Factored Models for Probabilistic Modal Logic

Afsaneh Shirazi and **Eyal Amir**
Computer Science Department, University of Illinois at U-C
Urbana, IL 61801, USA {hajiamin, eyal}@uiuc.edu

Abstract

Modal logic represents knowledge that agents have about other agents' knowledge. *Probabilistic modal logic* further captures probabilistic beliefs about probabilistic beliefs. Models in those logics are useful for understanding and decision making in conversations, bargaining situations, and competitions. Unfortunately, probabilistic modal structures are impractical for large real-world applications because they represent their state space explicitly. In this paper we scale up probabilistic modal structures by giving them a factored representation. This representation applies conditional independence for factoring the probabilistic aspect of the structure (as in Bayesian Networks (BN)). We also present two exact and one approximate algorithm for reasoning about the truth value of probabilistic modal logic *queries* over a model encoded in a factored form. The first exact algorithm applies inference in BNs to answer a limited class of queries. Our second exact method applies a variable elimination scheme and is applicable without restrictions. Our approximate algorithm uses sampling and can be used for applications with very large models. Given a query, it computes an answer and its confidence level efficiently.

1 Introduction

Reasoning about knowledge plays an important role in various contexts ranging from conversations to imperfect-information games. Formal models of reasoning about knowledge use modal operators and logic to express *knowledge* and *belief*. These enable agents to take into account not only facts that are true about the world, but also the knowledge of other agents. For example, in a bargaining situation, the seller of a car must consider what the potential buyer knows about the car's value. The buyer must also consider what the seller knows about what the buyer knows about the value, and so on.

In many applications, it is not enough to include certain knowledge or lack thereof. For example, the seller of a car may not know the buyer's estimate of the car's value, but may have a probability distribution over it. Current formal logical systems (especially modal logics) (Fitting 1993; Fagin *et al.* 1995) cannot represent such scenarios. On the other hand, probabilistic graphical models (Pearl 1988) can represent distributions over distributions (Blei, Ng, & Jordan 2003), but are not granular enough for multiple levels of complex beliefs.

A number of works have presented frameworks capable of capturing probabilistic beliefs about probabilistic beliefs (Fagin & Halpern 1988; Heifetz & Mongin 1998; Aumann & Heifetz 2001; Shirazi & Amir 2007). These rely on probabilistic modal structures that combine accessibility graphs and probabilistic distributions. However, reasoning with such structures does not scale to domains of many states because it accesses every state explicitly. Consequently, reasoning is impractical even for simple scenarios such as Poker ($\binom{52}{5}.\binom{47}{5} > 10^{12}$ states).

In this paper we provide a compact model and efficient reasoning algorithms for nested probabilistic modalities. Our model is capable of representing applications with large state spaces. We describe syntax and semantics of our representation, as well as reasoning algorithms for evaluating a query on a model.

More specifically, we introduce a framework for modeling probabilistic knowledge that uses a Bayesian Network (BN) (Pearl 1988) to represent the probabilistic relationship between states. We provide two exact methods for answering queries based on the class of queries of interest. Our first method performs inference in a BN representation of the combined query and model. Our second method is a variable elimination scheme in which we compute values for sub-formulas of the query in a dynamic-programming fashion. We show theoretically and experimentally that these methods are faster than earlier techniques (Shirazi & Amir 2007). We also introduce a sampling method that is tractable on larger models for which the exact methods are intractable. We show that for a specific group of queries the probability of error in the estimated value of the query does not decrease when the number of samples increases. Our method addresses this issue by computing the confidence in the answer to the query.

Most previous related works are limited to combining probability with a special case of modal logic in which accessibility relations are equivalence relations, which is called *probabilistic knowledge*. Among those, (Fagin & Halpern 1988; Heifetz & Mongin 1998) are mainly concerned with providing a sound and complete axiomatization for the logic of knowledge and probability. (Shirazi & Amir

2007) provides reasoning methods for general probabilistic modal structures but does not scale up to large domains.

Another related work is (Milch & Koller 2000) in which probabilistic epistemic logic is used to reason about the mental states of an agent. This logic is a special case of probabilistic knowledge with the additional assumption of agents having a common prior probability distribution over states.

Adding probabilistic notions to modal logic is also considered in (Herzig 2003; Nie 1992). The former adds a unary modal operator expressing that a proposition is more probable than its negation, whereas the latter defines an extension of fuzzy modal logic to perform probabilistic semantic-based approaches for finding documents relevant to a query.

Related works in the game theory literature mainly focus on imperfect information games. For example, (Koller & Pfeffer 1995) provides an algorithm for finding optimal randomized strategies in two-player imperfect information competitive games. The state of these games can be represented with our model.

2 Factored Probabilistic Modal Structures

In this section, we provide a compact Bayesian Network (BN) representation for probabilistic Kripke structures. A BN is a directed acyclic graph in which nodes represent random variables, and the joint distribution of the node values can be written as $Pr(X_1, \ldots, X_n) = \Pi_{i=1}^{n} Pr(X_i|\text{parents}(X_i))$. According to (Shirazi & Amir 2007), a probabilistic Kripke structure consists of a set of states and a probabilistic accessibility relation (we redefine it in the next section). The probabilistic accessibility relation is defined for each pair of states. The probabilistic Kripke structure is basically a graph of states with labeled edges. Therefore, the size of this structure is quadratic in the number of states and it is not scalable to large domains.

In the following section we review the probabilistic Kripke structures. Then, in Section 2.2 we provide complete specifications of our representation.

2.1 Probabilistic Kripke Structures

For simplicity we assume that the agent wishes to reason about a world that can be described in terms of a nonempty set, $\mathbf{Z}$, of state variables. *Probabilistic modal formulas* are built up from a countable set of state variables $\mathbf{Z}$ using equality (=), propositional connectives ($\neg$, $\wedge$), and the modal function K. We use $\top$ and $\bot$ for truth and falsehood constants. First, we need to define *non-modal* formulas. The formation rules are:

1. For every state variable X and value x, $X = x$ is a non-modal formula.
2. $\top$ and $\bot$ are non-modal formulas.
3. If X is a non-modal formula so is $\neg X$.
4. If X and Y are non-modal formulas so is $X \wedge Y$.
5. Every non-modal formula is a probabilistic modal formula.
6. If X is a probabilistic modal formula so is $(K(X) \propto r)$ when $0 < r \leq 1$ and $\propto \in \{<, =\}$.

Note that we have a different modal function K_i for each agent i in the domain. We take $\vee$, $\supset$ and $\equiv$ to be abbreviations in the usual way.

We use Texas Holdem poker game as an example of a game with imperfect knowledge that can be modeled using our framework. In Holdem, players receive two downcards as their personal hand, after which there is a round of betting. Three boardcards are turned simultaneously (called the "flop") and another round of betting occurs. The next two boardcards are turned one at a time, with a round of betting after each card. A player may use any five-card combination from the board and personal cards. There are some rules that applied to the cards to rank hands. For more info refer to http://www.texasholdem-poker.com/handrank.

In the Holdem example, suppose that we introduce two new propositional symbols, w_1 and w_2, to show whether player 1 or player 2 wins the hand, respectively. The value of these symbols is determined based on the game rules applied to players' hands and boardcards (players' hands and boardcards are state variables). In this example there are two players, therefore we have two modal functions, K_1 and K_2 corresponding to player 1 and player 2. $K_1(w_1) < 1/2$ is an example of a probabilistic modal formula whose truth value can be evaluated on the current state of the world. $K_1(w_1) < 1/2$ demonstrates that the probability of player 1 winning the hand is less than $1/2$ from her perspective.

Now, we describe the semantics of our language. Our approach is in terms of possible worlds which is similar to *Kripke structures* in modal logic.

Definition 2.1 *A* probabilistic Kripke structure M *is a tuple* $(\mathcal{S}, \mathcal{P}, \mathcal{V})$ *in which*

1. $\mathcal{S}$ *is a nonempty set of* states *or possible worlds.*
2. $\mathcal{P}$ *is a* conditional probability function. $\mathcal{P}(s'|s)$ *denotes the probability of accessing state* s' *given that we are in state* s. $\mathcal{P}(s'|s) > 0$ *indicates that* s' *is accessible from* s. *Therefore, it is similar to the accessibility relation of modal logic. Since* $\mathcal{P}$ *is a probability function, we should ensure that the following constraints hold:*
   - $0 \leq \mathcal{P}(s'|s) \leq 1$
   - *For each state* $s \in \mathcal{S}$: $\sum_{s' \in \mathcal{S}} \mathcal{P}(s'|s) = 1$
3. $\mathcal{V}$ *is an* interpretation *that associates with each state in* $\mathcal{S}$ *a value assignment to the state variables in* $\mathbf{Z}$.

Probabilistic knowledge is a special case of probabilistic modal logic in which the accessibility relation is an equivalence relation. This model captures the intuition that agent i considers state t accessible from state s if in both s and t she has the same knowledge about the world.

The above definition is for one modal function. When we have j modal functions $(K_1, \ldots K_j)$ we need a probability function $\mathcal{P}_i$ for each modal function K_i. Intuitively, the probability function $\mathcal{P}_i$ represents the probability of accessing a state from the perspective of agent i.

In probabilistic modal logic, the true state of the world is a state in $\mathcal{S}$. An agent has a probability distribution over all the states that are possible to her given the true state of the world .

Example. Let (KJ, K3, KKQ32) be the cards of players and the boardcards. Since player 1 does not know her opponent's hand, $\mathcal{P}_1$((KJ, 65, KKQ32)|(KJ, K3, KKQ32)) should be > 0. $\mathcal{P}_1$ is uniform on all the possible states since the player does not have any information about her opponents hand.

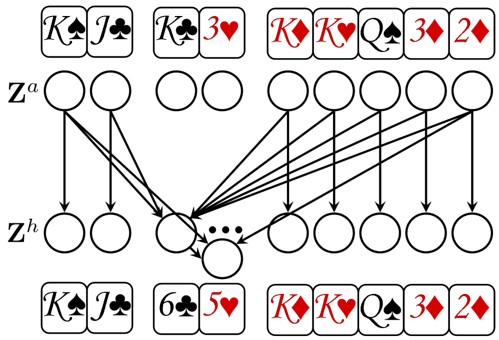

Figure 1: GKM of Holdem.

The truth value of any formula in a given state is evaluated with the following definitions.

Definition 2.2 *Let $M = (\mathcal{S}, \mathcal{P}, \mathcal{V})$ be a probabilistic Kripke structure. Let s be a state in $\mathcal{S}$. For any probabilistic modal formula φ, we define $val(s, \varphi)$ recursively as follows.*

1. $val(s, X = x) = (\mathcal{V}(s, X) = x)$ *for a state variable* X.
2. $val(s, \top) = true$.
3. $val(s, \bot) = false$.
4. $val(s, \neg X) = \neg val(s, X)$.
5. $val(s, X \wedge Y) = val(s, X) \wedge val(s, Y)$.
6. $val(s, K(X) \propto r) = true$ *iff* $\sum_{s' \in \mathcal{S}, val(s', X) = true} \mathcal{P}(s'|s) \propto r$.

We use val's definition to define logical entailment, $\models$.

Definition 2.3 $M, s \models X$ *iff* $val(s, X)$ *is true.*

2.2 Graphical Kripke Models

A probabilistic Kripke structure as defined above, $(\mathcal{S}, \mathcal{P}, \mathcal{V})$, has size $O(|\mathcal{S}|^2)$. This representation is impractical for large state spaces. In this section we provide a more compact representation for probabilistic Kripke structures.

In our new model, a state is represented by a set of state variables, $\mathbf{Z}$. $\mathcal{P}$ is represented by a BN with $2|\mathbf{Z}|$ variables: $Z^a_{(i)}$ and $Z^h_{(i)}$ for each $Z_{(i)} \in \mathbf{Z}$. $Z^a_{(i)}$ stands for a state variable for the *actual state* of the world, whereas $Z^h_{(i)}$ represents a variable for a *hypothetical state* of the world (an agent cannot distinguish with certainty between this state and the actual state). $Pr(\mathbf{Z}^h|\mathbf{Z}^a)$ is represented by the BN which serves as $\mathcal{P}$ of probabilistic Kripke structures (*e.g.,* see Figure 1).

Definition 2.4 *A* graphical Kripke model *(GKM) M on a set of random variables $\mathbf{Z}$ is a BN which is defined as follows:*

1. *The nodes are:* $Z^a_{(1)}, \ldots, Z^a_{(|Z|)}, Z^h_{(1)}, \ldots, Z^h_{(|Z|)}$
2. $Pr(\mathbf{Z}^h|\mathbf{Z}^a)$ *is defined by the edges of* M.
3. *There are no edges between the nodes in* $\mathbf{Z}^a$.

The above definition is for one modal function, K. For cases with j modal functions, $K_1, \ldots K_j$, we need to define $\mathbf{Z}^{h_i}$ and $Pr(\mathbf{Z}^{h_i}|\mathbf{Z}^a)$ for each modal function K_i.

Figure 1 shows the GKM of our Holdem example. The nodes in the first row represent the actual state of the world, whereas the second row represents a possible state of the world. Each node takes values from {A,2,...,10,J,Q,K} $\times$ {♠, ♡, ♣, ♢}. The first and second nodes are observed by player 1 to have values K♠ and J♣, respectively. In each row, the first two nodes correspond to player 1's hand, the second two nodes correspond to player 2's hand, and the last five are the boardcards. From the perspective of player 1, player 2 can have any cards except the boardcards and the cards in her hand. In the BN, this is shown by the edges to the third and forth node in $\mathbf{Z}^h$. The boardcards and player 1's hand cards are the same in the actual state of the world and the hypothetical state of the world. Let $Z_1, \ldots, Z_9$ stand for the nodes in each row (Z_1 be the leftmost node), the conditional probability functions are:

$$Pr(Z^h_i|Z^a_i) = \begin{cases} 1 & \text{if } Z^h_i = Z^a_i; \\ 0 & \text{otherwise.} \end{cases} \text{ for } i \in \{1, 2, 5, \ldots, 9\}$$

$$Pr(Z^h_3|Z^a_1, Z^a_2, Z^a_5, \ldots, Z^a_9) = \frac{1}{\alpha} \begin{cases} 1 & \text{if } Z^h_3 \notin \{Z^a_1, Z^a_2, Z^a_5, \ldots, Z^a_9\}; \\ 0 & \text{otherwise.} \end{cases}$$

As shown in the above equation, Z^h_3 has a uniform distribution. α is the normalization factor. The conditional probability function for Z^h_4 is the same as Z^h_3 except that Z^h_4 is a child of Z^h_3 and should not be equal to Z^h_3 as well.

Theorem 2.5 *Let $\mathbf{Z}$ be the set of state variables and k be the number of agents. GKM has $O(k|\mathbf{Z}|)$ nodes and $O(k|\mathbf{Z}|^2)$ edges and each node has* at most $2|Z| - 1$ *parents.*

Note that this model is most useful when the size of the largest Conditional Probability Table (CPT) is much smaller than $|\mathcal{S}|$ or when the CPTs can be represented compactly (*e.g.,* uniform distribution). In those cases, the size of GKM is much smaller than the size of the corresponding probabilistic Kripke structure ($O(2^{|\mathbf{Z}|})$ nodes when state variables are binary).

Example. We define a simpler 2-player game, named War, for the purpose of exposition with a smaller set of states. In War, there is a deck of three cards, A, B, and C. One card is dealt to each player and the third card is face down. The player with the highest card wins the game ($A > B > C$). The Kripke model has six states, and so can be easily analyzed. The equivalence classes for player 1 are shown in the right part of Figure 2. The first rounded rectangle corresponds to the class in which player 1 has A and player 2 either has B or C (the actual state of the world is either AB or AC). In this equivalence class, player 1 knows that she is winning with probability 1. Player 1 has a probability distribution over each of these equivalence classes.

The GKM representing the equivalence classes of player 1 is shown in the left part of Figure 2. $\mathbf{Z}^a$ represent the actual

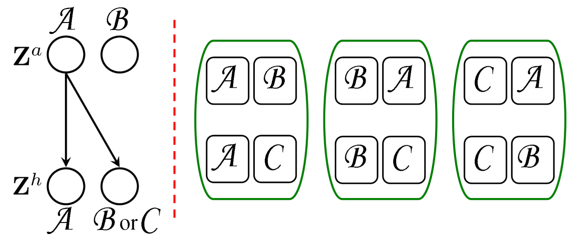

Figure 2: Left: GKM of War. Right: Kripke structure of War.

FUNCTION Q2BN[a](query q, set of state variables $\mathbf{Z}$)
Pa: associates with each node a set of parent nodes.
This function returns the query node of the BN.
if $q = K_i(x)$ **then**
  $\mathbf{Z}^{h_i} \leftarrow$ a set of new nodes for all state variables
  $Pa(\mathbf{Z}^{h_i}) \leftarrow \mathbf{Z}$ with conditional probability of $\mathcal{P}_i$
  return Q2BN(x, $\mathbf{Z}^{h_i}$)
else
  $n \leftarrow$ new node
  if q is $Z_{(i)} = x$ **then** $Pa(n) = Z_{(i)}$
  else if $q = \neg x$ **then** $Pa(n) =$ Q2BN(x, $\mathbf{Z}$)
  else if $q = x \wedge y$ **then**
    $Pa(n) = \{$Q2BN(x, $\mathbf{Z}$), Q2BN(y, $\mathbf{Z}$)$\}$
  return n

[a] We do not delve into the details of CPTs in this function.

FUNCTION QuAn(query q, state s)
1. $\mathbf{Z}^a \leftarrow$ a set of new nodes corresponding to all state variables
2. **return** $Pr($Q2BN(q, $\mathbf{Z}^a$)$= 1 | \mathbf{Z}^a = s)$
   //the probability is computed using any BN inference method

Figure 3: Query Answering (QuAn) algorithm.

state of the world and $\mathbf{Z}^h$ represent the hypothetical state of the world that player 1 considers possible. In this example $P(Z_2^h|Z_1^a)$ is uniform when $Z_2^h \neq Z_1^a$ ($P(Z_2^h = B|Z_1^a = A) = \frac{1}{2}$ and $P(Z_2^h = C|Z_1^a = A) = \frac{1}{2}$). $P(Z_1^h|Z_1^a)$ is equal to 1 when $Z_1^h = Z_1^a$ and is 0 otherwise.

3 Query Answering

In this section we provide reasoning methods for answering queries over GKMs. Previous sections showed that using GKMs potentially reduces the size of the model exponentially. The reasoning methods known for probabilistic Kripke structures cannot be used on GKMs in practice. This is because they enforce explicit access to every state in the probabilistic Kripke structure. In this section we design new methods for reasoning with GKMs and show that they are more efficient than their counterparts for probabilistic Kripke structures.

In Section 3.1 we investigate a class of queries that can be answered by inference in BNs. We also introduce a method that answers these queries by taking advantage of the BN structure of GKMs. In Sections 3.2 and 3.3 we provide an exact and a sampling algorithm for answering probabilistic modal queries (defined in Section 2.1), respectively.

3.1 Answering Expectation Queries

Any probabilistic modal query with no nested modal function can be answered by computing the marginal probability of a node in a BN. For example, for $K_1(x) < r$ we add a node x as a child of the hypothetical state of the corresponding GKM. Since x is a non-modal formula, it can be easily added as a node to the BN. In this new model $Pr(x|\mathbf{Z}^a = s)$ is equal to the value of $K_1(x)$ on state s which can be compared to r, thus answering $K_1(x) < r$.

A probabilistic modal query with nested modal functions cannot be modeled with a BN since the parameter of its modal function is inequality. For example, BNs cannot represent queries such as $K_1(K_2(x) < \frac{1}{2}) < \frac{3}{4}$. In this example $K_2(x)$ can be represented by the marginal probability of a node given the actual state of the world in a BN. However, we do not know a way to introduce another node that compares this value with a number.

Inference can answer queries with no inequality (*e.g.*, $K_1(K_2(x))$). The answer to such queries is a number between 0 and 1. In these queries we treat the modal function as an expectation function. $K_1(K_2(x))$ denotes the expected value from the perspective of agent 1 of the expected value of x from the perspective of agent 2.

Based on Definition 2.2, the value of $K_1(x)$ is equal to the expected value of x from the perspective of agent 1.

$$K_1(x) = \sum_{s' \in \mathcal{S}} x(s')\mathcal{P}_1(s'|s) = E_1(x)$$

where $x(s')$ denotes the value of x on state s', and $\mathcal{P}_1(s'|s)$ is the probability of accessing state s' when the true state of the world is s from the perspective of agent 1.

The expectation queries have the following format:
1. Every non-modal formula is a query.
2. If Q is a query so is $K(Q)$.

Algorithm QuAn of Figure 3 computes the answer to such queries. First, it transforms the query into a BN (using the GKM), and then computes the answer to the query by performing inference in the BN (any inference method can be used (Zhang & Poole 1994; Jordan *et al.* 1999; Yedidia, Freeman, & Weiss 2004)).

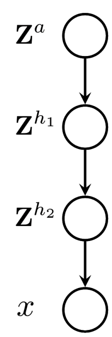

Figure 4: BN for $K_1(K_2(x))$

Example. Suppose that the query is $K_1(K_2(x))$ on state s. First, we need to transform this query into a BN. The BN is shown in Figure 4. $\mathbf{Z}^a$, $\mathbf{Z}^{h_1}$, and $\mathbf{Z}^{h_2}$ each represent a set of nodes. $\mathbf{Z}^a$ is the actual state of the world. $\mathbf{Z}^{h_1}$ is the hypothetical state from the perspective of player 1 (since the first modal function is K_1). $\mathbf{Z}^{h_2}$ is the hypothetical state from the perspective of player 2. The last node, x, is a non-modal formula on the state variables, therefore, can be represented by a node. After creating the BN, the value of $Pr(x|\mathbf{Z}^a = s)$ is computed by performing inference in the BN. This value is equal to the value of $K_1(K_2(x))$ on state s and so is the answer to the query.

The following equations justify our method. It shows that the value of $Pr(x|\mathbf{Z}^a = s)$ is equal to the value of $K_1(K_2(x))$ on state s.

$$\begin{aligned} Pr(x|\mathbf{Z}^a = s) &= \sum_{\mathbf{Z}^{h_1}} \sum_{\mathbf{Z}^{h_2}} Pr(x, \mathbf{Z}^{h_1}, \mathbf{Z}^{h_2}|\mathbf{Z}^a = s) \\ &= \sum_{\mathbf{Z}^{h_1}} \sum_{\mathbf{Z}^{h_2}} Pr(x|\mathbf{Z}^{h_2})Pr(\mathbf{Z}^{h_2}|\mathbf{Z}^{h_1})Pr(\mathbf{Z}^{h_1}|\mathbf{Z}^a = s) \\ &= \sum_{\mathbf{Z}^{h_1}} Pr(\mathbf{Z}^{h_1}|\mathbf{Z}^a = s) \sum_{\mathbf{Z}^{h_2}} Pr(x|\mathbf{Z}^{h_2})Pr(\mathbf{Z}^{h_2}|\mathbf{Z}^{h_1}) \\ &= \sum_{\mathbf{Z}^{h_1}} Pr(\mathbf{Z}^{h_1}|\mathbf{Z}^a = s) \left(K_2(x) \text{ on } \mathbf{Z}^{h_1}\right) \\ &= K_1(K_2(x)) \text{ on } s \end{aligned}$$

In the following sections we provide efficient algorithms for queries with inequalities (probabilistic modal queries).

3.2 Ordered Variable Elimination

In this section we provide an algorithm to answer probabilistic modal queries. In the previous section we mentioned that existing BN inference methods cannot answer these queries. The algorithm that we introduce is called *Ordered Variable Elimination* (OVE). The following example justifies that the original variable elimination (see (Pearl 1988)) does not answer the following query because some of the summations participate in inequalities. Therefore, the order of some of the summations cannot be changed.

Example. Assume the query $K_1(K_2(x) < \frac{1}{2}) < \frac{3}{4}$ on s. This query is calculated as follows:

$$\begin{aligned} & K_1(K_2(x) < \frac{1}{2}) < \frac{3}{4} \text{ on } s \\ = & \Big(\sum_{\mathbf{Z}^{h_1}} Pr(\mathbf{Z}^{h_1}|\mathbf{Z}^a = s)\Big(K_2(x) < \frac{1}{2} \text{ on } \mathbf{Z}^{h_1}\Big)\Big) < \frac{3}{4} \\ = & \Big(\sum_{\mathbf{Z}^{h_1}} Pr(\mathbf{Z}^{h_1}|\mathbf{Z}^a = s) \\ & \quad \Big(\big(\sum_{\mathbf{Z}^{h_2}} Pr(x|\mathbf{Z}^{h_2}) Pr(\mathbf{Z}^{h_2}|\mathbf{Z}^{h_1})\big) < \frac{1}{2}\Big)\Big) < \frac{3}{4} \end{aligned}$$

In this formula we cannot move $\sum_{\mathbf{Z}^{h_1}}$ inside $\sum_{\mathbf{Z}^{h_2}}$, since the latter participates in an inequality. OVE performs variable elimination on this formula in two rounds. It eliminates variables $\mathbf{Z}^{h_2}$ in the first round and variables $\mathbf{Z}^{h_1}$ in the second round. Assume that $\mathbf{Z}'$ which is a subset of $\mathbf{Z}^{h_1}$ is the set of parents of $\mathbf{Z}^{h_2}$ in the BN calculated by Function Q2BN of the previous section. After the first round of variable elimination $\big(\sum_{\mathbf{Z}^{h_2}} Pr(x|\mathbf{Z}^{h_2})Pr(\mathbf{Z}^{h_2}|\mathbf{Z}^{h_1})\big) < \frac{1}{2}$ is replaced by $f(\mathbf{Z}') < \frac{1}{2}$. The result is a summation over $\mathbf{Z}_{h_1}$ which is computed in the second round of variable elimination.

The algorithm is shown in figure 5. There are a few standard ways to speed up this function. For example, instead of summing over all $\mathbf{Z}$s we can sum over those in which $Pr(\mathbf{Z}|\mathbf{Z}')$ is not zero. This will provide a faster approach when $Pr(\mathbf{Z}|\mathbf{Z}')$ is sparse.

Theorem 3.1 *Let q be a query, s be a state, v be the maximum size of the domain of random variables, and t be the size of the largest factor. Function OVE calculates the value of q on s in $O(v^t)$ time.*

Elimination is deriven by an ordering on variables. OVE does not allow all the orders. Therefore, for some graphs its running time is worse than the variable elimination's. Typically, $v^t << |\mathcal{S}|$. However, the worst-case running time of this algorithm is the same as the running time of GBU in (Shirazi & Amir 2007) which is the fastest exact method in that paper.

3.3 Sampling with Confidence

In this section we provide a sampling method to answer queries on GKMs. Our method is based on probabilistic logic sampling of (Henrion 1986) which is the simplest and first proposed sampling algorithm for BNs. This method is optimal for our query answering because our evidence nodes are root nodes. The details of the method is provided in the rest of this section.

FUNCTION OVE(query q, state s)
q: the query of the form $K^{(1)}(\dots(K^{(m)}(x) < n^{(m)})\dots) < n^{(1)}$ in which $K^{(i)} \in \{K_1, \dots, K_j\}$ for j agents[a]
s: list of actual-state values
$\mathbf{Z}^a$: list of actual-state variables
Y: query node[b]
$\mathbf{Z}^{(i)}$: list of hypothetical-state variables corresponding to $K^{(i)}$
$\mathbf{Z}^a \leftarrow$ set of new nodes for all state variables
$Y \leftarrow$ Q2BN (q, $\mathbf{Z}^a$)
$F \leftarrow$ list of conditional probabilities in the model
for $i \leftarrow m$ to 1
  while $\mathbf{Z}^{(i)}$ is not empty
    remove node z from $\mathbf{Z}^{(i)}$
    sum-out(F, z)
  $f(\mathbf{Z}') \leftarrow$ the result of previous loop /*$\mathbf{Z}' \subseteq \mathbf{Z}^{(i-1)}$*/
  add $f(\mathbf{Z}') < n^{(i)}$ to F
$h \leftarrow$ the multiplication of all factors in F
return $\frac{h(Y)}{\sum_Y h(Y)}$

[a]For simplicity we only treat $<$. The inequality can be $>$ as well.

[b]Q2BN is similar to the one in the previous section except that these queries have inequalities. But it does not affect the BN.

Figure 5: Ordered Variable Elimination (OVE) algorithm.

First, we show that the estimated values of some queries may not converge to the true values by increasing the number of samples. Consequently, the only way to answer these queries is to use an exact method. The following theorem states this result.

Theorem 3.2 *Let $K(x) < n$ be a query, s be a state, and $s_1, s_2, \dots$ be a sequence of independent and identically distributed states sampled from $Pr(\mathbf{Z}^h|\mathbf{Z} = s)$. Define $\hat{K}_m = \frac{x(s_1)+\ldots+x(s_m)}{m}$ to be the observed value of $K(x)$ using m samples. $Pr(\lim_{m\to\infty}(\hat{K}_m < n)$ does not exist $) = 1$ when n is equal to the value of $K(x)$ on s and $0 < n < 1$.*

Proof Sketch. We show the proof for $n = \frac{1}{2}$. We show that $Pr(\lim_{m\to\infty}(\hat{K}_m < \frac{1}{2})$ does not exist $) = 1$ when the value of $K(x)$ on s is equal to $\frac{1}{2}$.

Since the value of $\hat{K}_m < n$ for a specific m is either 0 or 1, $\lim_{m\to\infty}(\hat{K}_m < \frac{1}{2})$ (if exists) should be either 0 or 1. First, we show that $Pr(\lim_{m\to\infty}(\hat{K}_m < \frac{1}{2}) = 1) = 0$ (the proof for value 0 is similar to the proof for value 1).

By definition of limit of a function at infinity, $\lim_{m\to\infty}(\hat{K}_m < \frac{1}{2}) = 1$ if and only if for each $\epsilon > 0$ there exists an N such that $|(\hat{K}_m < \frac{1}{2}) - 1| < \epsilon$ whenever $m > N$. Since $\hat{K}_m < \frac{1}{2}$ is binary, our definition would be $\hat{K}_m < \frac{1}{2}$ for $m > N$.

Each sample is drawn from a Bernoulli distribution with mean $\frac{1}{2}$. To compute the above probability we need to answer the following question. In a one-dimensional random walk, what is the probability that no return to the origin occurs up to and including time $2m$? A random walk is modeled with $X(t+1) = X(t) + \Phi(t)$. In this notation, $\Phi(t)$s are independent and identically distributed Bernoulli random variables that have value $+1$ and -1 with probability $1/2$ at all times.

By lemma 1 of chapter III.3 of (Feller 1968), the probability that no return to the origin occurs up to and including time $2m$ is the same as the probability that a return occurs at time $2m$ (*i.e.*, the number of +1s is equal to the number of -1s). This probability is:

$$Pr(\text{return at time } 2m) = \frac{\binom{2m}{m}}{2^{2m}}$$

We calculate the probability that in an infinite sequence no return to the origin occurs by computing the limit of above probability at infinity.

$$Pr(\lim_{m\to\infty}(\hat{K}_m < \frac{1}{2}) = 1) = \lim_{n\to\infty}\frac{\binom{2m}{m}}{2^{2m}} = 0$$

Similarly we can show that $Pr(\lim_{m\to\infty}(\hat{K}_m < \frac{1}{2}) = 0) = 0$. Using these results, it holds that $Pr(\lim_{m\to\infty}(\hat{K}_m < n)$ does not exist $) = 1$. □

Based on this theorem, the accuracy of a sampling method does not necessarily increase with the number of samples. To estimate the accuracy of the value of a query, our sampling method not only calculates the truth value of the query but also returns the confidence level of the method in that value. The confidence level is the probability of the query being true given the set of samples.

Function CoSa shown in Figure 6 presents our sampling method. It returns the probability of the query being true. The function first transforms the query to its BN representation. Then, it calculates the probability of $q = 1$ recursively using Function RecCS (*e.g.,* if RecCS returns 0 the truth value of the query is equal to 0).

For queries with no modal function, RecCS calculates the value of q on s (the details of this calculation is not shown in the function) and returns the probability of $q = 1$ based on this value. For queries with modal functions such as q = $K_i(q') < n$, RecCS repeats the following step m times. It samples a new state s' accessible from s and recursively computes the probability of q' on s'. Then, RecCS calculates the probability of $q = 1$ using the probabilities of the values of these samples. Function RecCS calls Function CalculateProb to perform this calculation. In the next few paragraphs we explain how CalulateProb computes the probability of the query.

Imagine the query $q = K_1(x) < 0.4$ on s. First, we sample m states from the probability distribution $Pr(\mathbf{Z}^h|\mathbf{Z} = s)$ and we calculate the value of x on each sampled state. Then, we compute the probability of $q = 1$ given these values. Let there be k samples with value 1. Each sample is a Bernoulli trial whose probability of success is $p = K_1(x)$. The sample proportion $\hat{p}$ is the fraction of samples with value 1 so $\hat{p} = \frac{k}{m}$. When m is large, $\hat{p}$ has an approximately normal distribution. The standard deviation of the sample proportion is $\sigma = \sqrt{\frac{p(1-p)}{m}}$. Since the true population proportion (p) is unknown, we use standard error instead of σ. The standard error provides an unbiased estimate of the standard deviation. It can be calculated from the equation $SE = \sqrt{\frac{\hat{p}(1-\hat{p})}{m}}$. Therefore, $\frac{K_1(x)-\hat{p}}{SE} \sim N(0,1)$and the probability of the query is calculated from:

$$Pr(K_1(x) < n) = \Phi\Big(\frac{n-\hat{p}}{\sqrt{\frac{\hat{p}(1-\hat{p})}{m}}}\Big) \quad (1)$$

where Φ is the cumulative distribution function of the standard normal distribution.

For queries with nested modal functions, the exact values of the sub-query on the sampled states are unknown. We only have the probability of the sub-query on those states. To calculate the probability of $K_1(q') < n$ where q' has a modal function, we use the equation bellow:

$$Pr(K_1(q') < 0.4) = \sum_{(v_1,\ldots,v_m)\in\{0,1\}^m} Pr(K_1(q') < 0.4 | q'_{s_1} = v_1, \ldots, q'_{s_m} = v_m) \; Pr(q'_{s_1} = v_1)\ldots Pr(q'_{s_1} = v_1)$$

where $s_1, \ldots, s_m$ are sampled states and $q'_{s_i} = v_i$ means the value of q' on sampled state s_i is equal to v_i. $Pr(q'_{s_i} = v_i)$ is calculated recursively. $Pr(K_1(q') < 0.4 | q'_{s_1} = v_1, \ldots, q'_{s_m} = v_m)$ is calculated using Formula 1 with $\hat{p}$ equal to sample proportion.

Theorem 3.3 *Let q be a query, s be a state, k be the number of nested modal functions, and m be the number of samples at each stage. Function CoSa calculates the truth value of q on s in $O(m^{k+2})$ time.*

FUNCTION CoSa(query q, state s)
q: the query of the form $K^{(1)}(\ldots(K^{(m)}(x) < n^{(m)})\ldots) < n^{(1)}$ in which $K^{(i)} \in \{K_1, \ldots, K_j\}$ for j agents
s: list of actual-state values
$\mathbf{Z}^a$: list of actual-state variables
Y: query node
$\mathbf{Z}^{(i)}$: list of hypothetical-state variables corresponding to $K^{(i)}$

```
Z^a ← set of new nodes for all state variables
Y ← Q2BN (q, Z^a)
return RecCS(q, s, Z^a)
```

FUNCTION RecCS(query q, state s, set of nodes $\mathbf{Z}$)

```
if q = (K^(i)(q') < n) then
    T ← ∅
    for j ← 1 to m
        s_j ← sample according to Pr(Z^(i)|Z = s)
        s' ← value of non-leaf nodes in Z^(i)
        if s' ∉ T then
            T ← add s' to T with weight 1
            conf(s') ← RecCS(q', s_j, Z^(i))
        else
            weight(s') ← weight(s') +1
    return CalculateProb(q, T)
else if q = x then
    if value x on s is true then return 1
    else return 0
```

Figure 6: Confidence Sampling (CoSa) algorithm.

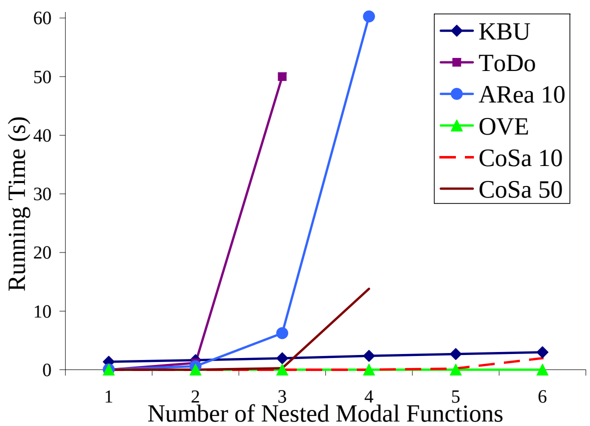

Figure 7: Running time comparison of exact methods (KBU, ToDo, OVE) with approximate methods (ARea 10, CoSa 10, CoSa 50).

4 Experimental Results

In this section we compare the running time of OVE and CoSa with their counterparts in (Shirazi & Amir 2007)(KBU (Knowledge Bottom-Up), ToDo (Top-Down), and ARea 10 (Approximate Reasoning)). This section confirms the theoretical results of section 3.3 about the running time of our algorithms. Figure 7 shows the running time of these methods on Holdem. ToDo and KBU are exact methods defined for probabilistic Kripke structures and ARea 10 is a sampling method in which the number of samples used to evaluate each modal function is equal to 10. As shown in the figure, OVE and KBU grow linearly with the degree of nesting in queries. Moreover, OVE takes advantage of the uniform conditional probability distribution over states in our tests and therefore is even faster. Both approximate methods (ARea and CoSa) grow exponentially with the degree of nesting, however for the same number of samples Cosa is much faster (compare ARea 10 with CoSa 10). The figure shows that even when we increase the number of samples to 50, CoSa 50 returns the answer faster than ARea 10. Note that in typical real-world situations the degree of nesting in queries is small (*e.g.,* less than 4; in poker a player at most cares about what the opponent knows about what the player knows).

In Holdem the number of states is small, so CoSa does not have any advantage over OVE. OVE is slow when the size of the state space is large, since its running time is linear in the size of the state space. In those cases CoSa returns the approximate answer much faster. Consequently, CoSa should be used only for domains with large state spaces (*e.g.*, five card poker with $\binom{52}{5}.\binom{47}{5}$ states). In small domains, however, our exact method returns the answer reasonably fast.

5 Conclusions & Future Work

We provided a factored representation for probabilistic modal structures. We also introduced exact and approximate reasoning methods for answering queries on such models. Our model is more compact than previous representations enabling larger-scale applications. Further, we show theoretically and experimentally that our methods are faster than their counterparts in previous representations.

Investigating the belief update in our language is one of our immediate future-work directions. There, it is open how to update a model with observation of opponent actions. Also, so far we have assumed that the GKM is available to us. In future works we also aim to find methods that learn a GKM.

6 Acknowledgments

We wish to thank Chandra Chekuri and Reza Zamani Nasab for valuable comments and Omid Fatemieh for editorial assistance. We also wish to acknowledge support from Army W9132T-06-P-0068 (CERL Champaign), Army DACA42-01-D-0004-0014 (CERL Champaign), and NSF IIS 05-46663 (US National Science Foundation).

References

Aumann, R. J., and Heifetz, A. 2001. Incomplete information. In *Handbook of Game Theory with Economic Applications*.

Blei, D.; Ng, A.; and Jordan, M. 2003. Latent dirichlet allocation.

Fagin, R., and Halpern, J. Y. 1988. Reasoning about knowledge and probability. In *TARK*.

Fagin, R.; Halpern, J.; Moses, Y.; and Vardi, M. 1995. *Reasoning about knowledge*. MIT Press.

Feller, W. 1968. *An Introduction to Probability Theory and Its Applications, Volume 1*. Wiley, 3rd edition.

Fitting, M. 1993. Basic modal logic. In *Handbook of Logic in Artificial Intelligence and Logic Programming, Volume 1: Logical Foundations*.

Heifetz, A., and Mongin, P. 1998. The modal logic of probability. In *TARK*.

Henrion, M. 1986. Propagating uncertainty in bayesian networks by probabilistic logic sampling. In *UAI*, 149–164.

Herzig, A. 2003. Modal probability, belief, and actions. *Fundam. Inf.* 57(2-4):323–344.

Jordan, M. I.; Ghahramani, Z.; Jaakkola, T.; and Saul, L. K. 1999. An introduction to variational methods for graphical models. *Machine Learning* 37(2):183–233.

Koller, D., and Pfeffer, A. 1995. Generating and solving imperfect information games. In *IJCAI*.

Milch, B., and Koller, D. 2000. Probabilistic models for agents' beliefs and decisions.

Nie, J.-Y. 1992. Towards a probabilistic modal logic for semantic-based information retrieval. In *SIGIR*. ACM Press.

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

Shirazi, A., and Amir, E. 2007. Probabilistic modal logic. In *Proc. AAAI '07*. AAAI Press.

Yedidia, J.; Freeman, W.; and Weiss, Y. 2004. Constructing free energy approximations and generalized belief propagation algorithms.

Zhang, N. L., and Poole, D. 1994. A simple approach to bayesian network computations. In *Proc. of the Tenth Biennial Canadian AI Conference*.