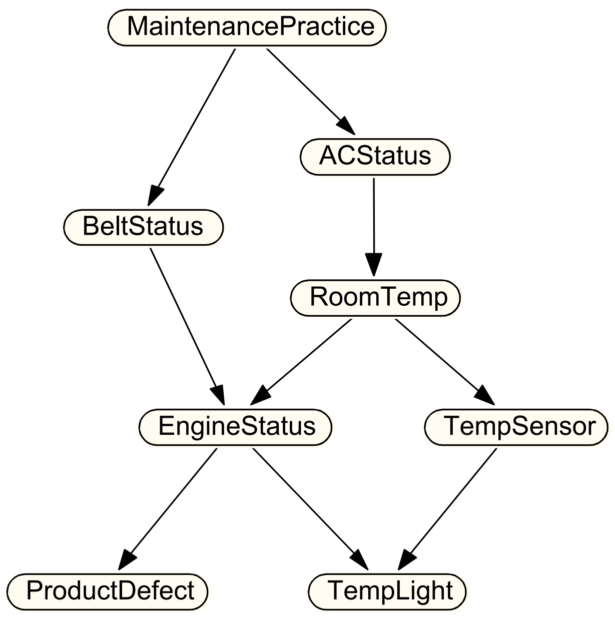


Figure 1: Bayesian Network for Diagnostic Task

A Bayesian network can be used to compute probabilities of some random variables given information about other random variables. For example, we might use the Bayesian network of Figure 1 to compute the probability of producing a defective product, and to update this distribution to incorporate evidence such whether the temperature light is blinking. Efficient algorithms have been developed for computing probabilities and propagating the impact of evidence (D'Ambrosio, 1999). Methods have also been developed for learning Bayesian networks from data and for combining observations with expert knowledge (e.g., Heckerman, et al., 1995; Dybowski, et al., 2003). By further reducing the dimensionality of the parameter space, use of local expressions can ease the specification burden, reduce the sample size required to learn the local distributions, and improve the tractability of inference.

The simple attribute-value representation of standard Bayesian networks is insufficiently expressive for many problems. For example, the Bayesian network of Figure 1 applies to a single piece of equipment located in a particular room and owned and maintained by a single organization. We may need to consider problems that involve multiple organizations, each of which owns and maintains multiple pieces of equipment of different types, some of which are in rooms that contain other items of equipment. The room temperature and air conditioner status random variables would have the same value for co-located items, and the maintenance practice random variable would have the same value for items with the same owner. Standard Bayesian networks provide no way of compactly representing the correlation between failures of co-located and/or commonly owned items of equipment or of properly accounting for these correlations when learning from observation. There has been a great deal of interest in extending the Bayesian network formalism to provide greater expressive power (e.g., Buntine, 1994; D'Ambrosio, et al, 2001; Getoor et al, 2000, 2001; Gilks et al, 1994; Heckerman, et al., 2004; Koller and Pfeffer, 1997; Laskey, et al, 2001; Laskey and Mahoney, 1997; Ngo and Haddawy, 1997; Pfeffer, 2001; Sato, 1998; Spiegelhalter et al., 1996). MEBN logic provides a unifying logical foundation for the emerging collection of more expressive probabilistic languages.

4 Multi-Entity Bayesian Networks

Like Bayesian networks, MTheories use directed graphs to specify joint probability distributions for a collection of interrelated random variables. Like Bayesian networks, MEBN logic represents relationships among hypotheses using directed graphs in which nodes represent uncertain hypotheses and edges represent probabilistic dependencies. MEBN logic extends ordinary Bayesian networks to provide first-order expressive power, and also extends first-order logic (FOL) to provide a means of specifying probability distributions over interpretations of first-order theories.

Knowledge in MEBN theories is expressed via *MEBN Fragments* (MFrags), each of which represents probability information about a group of related random variables. Just as first-order logic extends propositional logic to provide an inner structure for sentences, MEBN logic extends ordinary Bayesian networks to provide an inner structure for random variables. Random variables in MEBN logic take arguments that refer to entities in the domain of application. For example, *Manager*(*d*,*y*) might represent the manager of the department designated by the variable *d* during the year designated by the variable *y*. To refer to the manager of the maintenance department in 2003, we would fill in values for *d* and *y* to obtain an instance *Manager*(*Maintenance*,2003) of the *Manager* random variable. A given situation might involve any number of instances of the *Manager* random variable, referring to different departments and/or different years. As shown below, the Boolean connectives and quantifiers of first-order logic are represented as pre-defined MFrags whose meaning is fixed by the semantics. Any sentence that can be expressed in first-order logic can be represented as a random variable in MEBN logic. An MTheory implicitly expresses a joint probability distribution over truth-values of sets of FOL sentences. MEBN logic is modular and compositional. That is, probability distributions are specified locally over small groups of hypotheses and composed into globally consistent probability distributions over sets of hypotheses.

4.1 Entities and Random Variables

MEBN logic treats the world as being comprised of entities that have attributes and are related to other entities. Constant and variable symbols are used to refer to entities. There are three logical constants with meaning fixed by the semantics of the logic, an infinite collection of variable symbols, and an infinite collection of non-logical constant symbols with no pre-specified referents. MEBN logic uses random variables to represent features of entities and relationships among entities. MEBN logic has a collection of logical random variable symbols with meaning fixed by the semantics of the logic, and an infinite collection of non-logical random variable symbols with no pre-specified referents. The logical constants and random variables are common to all MTheories; the non-logical constants and random variables provide terminology for referring to objects and relationships in a domain of application.

Constant and variable symbols:

- (*Ordinary*) *variable symbols*: As in FOL, variables are used as placeholders to refer to non-specific entities. Variables are written as alphanumeric strings beginning with lowercase letters, e.g., *department*7. To avoid confusion, the adjective "ordinary" is sometimes used to distinguish ordinary variables from random variables.
- *Non-logical constant symbols*: Particular named entities are represented using constant symbols. As in our FOL notation, non-logical constant symbols are written as alphanumeric strings beginning with uppercase letters, e.g., *Machine*37, *Fernandez*.
- *Unique Identifier symbols*: The same entity may be represented by different non-logical constant symbols. MEBN logic avoids ambiguity by assigning a unique identifier symbol to each entity. The unique identifiers are the possible values of random variables. There are two kinds of unique identifier symbols:
  - *Truth-value symbols and the undefined symbol*: The reserved symbols T, F and ⊥, are logical constants with pre-defined meaning fixed by the semantics of MEBN logic. The symbol ⊥ denotes meaningless, undefined or contradictory hypotheses, i.e., hypotheses to which a truth-value cannot be assigned. The symbols T and F denote truth-values of meaningful hypotheses.
  - *Entity identifier symbols.* There is an infinite set $\mathcal{E}$ of entity identifier symbols. An interpretation of the theory uses entity identifiers as labels to refer to entities in the domain. Entity identifiers are written either as numerals or as alphanumeric strings beginning with an exclamation point, e.g., !*M*3, 48723.

Random variable symbols:

- *Logical connectives and the equality operator*: The logical connective symbols ¬, ∧, ∨, ⇒, and ⇔, together with the equality relation =, are reserved random variable symbols with pre-defined meanings fixed by the semantics of MEBN logic. Logical expressions may be written using prefix notation (e.g,, $\neg(x)$, $\vee(x,y)$, $=(x,y)$), or in the more familiar infix notation (e.g., $\neg x$, $(x \vee y)$; $(x=y)$). Different ways of writing the same expression (e.g., $=(x,y)$, $(y=x)$) are treated as the same random variable.
- *Quantifiers*: The symbols ∀ and ∃ are reserved random variable symbols with pre-defined meaning fixed by the semantics of MEBN logic. They are used to construct MEBN random variables to represent FOL sentences containing quantifiers.
- *Identity*: The reserved random variable symbol ◊ denotes the identity random variable. It is the identity function on T, F, ⊥, and the set of entity identifiers that denote meaningful entities in a domain. It maps meaningless, irrelevant, or contradictory random variable terms to ⊥.
- *Findings*: The *finding* random variable symbol, denoted Φ, is used to represent observed evidence, and also to represent constraints assumed to hold among entities in a domain of application.

- *Non-logical random variable symbols*: The domain-specific random variable symbols are written as alphanumeric strings beginning with an uppercase letter. With each random variable symbol is associated a positive integer indicating the number of arguments it takes. Each random variable also has an associated set of *possible values* consisting of a subset of the unique identifier symbols. The set of possible values may be infinite, but if so, there must exist an effective procedure (provably terminating algorithm) that lists all the possible values and an effective procedure for determining whether any unique identifier symbol is one of the possible values. A random variable for which the set of possible values is $\{\mathsf{T},\mathsf{F},\perp\}$ is called a *Boolean* random variable. The set of possible values for any non-Boolean random variable is contained in $\mathcal{E}\cup\{\perp\}$. Boolean random variables correspond to predicates and non-Boolean random variables correspond to functions in FOL.
- *Exemplar symbols*. There is an infinite set of exemplar symbols used to refer to representative entities in the range of quantifiers. A exemplar symbol is denoted by \$ followed by an alphanumeric string, e.g., \$*b*32.

Punctuation:
- MEBN random variable terms are constructed using the above symbols and the punctuation symbols comma, open parenthesis and close parenthesis.

A *random variable term* is a random variable symbol followed by a parenthesized list of arguments separated by commas, where the arguments may be variables, constant symbols, or (recursively) random variable terms. When α is a constant or ordinary variable, the random variable term $\Diamond(\alpha)$ may be denoted simply as α. If ϕ is a random variable symbol, a *value assignment term* for ϕ has the form $=(\psi,\alpha)$, where ψ is a random variable term and α is either an ordinary variable symbol or one of the possible values of ϕ. The strings $=(\alpha,\psi)$, $(\alpha=\psi)$, and $(\psi=\alpha)$ are treated as synonyms for $=(\psi,\alpha)$. A random variable term is *closed* if it contains no ordinary variable symbols and *open* if it contains ordinary variable symbols. An open random variable term is also called a *random variable class*; a closed random variable term is called a *random variable instance*. If a random variable instance is obtained by substituting constant terms for the variable terms in a random variable class, then it is called an instance of the class. For example, the value assignment term =(*BeltStatus*(!*B*1), !*OK*), also written (*BeltStatus*(!*B*1) = !*OK*), is an instance of both (*BeltStatus*(*b*)=*x*) and (*BeltStatus*(!*B*1)=*x*), but not of (*BeltStatus*(*b*) = !*Broken*). When no confusion is likely to result, the term random variable may be used to refer either to a class or to an instance. A random variable term is called *simple* if all its arguments are either unique identifier symbols or variable symbols; otherwise, it is called *composite*. For example, =(*BeltStatus*(!*B*1), !*OK*) is a composite random variable term containing the simple random variable term *BeltStatus*(!*B*1) as an argument. It is assumed that the sets consisting of ordinary variable symbols, unique identifier symbols, exemplar random variable symbols, non-logical constant symbols, and non-logical random variable symbols are all recursive.

4.2 MEBN Fragments

In MEBN logic, multivariate probability distributions are built up from *MEBN fragments* or MFrags (see Figure 2). An MFrag defines a probability distribution for a set of *resident* random variables conditional on the values of *context* and *input* random variables. Random variables are represented as nodes in a fragment graph whose arcs represent dependency relationships.

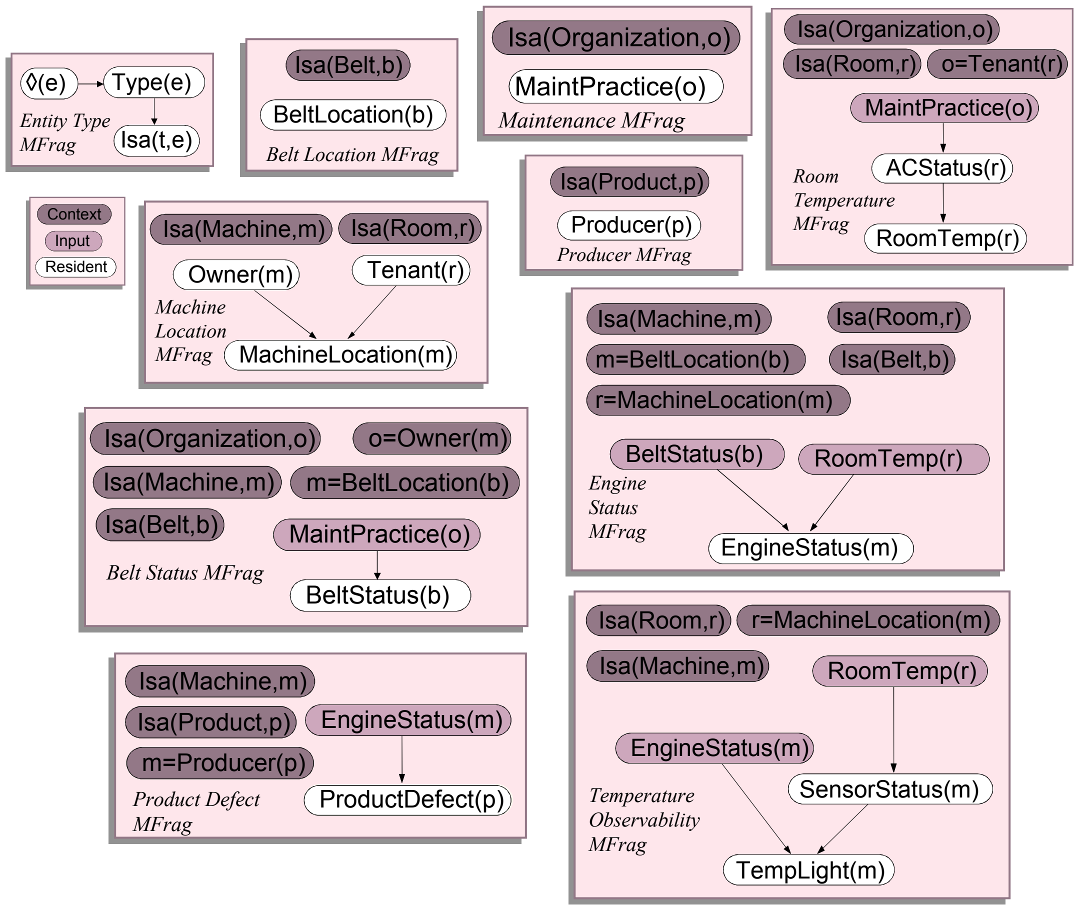


Figure 2: MEBN Fragments for Equipment Diagnosis Problem

Definition 1: An *MFrag* $\mathcal{F} = (\mathcal{C},\mathcal{I},\mathcal{R},\mathcal{G},\mathcal{D})$ consists of a finite set $\mathcal{C}$ of *context* value assignment terms;[7] a finite set $\mathcal{I}$ of *input* random variable terms; a finite set $\mathcal{R}$ of *resident* random variable terms; a *fragment graph* $\mathcal{G}$; and a set $\mathcal{D}$ of *local distributions*, one for each member of $\mathcal{R}$. The sets $\mathcal{C}$, $\mathcal{I}$, and $\mathcal{R}$ are pairwise disjoint. The fragment graph $\mathcal{G}$ is an acyclic directed graph whose nodes are in one-to-one correspondence with the random variables in $\mathcal{I}\cup\mathcal{R}$, such that random variables in $\mathcal{I}$ correspond to root nodes in $\mathcal{G}$. Local distributions specify conditional probability distributions for the resident random variables as described in Definition 3 below. ■

An MFrag is a schema for specifying conditional probability distributions for instances of its resident random variables given the values of instances of their parents in the fragment graph and given the context constraints. A collection of MFrags that satisfies the global consistency constraints defined in Section 4.3 below represents a joint probability distribution on an unbounded and possibly infinite number of instances of its random variable terms. The joint distribution is specified via the local distributions, which are defined formally below, together with the conditional independence relationships implied by the fragment graphs. Context terms are used to specify constraints under which the local distributions apply.

[7] If ϕ is a Boolean random variable, the context constraint ϕ=T may be abbreviated ϕ and the context constraint ϕ=F may be abbreviated $\neg\phi$.

As in ordinary Bayesian networks, a local distribution maps configurations of values of the parents of a random variable instance to probability distributions for its possible values. When all ordinary variables in the parents of a resident random variable term also appear in the resident term itself, as for the *RoomTemp* and *TempLight* random variables of the temperature observability MFrag of Figure 2, a local distribution can be specified simply by listing a probability distribution for the child random variable for each combination of values of the parent random variables. The situation is more complicated when ordinary variables in a parent random variable do not appear in the child. In this case, there may be an arbitrary, possibly infinite number of relevant instances of a parent for any given instance of the child. For example, in the engine status fragment of Figure 2, if it is uncertain where a machine is located, the temperature in any room in which it might be located is relevant to the distribution of the *EngineStatus* random variable. If a machine has more than one belt, then the status of any of its belts is relevant to the distribution of the *EngineStatus* random variable. Thus, any number of instances of the *RoomTemp* and *BeltStatus* random variables might be relevant to the distributions of the *EngineStatus* random variable. The local distribution for a random variable must specify how to combine influences from all relevant instances of its parents.

Definition 2: Let $\mathcal{F}$ be an MFrag containing ordinary variables $\theta_1, \ldots, \theta_k$, and let $\psi(\theta)$ denote a resident random variable in $\mathcal{F}$ that may depend on some or all of the θ_i.

2a. A *binding set* $\mathcal{B} = \{(\theta_1{:}\varepsilon_1), (\theta_2{:}\varepsilon_2), \ldots (\theta_k{:}\varepsilon_k)\}$ for $\mathcal{F}$ is a set of ordered pairs associating a unique identifier symbol ε_i with each ordinary variable θ_i of $\mathcal{F}$. The constant symbol ε_i is called the *binding* for variable θ_i determined by $\mathcal{B}$. The ε_i are not required to be distinct.

2b. Let $\mathcal{B} = \{(\theta_1{:}\varepsilon_1), (\theta_2{:}\varepsilon_2), \ldots (\theta_k{:}\varepsilon_k)\}$ be a binding set for $\mathcal{F}$, and let $\psi(\varepsilon)$ denote the instance of ψ obtained by substituting ε_i for each occurrence of θ_i in $\psi(\theta)$. A *potential influencing configuration* for $\psi(\varepsilon)$ and $\mathcal{B}$ is a set of value assignment terms $\{(\gamma{=}\phi(\varepsilon))\}$, one for each parent of ψ and one for each context random variable of $\mathcal{F}$. Here, $\phi(\varepsilon)$ denotes the instance of the context or parent random variable $\phi(\theta)$ obtained by substituting ε_i for each occurrence of θ_i;[8] and γ denotes one of the possible values of $\phi(\varepsilon)$ (as specified by the local distribution π_ψ; see Definition 3 below). An *influencing configuration* for $\psi(\varepsilon)$ and $\mathcal{B}$ is a potential influencing configuration in which the value assignments match the context constraints of $\mathcal{F}$. Two influencing configurations are *equivalent* if substituting θ_i back in for ε_i yields the same result for both configurations. The equivalence classes for this equivalence relation correspond to distinct configurations of parents of $\psi(\theta)$ in $\mathcal{F}$.

2c. Let $\{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n\}$ be a non-empty, finite set of unique identifier symbols. The *partial world* $\mathcal{W}$ for ψ and $\{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n\}$ is the set consisting of all instances of the parents of ψ and the context random variables of $\mathcal{F}$ that can be formed by substituting the ε_i for ordinary variables of $\mathcal{F}$. A *partial world state* $S_{\mathcal{W}}$ for a partial world is a set of value assignment terms, one for each random variable in the partial world.

2d. Let $\mathcal{W}$ be a partial world for ψ and $\{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n\}$, let $S_{\mathcal{W}}$ be a partial world state for $\mathcal{W}$, let $\mathcal{B} = \{(\theta_1{:}\varepsilon_{\mathcal{B}1}), (\theta_2{:}\varepsilon_{\mathcal{B}2}), \ldots (\theta_k{:}\varepsilon_{\mathcal{B}k})\}$ be a binding set for $\mathcal{F}$ with bindings chosen from

[8] If a context value assignment term $(\gamma{=}\phi)$ has no arguments, then no substitution is needed.

$\{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n\}$, and let $\psi(\varepsilon_{\mathcal{B}})$ be the instance of $\psi(\theta)$ from $\mathcal{B}$. The *influence counts* $\#S_{W_\psi}$ for $\psi(\alpha_{\mathcal{B}})$ in S_W consist of the number of influencing configurations S_W contains for each equivalence class of influencing configurations (i.e., each configuration of the parents of $\psi(\theta)$ in $\mathcal{F}$). ■

As an example, Table 1 shows a partial world state for the *EngineStatus*(*m*) random variable from Figure 2 with unique identifiers {!*M*1, !*R*1, !*R*2, !*B*1, !*B*2, !*O*1}. In the intended meaning of the partial world of Table 1, !*M*1 denotes a machine, !*B*1 and !*B*2 denote belts located in !*M*1, !*R*1 denotes the room where !*M*1 is located, !*R*2 denotes a room where !*M*1 is not located, and !*O*1 denotes an entity that is not a machine, a room, or a belt. The partial world state specifies the value of each random variable for each of the entity identifiers. Random variables map meaningless attributes (e.g., the value of *RoomTemp* for an entity that is not a room) to the absurd symbol $\perp$.

The partial world state of Table 1 contains two equivalent influencing configurations for *EngineStatus*(!*M*1):

IC1: { (*Isa*(*Machine*,!*M*1)=T), (*Isa*(*Belt*,!*B1*)=T), (*Isa*(*Room*,!*R*1)=T),
(*BeltLocation*(!*B1*)=!*M*1), (*MachineLocation*(!*M*1)=!*R*1), (*RoomTemp*(!*R*1)=!*Normal*),
(*BeltStatus*(!*B*1)=!*OK*)};

IC2: { *Isa*(*Machine*,!*M*1)=T), (*Isa*(*Belt*,!*B*2)=T), (*Isa*(*Room*,!*R*1)=T),
(*BeltLocation*(!*B*2)=*M*1)*,* (*MachineLocation*(!*M*1)=!*R*1), (*RoomTemp*(!*R*1)=!*Normal*),
(*BeltStatus*(!*B*2)=!*OK*)}.

It contains no other influencing configurations for *EngineStatus*(*M*1). Thus, the influence counts for *EngineStatus*(*M*1) in this possible world state are:

RoomTemp=!*Normal*,	*BeltStatus*=!*OK*	:	2
RoomTemp=!*Normal*,	*BeltStatus*=!*Broken*	:	0
RoomTemp=!*Hot*,	*BeltStatus*=!*OK*	:	0
RoomTemp=!*Hot*,	*BeltStatus*=!*Broken*	:	0 .

The local distribution assigned to *EngineStatus*(*M*1) in this partial world would thus be the one for a machine having two intact and no broken belts, and located in a room with normal room temperature.

***Definition 3*:** The *local distribution* π_ψ for resident random variable ψ in MFrag $\mathcal{F}$ is a function $\pi_\psi(\alpha|S)$ that maps unique identifiers α and partial world states S to real numbers, such that the following conditions are satisfied:

3a. For a given partial world state S, $\pi_\psi(\cdot|S)$ is a probability distribution on the unique identifier symbols. That is, $\pi_\psi(\alpha|S) \geq 0$ for all unique identifiers α, and $\sum_\alpha \pi_\psi(\alpha \mid S) = 1$.[9]

3b. For each instance $\psi(\varepsilon)$ of ψ, the set $\mathcal{V}_{\psi(\varepsilon)}$ of possible values of the instance $\psi(\varepsilon)$ is a recursively enumerable subset of the unique identifiers, and $\pi_{\psi(\varepsilon)}(\mathcal{V}_{\psi(\varepsilon)}|S) = 1$ for each partial world S.

3c. There is an algorithm such that for any recursive subset A of the possible values of ψ not containing $\perp$, and any partial world state S for ψ, either the algorithm halts with output

[9] Although random variables in MEBN logic have finite or countably infinite sample spaces, and local distributions are discrete, MEBN logic can represent continuous distributions (see Section 5 below).

$\pi_\psi(A|S)$ or there exists a value $N(A,S)$ such that if the algorithm is interrupted after a number of time steps greater than $N(A,S)$, the output is $\pi_\psi(A|S)$.[10]

3d. π_ψ depends on the partial world state only through the influence counts. That is, any two partial world states having the same influence counts map to the same probability distribution;

3e. Let $S_1 \subset S_2 \subset \ldots$ be an increasing sequence of partial world states for ψ. There exists an integer N such that if $k > N$, $\pi_\psi(S_k) = \pi_\psi(S_N)$.[11]

The probability distribution $\pi_\psi(\varepsilon|\varnothing)$ is called the *default distribution* for ψ. It is the probability distribution for ψ given that no potential influencing configurations satisfy the conditioning constraints of $\mathcal{F}$. If ψ is a root node in an MFrag $\mathcal{F}$ containing no context constraints, then the local distribution for ψ is just the default distribution. ■

Isa(*Machine*,!*M*1)=T *Isa*(*Belt*,!*M*1)=F *Isa*(*Room*,!*M*1)=F *BeltLocation*(!*M*1)=⊥ *MachineLocation*(!*M*1)=!*R*1 *RoomTemp*(!*M*1)=⊥ *BeltStatus*(!*M*1)=⊥	*Isa*(*Machine*,!*R*1)=F *Isa*(*Belt*,!*R*1)=F *Isa*(*Room*,!*R*1)=T *BeltLocation*(!*R*1)=⊥ *MachineLocation*(!*R*1)=⊥ *RoomTemp*(!*R*1)=!*Normal* *BeltStatus*(!*R*1)=⊥	*Isa*(*Machine*,!*R*2)=F *Isa*(*Belt*,!*R*2)=F *Isa*(*Room*,!*R*2)=T *BeltLocation*(!*R*2)=⊥ *MachineLocation*(!*R*2)=⊥ *RoomTemp*(!*R*2)=*Hot* *BeltStatus*(!*R*2)=⊥
Isa(*Machine*,!*B*1)=F *Isa*(*Belt*,!*B*1)=T *Isa*(*Room*,!*B*1)=F *BeltLocation*(!*B*1)=!*M*1 *MachineLocation*(!*B*1)=⊥ *RoomTemp*(!*B*1)=⊥ *BeltStatus*(!*B*1)=!*OK*	*Isa*(*Machine*,!*B*2)=F *Isa*(*Belt*,!*B*2)=T *Isa*(*Room*,!*B*2)=F *BeltLocation*(!*B*2)=!*M*1 *MachineLocation*(!*B*2)=⊥ *RoomTemp*(!*B*2)=⊥ *BeltStatus*(!*B*2)=!*OK*	*Isa*(*Machine*,!*O*1)=F *Isa*(*Belt*,!*O*1)=F *Isa*(*Room*,!*O*1)=F *BeltLocation*(!*O*1)=⊥ *MachineLocation*(!*O*1)=⊥ *RoomTemp*(!*O*1)=⊥ *BeltStatus*(!*O*1)=⊥

Table 1: Partial World State for *EngineStatus* Partial World

Conditions such as 3*c* and 3*e* are needed to ensure that a global joint distribution exists and can be approximated by a sequence of finite Bayesian networks. They are stronger than strictly necessary for this purpose, but they are satisfied in the MTheory presented in Section 4.5 below that specifies a probability distribution over interpretations of theories in first-order logic.

Table 2 shows an example of a local distribution for the engine status MFrag. The conditioning constraints imply there can be at most one *RoomTemp* parent that satisfies the context constraint *MachineLocation*(*m*) = *r*. When this parent has value !*Normal*, probability $\alpha_{k,n}$ is assigned to !*Normal* and probability 1-$\alpha_{k,n}$ is assigned to !*Overheated*, where *k* is the number of distinct *BeltStatus* parents having the value *OK*, out of a total of *n*>0 distinct *BeltStatus* parents. When the *RoomTemp* parent corresponding to *MachineLocation*(*m*) has value !*Hot*, the probability of a satisfactory engine is $\beta_{k,n}$ and the probability of an overheated engine is 1-$\beta_{k,n}$, where again *k* denotes the number of distinct belts with value *OK* and *n*>0 denotes the total number of distinct belts. The default distribution applies when no combination of entities meets the conditioning constraints. Condition 3*e* implies that the local distribution for any instance of

[10] It is required that $N(A,S)$ exists, but there need not be an effective procedure for computing it. The author is indebted to an anonymous reviewer for pointing out that a minor modification of this condition is required for the proof of Theorem 2.

[11] Again, it is not required that there be an effective procedure for computing *N*.

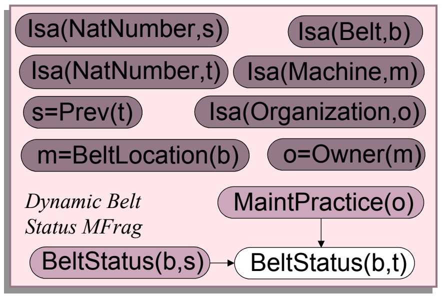


Figure 3: Recursive MFrag

MachineLocation(*m*) in any world can be calculated from at most finitely many *BeltStatus* parents. Given CC1, *EngineStatus*(!*M*1)=*!Satisfactory* has probability $\alpha_{2,0}$ and *EngineStatus*(!*M*2) = !*Overheated* has probability 1-$\alpha_{2,0}$. If CC1 were modified by changing *RoomTemp*(!*R*2) from !*Hot* to !*Normal*, the distribution would not change, because the influence counts for CC1 do not depend on *RoomTemp*(!*R*2). On the other hand, if *RoomTemp*(!*R*1) had value !*Hot*, then the probabilities of *EngineStatus*(!*M*1)=*!Satisfactory* and *EngineStatus*(!*M*2) =!*Overheated* would be $\beta_{2,0}$ and 1-$\beta_{2,0}$, respectively. The default distribution applies when there are no influencing configurations. The default distribution assigns probability 1 to ⊥, meaning that *EngineStatus*(*m*) is meaningless when the context constraints are not met (i.e., *m* does not denote a machine, *m* is not located in a room, or *m* has no belt). Default distributions are not required to assign probability 1 to ⊥. For example, the default distribution could be used to represent the engine status of beltless machines. Note, however, that the default distribution does not distinguish situations in which *m* refers to a machine with no belt from situations in which *m* is not a machine. Thus, this modeling approach would assign the same *EngineStatus* distribution to non-machines as to machines with no belt.

MFrags may contain recursive influences. Recursive influences allow instances of a random variable to depend directly or indirectly on other instances of the same random variable. One common type of recursive graphical model is a dynamic Bayesian network (Ghahramani, 1998; Murphy, 1998). Recursion is permissible as long as no random variable instance can directly or indirectly influence itself. This requirement is satisfied when the conditioning constraints prevent circular influences. For example, Figure 3 modifies the belt status MFrag from Figure 2 so that the status of a belt depends not only on the maintenance practice of the organization, but also on the status of the belt at the previous time. The function *Prev*(*n*), defined for natural numbers, maps a positive natural number to the previous natural number, and has value ⊥ when *n* is zero. The context constraint *s* = *Prev*(*t*), prevents circular influences in instances of the MFrag. If the variable *t* is bound to zero, there will be no influencing configurations satisfying the context constraints (because *Prev*(0) has value ⊥ and *NatNumber*(⊥)=⊥.). Thus, any instance of the *BeltStatus* random variable for which *s* is bound to zero will have no parents, and its local distribution will be the default distribution.

MFrags can represent a rich family of probability distributions over interpretations of first-order theories. The ability of MFrags to represent uncertainty about parameters of local distributions provides a logical foundation for parameter learning in first-order probabilistic

Context	***RoomTemp(r)***	***BeltStatus(b)***	***EngineStatus(m)***		
			Satisfactory	***Overheated***	⊥
Belt b located in machine m, located in room r	*!Normal*	*!OK* : *k*	$\alpha_{k,n}$	1-$\alpha_{k,n}$	0
		!Broken : *n-k*			
	!High	*!OK* : *k*	$\beta_{k,n}$	1-$\beta_{k,n}$	0
		!Broken : *n-k*			
Default			0	0	1

Table 2: Local Distribution as Function of Influence Counts

theories. Uncertainty about structure can be represented by sets of MFrags having mutually exclusive context constraints and different fragment graphs, thus providing a logical foundation for structure learning. Further discussion of learning from observation can be found in Section 6 below.

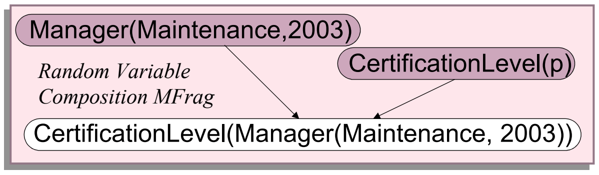
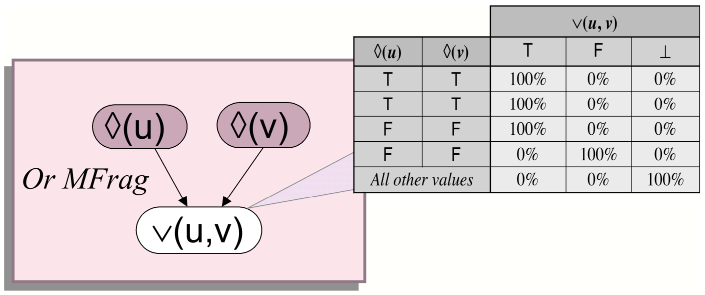

Manager(Maintenance,2003)
CertificationLevel(p)
Random Variable Composition MFrag
CertificationLevel(Manager(Maintenance, 2003))

Figure 4: Indirect Reference

MEBN logic comes equipped with a set of built-in MFrags representing logical operations, function composition, and quantification. There are also constraints that must be satisfied by domain-specific MFrags. The built-in MFrags, the constraints on domain-specific MFrag definitions, and the rules for combining MFrags and performing inference provide the logical content of pure MEBN logic. An applied MTheory specifies a set of domain-dependent MFrags that provide empirical and/or mathematical content.

The built-in MFrags are defined below:

- *Indirect reference*. The rules for instantiating MFrags allow only unique identifier symbols to be substituted for the ordinary variable symbols. Probability distributions for indirect references are handled with built-in composition MFrags, as illustrated in Figure 4. These MFrags enforce logical constraints on function composition. Let $\psi(\phi_1(\alpha_1), \ldots, \phi_k(\alpha_k))$ be a random variable instance, where ψ and ϕ_i are random variable symbols and each α_i is a list of arguments. The random variable instance $\psi(\phi_1(\alpha_1), \ldots ,\phi_k(\alpha_k))$ has a parent $\phi_i(\alpha_i)$ for each of the arguments and a *reference parent* $\psi(y_1, \ldots, y_k)$, where the y_i denote ordinary variable symbols such that y_i may be the same as y_j only if $\phi_i(\alpha_i)$ and $\phi_j(\alpha_j)$ are logically equivalent expressions.[12] The local distribution for $\psi(\phi_1(\alpha_1),\ldots,\phi_k(\alpha_k))$ assigns it the same value as $\psi(y_1,\ldots,y_k)$ when the value of y_i is the same as the value of $\phi_i(\alpha_i)$. Although there are infinitely many possible substitutions for $\psi(y_1,\ldots,y_k)$ and hence infinitely many potential influencing configurations, in any given world only one of the influences is active. Thus, condition 3*e* is satisfied. The default distribution specifies a value for $\psi(\phi_1(\alpha_1),\ldots,\phi_k(\alpha_k))$ when there are no influencing configurations.
- *Equality random variable*. The resident random variable in the equality MFrag has the form $=(u,v)$, also written $(u=v)$. There are two parents, one for each argument. The equality operator has value $\perp$ if either u or v has value $\perp$, T if ϕ and ψ have the same value and are not equal to $\perp$, and F otherwise. It is assumed that meaningful entity identifiers are distinct. That is, if ε_1 and ε_2 are distinct entity identifiers, then $(\varepsilon_1=\varepsilon_2)$ has value $\perp$ if $\Diamond(\varepsilon_1)$ or $\Diamond(\varepsilon_2)$ has value $\perp$, and F otherwise.
- *Logical connectives*. The random variable $\neg(u)$ has a single parent, $\Diamond(u)$; the other logical connectives have two parents, $\Diamond(u)$ and $\Diamond(v)$. The value of $\neg(u)$ is T if its parent has value F, F if its parent



		$\vee(u, v)$		
$\Diamond(u)$	$\Diamond(v)$	T	F	$\perp$
T	T	100%	0%	0%
T	T	100%	0%	0%
F	F	100%	0%	0%
F	F	0%	100%	0%
All other values		0%	0%	100%

Figure 5: Logical Connective MFrag

[12] It is always permissible to use distinct variables in a composition MFrag, but it is more efficient to use the same variable when the expressions are known to be logically equivalent.

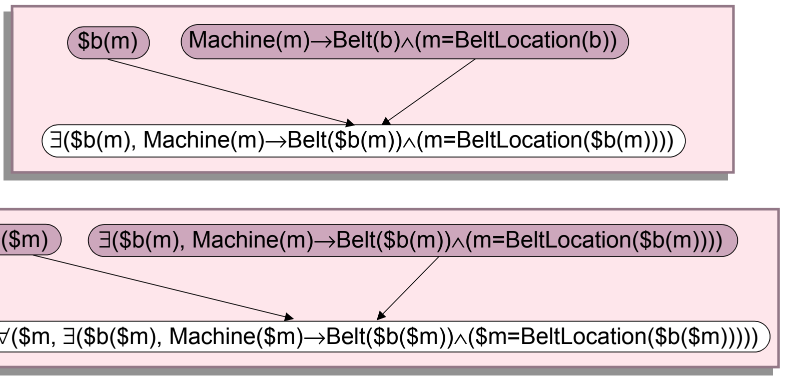

$b(m)
Machine(m)→Belt(b)∧(m=BeltLocation(b))
∃($b(m), Machine(m)→Belt($b(m))∧(m=BeltLocation($b(m))))
◊($m)
∃($b(m), Machine(m)→Belt($b(m))∧(m=BeltLocation($b(m))))
∀($m, ∃($b($m), Machine($m)→Belt($b($m))∧($m=BeltLocation($b($m)))))

Figure 6: Quantifier MFrags

has value T, and ⊥ otherwise. The other logical connectives map truth-values according to the usual truth tables and parents other than T or F to ⊥ (see Figure 5).

- *Quantifiers*. Let $\phi(\gamma)$ be an open Boolean random variable term containing the ordinary variable γ. A *quantifier random variable* has the form $\forall(\sigma, \phi(\sigma))$ or $\exists(\sigma, \phi(\sigma))$, where $\phi(\sigma)$ is obtained by substituting the *exemplar term* σ into $\phi(\gamma)$. A quantifier random variable instance has a single parent $\phi(\gamma)$. The value of $\forall(\sigma, \phi(\sigma))$ is T by default and F if any instance of $\phi(\gamma)$ has value F. The value of $\exists(\sigma, \phi(\sigma))$ is F by default and T if any instance of $\phi(\gamma)$ has value T. It is assumed that a unique exemplar symbol is assigned to each ordinary variable of each Boolean random variable term of the language.[13] Figure 6 shows quantifier MFrags representing the hypothesis that every machine has a belt. In FOL, the corresponding sentence is:

$$\forall m \exists b\ (Isa(Machine,m) \Rightarrow Belt(b) \wedge (m=BeltLocation(b))).$$

An important feature of MEBN logic is its logically consistent treatment of reference uncertainty. For example, suppose the random variable instance *CertificationLevel*(*Manager*(*Maintenance*, 2003)) is intended to refer to the individual who managed the maintenance department in 2003. If the possible managers are *!Employee*37 and *!Employee*49, MEBN logic ensures that the probability distribution for *CertificationLevel*(*Manager*(*Maintenance*, 2003)) will be a weighted average of the probability distributions for *CertificationLevel*(*!Employee*37) and *CertificationLevel*(*!Employee*49), where the weights are the probabilities that *Manager*(*Maintenance*, 2003) has value *!Employee*37 and *!Employee*49, respectively. Furthermore, if *!Employee*39 refers to an individual who is also referred to as *Carlos*, *Fernandez*, and *Father*(*Miguel*), any information germane to the certification level of *Carlos*, *Fernandez* or *Father*(*Miguel*) will propagate consistently to *CertificationLevel*(*Manager*(*Maintenance*, 2003)) when Bayesian inference is applied (see Figure 7).

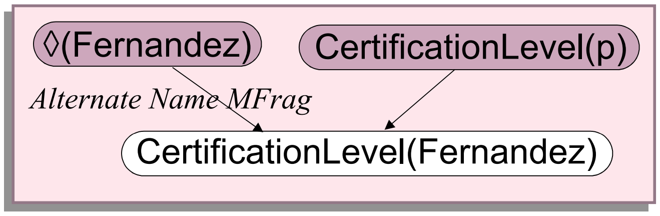


Figure 7: Relating a Name to a Unique Identifier

The built-in MFrags defined above provide sufficient expressive power to represent a probability distribution over interpretations of any finitely axiomatizable FOL theory, and to use Bayesian conditioning to generate a sequence of MTheories, where each MTheory in the sequence is obtained by conditioning the preceding MTheory on

[13] A countable infinity of exemplar symbols is sufficient for this purpose.

new axioms that are consistent with the preceding MTheories. MTheories can be used to define special-purpose logics such as the planning and decision-making logics described in Section 5 below. Each such special-purpose logic is a subclass of MEBN logic containing a set of MFrags common to all theories in the subclass.

In MEBN logic, there are two kinds of domain-specific MFrags: generative MFrags and finding MFrags. The distinction between generative MFrags and finding MFrags corresponds roughly to the *terminological box*, or T-box, and the *assertional reasoner*, or A-box (Brachman, et al., 1983). The generative domain-specific MFrags specify information about statistical regularities characterizing the class of situations to which an MTheory applies. Findings can be used to specify particular information about a specific situation in the class defined by the generative theory. Findings can also be used to represent constraints assumed to hold in the domain (cf., Jensen, 2001; Heckerman, et al., 2004), although there are both computational and interpretation advantages to using generative MFrags when "constraint findings" can be avoided.

***Definition* 4:** A *finding* MFrag satisfies the following conditions:

4a. There is a single resident random variable, $\Phi(\psi)$, where ψ is a closed value assignment term. For Boolean random variable instances, we may abbreviate $\Phi(\phi{=}\mathsf{T})$ as $\Phi(\phi)$, and $\Phi(\phi{=}\mathsf{F})$ as $\Phi(\neg(\phi))$.

4b. There are no context random variable terms. There is a single input random variable term ψ, which is a parent of the resident random variable $\Phi(\psi)$.

4c. The local distribution for $\Phi(\psi)$ is deterministic, assigning value T if ψ has value T and $\perp$ if it has value F or $\perp$. ■

***Definition* 5:** A generative domain-specific MFrag $\mathcal{F}$ must satisfy the following conditions.

5a. None of the random variable terms in $\mathcal{F}$ is a finding random variable term.

5b. Each resident random variable term in $\mathcal{F}$ is a simple open random variable term, i.e., a constant symbol, an ordinary variable symbol, or a random variable term that consists of a random variable symbol followed by a parenthesized list of ordinary variable symbols.

5c. The only possible values for the identity random variable $\Diamond(\varepsilon)$ are ε and $\perp$. Furthermore, $\Diamond(\mathsf{T})=\mathsf{T}$; $\Diamond(\mathsf{F})=\mathsf{F}$; and $\Diamond(\perp)=\perp$.[14]

5d. For any resident random variable term ψ other than the identity, the local distribution for ψ must assign probability zero to any unique identifier ε for which $\Diamond(\varepsilon) \neq \varepsilon$. One way to ensure this constraint is met is to make $\Diamond(\varepsilon)$ a parent of ψ for any possible value ε for which there is non-zero probability that $\Diamond(\varepsilon) \neq \varepsilon$, and to specify a local distribution that assigns probability zero to ε if $\Diamond(\varepsilon) \neq \varepsilon$. ■

In summary, MFrags represent influences among clusters of related random variables. Repeated patterns can be represented using ordinary variables as placeholders into which entity identifiers can be substituted. Probability information for an MFrag's resident random variables are specified via local distributions, which map influence counts for a random variable's parents to probability distributions over its possible values. When ordinary variables appear in a parent but not in a child, the local distribution specifies how to combine influences from multiple copies of the parent random variables. Restricting variable bindings to unique identifiers prevents double counting of repeated instances. Multiple ways of referring to an entity are handled

[14] A finite domain can be represented by specifying an ordering ε_1, ε_2,… on the unique identifiers, and specifying a probability of 1 that $\Diamond(\varepsilon_{i+1}) = \perp$ if $\Diamond(\varepsilon_i) = \perp$. In this case, the cardinality of the domain is the last i for which $\Diamond(\varepsilon_i) \neq \perp$. The cardinality may of course be uncertain.

through built-in MFrags that enforce logical constraints on function composition. Context constraints permit recursive relationships to be specified without circular references.

4.3 MEBN Theories

A *MEBN theory*, or MTheory, is a collection of MFrags that satisfies consistency constraints ensuring the existence of a unique joint probability distribution over the random variables mentioned in the theory. The built-in MFrags provide logical content and the domain-specific MFrags provide empirical content. This section defines an MTheory and states the main existence theorem, that a joint distribution exists for the random variable instances of an MTheory. A proof is given in the Appendix.

An MTheory containing only generative domain-specific MFrags is called a *generative MTheory*. Generative MTheories can be used to express domain-specific ontologies that capture statistical regularities in a particular domain of application. MTheories with findings can augment statistical information with particular facts germane to a given reasoning problem. MEBN logic uses Bayesian learning to refine domain-specific ontologies to incorporate observed evidence.

The MFrags of Figure 2 specify a generative MTheory for the equipment diagnosis problem. These MFrags specify local probability distributions for their resident random variables. The conditioning constraints in each MFrag specify type restrictions (e.g., the symbol *m* must be replaced by an identifier for an entity of type *Machine*) and functional relationships an influencing configuration must satisfy (e.g., the room identifier *r* must be equal to the value of *MachineLocation*(*m*)). Each local distribution provides a rule for calculating the distribution of a resident random variable given any instance of the MFrag.

Reasoning about a particular task proceeds as follows. First, finding MFrags are added to a generative MTheory to represent task-specific information. Next, random variables are identified to represent queries of interest. Finally, Bayesian inference is applied to compute a response to the queries. Bayesian inference can also be applied to refine the local distributions and/or MFrag structures given the task-specific data (see Section 6 below). For example, to assert that the temperature light is blinking in the machine denoted by *!Machine*37, which is located in the room denoted by *!Room*103*A*, we could add the findings Φ(*TempLight*(*!Machine*37)=*!Blinking*) and Φ(*MachineLocation*(*Machine*37)=*!Room*103*A*) to the generative MTheory of Figure 2. To inquire about the likelihood that there are any overheated engines, the FOL sentence ∃*m* (*Isa*(*Machine*,*m*)∧(*EngineStatus*(*m*)=*!Overheated*)) would be translated into the quantifier random variable instance ∃($*m*, *Isa*(*Machine*,$*m*)∧(*EngineStatus*($*m*)=*!Overheated*)). A Bayesian inference algorithm would be applied to evaluate its posterior probability given the evidence.

As with ordinary Bayesian networks, global consistency conditions are required to ensure that the local distributions collectively specify a well-defined probability distribution over interpretations. Specifically, the MFrags must combine in such a way that no random variable instance can directly or indirectly influence itself, and initial conditions must be specified for recursive definitions. Non-circularity is ensured in ordinary Bayesian networks by defining a partial order on random variables and requiring that a random variable's parents precede it in the partial ordering. In dynamic Bayesian networks, random variables are indexed by time, an unconditional distribution is specified at the first time step, and each subsequent distribution may depend on the values of the random variables at the previous time step. Non-circularity is ensured by prohibiting links from future to past and by requiring that links within a time step respect the random variable partial ordering. Other kinds of recursive relationships, such as genetic inheritance, have been discussed in the literature (cf., Pfeffer, 2000). Recursive Bayesian networks (Jaeger, 2001) can represent a very general class of recursively specified probability distributions for Boolean random variables on finite domains. No previously published probabilistic knowledge representation language provides general-purpose rules for defining probability distributions that can include both recursive and non-recursive influences for random variables Boolean and non-Boolean random variables on finite and/or countably infinite domains.

Definition 6: Let $\mathcal{T} = \{\mathcal{F}_1, \mathcal{F}_2 \ldots\}$ be a set of MFrags. The sequence $\phi_d(\varepsilon_d) \rightarrow \phi_{d-1}(\varepsilon_{d-1}) \rightarrow \ldots \rightarrow \phi_0(\varepsilon_0)$ is called an *ancestor chain* for $\mathcal{T}$ if there exist $\mathcal{B}_0, \ldots, \mathcal{B}_d$ such that:

6a. Each $\mathcal{B}_i$ is a binding set for one of the MFrags $\mathcal{F}_{ji} \in \mathcal{T}$;

6b. The random variable instance $\phi_i(\varepsilon_i)$ is obtained by applying the bindings in $\mathcal{B}_i$ to a resident random variable term $\phi_i(\theta_i)$ of $\mathcal{F}_{ji}$;

6c. For $i<d$, either:

- $\phi_{i+1}(\varepsilon_{i+1})$ is obtained by applying the bindings in $\mathcal{B}_i$ to an input random variable term $\phi_{i+1}(\theta_{i+1})$ of $\mathcal{F}_{ji}$, and there is an influencing configuration for $\phi_i(\varepsilon_i)$ and $\mathcal{B}_i$ that contains $\phi_{i+1}(\theta_{i+1})$, or
- $\phi_{i+1}(\varepsilon_{i+1})$ is obtained by applying the bindings in $\mathcal{B}_i$ to a context value assignment term $\phi_{i+1}(\theta_{i+1})$ of $\mathcal{F}_{ji}$.

The integer d is called the *depth* of the ancestor chain. The random variable instance $\phi_j(\varepsilon_j)$ is an *ancestor* of $\phi_0(\varepsilon_0)$ if there exists an ancestor chain $\phi_d(\varepsilon_d) \rightarrow \ldots \rightarrow \phi_j(\varepsilon_j) \rightarrow \ldots \rightarrow \phi_0(\varepsilon_0)$ for $\mathcal{T}$. ■

Definition 7: Let $\mathcal{T} = \{ \mathcal{F}_1, \mathcal{F}_2 \ldots \}$ be a set of MFrags. Let $\mathcal{V}_\mathcal{T}$ denote the set of random variable terms contained in the $\mathcal{F}_i$, and let $\mathcal{N}_\mathcal{T}$ denote the set of random variable instances $\mathcal{T}$ that can be formed from $\mathcal{V}_\mathcal{T}$. $\mathcal{T}$ is a *simple MTheory* if the following conditions hold:

7a. No cycles. No random variable instance is an ancestor of itself;[15]

7b. Bounded causal depth. For any random variable instance $\phi(\varepsilon) \in \mathcal{N}_\mathcal{T}$ containing the (possibly empty) unique identifier symbols ε, there exists an integer $N_{\phi(\varepsilon)}$ such that if $\phi_d(\varepsilon_d) \rightarrow \phi_{d-1}(\varepsilon_{d-1}) \rightarrow \ldots \rightarrow \phi(\varepsilon)$ is an ancestor chain for $\mathcal{T}$, then $d \leq N_{\phi(\varepsilon)}$. The smallest such $N_{\phi(\varepsilon)}$ is called the *depth* $d_{\phi(\varepsilon)}$ of $\phi(\varepsilon)$.

7c. Unique home MFrags. For each $\phi(\varepsilon) \in \mathcal{N}_\mathcal{T}$, there exists exactly one MFrag $\mathcal{F}_{\phi(\varepsilon)} \in \mathcal{T}$, called the *home MFrag* of $\phi(\varepsilon)$, such that $\phi(\varepsilon)$ is an instance of a resident random variable $\phi(\theta)$ of $\mathcal{F}_{\phi(\varepsilon)}$.[16]

7d. Recursive specification. $\mathcal{T}$ may contain infinitely many domain-specific MFrags, but if so, the MFrag specifications must be recursively enumerable. That is, there must be an algorithm that lists a specification (i.e., an algorithm that generates the input, output, context random variables, fragment graph, and local distributions) for each MFrag in turn, and eventually lists a specification for each MFrag of $\mathcal{T}$. ■

Theorem 1: Let $\mathcal{T} = \{ \mathcal{F}_1, \mathcal{F}_2 \ldots \}$ be a simple MTheory. There exists a joint probability distribution $\mathcal{P}^{\text{gen}}_\mathcal{T}$ on the set of instances of the random variables of its MFrags that is consistent with the local distributions assigned by the MFrags of $\mathcal{T}$. ■

[15] This condition can be relaxed as long as it can be demonstrated that the local distributions are specified non-circularly.

[16] It may be desirable to relax this condition. For example, in an independence of causal influence model, it might be convenient to specify influences due to different clusters of related causes to be specified in separate MFrags. In a polymorphic version of MEBN logic, it might be convenient to specify local distributions for separate subtypes in separate MFrags. It is clear that the main results would remain valid under appropriately weakened conditions.

The proof of Theorem 1 is found in the appendix.

MEBN inference is defined as conditioning the joint probability distribution implied by Theorem 1 on the proposition that all findings have value T. This conditional distribution clearly exists if there is a non-zero probability that all findings have value T. However, when there is an infinite sequence of findings or there are findings on quantifier random variables, then any individual sequence of findings may have probability zero even though some such sequence is certain to occur. For example, each possible realization of an infinite sequence of rolls of a fair die has zero probability, yet some such sequence will occur if tossing continues indefinitely. Although any individual sequence of tosses has probability zero, the assumption that the die is fair allows us to draw conclusions about properties of the sequences of tosses that will actually occur. In particular, it is a practical (although not a logical) certainty that if the die is fair, then the limiting frequency of rolling a four will be once in every six trials. That is, although a sequence having limiting probability 1/6 and a sequence having limiting probability 1/3 both have probability zero, the former is infinitely more probable than the latter. Practical certainties about stochastic phenomena are formalized as propositions that are true "almost surely" or "except on a set of measure zero" (Billingsley, 1995). Almost sure propositions are not true in all possible interpretations of the FOL theory corresponding to an MTheory, but the set of worlds in which they are true has probability 1 under the probability distribution represented by the MTheory.

Definition 8: The distribution $\mathcal{P}_{\mathcal{T}'}^{\text{gen}}$ is called the *generative or prior distribution* for $\mathcal{T}'$. Let $\underline{\Phi}$={Φ(ψ_1=α_1), Φ(ψ_2=α_2), … } be the finding MFrags for $\mathcal{T}'$. A *finding alternative* for $\mathcal{T}'$ is a set {Φ(ψ_1=α'_1), Φ(ψ_2=α'_2), … } of values for the finding random variables of $\mathcal{T}'$, possibly assigning different values to the finding random variables from the values assigned by $\mathcal{T}'$. Finding alternatives represent counterfactual worlds for $\mathcal{T}'$ – that is, worlds that were *a priori* possible but are different from the world asserted by the findings to have occurred. ■

Corollary 2: Let $\mathcal{T}'$ be an MTheory with findings {Φ(ψ_1=α_1), Φ(ψ_2=α_2), … }. Then a conditional distribution exists for $\mathcal{P}_{\mathcal{T}'}^{\text{gen}}$ given {ψ_1, ψ_2, …}. This distribution is unique in the sense that any two such distributions differ at most on a set of finding alternatives assigned probability zero by $\mathcal{P}_{\mathcal{T}'}^{\text{gen}}$. ■

Corollary 2 follows immediately from Theorem 1 and the Radon-Nikodym Theorem (Billingsley, 1995). The distribution $\mathcal{P}_{\mathcal{T}'}\left(\xi_1,\xi_2 \ldots \mid \Phi(\psi_1=\alpha_1),\Phi(\psi_2=\alpha_2),\ldots\right)$ for {ξ_1, ξ_2, …} obtained by conditioning $\mathcal{P}_{\mathcal{T}'}^{\text{gen}}$ on all findings having value T is called the posterior distribution for $\mathcal{T}'$ given its findings. The posterior distribution is abbreviated $\mathcal{P}_{\mathcal{T}'}\left(\xi \mid \Phi(\psi=\alpha)\right)$. The following corollary states that even when the joint probability of an infinite sequence of findings is zero, if the individual findings have positive probability and a limiting posterior distribution exists, it is unique.

Corollary 3: Suppose $\mathcal{P}_{\mathcal{T}'}^{\text{gen}}$ assigns strictly positive probability to the event that the first *n* findings Φ(ψ_1=α_1), Φ(ψ_2=α_2), …, Φ(ψ_n=α_n) all have value T. Then there is a unique conditional distribution for $\mathcal{P}_{\mathcal{T}'}^{\text{gen}}$ given that the first *n* findings Φ(ψ_1=α_1), Φ(ψ_2=α_2), …, Φ(ψ_n=α_n) all have value T. Furthermore, if the positivity condition holds for all *n* and a limiting distribution $\lim_{n\to\infty}\mathcal{P}_{\mathcal{T}'}\left(\xi_1,\xi_2 \ldots \mid \Phi(\psi_1=\alpha_1),\Phi(\psi_2=\alpha_2),\ldots,\Phi(\psi_n=\alpha_n)\right)$ exists, then the limit is unique. ■

Corollary 3 is a straightforward consequence of basic identities of conditional probability.

MTheories represent a conjugate family of probability distributions. That is, if finding random variables are added to an MTheory, the result is another MTheory. Section 5 below discusses some operations that can be performed on MFrags to transform an MTheory into another MTheory that represents the same probability distribution. Such transformations can be used to improve the computational efficiency of MEBN inference, to find a more compact and/or cognitively natural representation for an MTheory, or to translate between different partially overlapping MTheories.

Although simple MTheories are adequate to express probability distributions over interpretations of arbitrary finitely axiomatizable FOL theories, expressing structural uncertainty with simple MTheories is cumbersome. Structural uncertainty can be more compactly expressed using mixture MTheories, which provide the logical basis for a typed version of MEBN (Costa and Laskey, 2005).

***Definition* 9:** If the posterior distribution for $\mathcal{T}$ $\mathcal{P}_{\mathcal{T}}\left(\xi \mid \Phi(\psi=\alpha)\right)$ is not unique, $\mathcal{T}$ is said to be *disconfirmed by its findings*. ■

***Definition* 10:** A *mixture MTheory* is a set $\mathcal{T} = \{ (\mathcal{T}_1, p_1), (\mathcal{T}_2, p_2), \ldots \}$ of MFrags satisfying the following conditions:

10a. Each $\mathcal{T}_i$ is a simple MTheory;

10b. None of the $\mathcal{T}_i$ is disconfirmed by its findings;

10c. The p_i are positive numbers that sum to 1;

10d. There must be an effective procedure for computing each p_i;

10e. For each finding $\Phi(\psi{=}\varepsilon)$ of one of the $\mathcal{T}_i$, and for each $j{\neq}i$, the posterior distribution of $\mathcal{T}_j$ assigns probability 1 to $\psi{=}\varepsilon$.

The $\mathcal{T}_i$ are called *mixture components* with *mixture weights* p_i. An *MTheory* is either a simple MTheory or a mixture MTheory. ■

***Corollary* 4:** Let $\mathcal{T}$ be an MTheory. Then there exists a joint probability distribution on the set of instances of the random variables in its MFrags that is consistent with the local distributions assigned by the MFrags of $\mathcal{T}$. ■

Corollary 4 is an immediate consequence of Theorem 1.

4.4 Random Variable Semantics and Tarski Semantics

In the standard semantics for first-order logic developed by Tarski (1944), a FOL theory is interpreted in a domain by assigning each constant symbol to an element of the domain, each function symbol on *k* arguments to a function mapping *k*-tuples of domain elements to domain elements, and each predicate symbol on *k* arguments to a subset of *k*-tuples of domain elements corresponding to the entities for which the predicate is true (or, equivalently, to a function mapping *k*-tuples of domain elements to truth-values). If the axioms are consistent, this can be done in such a way that all the axioms of the theory are true assertions about the domain, given the correspondences defined by the interpretation. Such an interpretation is called a model for the axioms.

MTheories define probability distributions over interpretations of an associated FOL theory. Each *k*-argument random variable in an MTheory represents a function mapping *k*-tuples of unique identifiers to possible values of the random variable. Any function consistent with the logical constraints of the MTheory is allowable, and the probability that the function takes on given values is specified by the joint probability distribution represented by the MTheory. For Boolean random variables, the possible values of the function are T, F, and $\perp$; for non-Boolean random variables, the possible values are entity identifiers. Through the correspondence between entity identifiers and entities in the domain, a non-Boolean random variable also represents a function mapping *k*-tuples of domain entities either to domain entities (for non-Boolean random variables) or to truth-values of assertions about the domain (for random variables).

Interpreting random variable symbols as functions on the unique identifiers is consistent with the way random variables are formalized in mathematical statistics. A random variable is defined as a function that maps a sample space endowed with a probability measure to a set of possible outcomes (e.g., Billingsley, 1995; DeGroot and Schervish, 2002). In the standard definition, the global joint distribution is taken as given, and distributions for smaller sets of random variables are obtained by marginalizing the global joint probability measure. MEBN logic provides a logically coherent means of specifying a global joint distribution by composing local conditional distributions involving small sets of random variables. Formerly, this could be achieved only for restricted kinds of distributions. Standard Bayesian networks allow joint distributions on a finite number of random variables to be composed from locally defined conditional distributions. There are well-known special cases, such as independent and identically distributed trials or Markov chains, for which joint distributions on infinite sets of random variables can be composed from locally defined conditional distributions. MEBN logic provides the ability to construct joint distributions from local elements for a very general class of distributions on infinite collections of random variables, and is the first Bayesian logic that has been shown to be capable of defining a joint distribution over interpretations of any finitely axiomatizable theory in classical first-order logic.

Consider an MTheory $\mathcal{T}_M'$ in a language $\mathcal{L}_M$ having domain-specific non-Boolean random variable symbols $\mathcal{X}=\{\xi_i\}$, domain-specific constant symbols $\mathcal{A}=\{\alpha_i\}$, domain-specific Boolean random variable symbols $\mathcal{B}=\{\beta_i\}$, exemplar symbols $S=\{\sigma_{\phi i}\}$ and entity identifier symbols $\mathcal{E}=\{\varepsilon_i\}$. It is assumed that the sets $\mathcal{X}$, $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{E}$ are pairwise disjoint, are either finite or countably infinite, and do not contain the symbols T, F, or $\perp$. It is assumed that S contains a distinct exemplar symbol $\sigma_{\phi i}\notin \mathcal{X}\cup\mathcal{A}\cup\mathcal{B}\cup\mathcal{E}\cup\{\mathsf{T},\mathsf{F},\perp\}$ for each pair consisting of an open Boolean random variable term $\phi(\gamma_1,\ldots,\gamma_n)$ of $\mathcal{L}_M$ and index i of an ordinary variable γ_i occurring in $\phi(\gamma_1,\ldots,\gamma_n)$.

To facilitate the comparison with Tarski semantics, we begin by considering only the quantifier-free part of $\mathcal{T}_M'$. Suppose $\mathcal{T}_M'$ satisfies the following conditions:

FOL1: There are no quantifier random variable terms among the context terms in any of the MFrags of $\mathcal{T}_M'$, and no simple random variable term of $\mathcal{T}_M'$ has a quantifier random variable term as a parent.

FOL2: Random variables $\xi\in\mathcal{X}$ or $\beta\in\mathcal{B}$ have value $\perp$ if any of their arguments belong to $\{\mathsf{T}, \mathsf{F}, \perp\}$;

FOL3: If the values of all arguments to a non-Boolean random variable ξ belong to $\mathcal{E}$, then the value of ξ belongs to $\mathcal{E}$ with probability 1;

FOL4: Any constant symbol $\alpha\in\mathcal{A}$ has value in $\mathcal{E}$ with probability 1;

FOL5: If the values of all arguments to a Boolean random variable β belong to $\mathcal{E}$, then the value of β belongs to $\{\mathsf{T}, \mathsf{F}\}$ with probability 1.

Given these conditions, $\mathcal{P}^{\text{gen}}_{\mathcal{T}_M'}$ generates random interpretations of the domain-specific random variable symbols of $\mathcal{L}_M$ in the domain $\{\varepsilon\in\mathcal{E} : \Diamond(\varepsilon)\neq\perp)\}$ of meaningful entity identifiers. That is, for each constant symbol, $\mathcal{P}^{\text{gen}}_{\mathcal{T}_M}$ generates a meaningful entity identifier. For each non-Boolean random variable symbol, $\mathcal{P}^{\text{gen}}_{\mathcal{T}_M}$ generates a random function mapping k-tuples of meaningful entity identifiers to meaningful entity identifiers. For each Boolean random variable symbol, $\mathcal{P}^{\text{gen}}_{\mathcal{T}_M}$ generates a random function mapping k-tuples of meaningful entity identifiers to $\{\mathsf{T}, \mathsf{F}\}$ (or equivalently, the subset of k-tuples for which the randomly generated function has value T).

A classical first-order theory $\mathcal{T}_F'$ that represents the logical content of $\mathcal{T}_M'$ is defined as follows:

1. The language $\mathcal{L}_F$ for $\mathcal{T}_F'$ has function symbols $\mathcal{X}$, constant symbols $\mathcal{A}\cup\mathcal{E}\cup\{\perp\}$, and predicate symbols $\mathcal{B}$, where the number of arguments for functions and predicates in $\mathcal{L}_F$ is the same as the number of arguments for the corresponding random variables in $\mathcal{T}_M'$.
2. For each pair ε_1 and ε_2 of distinct entity identifiers, $\mathcal{T}_F'$ contains an axiom $(\varepsilon_1=\varepsilon_2)\Rightarrow (\varepsilon_1=\perp) \wedge (\varepsilon_2=\perp)$.
3. For each non-Boolean random variable symbol ξ, $\mathcal{T}_F'$ contains axioms asserting that no instance of ξ may take on values outside the set of possible values as defined in the home MFrag for ξ.
4. If a local distribution in a domain-specific MFrag of $\mathcal{T}_M'$ assigns probability zero to possible value ε of a non-Boolean resident random variable $\xi(x)$ for some set $\#S_{W\xi(x)}$ of influence counts, there is an axiom of $\mathcal{T}_F'$ specifying that the function corresponding to $\xi(x)$ is not equal to ε when the context constraints hold and the parents of $\xi(x)$ satisfy $\#S_{W\xi(x)}$. Each such axiom is universally quantified over any ordinary variables appearing in ξ and/or its parents and/or the context random variables in the home MFrag of ξ. Formally, $\mathcal{T}_F'$ contains an axiom $\forall x\ ((\kappa(x)\wedge\#S_{W\xi(x)}) \Rightarrow \neg(\xi(x)=\varepsilon))$. Here, $\kappa(x)$ and $\#S_{W\xi(x)}$ denote formulae in $\mathcal{L}_F$ asserting that the context constraints hold and that the influence counts for the parents of $\xi(x)$ are equal to $\xi(x)$; and x denotes any ordinary variables on which ξ, κ, and/or the parents of ξ depend.
5. If a local distribution in a domain-specific MFrag of $\mathcal{T}_M'$ assigns probability one to T for a Boolean random variable $\beta(x)$ for some set $\#S_{W\beta(x)}$ of influence counts, there is an axiom of $\mathcal{T}_F'$ specifying that the predicate $\beta(x)$ is true under these conditions. That is, $\mathcal{T}_F'$ contains an axiom $\forall x\ ((\kappa(x)\wedge\#S_{W\beta(x)}) \Rightarrow \beta(x))$. Here, $\kappa(x)$ and $\#S_{W\beta(x)}$ denote formulae in $\mathcal{L}_F$ asserting that the context constraints hold and that the influence counts for the parents of $\beta(x)$ are equal to $\beta(x)$, respectively; and x denotes any ordinary variables on which β, κ, and/or the parents of β depend.
6. If a local distribution in a domain-specific MFrag of $\mathcal{T}_M'$ assigns probability one to F for a Boolean random variable $\beta(x)$ for some set $\#S_{W\beta(x)}$ of influence counts, there is an axiom of $\mathcal{T}_F'$ specifying that the predicate $\beta(x)$ is false under these conditions. That is, $\mathcal{T}_F'$ contains an axiom $\forall x\ ((\kappa(x)\wedge\#S_{W\beta(x)}) \Rightarrow \neg\beta(x))$. Here, $\kappa(x)$ and $\#S_{W\beta(x)}$ denote formulae in $\mathcal{L}_F$ asserting that the context constraints hold and that the influence counts for the parents of $\beta(x)$ are equal to $\beta(x)$, respectively; and x denotes any ordinary variables on which β, κ, and/or the parents of β depend.

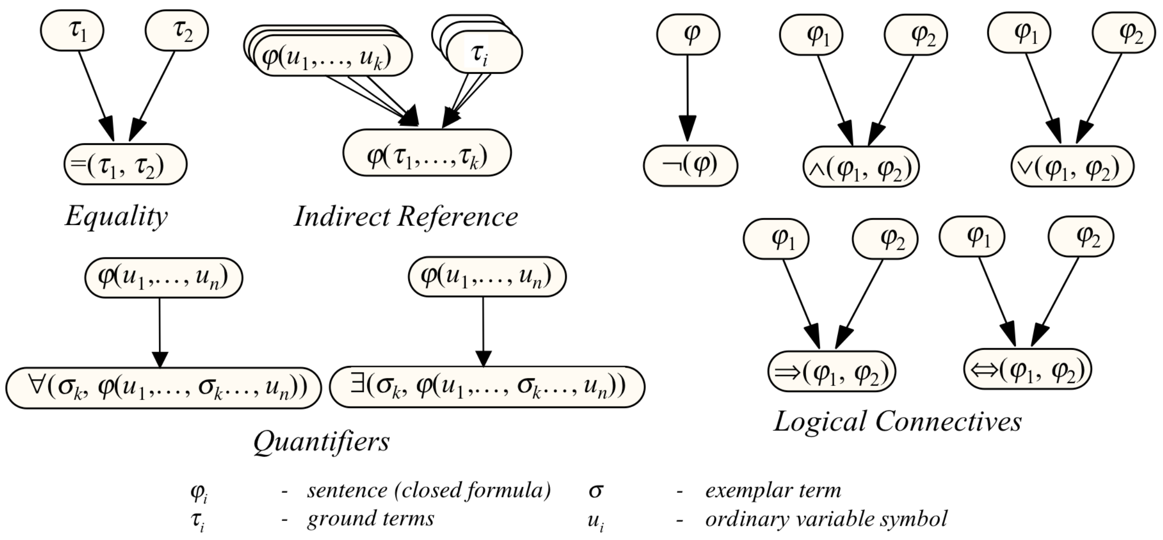

τ_1
τ_2
$=(\tau_1, \tau_2)$
Equality
$\varphi(u_1,\ldots,u_k)$
τ_i
$\varphi(\tau_1,\ldots,\tau_k)$
Indirect Reference
φ
$\neg(\varphi)$
φ_1
φ_2
$\wedge(\varphi_1, \varphi_2)$
$\vee(\varphi_1, \varphi_2)$
$\Rightarrow(\varphi_1, \varphi_2)$
$\Leftrightarrow(\varphi_1, \varphi_2)$
Logical Connectives
$\varphi(u_1,\ldots,u_n)$
$\forall(\sigma_k, \varphi(u_1,\ldots,\sigma_k\ldots,u_n))$
$\exists(\sigma_k, \varphi(u_1,\ldots,\sigma_k\ldots,u_n))$
Quantifiers
φ_i - sentence (closed formula)
τ_i - ground terms
σ - exemplar term
u_i - ordinary variable symbol

Figure 8: Logical MFrags

The logical combination MFrags (see Figure 8) ensure that any interpretation generated by $\mathcal{P}^{\text{gen}}_{\mathcal{T}'_M}$, specifies a well-defined truth-value for any sentence of $\mathcal{T}'_F$. The assumptions FOL1-FOL5 ensure that these truth-values satisfy the axioms defining $\mathcal{T}'_F$. That is, $\mathcal{P}^{\text{gen}}_{\mathcal{T}'_M}$ generates random models of the axioms of $\mathcal{T}'_F$. However, there may be sentences satisfiable under the axioms of $\mathcal{T}'_F$ to which $\mathcal{P}^{\text{gen}}_{\mathcal{T}'_M}$ assigns probability zero. When a satisfiable sentence of $\mathcal{T}'_F$ is assigned probability zero by $\mathcal{P}^{\text{gen}}_{\mathcal{T}'_M}$, there is no assurance that a well-defined conditional distribution exists given that the corresponding Boolean random variable has value T. The following additional condition ensures that a well-defined conditional distribution exists given any finite set of logically possible findings on random variables of $\mathcal{T}'_M$.

FOL6: If $\phi(\gamma_1,\ldots,\gamma_n)$ is a Boolean random variable of $\mathcal{T}'_M$ that corresponds to a satisfiable formula of $\mathcal{T}'_F$, and $\sigma_{\phi i}$ is the exemplar symbol for ordinary variable γ_i in $\phi(\gamma_1,\ldots,\gamma_n)$, then $\mathcal{P}^{\text{gen}}_{\mathcal{T}'_M}$ assigns strictly positive probability to the value T for the quantifier random variables $\theta(\sigma_{\phi 1}, \theta(\sigma_{\phi 2}, \ldots, \theta(\sigma_{\phi n}, \phi(\sigma_{\phi 1}, \sigma_{\phi 2}, \ldots, \sigma_{\phi n}))))$, where θ is one of the quantifier symbols $\exists$ or $\forall$.

***Corollary 5*:** Suppose $\mathcal{T}'_M$ satisfies FOL1-FOL6, and suppose that $\mathcal{T}'_F$ is the first-order theory, constructed as above, expressing the logical content of $\mathcal{T}'_M$. Let $\{\Phi(\psi_1=\alpha_1), \Phi(\psi_2=\alpha_2), \ldots, \Phi(\psi_n=\alpha_n)\}$ be a finite set of findings such that the conjunction of the $(\psi_i=\alpha_i)$ is satisfiable as a sentence of $\mathcal{T}'_F$. Then the posterior distribution $\mathcal{P}_{\mathcal{T}'_M}\left(\xi \mid \Phi(\psi=\alpha)\right)$ exists and is unique. ■

Corollary 5 is a straightforward consequence of Corollary 3. Specifying a generative distribution that satisfies FOL1-FOL5 is relatively straightforward. A construction is provided in Section 4.5 of an MTheory $\mathcal{T}'_{M*}$ for which $\mathcal{P}^{\text{gen}}_{\mathcal{T}'_{M*}}$ satisfies FOL6.

An MTheory is interpreted in a domain of application by associating each entity identifier symbol with an entity in the domain. Through this correspondence between identifiers and the entities they represent, the probability distribution on entity identifiers induces a probability distribution on attributes of and relationships among entities in the domain of application. In particular, although the generative distribution for an MTheory constructs interpretations in the countable domain of entity identifiers, an MTheory can be applied to reason about domains of any cardinality. Under the assumption that the entities associated with the entity identifiers constitute a representative sample of entities in the domain, statistical conclusions drawn about the domain are valid for domains of any cardinality.

Because MEBN allows joint distributions to be expressed over arbitrary first-order theories, MEBN can be used to define a Bayesian semantics for constructive mathematics. Mathematics is commonly (although not universally) viewed as being founded on set theory and first-order logic. When a mathematician claims to have found a proof for a theorem, typically what he or she means is a proof that the community of mathematicians agrees could be formalized, with sufficient diligence, as a formal derivation using the rules of first-order logic from the axioms of set theory (cf., Enderton, 2001). The most commonly used axiom system for set theory, the Zermelo-Frankel system with the axiom of choice (ZFC), has infinitely many axioms. There is another axiom system for set theory, the von Neumann-Bernays-Gödel system (NBG), that has finitely many axioms. Although no contradiction has been found in either of these axiom systems, mathematicians are not certain of their consistency. If one is consistent, then so is the other, and the two axiom sets have been shown to be essentially of equal strength (c.f., Stoll, 1963). A construction is provided below that implicitly defines a joint distribution on models of any consistent finite set of first-order logic sentences. Thus, one can construct an MTheory that, if NBG is consistent, implicitly represents a joint distribution on models of NBG. If NBG is inconsistent, then SSBN construction would, in principle, if continued persistently for a long enough period of time, find a proof that there is a contradiction in NBG (as would any refutation-complete proof procedure for classical first-order logic). We could conceive of the enterprise of mathematics as a collective process of performing approximate SSBN construction for MTheories whose findings consist of the NBG axioms together with proper axioms defining theory-specific mathematical content. In this view, finding a proof of a theorem would correspond to constructing a SSBN in which the proven sentence has value T with probability 1. Mathematicians sometimes say informally that a proposition they have not proven is "probably true." Such a statement is meaningless in the standard formalization of classical FOL. The proposition may follow from the axioms; it may generate a contradiction when conjoined with the axioms; or its truth-value given the axioms may be indeterminate. According to classical FOL, one of these situations is the actual state of affairs, and it is meaningless to speak of probabilities. Taking a Bayesian view, if we assume that the mathematician's axiom set is consistent, an MTheory implicitly represents a probability distribution over models of the axioms. Any given proposition has a probability in the closed unit interval. A probability of one corresponds to a theorem; a probability of zero corresponds to a statement inconsistent with the axioms, and intermediate values correspond to propositions having indeterminate truth-values. The probability assigned by the mathematician's current SSBN may be equal to the probability implicitly defined by the MTheory, or it may approximate that probability. In the latter case, the mathematician might say the proposition is "probably true" if the approximate SSBN assigns it a high probability. This might mean that the mathematician thinks it likely that a proof will eventually be found, or that a proof could likely be found by adding additional high probability axioms. These two alternatives can themselves be formalized as higher-order probability statements.

Of course, there is no scientific justification for claiming that the brains of mathematicians actually store MTheories, or that mathematicians are actually performing SSBN construction when they are developing proofs. Nevertheless, there are a number of advantages to formalizing mathematics as approximate Bayesian logic. In this view, it is perfectly meaningful for a mathematician to say he or she thinks NBG is probably consistent, or that any given mathematical hypothesis is probably a theorem. The mathematical enterprise is both meaningful and useful even if it should turn out that NBG is inconsistent. Alternative proposals for the foundations of mathematics, such as category theory, can also be formalized as Bayesian logic (although there are no guarantees that a joint distribution exists over models of proposed foundational theories with infinite numbers of axioms, even if the axioms are not mutually contradictory). One can formalize exploratory mathematics decision theoretically, as well as the collective process of identifying and eliminating errors in proofs.

Important advantages of MEBN random variable semantics are clarity and modularity. For example, we could add a new collection of MFrags to our equipment diagnosis MTheory, say for reasoning about the vacation and holiday schedule of maintenance technicians, without affecting the probabilities of any assertions unrelated to the change. Furthermore, the probability distribution represented by an MTheory is a well-defined mathematical object independent of its correspondence with actual objects in the world, having a clearly specified semantics as a probability distribution on $\mathcal{E}\cup\{\perp\}$. Its adequacy for reasoning about the actual world rests in how well the relationships in the model reflect the empirical relationships among the entities to which the symbols refer in a given domain of application. Our approach thus enforces a distinction between logical and empirical aspects of a representation and provides a clearly defined interface between the two. This supports a principled approach to empirical evaluation and refinement of domain ontologies.

4.5 A Generative Distribution for First-Order Logic

This section constructs a generative MTheory $\mathcal{T}_{M^*}$ such that $\mathcal{P}^{\text{gen}}_{\mathcal{T}_{M^*}}$ places positive probability on value T for any Boolean random variable ϕ that corresponds to a satisfiable sentence in the first-order theory $\mathcal{T}_{F^*}$ constructed from $\mathcal{T}_{M^*}$ as described in Section 4.4 above.

We assume there is a total ordering φ_1, φ_2, … of the domain-specific constant, non-Boolean and Boolean random variable terms $\varphi_i\in\mathcal{A}\cup\mathcal{X}\cup\mathcal{B}$, and a total ordering ε_1, ε_2, …$\in\mathcal{E}$ of entity identifiers. The domain-specific MFrags of a generative MTheory must define a distribution for each simple open random variable term $\varphi_i(u_1,\ldots,u_{n_i})$, where the u_j are ordinary variables and n_i is the number of arguments taken by φ_i. A distribution is also defined for the exemplar constants. The remaining random variables are defined via the logical MFrags of Figure 8.

The joint distribution for simple open random variables and exemplar constants is defined as follows. Let ψ_1, ψ_2, … be a total ordering of the quantifier random variables; let π_1, π_2, … be a strictly positive probability distribution on the entity identifiers, and let $0 < \theta, \rho < 1$ be real numbers. We use the notation ψ_k to refer a quantifier random variable and σ_{ψ_k} to refer to the exemplar constant for ψ_k. That is, ψ_k denotes a Boolean random variable of the form $\forall(\sigma_{\psi_k},\phi(\sigma_{\psi_k}))$ or $\exists(\sigma_{\psi_k},\phi(\sigma_{\psi_k}))$, where $\phi(u)$ is an open Boolean random variable called the *body* of ψ_k.

Exemplar constant distributions: The distributions for exemplar constants are defined inductively such that the exemplar term $\Diamond(\sigma_{\psi_k})$ has value $\perp$ in models in which ψ_k is constrained to have value F, and otherwise is sampled randomly from the entity identifiers that are logically possible values for σ_{ψ_k}. Specifically:

- The parents of $\Diamond(\sigma_{\psi_k})$ are $\Diamond(\sigma_{\psi_1}),\Diamond(\sigma_{\psi_2}),\ldots$, and $\Diamond(\sigma_{\psi_{k-1}})$.
- It is assumed $i<k$ for the inductive definition that if $\Diamond(\sigma_{\psi_i})=\perp$ then ψ_k has value F. Conditional $\Diamond(\sigma_{\psi_1}),\Diamond(\sigma_{\psi_2}),\ldots$, and $\Diamond(\sigma_{\psi_{k-1}})$, the distribution of $\Diamond(\sigma_{\psi_k})$ is defined as follows

   - If ψ_k is unsatisfiable as a formula of $\mathcal{L}_{F^*}$ given the constraints on $\psi_1, \ldots, \psi_{k-1}$ implied by the values of its parents, then $\Diamond(\sigma_{\psi_k})$ has value $\perp$ with probability 1.
   - If $\neg\psi_k$ is unsatisfiable as a formula of $\mathcal{L}_{F^*}$ given the constraints on $\psi_1, \ldots, \psi_{k-1}$ implied by the values of its parents, then $\Diamond(\sigma_{\psi_i})$ has value ε_j with probability π_j.
   - Otherwise, $\Diamond(\sigma_{\psi_i})$ has value$\perp$ with probability θ and ε_j with probability $(1-\theta)\pi_j$.

Domain-specific random variable distributions: The distribution of $\varphi_k(u_1,\ldots,u_{n_k})$ is defined as follows.

- The parents of $\varphi_k(u_1,\ldots,u_{n_k})$ are:
   - $\varphi_i(v_1,\ldots,v_{n_i})$ for all $i<k$, where v_j is a different ordinary variable than u_j, implying that all instances of $\varphi_i(v_1,\ldots,v_{n_i})$ are parents of each instance of $\varphi_k(u_1,\ldots,u_{n_k})$;
   - Instances of $\varphi_k(v_1,\ldots,v_{n_k})$ such that the entity identifier bound to each u_j is equal to or precedes the entity identifier bound to v_j, and strictly precedes it for at least one j. (This can be specified by a recursive definition with appropriate context constraints);
   - The identity random variables $\Diamond(e)$.
- If $\varphi_k(u_1,\ldots,u_{n_k})$ is a non-Boolean random variable, its probability distribution is calculated as follows. For any binding $\varepsilon_1,\ldots,\varepsilon_{n_k}$ of entity identifiers to the variables $u_1,\ldots,u_{n_k}$, the value $\varphi_k(\varepsilon_1,\ldots,\varepsilon_{n_k})=\varepsilon_j$ is assigned randomly, with probability proportional to π_j, from among the entity identifiers whose value is consistent with the satisfiability constraints implied by the assignment of values to the parents of $\varphi_k(\varepsilon_1,\ldots,\varepsilon_{n_k})$.
- If $\varphi_k(u_1,\ldots,u_{n_k})$ is a Boolean random variable, its probability distribution is calculated as follows. For any binding $\varepsilon_1,\ldots,\varepsilon_{n_k}$ of entity identifiers to the variables $u_1,\ldots,u_{n_k}$:
   - $\varphi_k(\varepsilon_1,\ldots,\varepsilon_{n_k})$ has value T if $\neg\varphi_k(\varepsilon_1,\ldots,\varepsilon_{n_k})$ is inconsistent with the satisfiability constraints implied by the assignment of values to the parents of $\varphi_k(\varepsilon_1,\ldots,\varepsilon_{n_k})$;
   - $\varphi_k(\varepsilon_1,\ldots,\varepsilon_{n_k})$ has value F if $\varphi_k(\varepsilon_1,\ldots,\varepsilon_{n_k})$ is inconsistent with the satisfiability constraints implied by the assignment of values to the parents of $\varphi_k(\varepsilon_1,\ldots,\varepsilon_{n_k})$;
   - Otherwise, $\varphi_k(u_1,\ldots,u_{n_k})$ has value T with probability ρ and F with probability $(1-\rho)$.

Theorem 6: If ψ is a closed Boolean random variable corresponding to a satisfiable sentence of $\mathcal{L}_{F^*}$, then $\mathcal{P}^{\text{gen}}_{\mathcal{T}_{M^*}}$ places non-zero probability on the value T for ψ. ■

Proof: The above construction ensures that if ψ corresponds to a satisfiable sentence of $\mathcal{T}'_{F^*}$, then there is a non-zero probability that $\Diamond(\sigma_{\neg\psi})$ has value $\perp$. When $\Diamond(\sigma_{\neg\psi})$ has value $\perp$, the local distributions for the domain-specific random variables are assigned in a way that constrains ψ to have value T. Therefore, there is a non-zero probability that ψ has value T.

4.6 Inference in MEBN Logic: Situation-Specific Bayesian Networks

As noted above, MEBN inference conditions the prior distribution represented by an MTheory on its findings. The Appendix presents an inference algorithm that uses knowledge-based model construction (Wellman, et al., 1992) to produce a sequence of *approximate situation-specific Bayesian networks*. Mahoney and Laskey (1998) define a situation-specific Bayesian network (SSBN) as a minimal Bayesian network sufficient to compute the response to a query, where a query consists of obtaining the posterior distribution for a set of target random variable instances given a set of finding random variable instances. Their *simple bottom-up construction* algorithm for constructing situation-specific Bayesian networks is provided in the Appendix. The algorithm begins with a *query set* consisting of a finite set of target random variable instances and a finite set of finding random variable instances. These are combined to construct an approximate SSBN. The approximate SSBN has an arc between a pair of random variables when one is an instance of an influencing configuration for the other in its home MFrag. At each step, the algorithm obtains a new approximate SSBN by adding findings, instantiating the home MFrags of the random variables in the query set and their ancestors, adding the resulting random variable instances to the query set, removing any that are not relevant to the query, and combining the resulting set of random variable instances into a new approximate SSBN. This process continues until either there are no changes to the approximate SSBN, or a stopping criterion is met. If the algorithm is run without a stopping criterion, then if SSBN construction terminates, the resulting SSBN provides an exact response to the query or an indication that the findings are inconsistent. When the algorithm does not terminate, it defines an anytime process that yields a sequence of approximate SSBNs converging to the correct query response if one exists. In general, there may be no finite-length proof that a set of findings is consistent, but inconsistent findings can be detected in a finite number of steps of SSBN construction.

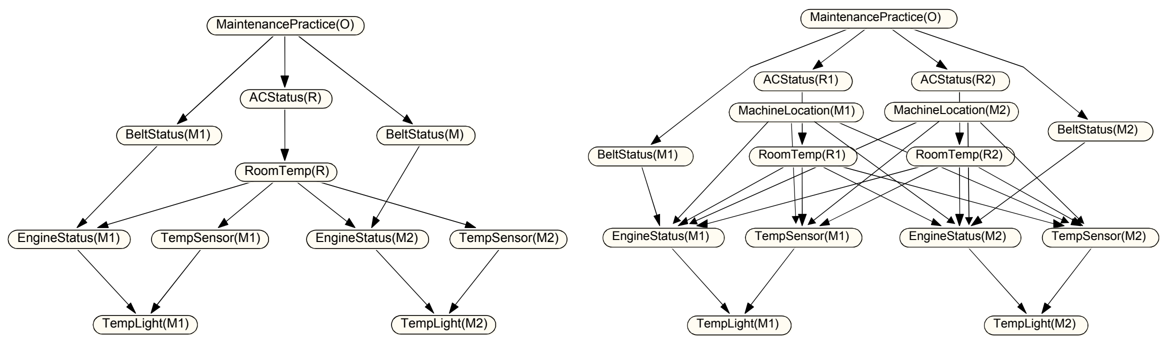
Figure 9 shows two SSBNs constructed from the MTheory of Figure 2 for a query on the engine status of two machines, the first for the case in which the two machines are known to be in the same room, and the second for the case in which the two machines are known to be in different rooms. In the first case, learning that the engine in one machine is overheated results in an increase in the probability that the other engine is overheated; in the second case, the same information has almost no effect on the probability distribution for the other machine (there is a small impact because of the influence of the evidence on beliefs about the maintenance practices of the owner).

As noted above, when an ordinary variable appears in a parent but not in its child, the random variable can have an unbounded number of parent instances in the constructed approximate SSBN. Each step of SSBN construction instantiates finitely many parents of any random variable. When there are infinitely many computationally relevant parent instances, additional instances are added at each step until a termination condition is reached. Even when a finite-size SSBN exists, constructing it and computing a query response is often intractable. It is typically necessary to approximate the SSBN by pruning arcs and random variables that have little influence on a query, and/or compiling parts of the SSBN to send to inference engines optimized for special problem types. The process of controlling the addition and pruning of random variable instances and arcs is called *hypothesis management*. More generally, *execution management* controls the inference process to balance accuracy against computational resources. Often, portions of an inference task can be solved exactly or approximately using efficient special-purpose reasoners. Such reasoners include constraint satisfaction systems, deductive theorem provers, differential equation solvers, heuristic search and optimization algorithms, Markov chain Monte Carlo algorithms, particle filters, etc. At the meta level, MEBN logic itself can be used to reason about which approximation method to apply to a given query. Online reasoning systems may interleave addition of new findings, refinement of the current approximate SSBN, computation of query responses given the current approximate SSBN, and learning (see Section 6 below).

Laskey, et al. (2000, 2001) treat hypothesis management as a problem of balancing the computational overhead of representing additional random variable instances against accuracy in responding to queries. Charniak and Goldman (1993) and Levitt et al. (1995; Binford and Levitt, 2003) also consider hypothesis management in open-world computational probabilistic reasoning systems. Hypothesis management is discussed extensively in the literature on tracking and multi-source fusion (e.g., Stone, et al., 2000).

We are justified in applying MEBN inference to draw conclusions about the world when: (*i*) the generative theory accurately represents the process by which outcomes of the finding random variables occur in the world and (*ii*) the process by which findings come to be observed is *ignorable*. A data generating process is ignorable if either there is no systematic relationship between the random variable(s) of interest and the process by which findings come to be observed, or if the relationship can be accounted for by the observed random variables (Little and Rubin, 1987). The following is an example of a non-ignorable observation process: (*i*) a finding on the *TempLight* random variable may not be observed if the temperature light is not working properly; (*ii*) an overheated engine can cause the temperature light to malfunction; and (*iii*) the MTheory does not account for the relationship between whether there is a finding and whether the engine is overheated. Whether findings are ignorable depends both on the generative theory and on the process that determines which random variable instances have findings. Inferences can be adjusted for a non-ignorable observation process by explicitly modeling the observation process and by collecting information on additional random variables that may be related to the observation process. MEBN fragments are a useful tool for representing common patterns in observation mechanisms, data gathering conditions, and data reporting processes. Libraries of common patterns can be developed, tailored for specific applications, and used to model and adjust for the effects of the conditions of observation. This is especially important when an MTheory combines information from different sources that may have been gathered under different conditions (e.g., Schum, 1994).

Along with the response to a query, MEBN inference can also return a conflict indicator (Jensen, 1991; Laskey, 1991). Unusually large values of a conflict indicator indicate that the findings are a poor fit to the generative theory. Commonly applied conflict indicators measure calibration of predictions against observed findings. Typically, the probability assigned by an



a. Two machines in the same room

b. Two machines that might or might not be in the same room

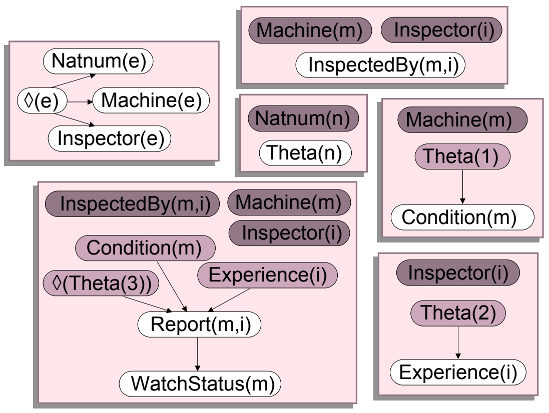
Figure 9: Situation-Specific Bayesian Networks

MTheory $\mathcal{T}'$ to the event that all findings have value T is compared with the probability under an alternative model that is both simple to compute and expected to fit more poorly than $\mathcal{T}'$ if $\mathcal{T}'$ is correct (Laskey, 1991). While occasional short-term runs of poor calibration sometimes occur in an probabilistic process, if the generative MTheory can represent the statistical regularities in the sequence of findings, there is zero probability that findings will be uncalibrated in the infinite limit (Dawid, 1984). A conflict indicator can be used to control SSBN construction, performing additional construction when the current SSBN provides a poor fit to findings (Laskey, et al., 2001).

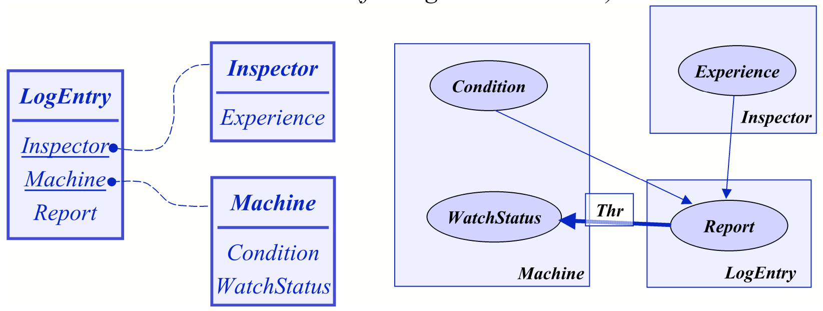
4.7 Relationship of MEBN to Other Probabilistic Logics and Languages

There is a growing literature on languages for representing probabilistic knowledge, the semantics of probabilistic representations, and well-foundedness, tractability and decidability of inference in probabilistic theories. The success of graphical models for parsimonious representation and tractable inference has generated strong interest in more expressive languages for reasoning with probability. Work in knowledge-based model construction (e.g., Wellman, et al., 1992) focused on constructing Bayesian networks from knowledge bases consisting of modular elements representing knowledge about small clusters of variables. Early KBMC systems were not built on decision theoretically coherent declarative domain theories, and relied on heuristic knowledge, typically encoded as procedural rules, for constructing complex models from simpler components. As work in knowledge-based model construction progressed, interest grew in the theoretical foundations of probabilistic representation languages, and in their relationship to classical first-order logic. A number of authors have investigated approaches to integrating classical logic with probability. A common approach has been to provide language constructs that allow one to express first-order theories not just about objects in a domain of discourse, but also about proportions and/or degrees of belief for statements about these objects. Bacchus et al. (1997; Bacchus, 1990) augment first-order logic with proportion expressions that represent the knowledge that a given proportion of objects in a domain have a certain property. A principle of indifference is applied to assign degrees of belief to interpretations satisfying the constraints imposed by ordinary first-order quantification and the proportion expressions. Halpern's (1991) logic can express both proportion expressions and degrees of belief, and provides a semantics relating proportions to degrees of belief. Neither of these logical systems provides a natural way to express theories in terms of modular and composable elements. Unlike Bayesian networks, which have easy to verify conditions ensuring the existence of a complete and consistent domain theory, it is in general quite difficult in these logical systems to specify complete and consistent probabilistic domain theories, or to verify that a theory is complete and consistent.

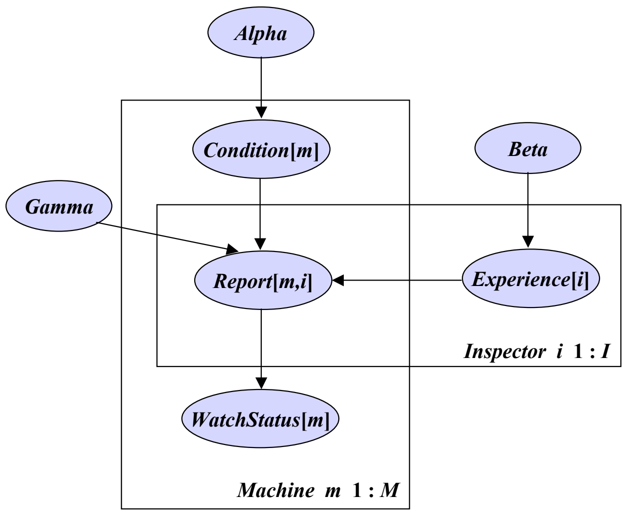
A number of languages have been developed that represent probabilistic knowledge as modular units that can have repeated substructures, and that can be composed into complex domain models. These include pattern theory (Grenander, 1995), hidden Markov models (Elliott, et al., 1995), the plates language implemented in BUGS (Gilks, et al., 1994; Buntine, 1994; Spiegelhalter, et all, 1996), object-oriented Bayesian networks (Koller and Pfeffer, 1997; Bangsø and Wuillemin, 2000; Langseth and Nielsen, 2003), and probabilistic relational models (Getoor, et al., 2000, 2001; Pfeffer, 2001). There is a great deal of commonality among languages for compactly expressing complex probabilistic domain theories (cf., Heckerman, et al., 2004). Plates in BUGS, object classes in object-oriented Bayesian networks, and PRM structures in probabilistic relational models all correspond to MFrag classes.



a. MEBN Fragments
(findings are not shown)



c. Probabilistic Relational Model – Relational Schema & PRM Structure
(skeleton and instances are not shown)



c. Plates

Figure 10: MFrags, PRM and Plates for Equipment Diagnosis Domain

Figure 10 compares MEBN, PRM and plate representations for a theory fragment in the equipment diagnosis domain. Like Bayesian networks, plates represent a joint distribution as an acyclic directed graph in which nodes represent random variables, arcs represent direct dependence relationships, and each node is annotated with a specification of a conditional distribution of the random variable given its parents. Repeated structure in a plates model is represented by indexing repeated random variables with subscripts, and enclosing the set of random variables indexed by a given subscript in a rectangle called a "plate." These indices play the role of the ordinary variables in an MFrag. As in MEBN, a random variable's parents may contain indices not mentioned in the random variable, in which case the local distribution for the child random variable must specify how to aggregate influences from multiple instances of the parent random variable. Plate models are restricted to a finite number of instances of each random variable. The number of instances of each random variable is a fixed attribute of the plate model. BUGS has sophisticated capability for parameter learning, and although there is no built-in mechanism for structure learning, plate models can be constructed to represent the problem of reasoning about the presence or absence of conditional dependency relationships between random variables.

A PRM contains the following elements (Heckerman, et al., 2004; see Figure 10*b*):

- A *relational schema* that specifies the types of objects and relationships that can exist in the domain;
- A *PRM structure* that represents probabilistic dependencies and numerical probability information;
- A *skeleton* that specifies a unique identifier and a blank template for each individual entity instance;
- The *data* to fill the entries in the blank template.

Like an MTheory, a PRM represents a probability distribution over possible worlds. Any given PRM can be expanded into a finite Bayesian network over attributes of and relationships between the individuals explicitly represented in the skeleton. PRMs use aggregation rules to combine influences when multiple instances of a parent random variable influence a child random variable (as when multiple reports influence the *WatchStatus* random variable in Figure 10). In addition to attribute value uncertainty, PRMs have been extended to handle type uncertainty, reference uncertainty, and identity uncertainty. PRM learning theory provides a formal basis for both parameter and structure learning. Learning methods have been published (e.g., Getoor, et al., 2001) for learning both the structure and parameters of PRMs from instances in the skeleton. If the probability distribution represented by a PRM is assumed to apply to similar entities not explicitly represented in the skeleton, then PRM learning methods can be extended to allow sequential learning as new individuals are added to the skeleton over time, thus providing the logical basis for a form of open-world reasoning. One can also extend the relational schema and PRM structure "by hand" to add new entity types.

Heckerman, et al. (2004) introduce a new language, DAPER, for expressing probabilistic knowledge about structured entities and their relationships. DAPER combines the entity-relation model from database theory with directed graphical models for expressing probabilistic relationships. DAPER is capable of expressing both PRMs and plates, thus providing a unified syntax and semantics for expressing probabilistic knowledge about structured entities and their relationships. As presented in Heckerman, et al. (2004), DAPER expresses probabilistic models over finite databases, and cannot express arbitrary first-order formulas involving quantifiers. That is, DAPER (and by extension PRMs and plates) is a macro language for compactly expressing finite Bayesian networks with repeated structure, and not a true first-order probabilistic logic. On the other hand, the random variable semantics described in Section 4.4 could provide a theoretical basis for extending DAPER, and thus PRMs and plates, into a true first-order logic. Conditions could be identified under which DAPER models of unbounded cardinality express well-defined probability distributions over models. If developed more fully, the relationship sketched here between MTheories, PRMs and plates would facilitate construction of such an extension.

Object-oriented Bayesian networks represent entities as instances of object classes with class-specific attributes and probability distributions. Reference attributes allow representation of function composition. Although OOBNs do not have multi-place relations, these can be handled by defining new object types to represent multi-place relations. Structure and parameter learning methods for OOBNs have been developed (e.g., Langseth and Nielsen, 2003; Langseth and

Bangsø, 2001). The current literature on OOBNs does not treat type and reference uncertainty, although clearly it would be possible to extend OOBNs to handle these kinds of uncertainty. An advantage of OOBNs is the ability to represent *encapsulated* information, or random variables defined internally to an object that are independent of external random variables given the interface random variables that shield an object from its environment. The semantics of encapsulation is based on conditional independence relationships. Thus, the concept of encapsulation could be extended to other languages based on graphical models, including MTheories and DAPER models with encapsulated random variables. As with plates and PRMs, the random variable semantics described in Section 4.4 could provide a theoretical basis for extending OOBNs toward full first-order expressiveness.

A feature of MEBN not present in PRMs, plates or OOBNs is the use of context constraints to specify logical conditions that determine whether one random variable influences another. A similar effect can be achieved by using aggregation functions that ignore influences ruled out by the context, but this is more cumbersome. PRMs and OOBNs are founded on a type system, and sophisticated implementations have subtypes and inheritance (e.g., IET, 2004). MEBN can be extended to a typed logic that has many of the advantages of typed relational languages (Costa and Laskey 2005). Because there presently is no direct implementation of MEBN logic, several published applications have translated MTheories into relational models and used the Quiddity*Suite probabilistic relational modeling and KBMC toolkit (IET, 2004) to construct situation-specific Bayesian networks (e.g., Costa, et al., 2005; AlGhamdi, et al., 2005). There are some features of MEBN logic (most notably context constraints) that cannot be represented declaratively in standard relational languages, but the ability of Quiddity*Suite to combine Prolog-style rules with a frame-based relational modeling language provides the ability to specify much more powerful declarative representations (e.g., Fung, et al., 2005).

Like MEBN logic, relational Bayesian networks (Jaeger 1997; 1998; 2001) provides formal semantics for probability languages that extend Bayesian networks to achieve first-order expressiveness. Random variables in a relational Bayesian network are all Boolean. A RBN has a set of pre-defined relations used in defining the local distributions and a set of probabilistic relational symbols, which represent uncertain relations on the domain. A RBN defines a joint probability distribution on models of the uncertain relations. Probability formulas specify how to combine influences from multiple instances of the parents of a random variable to obtain a conditional distribution for the random variable given finite sets of instances of its parents. General relational Bayesian networks can represent probability distributions only over finite domains, although non-recursive RBNs have been extended to represent probability distributions over countably infinite domains (Jaeger, 1998).

Bayesian logic programs (e.g., Kersting and deRaedt, 2001*a*,*b*; deRaedt and Kersting, 2003; Sato, 1998) also express uncertainty over interpretations of first-order theories. To ensure decidability, BLPs have typically been restricted to Horn clause theories. The PRISM language is a powerful and efficient implementation of Bayesian logic programming. Bayesian logic programs and MTheories represent complementary approaches to specifying first-order probabilistic theories. BLPs represent fragments of Bayesian networks in first-order logic; MTheories represent first-order logic sentences as MFrags. Although the restriction to Horn clause logic limits the expressiveness of BLP languages, this limitation is balanced by the efficiency of algorithms specialized to Horn clause theories. Research in Bayesian logic programming is applicable to the problem of execution management in SSBN construction. That is, an execution manager can identify portions of an inference task that involve only Horn clauses, and send these to an inference engine specialized for efficient reasoning with Horn clauses. MEBN semantics could be used to develop extensions to BLP languages that could handle knowledge bases not limited to Horn clauses.

Other research on integrating logic and probability includes Poole's (2003) parameterized Bayesian networks, Ngo and Haddawy's (1997) work on context-specific probabilistic knowledge bases, and BLOG (Milch, 2005), a new language that enables reasoning about unknown objects. Parameterized Bayesian networks are designed to provide the ability to reason about individuals not explicitly named, an important capability lacking in most probabilistic languages. Like MEBN, random variables in a parameterized Bayesian network can take arguments; individuals in a population can be substituted for the parameters to form instances of the random variables. Like MEBN, the population over which the parameters range can be finite or infinite. Poole considers only models without recursion. Thus, a parameterized Bayesian network corresponds to an MTheory with no recursive links. Ngo and Haddawy represent probabilistic knowledge as universally quantified sentences that depend on context. Like MEBN, Ngo and Haddawy exploit context constraints to focus inference on relevant portions of the knowledge base. Unlike MEBN, Ngo and Haddawy separate context, which is non-probabilistic, from uncertain hypotheses, for which context-specific probability distributions are defined. A context-sensitive knowledge base corresponds to a partially specified MTheory in which there is a reserved subset of Boolean random variables that may appear as context random variables in MFrags, but that have no home MFrags and whose truth-values are assumed to be known at problem solving time. BLOG (Milch, et al., 2005) is a new language that enables probabilistic reasoning about unknown entities, and about domains that can contain unknown numbers of entities.

Hidden Markov models are applied extensively in pattern recognition tasks such as speech and handwriting recognition. Formally, a hidden Markov model can be represented as a dynamic Bayesian network in which an observable random variable depends on a latent or hidden variable that follows a Markov transition. Dynamic Bayesian networks and partially dynamic Bayesian networks (Bayesian networks containing both static and dynamic nodes) allow a richer range of representation possibilities, in that complex dependency structures for hidden and observable random variables can be compactly represented. There is a large literature on efficient estimation and inference methods for hidden Markov models. HMMs and DBNs represent temporal recursion. Pfeffer (2000) also considers recursive probabilistic models, which can express non-temporal recursive relationships. It is straightforward to express HMMs, DBNs, and recursive probabilistic models as MEBN theories (e.g., Figure 3).

Pattern theory (Grenander, 1993) is a graphical modeling language based on undirected graphs. There is an extensive literature on applications of undirected graphical models to image understanding, geospatial data, and other problems in which there is no natural direction of influence. A hybrid language could be defined that extends MEBN logic to permit both directed and undirected arcs. Such an extension is not considered here.

Many languages designed for implementation have taken the strategy of restricting expressiveness to ensure that answers to probabilistic queries are decidable. In an open world, the answer to many queries of interest will be undecidable, and the best that can be expected is an approximate answer. Languages that provide decidable, closed-form responses to limited classes of queries have an important place both theoretically and practically. Nevertheless, intelligent reasoning in a complex world requires principled methods of coping with undecidable or intractable problems. MEBN logic exploits the language of graphical models to compose consistent domain theories out of modular components connected via clearly defined interfaces, and thus can support efficient implementations of tractable domain theories. Yet, MEBN logic can represent highly complex, intractable, and even undecidable domain theories. Although the answer to a probabilistic query may be undecidable, and may be intractable even when it is decidable, Bayesian decision theory provides a sound mathematical basis for designing and analyzing the properties of processes that converge to the correct response to undecidable queries, and resource-bounded processes that balance efficiency against accuracy. Bayesian theory also provides semantics for the relationship between empirical proportions and probabilities, as well as a logically justified and theoretically principled way to combine empirical frequencies with prior knowledge to refine theories in the light of observed evidence.

5 Representing Knowledge using MEBN Theories

A logic for knowledge representation requires both a mechanism for representing individual assertions and an ability to organize assertions into structures that permit a reasoner to derive related assertions from existing assertions. FOL is the *de facto* standard logic for formalizing both individual assertions and knowledge structures. A number of references describe how to translate individual natural language sentences into first-order logic (e.g., Enderton, 2001; Quine, 1982). Special-purpose knowledge structures have evolved for dealing with many generally useful aspects of the world, such as types and subtypes, parts and wholes, structured objects and their attributes, space, time, events, values, decisions, actions, and plans. A typical progression in the development of a generically useful knowledge structure begins with independent emergence of several informally specified and closely related variants, followed by increasing formalization, and eventual convergence on core standards. These core standards have typically been formalized either directly in FOL, or in some language for which a translation into FOL has been worked out. Such an evolution process is occurring with probabilistic logic, and there appears to be convergence toward first-order languages based on graphical models. MEBN logic is to our knowledge the first such language having all the following properties: (1) the ability to express a globally consistent joint distribution over models of any consistent FOL theory; (2) a proof theory capable of identifying inconsistent theories in finitely many steps and converging to correct responses to probabilistic queries; and (3) a built-in mechanism for refining theories in the light of observations.

Beyond viewing the world as composed of entities having attributes and existing in relationship to other entities, bare first-order logic makes no commitments about the best way to represent knowledge. This provides a knowledge base designer with great flexibility, but also with the responsibility for defining adequate representations for an enormous variety of entity types. Considerable savings are possible in development, maintenance, and integration to the extent that existing representations can be adapted rapidly and efficiently for use on related problems. However, the potential savings is often outweighed by the effort and risk involved in adapting a representation for reuse in another context. These difficulties would be mitigated if the *de facto* standard logic had principled uncertainty handling as a core capability built into the structure of the logic. This argues for moving toward first-order probabilistic logic as a standard formal basis for knowledge representation and interchange.

Because MEBN logic includes FOL as a subset, any knowledge structure formalized in FOL can automatically be translated to a family of MEBN theories all having the same logical content. Bare MEBN logic makes no specific commitment regarding probability assignments, leaving this up to the designer of an integration architecture, although extensions can be defined that make default assignments according to popular heuristics such as minimum description length or maximum entropy. Modulo the specification of intelligent default probability assignments, MEBN logic can fully exploit previous efforts at formalizing standard knowledge structures. For an example of an early effort at converting a large knowledge base from logic to probabilities, see the literature on the QMR-DT project (e.g., Parker and Miller, 1987). A similar approach could be applied to translating other large knowledge bases into theories expressed in a probabilistic logic.

Following the approach described in standard references (e.g., Sowa, 2000), a typed version of MEBN logic can be defined by specifying a lattice of types, a *Type* random variable that maps an entity to the label for its base type, and a predicate for each type that maps an entity to T if it is of the type indicated by the type label. Context random variables for a random variable's home fragment perform type checking, and a random variable maps entities to $\perp$ that do not match its input type. Because MEBN logic includes first-order logic as a subset, any abstract type that can be formalized using first-order logic can be represented in MEBN logic. Efficient specialized inference engines could be used to perform rapid type-checking as part of SSBN construction.

Standard techniques can be applied to define a polymorphic version of MEBN logic. For example, we might define the *BeltedMachineEngineStatus* and *BeltlessMachineEngineStatus* random variables to represent the status of the engine in machines that do and do not have belts, respectively. The former has *BeltStatus* as a parent; the home MFrag of the latter does not have a *BeltStatus* random variable. An *EngineStatus* random variable would then be defined that applies to all machines. It maps a machine *m* to *BeltedMachineEngineStatus*(*m*) if it has a belt and to *BeltlessMachineEngineStatus*(*m*) if it has no belt. Polymorphic MEBN (Costa and Laskey, 2005) would make this mapping automatic and invisible to the knowledge base designer.

A typed MEBN logic would have several powerful capabilities lacked by typed languages founded on FOL. First, typed MEBN logic can express and reason with uncertainty about the type of an entity. If we don't know whether an engine has a belt, then a query on *BeltStatus*(*m*) results in a probability weighted average of the result of a query on *BeltedMachine-EngineStatus*(*m*) and *BeltlessMachineEngineStatus*(*m*). Second, we often know some but not all of the attributes of an object. In such situations, standard typed logics and object-oriented systems must either assign values of unknown attributes by default or leave them unspecified. The former leads to brittle, *ad hoc* rules for retracting default assignments when additional information renders them implausible; the latter renders the logic too weak for interesting practical problems. MEBN logic assigns probabilities to the possible values of unspecified attributes. These probabilities incorporate all relevant knowledge about the entity itself and other related entities. This leads to theoretically principled nonmonotonic reasoning, in which probabilities for unknown attribute values move up and down appropriately as relevant evidence accrues. Third, it is often useful to generate a representative instance of a given type. MEBN logic provides a probability distribution for randomly generating an entity instance given its type, any pre-specified attributes, and any relevant information about related entities. Finally, MEBN logic has a built-in learning theory for refining type-specific probability distributions. Bayesian learning provides a principled mechanism for learning probability distributions from a sample of observed instances. When there are few observations, a type-specific distribution can "borrow strength" from samples of entities of similar types, weighting the information appropriately according to degree of similarity (Gelman, et al, 1995).

Another advantage of MEBN logic is its ability to represent hypothetical entities, such as the belt in a machine that may or may not have a belt, or the person who tripped the security alarm if the alarm may have been tripped by a power surge. We can define an attribute *Exists*(*x*) that has value T if and only if *x* refers to an actual entity. For some applications, we may want to assign the value $\perp$ to all attributes of a hypothetical entity; for other applications it may be useful to assign probabilities that are representative of what would be expected if the hypothetical entity were real. This ability to represent and reason with hypothetical entities makes MEBN logic a natural tool for counterfactual reasoning and reasoning about causality (cf., Druzdzel and Simon, 1993; Pearl, 2000). MEBN logic also can represent identity uncertainty, or uncertainty about whether two expressions refer to the same entity.

Other categories of knowledge for which specialized logics have been developed include parts and wholes, actors and roles, and space and time. MEBN logic makes no definite commitments regarding the proper way to treat these categories, but is compatible with many common approaches.

Actions, plans and decisions can be represented with an extension to MEBN logic called *multi-entity decision graph* (MEDG, or "medge") logic. A *MEDG theory* is an MEBN theory in which:

- Each random variable is labeled as a *world state*, *decision*, or *value* random variable.
- The possible values of value random variables are numbers.
- If a value random variable has children, they must be value random variables.
- The *total value* for a MEDG theory is the expected value of the sum of all value random variables.

- A *MEDG policy set* is a set of MEDG theories that differ only in the local distributions assigned to decision random variables. A MEDG theory is optimal for a policy set if its total value is at least as great as any other MEDG theory in the policy set.

A *multi-agent MEDG theory* has different actors who may play different roles. Each actor has his/her own value and decision random variables. An agent's value and decision random variables are world state random variables to all other agents. Each actor's optimal course of action is to maximize its total expected value given the probability distributions it assigns to the actions of the other actors. It is common in economics and game theory to assume *rational expectations* (Sargent, 2003), i.e., that an actor's probability distributions are obtained by conditioning a global MEDG probability distribution on the agent's information, and this global MEDG distribution is a generative distribution for the actual outcomes of both world state and action random variables. MEDG logic has no built-in requirement for rational expectations, but a knowledge base designer may include axioms that imply rational expectations. MEDG logic could provide a logical foundation for first-order graphical models for economic and game theoretic problems (see Kearns and Mansour, 2002).

Many different special-purpose logics have been proposed for space and time. MEBN logic provides a unifying formal basis for expressing probabilistic logics for space and time. MEBN logic can represent stochastic processes such as dynamic Bayesian networks (Murphy, 1998; Gharamani, 1998) and partially dynamic Bayesian networks (Takikawa, et al, 2002), as well as Bayesian network models for spatial reasoning (e.g., Wright, 2002). Many common representations for spatial and temporal reasoning can be formalized in FOL (see Davis, 1990 or Sowa, 2000) and then translated to MEBN logic.

Spatio-temporal reasoning and parameter learning typically require probability distributions on uncountably infinite sets such as the real or complex numbers. Just as classical first-order languages with countable symbol sets can represent theories about uncountable infinities, so can MEBN logic. It is important to note that an identifier in MEBN logic is a label for the entity, not the entity itself. Although there are uncountably many real numbers, mathematicians have been able to develop powerful theories for reasoning about real numbers using languages with only countably many symbols. Any application of an MTheory could represent at most countably many labeled objects that take values in the real numbers (e.g., lengths of objects; weights of objects; parameters of probability distributions). This does not imply that there are only countably many *possibilities* for object lengths, object weights, or parameters. In particular, arbitrary continuous distributions can be approximated as mixture distributions over a countable family of continuous density functions (c.f., Robert, 2001). Real-valued random variables in an MTheory could be approximated to arbitrary accuracy by defining their possible values to be indices referring to kernel density functions defined over the real numbers.

Another important application of MEBN logic is the problem of aligning ontologies and interchanging knowledge between different reasoners. If we think of the problem of integrating different reasoners as a problem in inference and decision making under uncertainty, MEDG is a natural logic for representing the integration process. A meta-level MEDG theory can be specified in which each of the reasoners to be integrated is viewed as an entity. A set of MEDG fragments can be defined to reason about the inputs, outputs, and performance characteristics of each reasoner. The context random variables for these MEDG fragments represent information such as the kind of problem, the time available for solution, the format of the inputs, the desired format of the outputs, the desired output quality, etc. MFrags can be developed to represent the flow of inputs and outputs among different reasoners (e.g., temporal reasoners, analogical reasoners, statistical reasoners, modal logics, etc.), where each reasoner can focus on aspects of the problem for which it is suited. The meta-MEDG can examine different architectures for combining the reasoners to evaluate which provide better overall solutions against the anticipated class of queries. Based on this model, a set of heuristic rules, or suggestors, can be specified that trigger calls to different reasoners based on features of the query and characteristics of intermediate results (D'Ambrosio, et al., 2001). As experience is gained, the meta-MEDG can be refined to provide a more accurate assessment of the predicted performance, improvements in the integration architecture can be identified, and improved suggestors can be developed.

6 Combining Knowledge with Observation

Cognitively natural and computationally tractable approaches to fusing knowledge and data are essential enabling technologies for large-scale application of probabilistic knowledge representations (Dybowski, et al., 2003; Druzdzel and van der Gaag, 2000). MEBN logic provides the ability to update and refine theories as observational evidence accrues. The generative MFrags in an MTheory represent general knowledge about statistical regularities in a class of problems. Findings represent specific information about particular situations drawn from the class. As findings accrue, MEBN logic draws implications about regular patterns and updates knowledge accordingly.

6.1 Overview of MEBN Learning

MEBN treats the learning problem as a sequence of MTheories, where each new theory in the sequence is obtained by adding additional findings to the previous theory and (optionally) restructuring the MFrags into a form amenable to efficient computation given the values of past observations. It is standard practice in the literature on learning graphical models to decompose learning into separate sub-problems of learning parameters conditional on a given structure and learning the structure. Structure of an MTheory includes the possible values of the random variables, their organization into MFrags, the fragment graphs, and the functional forms of the local distributions. Local distributions may be specified using parameterized families of distributions. If the value of a parameter is unknown, the parameter can be represented as a random variable that conditions the local distribution. In parameter learning, observations are used to refine the estimate of the value of the parameter.

Figure 11 illustrates parameter learning in MEBN logic. The figure depicts an MTheory for a simple statistical model of widths of entities. The generative part of the theory consists of a *parameter MFrag* that specifies a prior distribution for the average width of objects of different types and a *generative attribute MFrag* that predicts the width of an object conditional on the average widths for objects of its type. Each object type has a different prior distribution for widths. The generative MFrag for widths specifies a probability distribution for the width of an object as a function of the average width for objects of its type. The model implies that conditional on the average width, the width of an object is independent of the widths of other objects. The findings for this theory are the measured widths of four objects of type *!Belt*. The figure shows a situation-specific Bayesian network for a query on the width of the fifth belt. Note that context

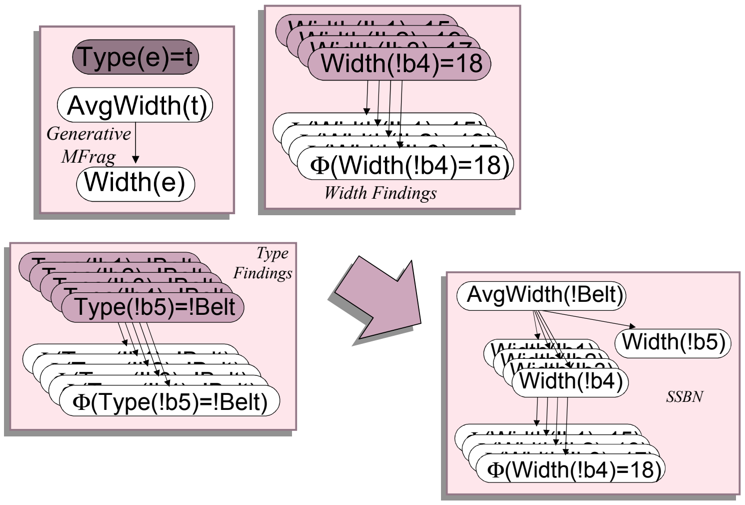


Figure 11: Parameter Learning

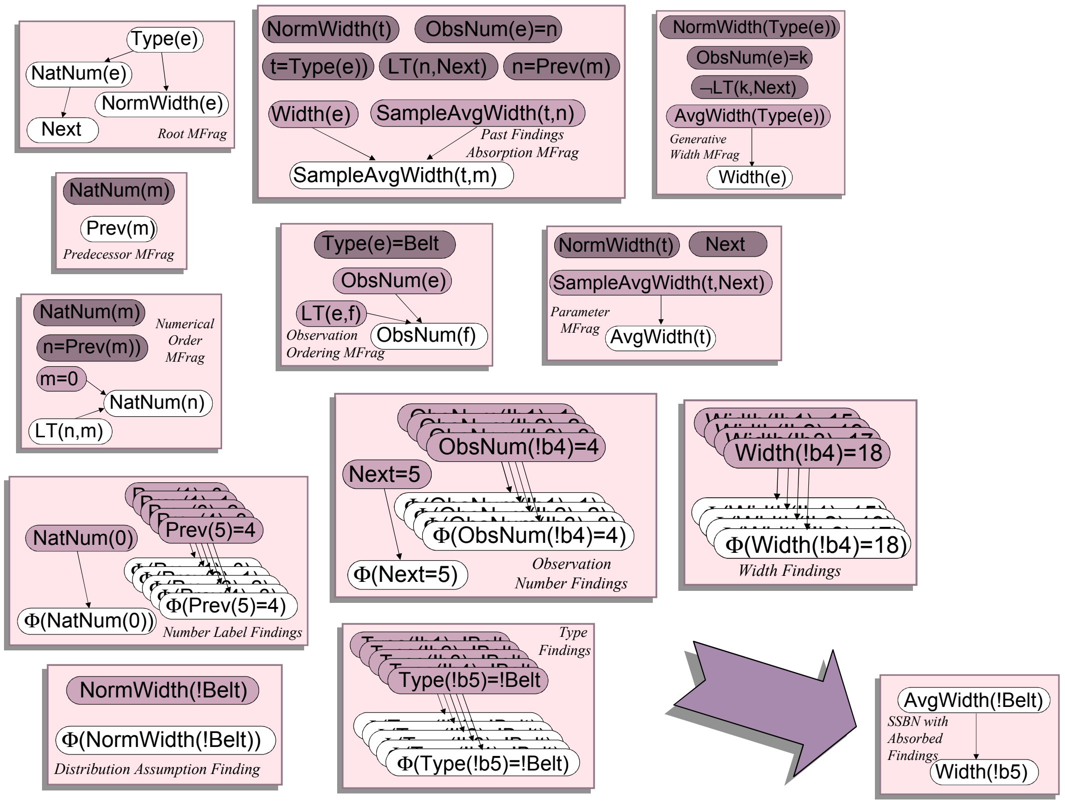
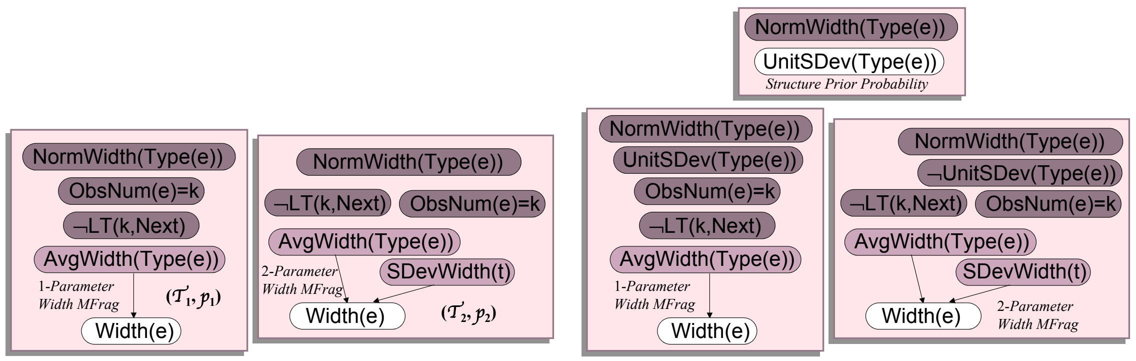


Figure 12: Finding Absorption

random variable instances with known values (e.g., *Type*(*!bi*)=*!Belt* in the generative width MFrag and the unique identifier instance *!Belt* in the parameter MFrag) do not appear in the SSBN. If more findings were added, the SSBN for predicting the width of the next belt would have exactly the same form as the SSBN of Figure 11, except that more findings would be added below *AvgWidth*(*!Belt*). If this model is an accurate representation of the observation generation process, then as observations accrue, the distribution of *AvgWidth*(*!Belt*) will become more and more concentrated about the population average. This theory is capable of learning the value of *AvgWidth*(*!Belt*) to arbitrarily high precision, and also of predicting the widths of not yet observed belts up to an accuracy limit determined by the dispersion of belt widths about the population average.

The type of MFrag shown in Figure 11 is common to a broad class of statistical models. A restructuring operation called finding absorption can be applied when the local distribution in the generative attribute fragment belongs to a family of distributions having a sufficient statistic (see Buntine, 1994). When applicable, finding absorption can greatly improve the tractability of learning and SSBN inference.

Figure 12 illustrates a MEBN representation of finding absorption under the assumption that the sample average of the previously observed belt weights is a sufficient statistic for the distribution of the random variable *AvgWdith*(*Belt*). The *finding absorption MFrag* specifies that the sample average of the first *m* observations depends on the sample average of the first *m*-1 observations and the *m*th observation. The parameter fragment specifies that the distribution of the average width for an object type depends on the sample average of previously observed objects of that type. Finally, the generative attribute fragment specifies a distribution only for objects for which findings have not yet been absorbed (i.e., objects *e* for which *ObsNum*(*e*) is greater than or equal to *Next*). Conditional on all findings having value T, this MTheory represents the same joint



a. Mixture MTheory *b. Polymorphic MTheory*

Figure 13: Example of Structure Learning

probability distribution over all remaining random variable instances as the MTheory of Figure 11.[17] Note that no findings are represented in the SSBN, which is simpler than the SSBN Figure 11. For complex models, finding absorption can result in substantial computational savings in both query processing and learning. Buntine (1994) provides an extensive discussion of re-structuring operations that can improve the efficiency of inference and learning in graphical models. With the translation between plate models and MTheories illustrated in Figure 10, these operations can be expressed as MFrags, thus providing a formal logical foundation for operations on graphical models.

MEBN logic can also be used to learn the structure of a MEBN theory from a set of findings. Structure learning can be represented mixture distributions over simple MTheories representing different structural hypotheses. A more compact representation for structure learning could be developed in an extension to MEBN logic that has types and polymorphism. In a typed language, different structural assumptions for a local distribution could be represented as different subtypes of a given random variable type.

Figure 13 shows an example of structure learning. Continuing the belt width example, we might consider two structural hypotheses: (1) widths are drawn from a one-parameter distribution that depends on the average width of belts, or (2) widths are drawn from a two-parameter distribution that depends on the mean and standard deviation of belt widths. Figure 13*a* shows a representation using a mixture MTheory. The two mixture components have all the same MFrags except for the *Width* MFrags, which differ as shown in the figure. This way of representing structure uncertainty becomes unwieldy when there are many different uncertain structural assumptions. A polymorphic MEBN would allow several home MFrags for the width of an entity, each applicable under different structural assumptions. In Figure 13*b*, there are two *Width* MFrags, each representing different structural assumptions for the distribution of object widths. The first depends on a single parameter, the average width of objects of the given type. The second depends on two parameters: the average and the standard deviation of widths of objects of the given type. Which distribution to use depends on the value of the context random variable *UnitSDev*(*Type*(*e*)), representing whether entities of the give type have standard deviation 1, or whether the standard deviation is a type-specific parameter. This kind of construction is capable of representing the standard approaches to Bayesian structure learning in graphical models (e.g., Cooper and Herskovits, 1992; Friedman and Koller, 2000; Heckerman, et al., 1995; Jordan, 1999). A polymorphic MEBN logic would need to have clearly defined rules for determining which home MFrag was applicable in a given situation.

[17] This is a complete micro-MTheory for this parameter learning problem, expressing from first principles a fragmentary theory of the natural numbers sufficient to provide the necessary logical basis for finding absorption.

6.2 The Dirichlet Process Conjugate Family

Standard methods for learning structure and parameters of ordinary Bayesian networks are based on the Dirichlet family of distributions (e.g., Heckerman, et al, 1995). Dirichlet distributions are attractive because they are conjugate distributions for multinomial sampling. That is, if the prior distribution for the parameters of a local distribution in a Bayesian network are Dirichlet distributions satisfying certain independence assumptions, then the posterior distributions given a sample of cases drawn from the network are also Dirichlet distributions satisfying the same independence relationships. Conjugate families of distributions are useful because they give rise to computationally tractable recursive updating formulas. To define a conjugate family of distributions for MEBN logic, we need to extend the Dirichlet distributions to cover the possibility that a random variable may have infinitely many possible values. We use *Dirichlet process* distributions (Ferguson, 1973), which extend the Dirichlet distribution to random variables that can have infinitely many possible values.

Definition 11: Let Ω be a set, let μ be a probability measure on a σ-field $\mathcal{A}$ of measurable subsets of Ω, and let N be a strictly positive number. The random probability measure P on $(\Omega,\mathcal{A})$ is a *Dirichlet process distribution* with *base probability measure* μ and *virtual sample size* N if for any finite partition $B_1, \ldots, B_n$ of Ω (i.e., $B_i \cap B_j = \emptyset$ for $i \neq j$ and $\cup_i B_i = \Omega$), the joint distribution of the random probabilities is Dirichlet with parameters $(N\mu(B_1), \ldots, N\mu(B_n))$. ■

To formalize the learning problem, we first consider a simple MTheory $\mathcal{T}'$. The MFrags of $\mathcal{T}'$ represent dependency relationships we expect to hold, and the local distributions represent our best estimate of the probability distributions that obtain in the domain to which we intend $\mathcal{T}'$ to apply. However, we may be unsure whether the distribution expressed by $\mathcal{T}'$ is adequate. We would like to use observations to refine both the structural relationships and the local distributions expressed by the MFrags of $\mathcal{T}'$.

Imagine that we are given a sequence of *situations*, where a situation is defined as a set of findings for some of the random variables of $\mathcal{T}'$. After observing a situation, we would like to use Bayesian conditioning to obtain a new MTheory $\mathcal{T}^*$ for which the probability distributions for structure and parameters incorporate information from the observed situation. This whole process can be represented by embedding $\mathcal{T}'$ in a *learning process MTheory,* denoted $\mathcal{T}' \rightarrow \mathcal{T}^*$. A learning process MTheory has the same random variable symbols as $\mathcal{T}'$, each depending on an additional integer *situation number* argument. We assume that completely specified situations are exchangeable, or equivalently, are drawn independently from a "true" generative distribution with unknown structure and parameters. The posterior distribution on structure and parameters given a sequence of completely or partially specified situations can then be obtained by Bayesian conditioning, whenever the posterior distribution is mathematically well defined.

Definition 12: Let N be a positive number and let μ be a probability measure on the set of simple[18] random variable instances $\mathcal{V}=\{\xi_1(\varepsilon_{1i}), \xi_2(\varepsilon_{1i}), \ldots\} \cup \{\beta_1(\varepsilon_{1i}), \beta_2(\varepsilon_{2i}), \ldots\}$ of $\mathcal{L}$. Here, $\xi_k(\varepsilon_{ki})$ $(\beta_k(\varepsilon_{ki}))$ denotes the l^{th} non-Boolean (Boolean) random variable symbol, and i indexes the combinations of entity identifiers forming arguments (if any) to ξ_j (β_j). It is assumed that the measure μ satisfies FOL1-FOL6. The *complete Dirichlet conjugate distribution* for $\mathcal{V}$ given N

[18] Recall that an instance of a random variable is simple if all its arguments are entity identifiers.

and μ, denoted $\Pi^c(N,\mu)$, is a Dirichlet process distribution such that each finite subset $\{\xi_{i1}(\varepsilon_{1i}), \xi_{i2}(\varepsilon_{2i}), \ldots, \xi_{im}(\varepsilon_{mi})\} \cup \{\beta_{i1}(\varepsilon_{1i}), \beta_{i2}(\varepsilon_{2i}), \ldots, \beta_{in}(\varepsilon_{ni})\}$ of $\mathcal{V}$ has a Dirichlet process distribution with base measure $\mu(\{\xi_{i1}(\varepsilon_{1i}), \xi_{i2}(\varepsilon_{2i}), \ldots, \xi_{im}(\varepsilon_{mi})\} \cup \{\beta_{i1}(\varepsilon_{1i}), \beta_{i2}(\varepsilon_{2i}), \ldots, \beta_{in}(\varepsilon_{ni})\})$ and virtual sample size N. ■

Lemma 7: The family of complete conjugate Dirichlet distributions for $\mathcal{V}$ is closed under independent, identically distributed sampling of completely specified worlds. Given the prior distribution $\Pi^c(N,\mu)$ and a completely specified world $\mathcal{W} = \{\xi_{1i}(\varepsilon_{1i}) = \gamma_{1i}, \xi_{2i}(\varepsilon_{2i}) = \gamma_{2i}, \ldots\} \cup \{\beta_{1i}(\varepsilon_{1i}) = \tau_{1i}, \beta_{2i}(\varepsilon_{2i}) = \tau_{2i}, \ldots\}$, where $\varepsilon_{ki} \in \mathcal{E}$; $\gamma_{ki} \in \mathcal{E}$, and $\tau_{ki} \in \{\mathsf{T}, \mathsf{F}\}$, the posterior distribution given $\mathcal{W}$ is $\Pi^c(N+1,\mu^*)$, where the marginal distribution of μ^* on any finite sub-collection of the random variables is calculated using the standard conjugate updating procedure for Dirichlet / multinomial sampling.[19] ■

Lemma 8: Let $\mathcal{T}'$ be a generative MTheory with generative probability distribution $\mathcal{P}^{\text{gen}}_{\mathcal{T}'}$. Let $\mathcal{T}'_N$ be a new MTheory defined as follows. For each MFrag $\mathcal{F}$ of $\mathcal{T}'$, we define a corresponding MFrag $\mathcal{F}_N$ of $\mathcal{T}'_N$ containing random variables with the same names, but each having an additional situation index argument. In addition to its parents in $\mathcal{F}$, each random variable φ contains a set of *parameter parents*, one for each influencing configuration of the parents of φ in $\mathcal{F}$. The possible values of the parameter parents are probability distributions on the possible values of φ. Each of these distributions has a Dirichlet process prior distribution with virtual sample size N and base distribution $\pi_\varphi(\varepsilon|S)$, the local distribution for φ in $\mathcal{P}^{\text{gen}}_{\mathcal{T}'}$. Let $\mathcal{W} = \{\xi_{1i}(\varepsilon_{1i}) = \gamma_{1i}, \xi_{2i}(\varepsilon_{2i}) = \gamma_{2i}, \ldots\} \cup \{\beta_{1i}(\varepsilon_{1i}) = \tau_{1i}, \beta_{2i}(\varepsilon_{2i}) = \tau_{2i}, \ldots\}$ be a completely specified world, and suppose $\mathcal{P}^{\text{gen}}_{\mathcal{T}'}(\bullet)$ assigns strictly positive probability to each $\xi_{ki}(\varepsilon_{ki}) = \gamma_{ki}$ and $\beta_{ki}(\varepsilon_{ki}) = \tau_{ki}$. Then the conditional distribution of $\mathcal{P}^{\text{gen}}_{\mathcal{T}'_N}$ given $\mathcal{W}$ exists as a well-defined limiting distribution conditional on an increasing sequence of subsets of $\mathcal{W}$. The MTheory with the same structure as $\mathcal{P}^{\text{gen}}_{\mathcal{T}'_N}$ and probabilities $\pi^*_\varphi(\varepsilon|S)$ obtained by conjugate updating of $\pi_\varphi(\varepsilon|S)$ represents the same generative distribution for worlds as the MTheory obtained by augmenting $\mathcal{T}'$ with the findings $\{\Phi(\xi_{i1}(\beta_{1i}) = \alpha_{1i}), \Phi(\xi_{i2}(\beta_{2i}) = \alpha_{2i}), \ldots\} \cup \{\Phi(\delta_{i1}(\beta_{1i}) = \beta_{1i}), \Phi(\delta_{i2}(\beta_{2i}) = \beta_{2i}), \ldots\}$. ■

The proof of Lemmas 7 and 8 follow from properties of the Dirichlet distribution (e.g., Heckerman, et al., 1995), extended in the natural way to Dirichlet processes.

We call $\mathcal{P}^{\text{gen}}_{\mathcal{T}'_N}$ the *natural conjugate distribution with virtual sample size N* for $\mathcal{T}'$. Lemmas 7 and 8 suggest the following heuristic interpretation of the natural conjugate distribution. Imagine a process that samples worlds according to an unknown distribution. We can think of $\mathcal{P}^{\text{gen}}_{\mathcal{T}'_N}$ as representing the prior information that (*i*) If $\pi_\varphi(\varepsilon|S) = 0$, then according to $\mathcal{T}'$, ε is a logically impossible value for φ when the parents of φ are in configuration S; (*ii*) we have observed N previous worlds; and (*iii*) the observed frequencies from the previously observed worlds are the frequencies specified by $\mathcal{P}^{\text{gen}}_{\mathcal{T}'}$. Of course, because $\mathcal{P}^{\text{gen}}_{\mathcal{T}'}$ may contain frequencies that are not

[19] That is, the virtual count for the observed configuration of a finite subset of the random variables is N times the prior base probability plus 1; the virtual count for each non-observed configurations is N times its prior count; these virtual counts are divided by N+1 to obtain the posterior base probabilities.

integral multiples of N, this heuristic interpretation is not an accurate description of the true state of affairs. Nevertheless, it can be helpful for gaining some intuition for the meaning of $\mathcal{P}^{\text{gen}}_{\mathcal{T}}$.

Lemma 9**:** A set $\mathcal{B} = \cup_i \mathcal{B}^i$ of consistent[20] binding sets for $\mathcal{T}$ induces a partial order $\preceq_{\mathcal{B}}$ on random variable instances constructed by applying the bindings in $\mathcal{B}$ to the random variable terms of $\mathcal{T}$. This partial order satisfies the condition that if $\phi_k(\varepsilon_k) \preceq_{\mathcal{B}} \phi_{k\text{-}1}(\varepsilon_{k\text{-}1}) \preceq_{\mathcal{B}} \ldots \preceq_{\mathcal{B}} \phi(\varepsilon_1) \preceq_{\mathcal{B}} \phi(\varepsilon)$ is any increasing sequences of instances ending in $\phi(\varepsilon)$, then $k < d_{\phi(\varepsilon)}$.[21]

Proof**:** This is an immediate consequence of the no cycles and finite causal depth conditions of Definition 8. ■

The following theorem extends Theorem 2 of Heckerman, et al. (1995).

Theorem 10**:** Let $\mathcal{V}=\{\xi_1(\varepsilon_{1i}), \xi_2(\varepsilon_{1i}), \ldots\} \cup \{\beta_1(\varepsilon_{1i}), \beta_2(\varepsilon_{2i}), \ldots\}$ be the set of domain-specific non-Boolean and Boolean random variable instances for language $\mathcal{L}$. Let $\mathcal{T}$ be an MTheory and let $\Pi^c(N, \mathcal{P}^{\text{gen}}_{\mathcal{T}})$ be the complete conjugate Dirichlet distribution for $\mathcal{T}$ with virtual sample size N. Let $\preceq$ be a partial ordering on the random variable instances consistent with the MFrags of $\mathcal{T}$ (such a partial order exists by Lemma 6). Suppose the prior distribution $\Pi(\mu)$ for a random probability measure μ on $\mathcal{V}$ satisfies the following assumptions:

1. *Mixture of conjugate distributions*: $\Pi(\mu) = \sum_i q_i \mathcal{P}^{\text{gen}}_{\mathcal{T}_i}(\bullet | N)$ is a finite or countably infinite mixture of natural conjugate distributions for MTheories with different structures, such that the weights q_i on the mixture elements are non-zero and sum to 1;

2. *Parameter independence*: For each MTheory $\mathcal{T}_i$ having positive weight in the mixture distribution, the local distributions $\pi_\varphi(\varepsilon | S, \mathcal{T}_i)$ for different φ are independent of each other. For a given φ, the local distributions $\pi_\varphi(\varepsilon | S, \mathcal{T}_i)$ for different equivalence classes of influence configurations are independent of each other.

3. *Parameter modularity*: If random variable instance φ has exactly the same parent configurations in two different MTheories, it has the same local distribution in each MTheory.

4. *Likelihood equivalence*: If two MTheories with different structures encode the same independence assertions, then the likelihood of any completely specified world $\mathcal{W}$ is the same given the two MTheories.

5. *Structure possibility*: Suppose $\{\varphi_1, \varphi_2, \ldots\}$ is a complete ordering of the random variable instances of $\mathcal{T}$ consistent with the partial ordering $\preceq$, and let $\{\varphi_{i1}, \varphi_{i2}, \ldots, \varphi_{in}\}$ be any permutation of the first n random variable instances $\{\varphi_1, \varphi_2, \ldots, \varphi_n\}$. All complete

[20] $\mathcal{B}^i$ is consistent with $\mathcal{B}^i$ if whenever a variable x is bound to unique identifer α in $\mathcal{B}^i$, if x appears in $\mathcal{B}^j$, it is also bound to α.

[21] Recall that $d_{\phi(\varepsilon)}$ is the depth of $\phi(\varepsilon)$ as defined in Definition 7.

structures with the random variable instances ordered $\{\varphi_{i1}, \varphi_{i2}, \ldots, \varphi_{in}, \varphi_{n+1}, \varphi_{n+2}, \ldots\}$ for all permutations of the first n instances and all n have non-zero probability.[22]

Then the distribution $\Pi^c(N, \mathcal{P}^{\text{gen}}_{\mathcal{T}})$ completely determines the parameter distribution for each structure.

Proof: The proof of Theorem 10 proceeds by noting that Theorem 2 of Heckerman, et al. (1995) implies the theorem is true for the complete MTheory that (*i*) agrees with $\mathcal{T}$ on any finite sub-collection of the random variable instances of $\mathcal{T}$ arranged in a completely connected graph and (*ii*) assigns value $\perp$ (not relevant) to all other random variable instances. All the distributions thus obtained are consistent with each other on the instances they share, and these jointly imply the existence of an infinite-dimensional distribution satisfying all these mutually consistent constraints on the marginal virtual counts. ■

The family of natural conjugate distributions is very restrictive in that it can represent only prior information equivalent to having sampled a known number of completely specified worlds. This is even more restrictive than in the finite-dimensional case, because worlds contain infinitely many entities. It is unrealistic to imagine that one could have observed even a single fully specified world, let alone a sample of size N. However, the family of countable mixtures of natural conjugate distributions is closed under sampling of worlds with missing observations (i.e., situations). This family of distributions can represent prior information in which observations consist of samples of different size for different sub-collections of random variables, and is closed under sampling of situations.

7 Summary and Discussion

Graphical models were initially limited to problems in which the relevant random variables and relationships could be specified in advance. Languages based on graphical models are rapidly reaching the expressive power required for general computing applications. It is becoming possible to base computational inference and learning systems on rationally coherent domain models implicitly encoded as sets of graphical model fragments, and to use such coherent deep structure models to guide reasoning and knowledge discovery. Probability theory provides a logically coherent calculus for combining prior knowledge with data to evolve an agent's knowledge as observations accrue. Probability theory also provides a principled approach to knowledge interchange among different reasoners. This paper presents a logical system that unifies Bayesian probability and statistics with classical first-order logic. An instance of a first-order Bayesian language called Multi-Entity Bayesian Networks (MEBN) is presented. The syntactic similarity of MEBN to standard first-order logic notation clarifies the relationship between first-order logic and probabilistic logic. A MEBN theory (MTheory) assigns probabilities to models of an associated FOL theory. MTheories partition FOL theories into equivalence classes of theories with the same logical content but different probabilities assigned to models. Provable statements in FOL correspond to statements in the associated MTheory for which SSBN construction terminates with a probability of 1 assigned to the value T. An MTheory corresponding to an inconsistent FOL theory has at least one finding equal to $\perp$ with probability 1. If the associated MTheory is inconsistent, SSBN can determine in finitely many steps that it is inconsistent. When SSBN construction does not terminate but the MTheory represents a globally consistent joint distribution, the construction process gives rise to an anytime sequence of

[22] There are $n!$ structures for each n, all of which have non-zero probability. Thus, most of these completely connected structures have astronomically small probabilities, but the structure possibility assumption says none has probability equal to zero.

approximations that converges in the infinite limit to the correct response to the query. MEBN logic is inherently open. Bayesian learning theory provides an inbuilt capability for MEBN-based systems to learn better representations as observations accrue. Parameter learning can be expressed as inference in MTheories that contain parameter random variables. Structure learning can also be handled by introducing multiple versions of random variables having home MFrags with different structures. A more natural approach to structure learning, as well as a more flexible type system, requires a polymorphic extension of MEBN logic. Clearly, a typed MEBN with polymorphism would be desirable for many applications. We chose in this paper to focus on the basic version of the logic to highlight its relationship to classical first-order logic and demonstrate that the logic is sufficiently powerful to represent general first-order theories. Extensions of MEBN are planned to incorporate additional expressivity.

Appendix A: Proofs and Algorithms

This appendix proves that an MTheory represents a globally consistent joint distribution over random variable instances, proves that an MTheory constructed as described in Section 4.5 places non-zero probability of value T on Boolean random variables corresponding to satisfiable first-order sentences, presents the SSBN construction algorithm, shows that SSBN construction identifies an unsatisfiable set of findings in finitely many steps, and proves that when findings are consistent, SSBN construction converges with probability 1 to the posterior distribution over an MTheory's random variables given that all finding random variables have value T.

A.1. Proof of Existence Theorem

Theorem 1: Let $\mathcal{T} = \{ \mathcal{F}_1, \mathcal{F}_2 \ldots \}$ be a simple MTheory. Then there exists a joint probability distribution on the set of instances of its random variables that is consistent with the local distributions assigned by the MFrags of $\mathcal{T}$.

Proof: Let $\mathcal{Z}=\{\phi_1(\alpha_1), \ldots, \phi_m(\alpha_m)\}$ be a finite subset of $\mathcal{N}_\mathcal{T}$, and let $D = \max [d_{\phi(\alpha)} : \phi(\alpha)\in \{\phi_1(\alpha_1), \ldots, \phi_m(\alpha_m)\}]$ be the maximum depth of the instances of $\mathcal{Z}$. Suppose $D = 0$. Let $\pi_{\mathcal{T}1}(\phi_1(\alpha_1), \ldots, \phi_m(\alpha_m))$ be a distribution in which the $\phi_i(\alpha_i)$ are independent and distributed according to the default distributions $\pi_{\phi_i(\theta_i)}(\bullet|\varnothing)$ from their home MFrags $\mathcal{F}_{\phi_i(\alpha_i)}$. All finite-dimensional distributions constructed in this way from depth 0 elements of $\mathcal{N}_\mathcal{T}$ are consistent with each other and with the local distributions of $\mathcal{T}$. Therefore, Kolmogorov's existence theorem[23] implies that these finite-dimensional distributions can be extended to a joint distribution $\pi_{\mathcal{T}1}$ over all instances of depth zero random variables, and this joint distribution is consistent with the local distributions of $\mathcal{T}$.

Now, suppose $\mathcal{T}$ represents a joint distribution $\pi_{\mathcal{T}D}$ over all instances of all random variables of depth less than D. Let $\mathcal{Z} = \{\phi_1(\alpha_1), \ldots, \phi_m(\alpha_m)\}$ be a finite subset of $\mathcal{N}_\mathcal{T}$ such that no $\phi_i(\alpha_i) \in \mathcal{Z}$ has depth greater than D. Let $\mathcal{A}$ denote the (possibly infinite) subset of $\mathcal{N}_\mathcal{T}$ consisting of the ancestors of depth D elements of $\mathcal{Z}$, together with any elements of $\mathcal{Z}$ with depth strictly less than D. Clearly, any instance $\varphi(\beta)\in \mathcal{A}$ must have depth less than D. Therefore, the marginal

[23] Kolmogorov's existence theorem (c.f., Billingsley, 1995) states that if joint distributions exist for all finite subsets of a collection of random variables, and if all these finite-dimensional distributions are consistent with each other, then a joint distribution exists for the infinite collection of random variables.

distribution of $\pi_{\mathcal{T}D}$ represents a joint distribution for $\mathcal{A}$ consistent with the local distributions of $\mathcal{T}'$.

Let $S=\{\varphi(\beta)=\gamma : \varphi(\beta)\in\mathcal{A}\}$ be a set of value assignment terms, one for each element of $\mathcal{A}$. Suppose $\phi_i(\alpha_i) \in \mathcal{Z}$. If $\phi_i(\alpha_i)$ has depth less than D, then $\phi_i(\alpha_i)\in\mathcal{A}$ and S assigns a particular value to $\phi_i(\alpha_i)$ with probability 1. Otherwise, condition 3e implies that there is a finite subset $S_{\phi_i(\alpha_i)} \subset S$ such that $\pi_{\phi_i(\alpha_i)}(\bullet | S_{\phi_i(\alpha_i)}) = \pi_{\phi_i(\alpha_i)}(\bullet | S^*)$ whenever $S_{\phi_i(\alpha_i)} \subset S^* \subset S$.[24] Thus, given the value assignments in S, $\mathcal{T}'$ assigns a well-defined conditional distribution to each $\phi_i(\alpha_i) \in \mathcal{Z}$, which is denoted $\pi_{\phi_i(\alpha_i)}(\bullet | S)$. Define a joint conditional distribution

$$\pi_{\mathcal{T}(D+1)}(\phi_1(\alpha_1)=\gamma_1, \ldots, \phi_m(\alpha_m)=\gamma_m \mid S) = \prod_{i=1}^{m} \pi_{\phi_i(\alpha_i)}(\phi_i(\alpha_i)=\gamma_i \mid S).$$

in which the $\phi_i(\alpha_i)$ are independent and distributed as assigned by the local distributions in their home MFrags conditional on the value assignments in S. Existence of both a joint conditional distribution for the $\phi_i(\alpha_i)$ and a marginal distribution for S implies that the marginal joint distribution

$$\pi_{\mathcal{T}(D+1)}(\phi_1(\alpha_1), \ldots, \phi_m(\alpha_m)) = \int \prod_{i=1}^{m} \pi_{\phi_i(\alpha_i)}(\phi_i(\alpha_i) \mid S)\, d\pi_{\mathcal{T}D}(S). \qquad (1)$$

exists and is consistent with the local distributions of $\mathcal{T}'$. The marginal distribution (1) is expressed as an integral rather than a sum because there may be uncountably many different ways to choose the value assignments $S=\{\varphi(\beta)=\gamma : \varphi(\beta)\in\mathcal{A}\}$.

This construction can be carried out for any finite set of depth D instances, and it is clear that all the distributions thus defined are consistent with each other and with the local distributions of $\mathcal{T}'$. This implies that $\mathcal{T}'$ represents a joint distribution over arbitrary finite subsets of $\mathcal{N}_{\mathcal{T}'}$, and that the distributions constructed in this way are consistent with each other and with the local distributions of $\mathcal{T}'$. A second application of Kolmogorov's existence theorem implies that $\mathcal{T}'$ represents a joint distribution over all instances of random variables in $\mathcal{V}_{\mathcal{T}'}$. It is clear that this distribution is consistent with the local distributions of $\mathcal{T}'$. ■

A.2. SSBN Construction Algorithm

The situation-specific Bayesian network construction algorithm takes an MTheory $\mathcal{T}'$, a finite (possibly empty) set of *target* random variable instances, and a (possibly empty) set of *finding* random variable instances, and computes a sequence of Bayesian networks containing the target and finding random variable instances. If the algorithm terminates, and the findings are consistent, the last Bayesian network in the sequence can be used to compute the joint distribution of the target random variable instances given that all finding random variable instances have value T. That is, additional model construction would not change the result of the query. For many problems of interest, the algorithm never terminates, but the approximate SSBN can be used to compute approximate responses to queries.

We give the SSBN construction for simple MTheories only. The modification for mixture MTheories is straightforward. SSBN construction proceeds as follows:

[24] Theorem 1 holds under weaker conditions on the local distributions, but condition 3e suffices to show that MEBN can represent classical first-order logic.

SSBNConstruct: The inputs to SSBNConstruct are:

- A simple MTheory $\mathcal{T}$ with partial ordering $\preceq$ and modeler-defined MFrags $\mathcal{F}$ defined on a set $\mathcal{X}$ of random variable symbols and a set $\mathcal{A}$ of constant symbols;
- A finite (possibly empty) set $\{\tau_i\}_{i\leq T}$ of non-finding random variable instances called the *target* random variable instances;
- A (possibly empty) set $\{\phi_i\}_{i\leq F}$ of *finding* random variable instances.

The steps in SSBNConstruct are:

1. Set the *initial query set* Q_0 equal to the union of the target instances and a finite subset of the finding instances. Let N_0 be a positive integer.
2. Set $\mathcal{B}_0$ equal to a Bayesian network in which the nodes are the random variables in Q_0, and there is an arc from random variable α to β if α is an instance of a parent of β in the home fragment of β. Remove any arcs from $\mathcal{B}_0$ that do not correspond to influencing configurations. $\mathcal{B}_0$ is called the *approximate SSBN*. Set the cached marginal distribution for each node β to its default distribution.
3. Set the iteration number i equal to 0.
4. Do until no more changes to $\mathcal{B}_i$ occur:
   - Remove from $\mathcal{B}_i$ all *barren* nodes, that is, nodes having no descendants in Q_i;
   - Remove from $\mathcal{B}_i$ all nodes that are *d-separated* by finding nodes from any target nodes;
   - Remove from $\mathcal{B}_i$ any *nuisance nodes* for which the cached marginal distribution is the correct distribution for $\mathcal{B}_i$. A nuisance node (Lin

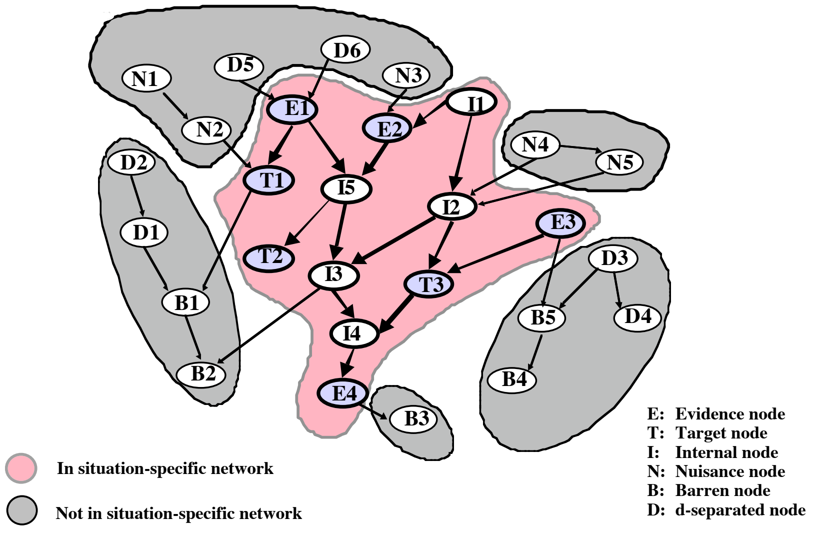


Figure 14: Situation-Specific Bayesian Network

and Druzdzel, 1997) is a node that is computationally relevant given the query, but is on no evidential trail[25] between an evidence and a target node.

5. Set the local distributions in $\mathcal{B}_i$. These distributions are calculated from the local distributions in the MFrags of $\mathcal{T}'$, with modifications to restrict random variables to have no more than N_i possible values and to approximate the effects of random variables that have not been enumerated.
   - If a random variable α has more than N_i possible values, the N_k^{th}, N_k+1^{st}, N_k+2^{nd}, etc. (not including $\perp$) values are grouped into a single aggregate value. In the local distribution for α in its home MFrag, for each influencing configuration IC_α of the parents of α, assign the aggregate value a probability equal to the sum of the probabilities of the N_k^{th}, N_k+1^{st}, N_k+2^{nd}, etc. possible values given IC_α. For any child β of α, and any influencing configuration IC_β of the parents of β for which the distribution of β depends on which of the N_k^{th}, N_k+1^{st}, N_k+2^{nd}, etc. is the case, assign β a default distribution. This may be the default distribution from β's home MFrag or an SSBN construction distribution, but it must satisfy the condition that none of the possible values represented in the approximate SSBN has probability zero.
   - The local distributions of $\mathcal{T}'$ provide a recipe for computing the probability of a random variable given all the parents that have been enumerated thus far. If a random variable α has computationally relevant, non-nuisance instances that are not included in $\mathcal{B}_i$, add a single "leak parent" λ_α to approximate these unrepresented influences. The leak parent has value T with probability ε and F with probability 1-ε, where ε is a small number. Conditional on $\lambda_\alpha = \mathsf{F}$, α has the distribution conditional on the enumerated parents, as modified above to restrict to N_i or fewer values. Conditional on $\lambda_\alpha = \mathsf{T}$, α is assigned a default distribution, also grouping values to restrict to N_i or fewer values. This may be the default distribution from β's home MFrag or an SSBN construction distribution, but it must satisfy the condition that none of the possible values represented in the approximate SSBN has probability zero.
6. Apply a standard Bayesian network inference algorithm to compute the joint distribution of the target random variables in $\mathcal{B}_i$. (If there are no target random variables, then apply Bayesian network inference to compute marginal distributions on all random variables in $\mathcal{B}_i$.)
   - If the inference algorithm returns a probability of zero that all findings have value T, then set the SSBN S equal to $\mathcal{B}_i$, output S, and

[25] A node is computationally relevant if it remains after iteratively removing all barren and *d*-separated nodes. An evidential trail between two sets of nodes is a minimal active undirected path from a node in one set to a node in the other. If a global joint distribution exists, then nuisance nodes can be marginalized out without affecting the result of the query.

stop with an indication that SSBN construction terminated and $\mathcal{T}'$ is inconsistent.
  - Else, if all computationally relevant random variables have been added and no random variable in $\mathcal{B}_i$ has more than N_i possible values, then set the SSBN S equal to $\mathcal{B}_i$. Output S and the joint distribution of the target random variables. Stop with a flag indicating that SSBN construction terminated.
  - Else, set the cached marginal distribution for each node β to its marginal distribution in $\mathcal{B}_i$ and go to Step 7.

7. If a stopping criterion is met, output $\mathcal{B}_i$ and the joint distribution computed in Step 6, and stop with an indication that SSBN construction did not terminate.
8. Else, for each random variable instance $\beta \in \mathcal{B}_i$ for which a change in the local distribution may occur if additional parents are added, add a finite number of instances of parents of β, using a process that ensures eventual addition of all instances of parents of β. If there are computationally relevant findings that have not been added, add a finite number of additional findings, using a process that ensures eventual addition of all computationally relevant findings.
9. Set N_{i+1} to a positive integer strictly greater than N_i, increment i, and go to Step 4.

The following theorem states that an inconsistent theory can be discovered in a finite number of steps of SSBN construction.

Theorem 11: If the logical constraints represented by $\mathcal{T}'$ are unsatisfiable and Step 7 of *SSBNConstruct* is set never to stop, then SSBN construction on a query set consisting only of the findings of $\mathcal{T}'$ terminates in finitely many steps with an indication that $\mathcal{T}'$ is inconsistent.

Proof: Each approximate SSBN $\mathcal{B}_i$ represents a probability distribution over interpretations of a theory for which the logical axioms form a subset of the logical axioms of $\mathcal{T}'$. The domain of this interpretation is a finite set consisting of all possible assignments of values to the random variables of $\mathcal{B}_i$ such that all finding random variables have value T. The approximate SSBN $\mathcal{B}_i$ assigns non-zero probability to the hypothesis that all finding random variables have value T if and only if there is at least one interpretation on this finite domain that satisfies all the logical axioms represented in $\mathcal{B}_i$, which in turn is the case if and only if the logical axioms represented in $\mathcal{B}_i$ are simultaneously satisfiable. For $k>i$, the approximate SSBN $\mathcal{B}_k$ includes all logical constraints included in $\mathcal{B}_i$, along with any additional constraints implied by the local distributions of random variables appearing in $\mathcal{B}_{i+1}$ but not in $\mathcal{B}_i$. The SSBN construction process eventually adds all computationally relevant random variables, and therefore eventually includes all logical constraints represented by the local distributions of any random variable instances that are either findings or ancestors of findings in the random variable partial ordering $\preceq$. Thus, if the findings are unsatisfiable, eventually there will be an approximate SSBN in the sequence that represents an unsatisfiable set of constraints. ■

It is well known that if a set of sentences in FOL is unsatisfiable, then there exists a finite set of ground instances of a set of logically equivalent sentences that is also unsatisfiable (see, for example, Russell and Norvig, 2002). The SSBN construction algorithm produces a sequence of Bayesian networks, each of which can be translated into a set of constraints on truth-values of a finite set of ground instances of FOL sentences implied by the MEBN theory $\mathcal{T}'$. Each of these Bayesian networks encodes a probability distribution that assigns non-zero probability to any assignment of truth-values consistent with the constraints it encodes. Each approximate SSBN includes all constraints represented in the preceding approximate SSBNs, together with additional constraints. If the query set contains only the findings, then eventually all logical constraints implied by the findings and their predecessors in the random variable instance partial order are enumerated. If the set of all logical constraints is unsatisfiable, then so is a finite subset, and eventually the constraints encoded in the SSBN will include a finite unsatisfiable subset.

Note that SSBN construction will never add random variables *d*-separated from the target random variables by findings. Therefore, if the query set contains non-finding target random variables, then inconsistencies that would be introduced only by adding *d*-separated random variables will not be discovered. It is often asserted in logic texts that an inconsistent theory is "useless" because anything can be proven from a contradiction. In practice, though, inconsistent theories can be quite useful. MEBN logic can be used to reason with inconsistent theories, as long as queries are structured so that the target of any given query is *d*-separated by a subset of the findings from any findings that contradict this subset. Thus, MEBN logic may turn out to be a useful tool for studying conditions under which inconsistent theories can provide accurate results to probabilistic queries.

Acknowledgements

Research for this paper was partially supported by DARPA & AFRL contract F33615-98-C-1314, Alphatech subcontract 98036-7488. Additional support was provided by the Advanced Research and Development Activity (ARDA), under contract NBCHC030059, issued by the Department of the Interior. The views, opinions, and findings contained in this paper are those of the author and should not be construed as an official position, policy, or decision, of DARPA, AFRL, ARDA, or the Department of the Interior, unless so designated by other official documentation. Appreciation is extended to Bruce D'Ambrosio, Suzanne Mahoney, Mike Pool, Bikash Sabata, Masami Takikawa, Dan Upper, and Ed Wright for many helpful discussions. Special thanks are due to Paulo Costa and Tod Levitt for extensive feedback on earlier drafts. The author is grateful to the anonymous reviewers of an earlier draft for many insightful comments and useful suggestions.

References

Ghazi AlGhamdi, Kathryn Laskey, Ed Wright, Daniel Barbará and K.C. Chang. Modeling Insider Behavior Using Multi-Entity Bayesian Networks. *10th Annual Command and Control Research and Technology Symposium*, 2005.

Fahiem Bacchus. *Representing and Reasoning with Probabilistic Knowledge.* MIT Press, Cambridge, Massachusetts, 1990.

Fahiem Bacchus, Adam Grove, Joseph Y. Halpern and Daphne Koller. From statistical knowledge bases to degrees of belief. *Artificial Intelligence* 87: 75-143, 1997.

Olav Bangsø and Pierre-Henri Wuillemin. Object Oriented Bayesian Networks: A Framework for Topdown Specification of Large Bayesian Networks and Repetitive Structures. Technical Report CIT-87.2-00-obphw1, Department of Computer Science, Aalborg University, 2000.

Patrick Billingsley. *Probability and Measure*. New York: Wiley, 1995.

Thomas Binford and Tod S. Levitt. Evidential reasoning for object recognition, *IEEE Transactions Pattern Analysis and Machine Intelligence*, 25(7): 837-851.

Craig Boutilier, Nir Friedman, Moises Goldszmidt, and Daphne Koller. Context-Specific Independence in Bayesian Networks. *Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence*, San Mateo, CA: Morgan-Kaufman, 1996.

Ronald J. Brachman, Richard E. Fikes, and Hector J. Levesque, KRYPTON: A Functional Approach to Knowledge Representation, *IEEE Computer*, 16(10): 67-73, 1983.

Wray Buntine. Operations for Learning with Graphical Models. *Journal of Artificial Intelligence Research,* 2: 159-225, 1994.

Eugene Charniak and Robert Goldman. A Bayesian Model of Plan Recognition. *Artificial Intelligence,* 64: 53-79, 1993.

Gregory Cooper and Edward Herskovits. A Bayesian Method for the Induction of Probabilistic Networks from Data. *Machine Learning,* 9: 309-347, 1992.

Paulo Costa and Kathryn B. Laskey. To Type or Not to Type: A Polymorphic Version of MEBN, Fairfax, VA: George Mason University, 2005, http://ite.gmu.edu/~klaskey/Costa_Laskey.

Paulo Costa, Kathryn B. Laskey, Francis Fung, Mike Pool, Masami Takikawa and Ed Wright, MEBN Logic: A Key Enabler for Network-Centric Warfare, *10th Annual Command and Control Research and Technology Symposium*, 2005.

Robert G. Cowell, A. Phillip Dawid, Steffen L. Lauritzen, and David J. Spiegelhalter, *Probabilistic Networks and Expert Systems*. Springer-Verlag, New York, NY, 1999.

Bruce D'Ambrosio: Inference in Bayesian Networks. *AI Magazine 20* (2): 21-36, 1999.

Bruce D'Ambrosio. Local expression languages for probabilistic dependency. In *Proceedings of the Seventh Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann Publishers, San Mateo, California, 1991.

Bruce D'Ambrosio, Masami Takikawa, Julie Fitzgerald, Daniel Upper, and Suzanne Mahoney. Security situation assessment and response evaluation (SSARE). In *Proceedings of the DARPA Information Survivability Conference & Exposition II*, volume I, pages 387–394. IEEE Computer Society, 2001.

Ernest Davis. *Representations of Commonsense Knowledge*. Morgan Kaufmann Publishers, Inc., San Mateo, CA, 1990.

A. Philip Dawid. "Statistical Theory, the Prequential Approach." *Journal of the Royal Statistical Society A 147*: 278-292, 1984.

Morris DeGroot and Mark J. Schervish. *Probability and Statistics* (3rd edition). Addison Wesley, Boston, Massachusetts, 2002.

Luc deRaedt and Kristen Kersting, Probabilistic Logic Learning. *ACM-SIGKDD Explorations: Special Issue on Multi-Relational Data Mining*, 5(1): 31-48, 2003.

Marek J. Druzdzel and Herbert A. Simon. Causality in Bayesian Belief Networks. In *Proceedings of the Ninth Annual Conference on Uncertainty in Artificial Intelligence* (UAI-93), pp. 3-11, Morgan Kaufmann Publishers, Inc., San Francisco, CA, 1993.

Marek J. Druzdzel and Linda C. van der Gaag. Building probabilistic networks: "Where do the numbers come from?" *IEEE Transactions on Knowledge and Data Engineering*, 12: 481-486, 2000.

Richard Dybowski, Kathryn B. Laskey, James W. Myers, and Simon Parsons. Introduction to the Special Issue on the Fusion of Domain Knowledge with Data for Decision Support. *Journal of Machine Learning Research* 4(Jul):293-294, 2003.

Robert J. Elliott, Lakhdar Aggoun and John B. Moore. *Hidden Markov Models: Estimation and Control*. Springer-Verlag, New York, New York, 1995.

H. B. Enderton. *A Mathematical Introduction to Logic*. New York: Academic Press, 2001.

T.S. Ferguson. A Bayesian Analysis of Some Nonparametric Problems. *Annals of Statistics 1*, 209-230, 1973.

Bruno de Finetti. *Theory of Probability: A Critical Introductory Treatment* (2 vols.), New York: Wiley, 1974-75.

Gottlob Frege. *Begriffsschrift*, 1879, translated in Jean van Heijenoort, ed., *From Frege to Gödel*, Cambridge, MA: Harvard University Press, 1967.

Nir Friedman and Daphne Koller. Being Bayesian about Network Structure. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann Publishers, San Mateo, California, 2000.

Francis Fung, Kathryn B. Laskey, Mike Pool, Masami Takikawa and Ed Wright. *PLASMA: Combining Predicate Logic and Probability for Information Fusion and Decision Support*, presented at *AAAI Spring Symposium on Challenges to Decision Support in a Changing World*, 2005.

Daniel Geiger and David Heckerman. Advances in probabilistic reasoning. In *Proceedings of the Seventh Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann Publishers, San Mateo, California, 1991.

Andrew Gelman, John B. Carlin, Hal S. Stern, and Donald B. Rubin. *Bayesian Data Analysis.* Chapman and Hall, London, 1995.

Michael R. Genesereth and Nils J. Nilsson. *Logical Foundations of Artificial Intelligence*. Morgan Kaufmann Publishers, San Mateo, California, 1987.

Lise Getoor, Nir Friedman, Daphne Koller, and Avi Pfeffer. Learning Probabilistic Relational Models. In Saso Dzeroski and Nada Lavrac, editors. *Relational Data Mining*, Springer-Verlag, New York, New York, 2001.

Lise Getoor, Daphne Koller, Benjamin Taskar, and Nir Friedman. Learning Probabilistic Relational Models with Structural Uncertainty. In *Proceedings of the ICML-2000 Workshop on Attribute-Value and Relational Learning:Crossing the Boundaries*, Stanford, California, 2000.

Zoubin Ghahramani. Learning Dynamic Bayesian Networks. In C.L. Giles and M. Gori (eds.), *Adaptive Processing of Sequences and Data Structures*. Lecture Notes in Artificial Intelligence, 168-197. Berlin: Springer-Verlag, 1998.

W. Gilks, A. Thomas, and David J. Spiegelhalter. A language and program for complex Bayesian modeling. *The Statistician,* 43: 169-178, 1994.

Sabine Glesner and Daphne Koller. Constructing Flexible Dynamic Belief Networks from First-Order Probabilistic Knowledge Bases,.In Proceedings of the European Conference on Symbolic and Quantitative Approaches to Reasoning under Uncertainty (ECSQARU '95), Fribourg, Switzerland, July 1995. In Lecture Notes in Artificial Intelligence, Ch. Froidevaux and J. Kohlas (Eds.), Springer Verlag, 1995, pages 217-226.

Ulf Grenander. *Elements of Pattern Theory*. Johns Hopkins University Press, Baltimore, MD, 1995.

Thomas R. Gruber. A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition*, **5**(2), 199-220, 1993.

Joseph Y. Halpern. An Analysis of First-Order Logics of Probability. Artificial Intelligence 46: 311-350, May 1991.

David Heckerman, Christopher Meek and Daphne Koller, *Probabilistic Models for Relational Data*, Technical Report MSR-TR-2004-30, Redmond, WA: Microsoft Corporation, 2004.

David Heckerman, Daniel Geiger, and David Maxwell Chickering. Learning Bayesian Networks: The Combination of Knowledge and Statistical Data. *Machine Learning,* 20: 197-243, 1995.

Colin Howson and Peter Urbach. *Scientific Reasoning: The Bayesian Approach*. Open Court, 1993.

IET. *Quiddity*Suite Technical Guide*. Technical Report, Information Extraction and Transport, Inc., Rosslyn, VA, 2004.

Manfred Jaeger. Complex Probabilistic Modeling with Recursive Relational Bayesian Networks, *Annals of Mathematics and Artificial Intelligence*, 32: 179-220, 2001.

Manfred Jaeger. Reasoning about infinite random structures with relational Bayesian networks, in: *Principles of Knowledge Representation and Reasoning: Proceedings of the 6th International Conference (KR'98)* (1998) pp. 570–581.

Edward T. Jaynes. *Probability Theory: The Logic of Science*. Cambridge, UK, Cambridge University Press, 2003.

Finn V. Jensen. *Bayesian Networks and Decision Graphs*. Springer-Verlag, New York, New York, 2001.

Finn V. Jensen, Bo Chamberlain, Nordahl Torsten, and Frank Jensen. Analysis in HUGIN of Data Conflict. In *Uncertainty in Artificial Intelligence: Proceedings of the Sixth Conference*, Elsevier Science Publishing Company, New York, New York, 1991.

Michael I. Jordan, editor. *Learning in Graphical Models*. MIT Press, Cambridge, Massachussets, 1999.

Michael Kearns and Y. Mansour. Efficient Nash Computation in Large Population Games with Bounded Influence. In *Uncertainty in Artificial Intelligence: Proceedings of the Eighteenth Conference*, Morgan Kaufmann Publishers, San Mateo, California, 2002.

Kristian Kersting and Luc de Raedt. Adaptive Bayesian Logic Programs. In C. Rouveirol and M. Sebag (eds), *Proceedings of the Eleventh International Conference on Inductive Logic Programming* (ILP 2001). Springer-Verlag, 2001*a*.

Kristian Kersting and Luc De Raedt. *Bayesian Logic Programs*. Technical Report 151, Institute for Computer Science, University of Freiburg, Germany, 2001*b*.

Daphne Koller and Avi Pfeffer. Object-Oriented Bayesian Networks. In *Uncertainty in Artificial Intelligence: Proceedings of the Thirteenth Conference*, pages 302-313, Morgan Kaufmann Publishers, San Mateo, California, 1997.

Helge Langseth and Olav Bangsø. Parameter learning in object oriented Bayesian networks. *Annals of Mathematics and Artificial Intelligence*, 31(1/4): 221-243, 2001.

Helge Langseth and Thomas Nielsen. Fusion of Domain Knowledge with Data for Structured Learning in Object-Oriented Domains. *Journal of Machine Learning Research*, 2003.

Kathryn B. Laskey. Conflict and Surprise: Heuristics for Model Revision, in *Uncertainty in Artificial Intelligence: Proceedings of the Seventh Conference*, pages 197-204, Morgan Kaufman Publishers, San Mateo, California, 1991.

Kathryn B. Laskey, Bruce D'Ambrosio, Tod S. Levitt, and Suzanne M. Mahoney. Limited Rationality in Action: Decision Support for Military Situation Assessment. *Minds and Machines*, *10(1)*, 53-77, 2000.

Kathryn B. Laskey and Suzanne M. Mahoney. Network Fragments: Representing Knowledge for Constructing Probabilistic Models. In *Uncertainty in Artificial Intelligence: Proceedings of the Thirteenth Conference*, Morgan Kaufmann Publishers, San Mateo, California, 1997.

Kathryn B. Laskey, Suzanne M. Mahoney, and Edward Wright. Hypothesis Management in Situation-Specific Network Construction. In *Uncertainty in Artificial Intelligence: Proceedings of the Seventeenth Conference*, Morgan Kaufmann Publishers, San Mateo, California, 2001.

Laskey, K.B., Alghamdi, G., Wang, X., Barbara, D., Shackleford, T., Wright, E., and Fitzgerald, J., Detecting Threatening Behavior Using Bayesian Networks, *Proceedings of the Conference on Behavioral Representation in Modeling and Simulation*, 2004.

Stephan Lauritzen. *Graphical Models*. Oxford Science Publications, Oxford, 1996.

Tod S. Levitt, C. Larrabee Winter, Charles J. Turner, Richard A. Chestek, Gil J. Ettinger, and Steve M. Sayre. Bayesian Inference-Based Fusion of Radar Imagery, Military Forces and Tactical Terrain Models in the Image Exploitation System/Balanced Technology Initiative. *International Journal of Human-Computer Studies,* 42, 1995.

Y. Lin and M. J. Druzdzel. Computational Advantages of Relevance Reasoning in Bayesian Belief Networks. In Geiger, D. and Shenoy, P. (eds) Uncertainty in Artificial Intelligence:

Proceedings of the Thirteenth Conference, San Francisco, CA: Morgan Kaufmann. pp. 342-350, 1997.

Roderick JA Little and Donald B. Rubin. *Statistical Analysis with Missing Data*. Wiley, New York, NY, 1987.

Suzanne M. Mahoney. *Network Fragments.* PhD thesis, George Mason University, Fairfax, Virginia, 1999.

Suzanne M. Mahoney and Kathryn B. Laskey. Representing and Combining Partially Specified Conditional Probability Tables. In *Uncertainty in Artificial Intelligence: Proceedings of the Fifteenth Conference*, Morgan Kaufmann Publishers, San Mateo, California, 1999.

Suzanne M. Mahoney and Kathryn B. Laskey. Constructing Situation Specific Networks. In *Uncertainty in Artificial Intelligence: Proceedings of the Fourteenth Conference*, Morgan Kaufmann Publishers, San Mateo, California, 1998.

Brian Milch, Bhaskara Marthi, Stuart Russell, David Sontag, Daniel L. Ong, and Andrey Kolobov. "BLOG: Probabilistic Models with Unknown Objects." *Proceedings of the Nineteenth Joint Conference on Artificial Intelligence*, 2005.

Elliot Mendelson. *Introduction to Mathematical Logic*. Lewis Publishers, Inc., 1997.

Murphy, K. 1998. *Dynamic Bayesian Networks: Representation, Inference and Learning*. Computer Science Division. Berkeley, University of California.

Richard E. Neapolitan *Learning Bayesian Networks*. New York: Prentice Hall, 2003.

Alan Newell and Herbert Simon. Computer Science as Empirical Inquiry: Symbols and Search, *Communications of the ACM*, 19(3), 1976.

Liem Ngo and Peter Haddawy. Answering Queries from Context-Sensitive Probabilistic Knowledge Bases. *Theoretical Computer Science*, 171:147-177, 1997.

D. Oakes. Self Calibrating Priors Do Not Exist. *Journal of the American Statistical Association*, 80(390), 339.

R. C. Parker and R. A. Miller. Using causal knowledge to create simulated patient cases: the CPCS project as an extension of Internist-1. *Proceedings of the Eleventh Annual Symposium on Computer Applications in Medica Care*. IEEE Comp Soc Press, 473-480, 1987.

Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, San Francisco, California, 1988.

Judea Pearl. *Causality*. Cambridge University Press, Cambridge, UK, 2000.

Charles Sanders Peirce. On the Algebra of Logic. *American Journal of Mathematics*, 7:180-202, 1885.

Avi Pfeffer. *Probabilistic Reasoning for Complex Systems*. PhD dissertation, Stanford University, 2000.

Avi Pfeffer. IBAL: A Probabilistic Rational Programming Language *International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 985-991, 2001.

David Poole. First-Order Probabilistic Inference. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI),* 2003.

David Poole, Probabilistic Horn abduction and Bayesian networks, *Artificial Intelligence*, 64(1), 81-129, 1993.

W. V. Quine. *Methods of Logic* (4th edition). Cambridge, MA, Harvard University Press, 1982.

Christian Robert. *The Bayesian Choice* (2nd edition). Springer-Verlag, New York, New York, 2001.

Stuart Russell and Peter Norvig. *Artificial Intelligence*: *A Modern Approach*. (2nd edition) Prentice-Hall, New York, New York, 2002. grevious

Thomas J. Sargent, Rational Expectations, in David R. Henderson (ed.) *The Concise Encyclopedia of Economics*. Liberty Fund, Inc.: Library of Economics and Liberty. 30 August 2003. http://www.econlib.org/library/Enc/RationalExpectations.html.

Taisuke Sato. Modeling scientific theories as PRISM programs. *ECAI98 Workshop on Machine Discovery*, pp.37–45, 1998.

Leonard J. Savage. *The Foundations of Statistics*. New York: Dover Publications, Inc., 1972.

David Schum. *Evidential Foundations of Probabilistic Reasoning*. Wiley, New York, New York, 1994.

John F. Sowa. *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Pacific Grove, California, Brooks/Cole Thomson Learning, 2000.

David J. Spiegelhalter, Andrew Thomas, and Nicky Best. Computation on Graphical Models. *Bayesian Statistics,* 5: 407-425,1996.

Robert P. Stoll. *Set Theory and Logic*. New York, Dover Publications Inc., 1963.

Larry D. Stone, C. A. Barlow and Thomas L. Corwin. *Bayesian Multiple Target Tracking,* Boston, MA: Artech House, 1999.

Masami Takikawa, Bruce D'Ambrosio and Ed Wright, Real-time Inference with Large-scale Temporal Bayes nets, *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 2002.

Alfred Tarski. The Semantical Concept of Truth and the Foundations of Semantics, *Philosophy and Phenomenological Research* 4, 1944.

Michael Wellman, Jack Breese, and Robert Goldman. From Knowledge Bases to Decision Models. *Knowledge Engineering Review 7*(*1*): 35-52, 1992.

Joe Whittaker. *Graphical Models in Applied Multivariate Statistics*. Wiley, Chichester, 1990.

Edward J. Wright. *Understanding and Managing Uncertainty in Geospatial Data for Tactical Decision Aids*. Doctoral dissertation, George Mason University, 2002.

Wright, E., Mahoney, S., Laskey, K., Takikawa, M. and Levitt, T. Multi-Entity Bayesian Networks for Situation Assessment, *Proceedings of the Fifth International Conference on Information Fusion*, July 2002.

Wright E., Mahoney S.M., and Laskey K.B., Use of Domain Knowledge Models to Recognize Cooperative Force Activities, *Proc. 2001 MSS National Symposium on Sensor and Data Fusion*, San Diego, CA, June 2001.