Pergamon

Comput. Biol. Med. Vol. 27, No. 5, pp. 453–476, 1997

Printed in Great Britain
0010-4825/97 $17.00+0.00

PII: S0010-4825(97)00015-2

EFFICIENT TEMPORAL PROBABILISTIC REASONING VIA CONTEXT-SENSITIVE MODEL CONSTRUCTION

LIEM NGO[1], PETER HADDAWY[1], ROBERT A. KRIEGER[2] and JAMES HELWIG[1]

[1] Department of Electrical Engineering and Computer Science, University of Wisconsin-Milwaukee, Milwaukee, WI 53201, U.S.A.
[2] Merge Technologies, Milwaukee, WI 53214, U.S.A.

(*Received* 20 *February* 1997)

Abstract—We present a language for representing context-sensitive temporal probabilistic knowledge. Context constraints allow inference to be focused on only the relevant portions of the probabilistic knowledge. We provide a declarative semantics for our language. We present a sound and complete algorithm for computing posterior probabilities of temporal queries, as well as an efficient implementation of the algorithm. Throughout we illustrate the approach with the problem of reasoning about the effects of medications and interventions on the state of a patient in cardiac arrest. We empirically evaluate the efficiency of our system by comparing its inference times on problems in this domain with those of standard Bayesian network representations of the problems. © 1997 Elsevier Science Ltd.

Temporal probability models Bayesian networks Logic programming Prognostic evaluation

1. INTRODUCTION

For accurate medical diagnosis and prediction, it is often necessary to model a patient's condition over time. Because there is great uncertainty in clinical medicine, a system for diagnostic or prognostic evaluation must be able to represent and reason with uncertainty. Bayesian networks are currently the most powerful and popular method for representing and reasoning with probabilistic information. A Bayesian network is a directed acyclic graph in which the nodes represent random variables and the links represent direct influences. The influences are quantified with conditional probabilities in the form of a link matrix associated with each node. A link matrix specifies the probabilities of all possible values of a node given all possible combinations of values of its parents. Researchers have recently applied Bayesian networks to the modeling of temporal processes [1,2]. This is typically done by representing time discretely and creating an instance of each time-varying random variable for each point in time.

Although Bayesian networks provide a relatively efficient method for representing and reasoning with probabilistic information, the process of computing posterior probabilities (inference) in Bayesian networks remains NP-hard [3]. This complexity becomes particularly problematic in large models such as those that arise in modeling temporal processes. We can greatly reduce the size of the network models if we can identify some deterministic information and use it as the context to index the probabilistic information. For example, in using Bayesian networks to determine the likely outcomes of a plan, actions are typically represented as nodes in the network [[4], ch7], [5,6]. For problems in which only one action can be performed at a time, the network contains a single action node at each time point, with states corresponding to the available actions. For problems in which actions can be performed concurrently, the network may contain multiple action nodes at each time point. Representing plans in this way often results in networks with large numbers of nodes and large link matrices. The reason is that we need two types of knowledge for each domain variable: a specification of how it is influenced by each action (causal rules), and a specification of how it behaves over time in the absence of actions that influence it (persistence rules). When evaluating a plan, the performance of one's own actions is deterministic knowledge—we know whether or not we plan to attempt an action—actions can be used as context information.

We propose representing a class of Bayesian networks with a knowledge base of probabilistic rules augmented with context constraints. A context constraint is a logical expression that determines the applicability of a probabilistic relation based on some deterministic knowledge. A context-constrained rule has the general form (Pr(consequent|antecedents) = prob)←context.

For example, we could represent the effect of cardiopulmonary resuscitation (CPR) on the heart rhythm of a patient in cardiac arrest with a set of context-constrained rules, one of which might be

$$(\Pr(\text{rhythm}(t, \text{normal\_sinus}) | \text{rhythm}(t-1, \text{asystole})) = .05) \leftarrow \text{CPR}(t-1)$$

and the persistence of the state of the heart rhythm with a set of rules, one of which might be

$$(\Pr(\text{rhythm}(t, \text{normal\_sinus}) | \text{rhythm}(t-1, \text{asystole})) = .00) \leftarrow \neg\text{CPR}(t-1).$$

Each of these rules has a link matrix half the size of that for a rule representing both the action effect and persistence, and only one of these rules is applicable at any time point.

This paper makes four main contributions. First, we define a logical language for representing classes of temporal probability models that permits the probabilistic information to be qualified by context information when available. Second, we provide a declarative semantics for the language. Third, we present a sound and complete query answering procedure that uses context information to construct small Bayesian networks to compute the posterior probabilities of temporal queries. Finally, we present an implementation of the algorithm, the BNG system, that uses additional pruning techniques to reduce the sizes of the constructed networks.

The rest of this paper is organized as follows. Section 2 provides an overview of the medical problem of evaluating interventions to cardiac arrest, which will be used as a running example throughout the paper. Section 3 presents an overview of the BNG system, shows its use in modeling the cardiac arrest domain, and motivates the need for a formal framework. Then in sections 4 and 5 we present the representation language and associated semantics for our context-sensitive temporal probabilistic knowledge bases. In section 6 we present the procedure for answering probabilistic queries and use the formal semantics to prove the procedure both sound and complete. In section 7 we discuss the implementation of the procedure in BNG and in section 8 we present empirical results evaluating the efficiency of BNG. Section 9 provides a discussion of related work and section 10 presents conclusions and directions for future research.

2. CARDIAC ARREST

We illustrate the capabilities of context-sensitive temporal probability model construction by modeling the effects of medications and other interventions on the condition of a patient in cardiac arrest. The goal of treatment is to maintain life and prevent anoxic injury to the brain. Fewer than 10% of cardiopulmonary resuscitation attempts result in survival without brain damage [7].

The observable variable is the electrocardiogram or rhythm strip. While not including all possible rhythms, we consider the range of rhythms most commonly presented: normal sinus rhythm (nsr), ventricular fibrillation (vf), ventricular tachycardia (vt), atrial fibrillation (af), supraventricular tachycardia (svt), bradycardia (b), and asystole (a).

While patient survival is of primary importance, cerebral damage must be taken into account and can be viewed as part of the cost in a resuscitation attempt. The length of time a patient has been without cerebral blood flow (cbf) determines the period of anoxia (poa). If the patient has ineffective circulation for more than 5 min, there is a likelihood of sustaining cerebral damage (cd). This damage is persistent and its severity increases as the period of anoxia increases. We take poa to have possible values {none, 1 min, 2 min, 3 min, 4 min, 5 min, sustained} and cd to have the range of values {none, mild, moderate, severe}, while cbf is either present or absent.

Medical personnel treat a patient experiencing a cardiac arrest with a variety of interventions and medications. We consider the two most common medical interventions: cardiopulmonary resuscitation (CPR) and defibrillation (DFIB). A number of medications help control the heart rhythm and rate, improve cardiac output and increase blood pressure. Many effective drugs are currently available, of which we chose to model the three most commonly used. Lidocaine (LIDO) is an anti-arrhythmic drug that helps restore a regular rhythm, it is usually used for ventricular tachycardia, ventricular fibrillation, or to prevent ventricular fibrillation. Atropine (ATRO) increases the heart rate during bradycardia or asystole. Epinephrine (EPI) overcomes heart block and helps restore cardiac function. We have simplified our model by assuming a standard bolus size and administration rate. So each intervention and medication is simply either present or absent.

We make the following modeling assumptions. Time is modeled discretely and the time step is taken to be 1 min. Heart rhythm is determined by the previous rhythm, as well as medication and intervention. Cerebral blood flow is determined by the current heart rhythm. The period of anoxia is determined by the previous period of anoxia and the previous cerebral blood flow. For example, if the previous period of anoxia was 3 min and there was no cerebral blood flow present, the current period of anoxia becomes 4 min. The extent of cerebral damage is determined by the previous cerebral damage and the current period of anoxia. For example, if the previous cerebral damage was mild and the period of anoxia is sustained, then the current cerebral damage becomes moderate.

3. THE BNG SYSTEM

The BNG (Bayesian Network Generator) system is capable of constructing temporal Bayesian networks from a knowledge base of context-constrained rules. BNG works roughly as follows. Given a knowledge base of rules, a set of evidence, a set of context information, and a query; BNG constructs a network to compute the probability of the query given the evidence in the specified context. The required posterior probability is then competed using any of the many available Bayesian network inference algorithms. After generation and before evaluation, the network is pruned to eliminate irrelevant nodes. We illustrate the functioning of BNG by showing how it can be used to model the cardiac arrest domain.

3.1. *Knowledge base*

Consider a knowledge base of rules for the cardiac arrest domain, depicted graphically in Fig. 1. The knowledge base contains rules for determining the probabilities of rhythm,

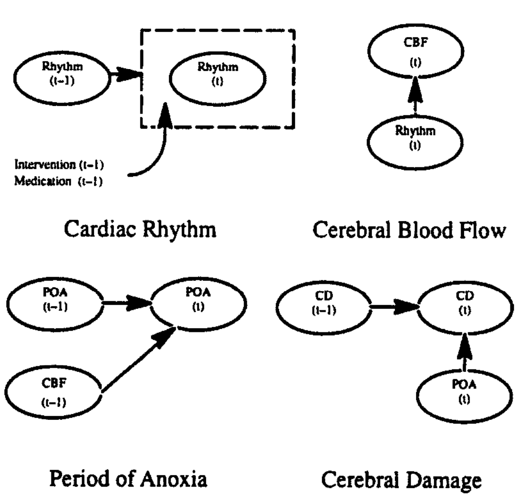

Fig. 1. Diagrammatic representation of the rules for the cardiac arrest domain. A context constraint is represented with an arrow entering a dashed box.

cbf, poa and cd at any point in time. Context constraints are used in representing the effects of medications and interventions on the heart rhythm. We have a set of rules specifying the probability of the rhythm at a time t given each possible combination of medication and intervention at time $t-1$ and the rhythm at time $t-1$. We also have a rule specifying the probability of the rhythm at t given only knowledge of the rhythm at $t-1$. This type of rule is called a persistence rule. The medication administered and the intervention applied are specified as context constraints on the rules. The BNG rule specifying the effect of epinephrine on rhythm is shown below. Each row represents the probabilities of a particular rhythm, conditioned on the previous rhythms labeling the columns. Thus the entries in each row sum to one.

Context: Intervention($t-1$, CPR),

Medication($t-1$, EP_1)

Ante: Rhythm($t-1$)

Conse: Rhythm(t)

Matrix:

;*NSR*	*VF*	*VT*	*AF*	*SVT*	*B*	*A*	
.03	.26	.10	.05	.55	.00	.01	;*normal*(*NSR*)
.12	.60	.10	.00	.08	.00	.10	;*v* − *fib*(*VF*)
.01	.80	.14	.00	.00	.00	.05	;*v* − *tach*(*VT*)
.01	.30	.05	.50	.05	.00	.09	;*a* − *fib*(*AF*)
.01	.10	.09	.35	.30	.00	.05	;*SVT*
.08	.15	.02	.05	.60	.05	.05	;*brady*(*B*)
.10	.15	.05	.00	.05	.05	.60	;*asystole*(*A*)

If actions were not represented as context constraints, the link matrix for rhythm would contain 588 entries. By representing actions as context constraints, we obtain 12 rules each with a link matrix of 49 entries. Although using context constraint does not reduce the amount of probabilistic information we must assess, it drastically reduces the size of the link matrices in the generated networks, which is the main determining factor in the inference complexity.

The knowledge base contains three other types of rules. The first states that the value of cbf at a time is determined by the value of rhythm at that time. The second states that the value of poa is determined by the previous values of poa and cbf. The last states that the current value of cd is determined by the value of poa at that time and the previous value of cd.

One of the more difficult aspects of developing a probabilistic model such as this one is securing complete and accurate knowledge. All domain knowledge was elicited from an expert ER physician. We started by eliciting the qualitative causal structure of the domain. We then used Bahill's techniques [8] for eliciting the needed conditional probability values.

3.2. *Modeling examples*

The procedure used when modeling the effects of medications and other interventions involves three steps. First, the evidence is set to specify the condition of the patient at the present time (time 0). The conditions specified are cardiac rhythm, known period of anoxia and previous extent of cerebral damage. Second, the actions (medications and/or other interventions) are specified for individual time segments[1]. Third, the queries are set for particular variables at specific times, usually examining the rhythm and cerebral damage.

[1] In practice a physician would decide on later actions after observing the effects of previous actions. But in order to choose the optimal next action, one must evaluate each choice in the context of the optimal sequence of future actions.

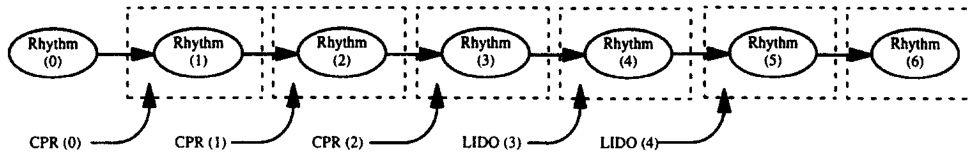

Fig. 2. BNG generated temporal Bayesian network for querying heart rhythm at time 6.

These specifications are used by BNG to construct a Bayesian network that can then be fed to an inference system. The queries are answered by providing the posterior probabilities given the evidence and actions.

Our first example simulates a response to a massive myocardial infraction. The following evidence is presented as the patient's state at time 0: the rhythm is ventricular fibrillation, there has been 2 min of anoxia and there exists no cerebral damage. The actions are represented as context information. CPR is administered until the administrstion of Lidocane. Lidocane is administered at times 3 and 4[2]. Notice that because of the negation-as-failure assumption, we need not specify what medications and interventions are not being performed at each time. We query the cardiac rhythm at time 6. Given this inference problem, BNG generates the network shown in Fig. 2. The computed posterior probabilities are Pr(nsr)=0.43, Pr(vf)=0.06, Pr(vt)=0.01, Pr(af)=0.05, Pr(svt)=0.06, Pr(b)=0.00, Pr(a)=0.39.

Next we evaluate an alternative course of treatment for the same patient. Rather than ceasing CPR at time 3, we continue while we initially administer lidocane. BNG generates a network with the same topology as that in Fig. 2 and computes the following posterior probabilities: Pr(nsr)=0.52, Pr(vf)=0.08, Pr(vt)=0.01, Pr(af)=0.05, Pr(svt)=0.05, Pr(b)=0.00, Pr(a) 0.29. Continuing the administration of CPR when introducing lidocane is determined to have a higher probability of resulting in normal sinus rhythm than immediately ceasing CPR.

The next example models a cardiac arrest due to drowning. The initial rhythm is asystole, the period of anoxia is known to be 5 min and there is no prior cerebral damage. Treatment consists of atropine administration at times 2 and 3, and continued CPR from time 0 to 4 The network generated in response to a query of cerebral damage at time 6 is shown in Fig. 3. The computed posterior probabilities of cerebral damage are Pr(none)=0.80, Pr(mild)=0.19, Pr(moderate)=0.01.

BNG has the ability to answer temporal interval queries, i.e. to determine the probability that a variable is in a particular state for at least one time point or for all time points within

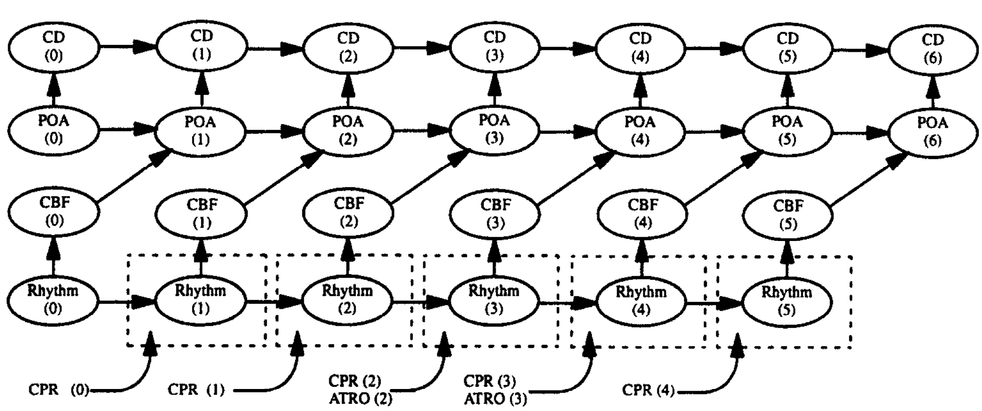

Fig. 3. BNG generated temporal Bayesian network for querying cerebral damage at time 6.

[2] While all the examples of plans that we present are open-loop plans, we can also represent closed-loop plans in which the action taken at some place is dependent on the information obtained by a previous action. this is done by modeling a branch point by an action in which the effects are determined by the state of a variable representing the observation resulting from the previous action.

an interval. This example models a patient initially presenting with asystole and treated with atropine and CPR. A query as to whether or not a cerebral blood flow state of present ever exists between times 1 and 5 reveals a probability of 0.51 that it will exist at some point and 0.49 that it will never exist during the interval.

The network generated for the first problem above (Fig. 2) is much smaller than that generated for the second problem (Fig. 3). Furthermore, the presence of loops in the network for the second problem makes its evaluation much more time consuming than the evaluation of the simple linear chain network for the first problem. BNG generates only that portion of the network which is relevant to computing the given probabilistic query. In this domain, BNG generates networks ranging in size from 7 to 28 nodes, depending on the particular query at time 6. If one wanted to use a stored, general network to answer all such queries, we would need a node for each random variable at every time point, in addition to nodes representing the choices of medication and other intervention. Such a network, containing 42 nodes, is shown in Fig. 4.

The difference in the sizes of the networks in Figs 3 and 4 is due strictly to the use of context, since no pruning of the network in Fig. 3 takes place. As can be seen from this example, the use of context constraints reduces the size of constructed models, and thus computation time, by selecting a specific probabilistic relation from a set of possible relations. Thus probabilistic information irrelevant to a particular problem can be ignored. In the above examples, context was used to determine the probabilistic relation between current and previous heart rhythm but did not affect the network structure. Context can affect structure when the set of influences on a variable changes with the context. The difference in the sizes of the networks in Figs 2 and 4 is due to both context and pruning of irrelevant nodes.

In general, the use of context provides significant computational savings when the number of influences on some random variables is large and can be selected by context. Pruning provides significant savings when a model covers a large and diverse domain, so that only a small portion of it is relevant to a given inference problem. For example, a model might encompass tens of diseases and hundreds of symptoms. If we are interested in determining the probability of one disease given a few symptoms, only a small portion of the model may be relevant. The above examples are intended to provide a qualitative sense for the effects of context and pruning. An empirical evaluation is provided in section 8.

While the capabilities provided by a system like BNG are appealing, for it to be an acceptable computational tool we must show that it performs computations correctly. In order to do this, several formal issues need to be addressed. First, since the user of such a system encodes his information in the language of the knowledge base, that language must be provided with a precise semantics. This semantics is also necessary in order to prove the correctness of model construction system. In order to be correct, the probabilities inferred

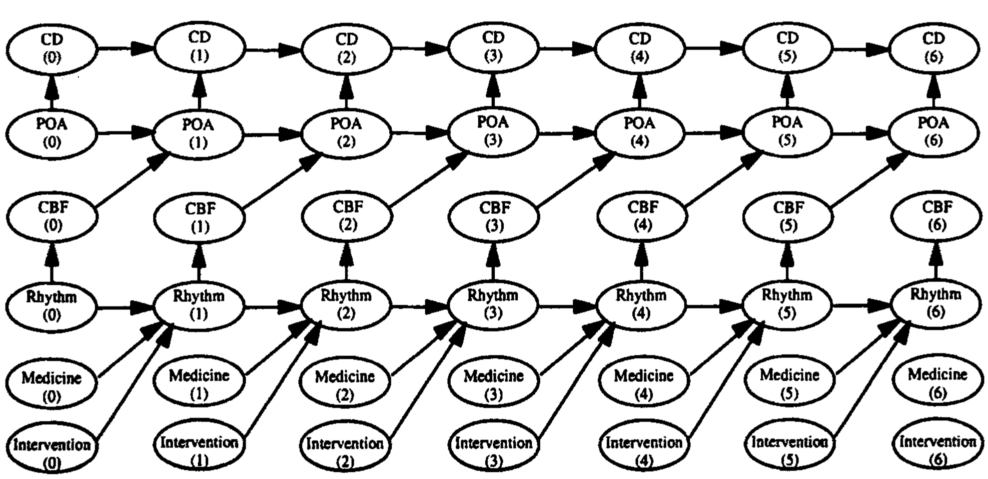

Fig. 4. General temporal Bayesian network for all queries up to time 6.

using the constructed networks must be identical to those entailed by the information in the knowledge base. We address these issues by presenting a formal syntax and semantics for the knowledge base representation language; formally specifying the network construction algorithm: and proving the algorithm correct.

4. THE REPRESENTATION LANGUAGE

There are two disjoint types of predicates: context and probabilistic predicates. Some predicates are timed predicates. A timed predicate always has one attribute indicating the time the associated event or relationship denoted by the predicate occurs. We model only discrete time points and throughout the paper we represent the set of time points by the set of integers. If t is a time point, $t+5$ denotes the fifth time point after t. If A is a ground timed atom and the time attribute is t, we say A happens at time t.

Context predicates (c-predicates) have value true or false and are deterministic. They are used to describe the context the agent is in and to eliminate unnecessary or inappropriate probabilistic information from consideration by the agent. An atom formed from a context predicate is called a context atom (c-atom). A context literal (c-literal) is either a c-atom or the negation of a c-atom. A context base is a normal logic program [9], which is a set of universally quantified sentences of the form $C_0 \leftarrow L_1, L_2, ..., L_n$, where n is a nonnegative integer, $\leftarrow$ stands for implication, comma for logical conjunction, C_0 for a c-atom, and L_i, $1 \le i \le n$, for c-literals.

Each probabilistic predicate (p-predicate) represents a class of similar random variables. P-predicates appear in probabilistic sentences and are the focus of probabilistic inference processes. An atom formed from a probabilistic predicate is called a probabilistic atom (p-atom). Queries and evidence are expressed as p-atoms. In the probability models we consider, each random variable can assume a value from a finite set and in each possible realization of the world, that variable can have one and only one value. We capture this property by requiring that each p-predicate has at least one attribute. The last attribute of a p-predicate represents the value of the corresponding random variable. For example, the variable rhythm of a person can have value nsr, vf, vt, af, svt, b, or a and can be represented by a two-position predicate, the first position indicating which person and the second indicating that person's type of cardiac rhythm. Associated with each p-predicate p must be a statement of the form $VAL(p)=\{v_1, ..., v_n\}$, where $v_1, ..., v_n$ are constants. Let $A=p(t_1, ..., t_{m-1}, t_m)$ be a p-atom, we use $obj(A)$ to designate the tuple $(p, t_1, ..., t_{m-1})$ and $VAL(A)$ to designate t_m. So if A is a ground p-atom then $obj(A)$ represents a concrete random variable or object in the model and $VAL(A)$ is its value. We also define $Ext(A)$, the extension of A, to be the set $\{p(t_1, ..., t_{m-1}, v_i) | 1 \le i \le n\}$. If A is an atom of predicate p then $VAL(A)$ means $VAL(p)$. We assume that p-predicates are typed, so that each attribute of a p-predicate is assigned values in some well-defined domain. We denote the set of all such predicate declarations in a knowledge base by PD.

Let A be a (p- or c-) atom. We define $ground(A)$ to be the set of all ground instances of A, i.e. the atoms obtained by substituting the variables in A by constants. If E is a set of atoms then $ground(E)$ is defined as $\cup_{A \in E} ground(A)$. We assume *mutual exclusivity* and *exhaustivity*: for each ground p-atom A and in each possible world, one and only one element in $Ext(A)$ is true. A set of ground p-atoms $\{A_i | 1 \le i \le n\}$ is called **coherent** if $\forall i,j$ $(1 \le i,j \le n)$: $obj(A_i)=obj(A_j) \rightarrow val(A_i)=val(A_j)$.

A p-sentence has the form $(Pr(A_0 | A_1, ..., A_n)=\alpha) \leftarrow L_1, ..., L_m$ where $n \ge 0$, $m \ge 0$, $0 \le \alpha \le 1$, A_i are p-atoms, and L_j are context literals. The sentence can have free variables and each free variable is universally quantified over the entire scope of the sentence. The above sentence is called context-free if $m=0$. If S is the above p-sentence, we define $context(S)$ to be the conjunction $L_1, ..., L_m$, $prob(S)$ to be the probabilistic statement $Pr(A_0 | A_1, ..., A_n)=\alpha$, $ante(S)$ to be the conjunction $A_1 \wedge ... \wedge A_n$, and $cons(S)$ to be A_0. Sometimes, we use $ante(S)$ as the set of conjuncts and call $cons(S)$ the consequent of S. A probabilistic base (PB) of a knowledge base is a finite set of p-sentences.

A PB will typically not be a complete specification of a probability distribution over the random variables represented by the p-atoms. One type of information which may be lacking is the specification of the probability of a variable given combinations of values of two or more variables which influence it. For real-world applications, this type of information can be difficult to obtain. For example, for two diseases D_1 and D_2 and a symptom S we may know $Pr(S|D_1)$ and $Pr(S|D_2)$ but not $Pr(S|D_1, D_2)$. Combining rules such as noisy-OR and noisy-AND [10] are commonly used to construct such combined influences. For example, the basic assumption of noisy-OR is the two causes D_1 and D_2 of S are independent. Let q_i, $i=1, 2$, be the probability that D_i fails to cause the symptom S. Then, the probability that D_1 and D_2 both fail to cause S is $q_1 \times q_2$. Hence, we need only two numbers q_1 and q_2 to specify the probability that S occurs under any possible states of D_1 and D_2. Noisy-OR has been shown useful for knowledge acquisition and efficient inference in medical and other applications. Several extensions have been proposed for more complex situations. e.g. generalized noisy-OR [11].

To incorporate the most general combining rules available, we define a **combining rule** as any algorithm that

(1) takes as input a set of ground p-sentences that have the same consequent $\{Pr(A_0|A_{i1}, \ldots, A_{in_i}=\alpha_i|1\leq i\leq m\}$ where m is a non-negative integer or infinity, and

(2) produces as output a set of ground p-sentences $OS=\{Pr\,(A_0|B_{i1}, \ldots, B_{ik_i}))=\beta_i|1\leq i\leq h$, h and k_i are some non-negative finite integers$\}$ such that:

(2.1) the antecedent of each p-sentence in OS is a coherent set,

(2.2) if S_1 and S_2 are two different p-sentences in OS then ante$(S_1)\cup ante(S_2)$ is not coherent, and

(2.3) if B appears in the antecedent of a p-sentence in OS then

$$obj(B)\in \bigcup_{i=1}^{m} obj(A_{i_1}),\ldots,obj(A_{\epsilon_i})$$

and

(3) the output set is empty if and only if the input set is empty.

Condition (2.2) enforces the constraint that at most one antecedent is true in any possible world. As a result, in each possible world there is at most one statement about the direct influences on the consequent A_0. Condition (2.3) is natural: we do not want to introduce new random variables.

We assume that for each p-predicate p, there exists a corresponding combining rule in CR, the combining rules component of the KB. Each combining rule is applied only once for any given predicate. The combining rules are specified by the designer of a knowledge base and they reflect his conception of the interaction between direct influences of the same variable.

A knowledge base (KB) consists of predicate descriptions, a probabilistic base, a context base, and a set of combining rules. We write $KB=\langle$PD, PB, CB, CR$\rangle$. Figure 5 shows a possible knowledge base for representing the cardiac arrest example introduced in the previous section.

We have the following p-predicates: *rhythm*, *cbf*, *poa*, and *cd*. The statement *rhythm(X: Person, V)* says that the first argument is a value in the domain *Person*, which is the set of all persons and *V* can take one value in the set {*nsr*, *vf*, *vt*, *af*, *svt*, *b*, *a*}. So *rhythm(john, t, nsr)* means the random variable *cardiac rhythm of John at time t*, indicated in the language by *obj(rhythm(john, t, nsr))*, is *normal sinus rhythm*, indicated in the language by *val(rhythm(john, t, nsr))=nsr*. Similar interpretations apply to other statements in *PD*.

PB contains the p-sentences. Due to space limitations, we show only a sample of the sentences. We have sentences specifying prior probabilities at time 0, assuming a patient in a state of cardiac arrest. We also have four sets of sentences, each specifying how the value of one of the random variables *rhythm*, *cbf*, *poa*, *cd* is inflenced by the values of the other random variables. The four sets of sentences are shown diagrammatically in Fig. 1.

CB defines the relationships among context information. The clause *NO_INTER(X, t)*$\leftarrow\neg$*DFIB(X, t)*,$\neg$*CPR(X, t)* allows us to imply that no intervention is being employed at a particular time if no intervention is specified. The negation in the antecedent encodes a non-monotonic deduction by negation as failure.

Finally, generalized noisy-OR is used as the combining rule.

5. SEMANTICS

It can be difficult for a user to guarantee global consistency of a large probabilistic knowledge base, especially when we allow context-dependent specifications. We define our semantics so that we need consider only that portion of a knowledge base relevant to a given problem. Thus if part of the knowledge base is inconsistent, this will not affect reasoning in all contexts. We define the semantics relative to an inference session, characterized by a set of evidence and a set of context, information.

Definition 1 *A set of context information C is any set of c-atoms. A set of evidence E is*

$$PD = \{rhythm(X : Person, t : time, V), VAL(rhythm) = \{nsr, vf, vt, af, svt, b, a\};$$
$$cbf(X : Person, t : time, V), VAL(cbf) = \{present, absent\};$$
$$poa(X : Person, t : time, V), VAL(poa) = \{none, 1min, 2min, 3min, 4min, 5min, sustained\};$$
$$cd(X : Person, t : time, V), VAL(cd) = \{none, mild, moderate, severe\}\};$$

$$PB = \{Pr(rhythm(X,0,nsr)) = 0.001, Pr(rhythm(X,0,vf)) = 0.74, \cdots$$
$$Pr(poa(X,0,none)) = 0.99, Pr(poa(X,0,1min)) = 0.005, \cdots$$
$$Pr(cd(X,0,none)) = 0.99, Pr(cd(X,0,mild)) = 0.005, \cdots$$
$$Pr(cbf(X,0,present)) = 0.99, Pr(cbf(X,0,absent)) = 0.01$$
$$Pr(rhythm(X,t,nsr)|rhythm(X,t-1,nsr)) = .05 \leftarrow NO\_INTER(X,t-1), EPI(X,t-1)$$
$$Pr(rhythm(X,t,nsr)|rhythm(X,t-1,vf)) = .01 \leftarrow NO\_INTER(X,t-1), EPI(X,t-1)$$
$$Pr(rhythm(X,t,nsr)|rhythm(X,t-1,vt)) = .01 \leftarrow NO\_INTER(X,t-1), EPI(X,t-1)$$
$$\cdots$$
$$Pr(rhythm(X,t,vf)|rhythm(X,t-1,af)) = .35 \leftarrow DFIB(X,t-1), ATRO(X,t-1)$$
$$\cdots$$
$$Pr(rhythm(X,t,a)|rhythm(X,t-1,vf)) = .15 \leftarrow NO\_INTER(X,t-1), NO\_MED(X,t-1)$$
$$\cdots$$
$$Pr(cbf(X,t,present)|rhythm(X,t,nsr)) = 1.0$$
$$Pr(cbf(X,t,absent)|rhythm(X,t,nsr)) = .0$$
$$Pr(cbf(X,t,present)|rhythm(X,t,vf)) = .0$$
$$Pr(cbf(X,t,absent)|rhythm(X,t,vf)) = 1.0$$
$$\cdots$$
$$Pr(poa(X,t,3min)|cbf(X,t-1,present), poa(X,t,2min)) = .0$$
$$Pr(poa(X,t,3min)|cbf(X,t-1,absent), poa(X,t,1min)) = .0$$
$$Pr(poa(X,t,3min)|cbf(X,t-1,absent), poa(X,t,2min)) = 1.0$$
$$\cdots$$
$$Pr(cd(X,t,mild)|poa(X,t,3min), cd(X,t-1,mild)) = .0$$
$$Pr(cd(X,t,mild)|poa(X,t,sustained), cd(X,t-1,severe)) = .0$$
$$Pr(cd(X,t,mild)|poa(X,t,sustained), cd(X,t-1,mild)) = .98$$
$$Pr(cd(X,t,moderate)|poa(X,t,sustained), cd(X,t-1,mild)) = .02$$
$$\cdots\}$$

$$CB = \{NO\_INTER(X,t) \leftarrow \neg DFIB(X,t), \neg CPR(X,t)$$
$$NO\_MED(X,t) \leftarrow \neg LIDO(X,t), \neg ATRO(X,t), \neg EPI(X,t)\}$$

$$CR = \{Generalized - Noisy - OR\}$$

Fig. 5. Knowledge base for the cardiac arrest domain.

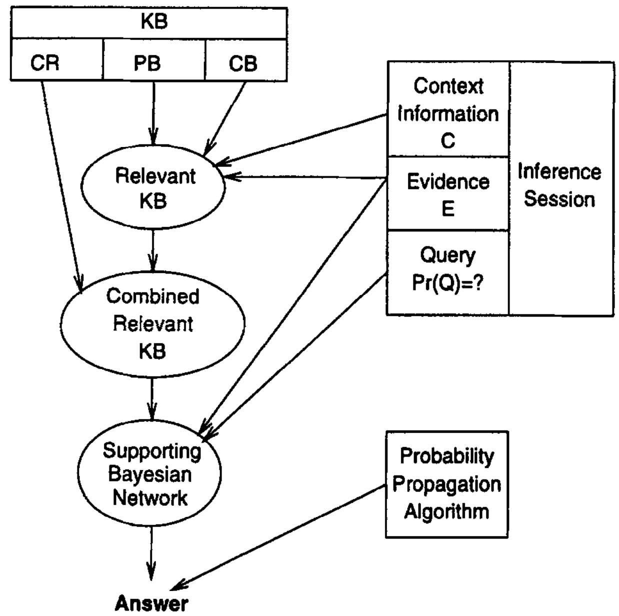

Fig. 6. The interpretation of a KB in a specific context.

simply a set of p-atoms. We always assume that ground(E) is coherent.

An inference session will be concerned with determining the posterior probability of some p-atoms Q given E, within context C. Figure 6 shows how the semantics of a KB is determined. We assume that a KB is given. In each inference session, the context C is used in conjunction with the context base *CB* to derive from *PB* a set of p-sentences which are *relevant*. The combining rules are used to combine each group of p-sentences which have the same random variable in their consequents. The result set of p-sentences is called the Combined Relevant Probabilistic Base (CRPB). In order to evaluate the posterior probability of Q given E, we construct a Bayesian network from the CRPB and apply a conventional propagation algorithm.

5.1. *The relevant knowledge base*

In a particular inference session, only a portion of the *KB* is relevant. The relevant part of *KB* is determined by the given context information and the set of evidence. The first thing we need to determine is the set of relevant random variables. This set is represented by the set of relevant p-atoms. The set of relevant p-atoms (RAS) is the set of evidence, the set of atoms whose marginal probability is directly stated, and the set of atoms influenced by these two sets, as indexed by the context information. In constructing the set of relevant p-atoms, we consider only the qualitative dependencies described by probabilistic sentences. If $(Pr(A_0|A_1, ..., A_n)=\alpha)\leftarrow L_1, ..., L_m$ is a ground instance of a sentence in PB and $L_1, ..., L_m$, can be deduced from the logic program [9] C ∪ CB, then that sentence confirms the fact that A_0 is directly influenced by $A_1, ..., A_n$. If, in addition, $A_1, ..., A_n$ are relevant atoms then it is natural to consider A_0 as relevant.

Informally, the Relevant Probabilistic Base (RPB) is determined in the following way. If $(Pr(A_0|A_1, ..., A_n)=\alpha\leftarrow L_1, ..., L_n$ is a ground instance of a sentence in PB, $L_1, ..., L_m$ can be deduced from the logic program C ∪ CB and $A_0, ..., A_n$ are relevant then $Pr(A_0|A_1, ..., A_n)=\alpha$ is a relevant p-sentence. RPB is the set of all such sentences. Formal definitions of RAS and RPB are presented in the Appendix.

The RPB contains the basic relationships between p-atoms in RAS. In the case of multiple influences represented by multiple sentences, we need combining rules to construct the combined probabilistic influence.

Definition 2 *Given a set of evidence E, a set of context information C, and a KB:* **the set of combined relevant p-sentences (CRPB)** *is constructed by applying the corresponding* *combining rule to each maximal set of sentences* $\{S_1, ..., S_n\}$ *in RPB for which the elements have the same consequent.*

Example 3 *Consider our example KB and suppose we have the evidence {rhythm(john, 0, vf), poa(john, 0 none), cd(john, 0, none), cbf(john, 0, present)} and the context {EPI(john, 0), EPI(john, 2), DFIB(john, 2)}. The RAS and RPB are shown in Fig. 7.*

We define a syntactic property of CRPB which is necessary in constructing Bayesian networks from the *KB*.

Definition 4 *A CRPB is* **completely quantified** *if*

(1) for all ground atoms A_0 *in RAS, there exists at least one sentence in CRPB with* A_0 *in the consequent; and*

(2) for all ground sentences S of the form $Pr(A_0|\vec{A}_1, ..., \vec{A}_n) = \alpha$ *in CRPB we have the following property: For all* $i = 0, ..., n$, *if* A_i *is* $p(\vec{t}, v)$, *where* $\vec{t}$ *is a list of ground terms and* v' *is a value in VAL(p) such that* $v \neq v'$, *then there exists another ground sentence* S' *in CRPB such that* S' *can be constructed from S by replacing* A_i *by* $p(\vec{t}, v')$ *and* α *by some* α'.

If we think of each ground *obj(A)*, where *A* is some p-atom, as representing a random

$$
\begin{aligned}
RAS = \{ & rhythm(john, 0, nsr), rhythm(john, 0, vf), rhythm(john, 0, vt) \cdots \\
& rhythm(john, 1, nsr), rhythm(john, 1, vf), rhythm(john, 2, vt) \cdots \\
& cbf(john, 0, present), cbf(john, 0, absent) \\
& cbf(john, 1, present), cbf(john, 1, absent) \cdots \\
& poa(john, 0, none), poa(john, 0, 1min), poa(john, 0, 2min) \cdots \\
& cd(john, 0, none), cd(john, 0, mild), cd(john, 0, moderate) \cdots \\
& \cdots \} \\
RPB = \{ & Pr(rhythm(john, 0, nsr) = 0.001, Pr(rhythm(john, 0, vf) = 0.74, \cdots \\
& Pr(poa(john, 0, none) = 0.99, Pr(poa(john, 0, 1min) = 0.005, \cdots \\
& Pr(cd(john, 0, none)) = 0.99, Pr(cd(john, 0, mild) = 0.005, \cdots \\
& Pr(cbf(john, 0, present) = 0.99, Pr(cbf(john, 0, absent)) = 0.01, \cdots \\
& \cdots \\
& Pr(rhythm(john, 1, nsr)|rhythm(john, 0, nsr)) = .05 \\
& Pr(rhythm(john, 1, nsr)|rhythm(john, 0, vf)) = .01 \\
& Pr(rhythm(john, 1, nsr)|rhythm(john, 0, vt)) = .01 \\
& Pr(rhythm(john, 1, nsr)|rhythm(john, 0, af)) = .01 \\
& \cdots \\
& Pr(rhythm(john, 1, vf)|rhythm(john, 0, nsr)) = .10 \\
& \cdots \\
& Pr(rhythm(john, 2, nsr)|rhythm(john, 1, nsr)) = 1.0 \\
& Pr(rhythm(john, 2, nsr)|rhythm(john, 1, vf)) = .05 \\
& \cdots \\
& Pr(poa(john, 1, 3min)|cbf(john, 0, present), poa(john, 0, 2min)) = 0.0 \\
& Pr(poa(john, 1, 3min)|cbf(john, 0, absent), poa(john, 0, 2min)) = 1.0 \\
& \cdots \\
& Pr(cd(john, 2, mild)|poa(john, 2, 3min), cd(john, 1, mild) = .0 \\
& Pr(cd(john, 2, mild)|poa(john, 2, sustained), cd(john, 1, mild) = .98 \\
& Pr(cd(john, 2, moderate)|poa(john, 2, sustained), cd(john, 1, mild) = .02 \\
& \cdots \}
\end{aligned}
$$

Fig. 7. RAS and RPB for the cardiac arrest problem, given particular context and evidence.

variable in a Bayesian network model then the above condition implies that we can construct a link matrix for each random variable in the model. This is a generalization of constraint (C1) in [12]. We do not require the existence of link matrix for every random variable, but only for the random variables that are relevant to an inference problem.

5.2. *Probabilistic independence assumption*

Besides the probabilistic quantities given in a PB, we assume some probabilistic independence relationships specified by the structure of p-sentences. Probabilistic independence assumptions are used in all related work [12–14] as the main device to construct a probability distribution from local conditional probabilities. Unlike Poole [14], who assumes independence on the set of consistent assumable atoms, we formulate the independence assumption in our framework by using the structure of the sentences in CRPB. We find this approach more natural since the structure of the CRPB tends to reflect the causal structure of the domain and independencies are naturally thought of causally.

Definition 5 *Given a set of ground context-free probabilistic sentences, let A and B be two p-atoms. We say A is influenced by B if (1) there exists a sentence S, an atom A′ in Ext(A) and an atom B′ in Ext(B) such that A′ = cons(S) and B′ ∈ ante(S) or (2) there exists another p-atom C such that A is influenced by C and C is influenced by B.*

Assumption 1 *Given a set of evidence E, a set of context information C and a KB, we can construct CRPB. We assume that if* $Pr(A_0|A_1,\dots, A_n) = \alpha$ *is in CRPB then for all ground p-atoms B which are not in* $Ext(A_0)$ *and not influenced by* A_0*,* A_0 *and B are probabilistically independent given* $A_1, \dots, A_n$*.*

This assumption is more intuitive and probably easier to check (for knowledge base builders) than the d-separation assumption in [12]. It is well known that these two ways of stating probabilistic independence assumptions are equivalent for finite Bayesian networks [10].

Example 6 *Continuing the cardiac arrest example, rhythm(john, 1, vf) is probabilistically independent of rhythm(john, 3, vf) given rhythm(john, 2, nsr).*

5.3. *Model theory*

The RAS contains all relevant atoms for an inference session. We assume that *in such a concrete situation, the belief of an agent can be formulated in terms of possible models on RAS.*

Definition 7 *Given a set of evidence E, a set of context information C and a KB, a possible model M of the corresponding CRPB is a set of atoms in RAS such that for all A in RAS,* $Ext(A) \cap M$ *has one and only one element.*

A probability distribution on the possible models is realized by a probability density assignment to each model. Let *Pr* be a probability distribution on the possible models, we define $Pr(A_1, \dots, A_n)$, where $A_1, \dots, A_n$ are atoms in RAS, as $\Sigma\{Pr(w)|w$ is a possible model containing $A_1, \dots, A_n\}$. We take a sentence of the form $Pr(A_0|A_1, \dots, A_n) = \alpha$ as shorthand for $Pr(A_0,A_1, \dots, A_n) = \alpha \times Pr(A_1, \dots, A_n)$, so that probabilities conditioned on zero are not problematic. We say *Pr* satisfies a sentence $Pr(A_0|A_1, \dots, A_n) = \alpha$ if $Pr(A_0, A_1, \dots, A_n) = \alpha \times Pr(A_1, \dots, A_n)$ and *Pr* satisfies CRPB if it satisfies every sentence in CRPB.

Definition 8 *A probability distribution induced by the set of evidence E, the set of context information C, and KB is a probability distribution on possible models of CRPB satisfying CRPB and the independence assumption implied by CRPB.*

We define the consistency property only on the relevant part of a KB. Since the entire KB contains information about various contexts, testing for consistency of such a KB may be very difficult.

Definition 9 *A completely quantified CRPB is* **consistent** *if*

(1) there is no atom in RAS which is influenced by itself;
(2) if $Pr(A_0|A_1, \dots, A_n) = \alpha$ *is in CRPB then* $Pr(A_0|A_1, \dots, A_n) = \alpha'$ *is not in CRPB, for any* $\alpha' \neq \alpha$*; and*
(3) for all $Pr(A_0|A_1, \dots, A_n) = \alpha$ *in CRPB,* $\Sigma\{\alpha_i|Pr(A_0'|A_1, \dots, A_n) = \alpha_i \in CRPB$ *and*

$obj(A_0') = obj(A_0)\} = 1$.

Condition (1) rules out cycles and condition (3) enforces the usual probability assignment constraint.

A possible model in temporal frameworks corresponds to a world history [15]. The set of world histories is infinite. In practice, particularly for plan projection problems, we typically consider only events occurring over finite periods. We assume that for each temporal reasoning problem there exist two integers ι, τ, $\iota \leq \tau$ such that things occurring outside $[\iota, \tau]$ are not of our concern. Hence, any timed atom in *E* or *C* or any timed atom whose prior probability is given in *PB* is at a time in the interval $[\iota, \tau]$.

In the following three definitions, we redefine the concepts of RAS, RPB, CRPB, possible model and induced probability distribution with respect to a given time interval $[\iota, \tau]$.

Definition 10 *Given two integers* ι, $\tau(\iota \leq \tau)$, *a set of evidence E, a set of context information C and a KB, the* (ι, τ)*-bounded RAS, denoted by* (ι, τ)*-RAS is the set* $\{A | A \in RAS$ *and if A is timed then A happens at t,* $\iota \leq t \leq \tau\}$. *The* (ι, τ)*-RPB and* (ι, τ)*-CRPB are confined versions of RPB and CRPB, correspondingly, on* (ι, τ)*-RAS.*

Definition 11 *Given two integers* ι, $\tau(\iota \leq \tau)$, *a set of evidence E, a set of context information C and a KB, a possible* (ι, τ)*-model M of the corresponding CRPB is a set of atoms in* (ι, τ)*-RAS such that for all A in* (ι, τ)*-RAS,* $Ext(A) \cap M$ *has one and only one element.*

Definition 12 *Given two integers* ι, $\tau(\iota \leq \tau)$. *A probability distribution which is* (ι, τ)*-bound induced by the set of evidence E, the set of context information C, and KB is a probability distribution on possible* (ι, τ)*-models of CRPB satisfying* (ι, τ)*-CRPB and the independence assumption implied by* (ι, τ)*-CRPB.*

The following theorem states the conditions under which a KB specifies a well-defined probability distribution.

Theorem 13 *Given two integers* ι, $\tau(\iota \leq \tau)$, *a set of evidence E, a set of context information C, and a KB, if the* (ι, τ)*-RAS is finite and the* (ι, τ)*-CRPB is completely quantified and consistent then there exists one and only one* (ι, τ)*-bound induced probability distribution.*

5.4. *Queries*

In one inference session, we can pose queries to ask for the posterior probabilities of some random variables.

Definition 14 A **complete ground query** *wrt the set of evidence E and the set of context information C is a query of the form* $Pr(Q) = ?$, *where the last attribute of Q is a variable and it is the only variable in Q. The meaning of such a query is: find the posterior probability distribution of obj(Q). If* $VAL(Q) = \{v_1, ..., v_n\}$ *then the answer to such a query is a vector* $(\alpha_1, ..., \alpha_n)$, *where* α_i *is the posterior probability that obj(Q) receives the value* v_i.

A **complete query** *is a query of the form* $Pr(Q) = ?$, *where the last attribute of Q is a variable and the other attributes may also contain variables.*

Example 15 *We can pose the complete ground query Pr(rhythm(john, 3, V)) = ? to the example KB to ask for the posterior probability of John's cardiac rhythm at time 3.*

To determine the correctness of the answers returned by a query-answering procedure, we need the concept of the true consequence of a KB.

Definition 16 *Assume a set of evidence E, a set of context information C, a KB, and a complete ground query* $Pr(Q) = ?$, *where* $VAL(Q) = \{v_1, ..., v_n\}$ *we say* $Pr(Q) = (\alpha_1, ..., \alpha_n)$ *is a* **logical consequence** *of (E, C, KB) if for all probability distributions* Pr^* (ι, τ)*-bound induced by E, C, and KB and for all i* $(0 \leq i \leq n)$: $Pr^*(\{M | Q_i$ *and ground(E) are true in* $M\}) = \alpha_i \times Pr^*(\{M | ground(E)$ *are true in* $M\})$, *where* Q_i *is Q after replacing val(Q) by* v_i *and M is an arbitrary possible* (ι, τ)*-model.*

Example 17 *Suppose we have the query Q = rhythm(john, 3, V). There are an infinite* *number of induced probability distributions but in all of them Pr(rhythm(john, 3, nsr)) = 0.41, Pr(rhythm(john, 3, vf) = 0.09, and Pr(rhythm(john, 3, vt) = 0.04. (See the first example in Section 3.2.) So they are logical consequences of CRPB.*

In order to prove that the answers returned by our query answering procedure are correct, we need the following definition.

Definition 18 *Assume a set of evidence E, a set of context information C, a KB, and a complete ground query Pr(Q) = ? are given. Then* $Pr(Q) = (\alpha_1, ..., \alpha_n)$ *is a correct answer to the complete ground query Pr(Q) = ? if* $Pr(Q) = (\alpha_1, ..., \alpha_n)$ *is a logical consequence of* $\langle E, C, KB \rangle$.

If Pr(Q) = ? is a complete query then $Pr(Q') = (\alpha_1, ..., \alpha_n)$ *is a correct answer to the complete query Pr(Q) = ? if Q' is formed from a ground instance of Q by substituting its last attribute by a new variable name and* $Pr(Q') = (\alpha_1, ..., \alpha_n)$ *is a correct answer to the complete ground query Pr(Q') = ?.*

6. QUERY ANSWERING PROCEDURE

In this section we present an algorithm for answering a complete query. Assume that we are given (*Pr*(*Q*)=?, *E*, *C*, *KB*), where *Pr*(*Q*)=? is a complete query. We call the query answering procedure Q-procedure.

Q-procedure has the following steps: build the necessary portion of the Bayesian network, each node of which corresponds to an *obj*(*A*), where *A* is a ground p-atom in RAS: update the network using the set of evidence E: and output the updated belief of the query nodes. The main idea of the algorithm is to build a *supporting network* for each random varaible corresponding to a ground instance of an evidence atom or the query. Let *A* be a ground p-atom and consider the set of all ground p-atoms *B* such that *A* is influenced by *B* in CRPB. The **supporting network** for *obj*(*A*) is a Bayesian network consisting of *obj*(*A*) and the set of all *obj*(*B*), with the relevant influenced by relationships presented as links or sequences of links.

Constructing the network by building supporting networks is justified by the fact that atoms which do not influence either the evidence or the query are irrelevant. To build the supporting networks for the evidence, we first generate the set of all ground instances of the evidence p-atoms. Then for each ground instance, we build the supporting network using PB, the set of ground instances of the evidence which have not been explored, and the current network.

The supporting networks are constructed via cells BUILD-NET, which receives as input an atom whose supporting network needs to be explored. It updates the NET, which might have been partially built. The return value of the function is a set of substitutions such that for each substitution there exists a supporting network for the ground instance of the atom corresponding to the substitution. Details of BUILD-NET are given in the Appendix.

In Q-procedure, we only construct the supporting networks, not the entire network for CRPB. We can show this portion of the network is enough for evaluating the given query.

Definition 19 *A PB is called acyclic if there is a mapping* $P_{ord}()$ *from the set of ground instances of p-atoms into the set of natural numbers such that (1) For any ground instance* $(Pr(A_0 \leftarrow A_1, ..., A_n) = \alpha a) \leftarrow L_1, ..., L_m$ *of some clause in PB,* $P_{ord}(A_0) > P_{ord}(A_i)$, $\forall i$: $1 \leq i \leq n$. *(2) For any ground p-atom A,* $P_{ord}(A) = P_{ord}(A')$, $\forall A' \in Ext(A)$.

Theorem 20 *(Soundness and completeness.) Assume a set of evidence E, a set of context information C, and a KB are given. Also assume that a complete query Pr(Q) = ?, where Q is at a constant time t,* $\iota \leq t \leq \tau$ *is given. If (1) the domains (except time) are finite, (2)* (ι, τ)*-CRPB is completely quantified and consistent, (3) the proof procedure for* $C \cup CB$ *always stops after a finite amount of time and is sound and complete wrt any context query generated by Q-procedure, and (4) PB is acyclic; then (1) Q-procedure is sound: the returned answers are correct; (2) Q-procedure is complete: every ground instance of Q which has an answer is returned.*

A proof procedure satisfying the condition (3) of theorem 20 exists for a large class of logic programs, which has been studied extensively by researchers in logic programming

[9]. We need the following definition.

Definition 21 *A CB is called acyclic if there is a mapping $C_{ord}()$ from the set of ground instances of c-atoms into the set of natural numbers such that for any ground instance $A \leftarrow L_1, ..., L_n$ of some clause in CB, $C_{ord}(A) > C_{ord}(L_i)$, $\forall i$: $1 \leq i \leq n$, where we extend $C_{ord}()$ by $C_{ord}(\neg A) = C_{ord}(A)$, for any c-atom A.*

A KB is acyclic if both PB and CB are acyclic.

Our cardiac domain is an acyclic KB. The expressiveness of acyclic normal programs is demonstrated in [16]. We hope that acyclic PBs also have an equivalent importance To the best of our knowledge, PBs with loops are considered problematic and all PBs considered in the literature are acyclic. We need a syntactic criterion called *allowedness* [9].

Theorem 22 *Assume a set of evidence E, a set of context information C, and a KB are given. Also assume that a complete query $Pr(Q) = ?$, where Q is at a constant time t, $\iota \leq t \leq \tau$ is given. If (1) the domains (except time) are finite, (2) KB is acyclic, and (3) (ι, τ)-CRPB is completely quantified and consistent: then there, is a proof procedure for $C \cup CB$ such that Q-procedure is sound and complete wrt complete queries.*

7. IMPLEMENTATION OF BNG

The BNG system described in section 3 is an implementation of a version of Q-procedure[3]. BNG is written in CommonLisp and interfaces to both the IDEAL [17] and HUGIN [18] Bayesian network inference systems. BNG differs from Q-procedure in that it performs d-separation based pruning of the constructed network to further eliminate irrelevant nodes.

The BNG system takes as input a probabilistic base (PB), a context base (CB), a set of combining rules (CR), a set of evidence atoms (E), a set of context atoms (C), and a query atom (Q) and creates a network to compute the probability of Q given E in the context C. A rule in a BNG knowledge base has the general form:

Context: $C_1,...,C_m$

Ante: $A_1,...,A_n$

Conse: A_0

Matrix: (conditional probabilities) where the A_i are p-atoms and the C_i are c-atom literals, both of which may contain variables that are implicitly universally quantified. The link matrix contains the probabilities of all possible values of the consequent given all possible combinations of values of the antecedents. Such a rule represents a set of context-constrained probability sentences of the form

$$\forall X Pr(A_0|A_1, ..., A_n) = \alpha \leftarrow C_1, ..., C_m.$$

Both the antecedent and context can be empty but the consequent must be present.

The algorithm proceeds by first generating the network and then pruning nodes d-separated from the query. By simply backward chaining on the query and on the evidence atoms, the generation phase generates all relevant nodes and avoids generating barren nodes, which are nodes below the query that have no evidence nodes below them. Such nodes are irrelevant to the computation of $Pr(Q|E)$ [19]. During network generation we keep track of whether a node is a predecessor of an evidence node. Such nodes are called epsilon nodes. This information will be used by the pruning algorithm in determining d-separation.

The pruning phase involves traversing the generated network using a modified depth-first search originating at the query. Only those nodes reachable via an actiove path from the

[3] The code is available at http://www.cs.uwm.edu/faculty/haddawy.

```
Q-PROCEDURE
BEGIN
   { Build the network that supports the evidence }
   NET:= {};
   FOR i:=1 TO number_of_elements_in_E DO
        temp := BUILD-NET(the i^th element E_i of E, NET);
   { Extend the network to support the ground instances of the query atom }
   SUBSS := BUILD-NET (Q, NET);
   IF SUBSS is empty THEN
        output "UNKNOWN"
   ELSE
        BEGIN UPDATE(NET,E);
                { Output posterior probabilities }
                FOR each θ in SUBSS
                     output the probability values at node obj(Qθ);
        END
END.
```

Fig. 8. Query processing procedure.

query are visited and marked as reachable. Upon termination of the search, only reachable nodes are retained.

As we approach each node within the search, we examine the direction of the incident edge into that node, whether the node is marked as an epsilon node, and whether the node is an evidence node. From this information we can determine the allowed directions of outgoing edges from that node. The relations between incoming edge (link from a parent), outgoing edge (link to a child), and the type of node are shown in Fig. 9. An incoming edge is represented by a + and outgoing node is represented by a -. For example, the first entry says that if a node is an evidence node and a path enters the node along an incoming edge then it can exit the node only along another incoming edge.

8. EMPIRICAL EVAULATION

In order to provide some evaluation of the computational efficiency of the network construction approach presented in this paper, we compared inference times using BNG with those using a standard Bayesian network representation for problems in the cardiac arrest domain. Since BNG is written in Lisp, to make the comparisons fair, we used IDEAL to perform the Bayesian network inference, both with BNG and without. The Jensen algorithm was used for all Bayesian network evaluations. The algorithm first triangulates the network to produce a junction tree and then propogates probabilities on the tree. We assumed that if standard Bayesian networks were used, they would be stored as junction trees, while use of BNG requires construction of the network, followed by triangulation. All experiments were conducted using Allegro Common Lisp 4.2 on a Sun Microsystems SPARC Station 20.

We compared the use of traditional Bayesian networks with BNG by processing queries for the cardiac arrest domain at times 1–6. We considered two approaches using a traditional Bayesian network to process these queries. The first approach used the single network shown in Fig. 4 that can be used to process all possible queries to time 6. The other approach stored six separate networks, one for each future time point through time 6 and accessed the smallest network necessary to process the query. So, for example, a query of rhythm at time 4 would be processed by a network that contained all nodes present in Fig. 4 up to time 4.

$+ (E) +$	$- (E)$	
$- (\neg E)\ - +$	$+ (\epsilon \wedge \neg E)\ + -$	$+ (\neg\epsilon \wedge \neg E) -$

Fig. 9. Traversal chart for identifying active paths.

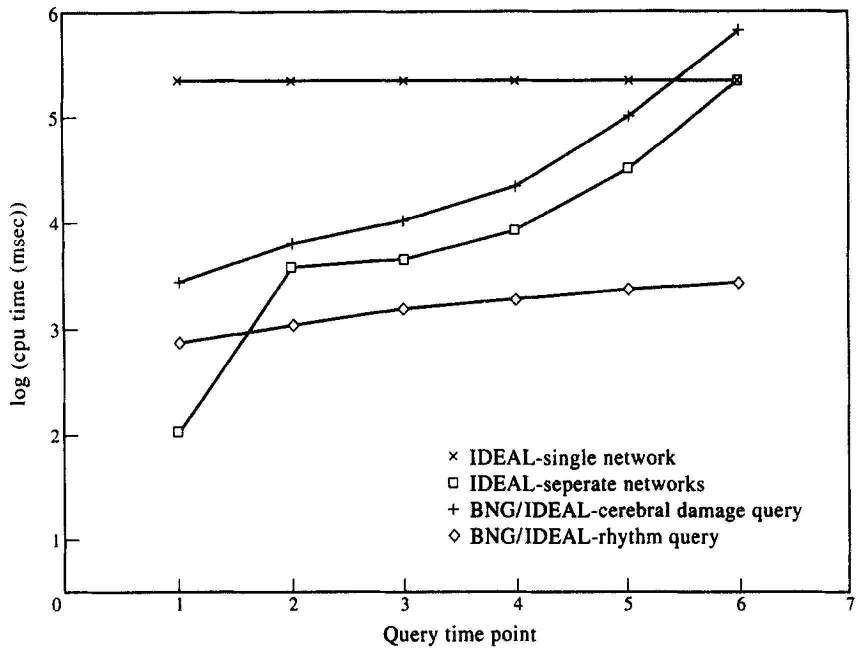

Fig. 10. Inference times for queries at various future times. Inference times for BNG include time required for network construction, triangulation of the constructed network, and propagation. Inference times for IDEAL include only propagation time.

Figure 10 shows the execution times for the three strategies. The first used BNG to generate the networks and then used IDEAL to evaluate them. To show the range of performance for this strategy, queries were performed on both the heart rhythm and level of cerebral damage at the given future time. These span the range of queries in terms of generated network size. The running time for the strategy of using a single stored network is constant and clearly worse in all but the most extreme cases. Using a separately stored network for queries at a given future time improves performance, yet all queries at that particular future time have the same running time. This strategy's performance falls in the middle of the performance range of the strategy that uses networks generated by BNG. General networks often contain information that is not necessary for answering a given query, while BNG always produces a minimal network.

The performance of BNG is clearly worse than that of IDEAL on some queries. This is due to the additional time needed to generate and triangulate the network. BNG outperforms IDEAL on other queries due to its ability to generate and use a minimal network and thus minimize the time needed to propagate evidence through the network.

Although BNG-generated networks are not saved from query to query, they can be used for multiple computations by changing the states of the set of evidence nodes and re-propagating the evidence. This can be useful when performing sensitivity analysis. So it is informative to separate out the propagation time from the time required for generation and triangulation of the network. Figure 11 shows the propagation time for the networks used to produce the four curves in Fig. 10. Networks generated by BNG always have a lower propagation time.

In addition to interfacing to IDEAL, the BNG system can also be run using HUGIN to evaluate the generated networks. To demonstrate the efficiency of using BNG with HUGIN, we compared the running time of the BNG/IDEAL combination with that of the BNG/HUGIN combination in evaluating queries for rhythm and cerebral damage at time 6. For the rhythm query BNG/HUGIN required only 0.84 s while BNG/IDEAL required 3.16 s while BNG/HUGIN required only 1.83 s. These numbers suggest that by writing BNG in an efficient language like C and coupling it with an efficient inference system like HUGIN, the performance can be made efficient enough for use in real-time clinical settings.

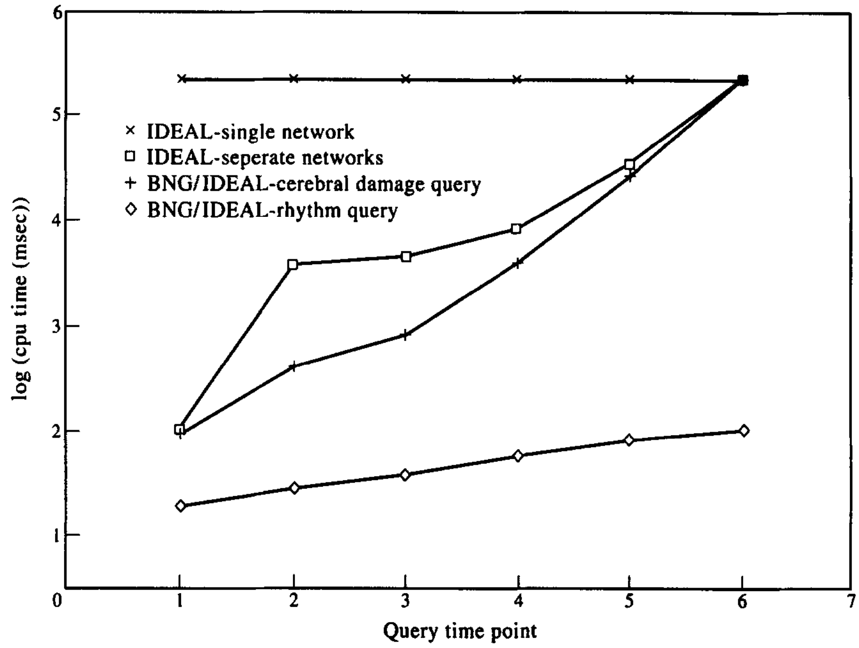

Fig. 11. Propagation times for queries at various future times.

9. RELATED WORK

In an earlier paper [12], we proposed a framework for constructing Bayesian networks from knowledge bases of probability logic sentences. The present work extends that framework in several significant ways. That work did not address the problem of representing context-sensitive knowledge or the representation of time. In addition, it imposed a number of constraints (Cl–C4) on the language in order to guarantee that a set of ground instances of the rules in the knowledge base was isomorphic to a Bayesian network. In this work we drop these constraints, which increases the expressive power of the language but requires us to specify combining rules. Our probabilistic independence assumption is simpler than that of d-separation for a knowledge base used in [12].

Our representation language has many similarities to Breese's Alterid [13]. Alterid can represent the class of Bayesian networks and influence diagrams with discrete-valued nodes. Influence diagrams are similar to Bayesian networks but include decision and value nodes. Breese also uses a form of predicates similar to our p-predicates. Alterid also has probabilistic rules with context constraints and uses logic program clauses to express relations among the context atoms. But in contrast to our framework, Breese mixes logic program clauses with probabilistic sentences by using the logic program clauses to infer information about evidence. In contrast, we separate logic program clauses from probabilistic sentences. In our framework, pure logical dependencies between p-atoms are represented by 0 and 1 probability values, which can be less efficient than representing them with logical sentences. Breese does not provide a semantics for the knowledge base. As a result, his paper cannot prove the correctness of his query answering procedure. Breese's procedure does both backward and forward chaining. Because Q-procedure only chains backwards, we can extend the procedure for some infinite domains. In Breese's framework, a set of ground instances of rules may not represent a single network. In this case, the construction algorithm must choose between networks to generate.

Goldman and Charniak [20,21] present a language called Frail3, for representing belief networks, and outline their associated network construction algorithm. In their framework, rules, evidence, and the emerging network are all stored in a common database. They also allow additional information to be stored in a separate knowledge base. For example, for the natural language processing applications they consider, the knowledge base can be used to store a dictionary of word meanings. The knowledge base plays a role similar to that of our context base.

Frail3 represents network dependencies by rules with variables. The rules in Frail3 differ from ours in that they contain embedded control information. Frail3 uses two operators to express network construction rules: → and →←. The → rules have the form (→ trigger consequent:prob pforms). When a statement unifying with trigger is added to the database, the bindings are applied to the consequent and the resulting statement is added to the database, along with a possible probabilistic link. The atoms within the rule with which the link is associated and the direction of the link are specified independent of the direction of rule application. The →← operator is used to incorporate information from a knowledge base and to perform network expansions conditional on the previous network contents. Its first function is similar to our notion of context constraints. The language includes numerous combining rules, such as noisy-OR. Networks are generated by a forward and backward chaining TMS type system. A rigorous semantics for the Frail3 representation language is not provided.

Nicholson and Brady [2] discuss a method of dynamically constructing temporal Bayesian networks for discreet event monitoring. They are able to control the size and complexity of the models by pruning states from nodes, arcs between nodes, or complete nodes. The resulting network may be a precise or approximate representation. They do not provide a formal semantics for their knowledge base representation language. They demonstrate the application of their system to the problem of monitoring the movement of robot vehicles and people in a simple environment.

Provan and Clark [1] present a system called DYNASTY for constructing dynamic influence diagrams. DYNASTY 's model construction algorithm is similar to that of ALTERID. It takes as input a Horn-clause knowledge base and a set of observations and constructs a temporal influence diagram. DYNASTY differs from ALTERID in that the probabilistic relations and the network topology of the constructed diagram can vary with time. Furthermore, the system can construct approximate diagrams in order to trade off precision for inference time. The representation language is not provided with a formal semantics. Provan and Clark demonstrate the application of their system to the diagnosis of acute abdominal pain.

Poole [14] expresses an intention similar to ours: there has not been a mapping between logical specifications of knowledge and Bavesian network representations... He provides such a mapping using probabilistic Horn abduction theory in which knowledge is represented by Horn clauses and the independence assumption of Bayesian networks is explicitly stated. His work is developed along a different track than ours, however, by concentrating on using the theory for abduction. Our approach has several advantages over Poole's. We do not impose as many constraints on our representation language as he does. Probabilistic dependencies are simply directly represented in our language, while in Poole's language they are indirectly specified through the use of special predicates in the rules. Our probabilistic independence assumption is more intuitively appealing since it reflects the causality of the domain. Our framework is representationally richer than Poole's by allowing context-dependent specification.

Kanazawa [22] presents a continuous-time logic for expressing knowledge about time and probability. He shows how to map his logic onto time nets, which are special kinds of Bayesian networks that encode the probabilities of facts and events over time. He does not provide a model construction algorithm and does not address the issue of the size of a model produced by mapping sentences in his logic onto a time net.

Dagum *et al.* [23] present a synthesis of belief network models with classical time-series analysis, which they call dynamic network models. A dynamic network model consists of a set of random variables connected by synchronic and diachronic links. These links are considered to be fixed over time. In our framework the links can vary over time. The conditional probabilities are specified in parameterized form and are updated over time to account for unmodeled erogenous influences. The dynamic network model framework emphasises lipdating of the actual model parameters, while our work assumes that the parameters, as specified in the knowledge base, are fixed. Model parameters in our constructed networks can change in response to varying context as a function of time.

10. DISCUSSION AND FUTURE RESEARCH

We have presented a theoretically well-founded method for constructing temporal Bayesian networks from context-constrained rules. The presence of a formal semantics for the representation language is necessary in order to prove the correctness of the network construction algorithm. Such proofs are important for the high-stakes decision making problems encountered in medicine. Our technique is capable of selecting that portion of a probability model that is relevant to a particular inference problem by using context information and by pruning the constructed network. The naturalness of the encoding of the cardiac arrest domain shows that the representation is relatively easy to use. The networks generated to solve the example problems illustrate the potential computational savings of the technique. Computational efficiency becomes a major issue as researchers attempt to model larger and more complex domains.

Our framework for temporal probabilistic reasoning is still in the preliminary stages. We are exploring extending the framework to allow infinite domains, continuous random variables, and function symbols. We are also investigating how to reason with quantified atoms, i.e. the p-sentences can have the form $\forall Y Pr(\exists X has\_cancer(Y, X) irradiated(Y)) = 0.8$. The current framework does not have the concepts of decision and utility. We plan to investigate the representation of decision models and efficient procedures for finding optimal or good decision strategies.

Acknowledgements—We would like to thank Hugin Expert A/S for providing us the use of the Hugin software. This work was partially supported by NSF grant IRI-9509165 (PH), by a University of Wisconsin-Milwaukee Graduate School Fellowship (LN), and by a Sun Microsystems AEG award (PH).

REFERENCES

1. G.M. Provan and J.R. Clarke, Dynamic network construction and updating techniques for the diagnosis of acute abdominal pain, *IBEE Trans. Patt. Anal. Mach. Intell.*, **15**, 299–307 (1993).
2. A.E. Nicholson and J.M. Brady, Dynamic belief networks for discrete monitoring, *IEEE Trans. Syst. Man Cybernet.*, **24**, 1593–1610 (1994).
3. G.F. Cooper, The computational complexity of probabilistic inference using Bayesian belief networks, *Artif. Intell.*, **42**, 393–405 (1990).
4. T.L. Dean and M.P. Wellman, *Planning and Control.* Morgan Kaufmann, San Mateo, CA (1991).
5. A. Darwiche and M. Goldszmidt, Action networks: a framework for reasoning about actions and change under uncertainty. In *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, Seattle, pp. 136-144 (1994).
6. J.S. Blythe, Planning with external events. In *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, Seattle (1994).
7. P. Safar, Cerebral resuscitation after cardiac arrest: research initiatives and future directions. *Ann. Emergency Med.* (1993).
8. A.T. Bahill, *Verifying and Validating Personal Computer-based Expert Systems.* Prentice-Hall, Englewood Cliffs, NJ (1991).
9. J.W. Lloyd, *Foundation of Logic Programming*, 2nd edn. Springer-Verlag, Berlin (1987).
10. J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann, San Mateo, CA (1988).
11. S. Srinivas, A generalization of the noisy-or model. In *Proceedings of the Ninth Conference on Uncertainty in AI*, pp. 208-217 (1993).
12. P. Haddawy, Generating Bayesian networks from probability logic knowledge bases. In *Proc. of the Tenth Conference on Uncertainty in Artificial Intelligence*, Seattle, pp. 262-269 (1994).
13. J.S. Breese, Construction of belief and decision networks, *Comput. Intell.*, **8**, 624–647 (1992).
14. D. Poole, Probabilistic horn abduction and Bayesian networks, *Artif. Intell.*, **64**, 81–129 (1993).
15. P. Haddawy, Representing plans under uncertainty: a logic of time, chance, and action. *Lecture Notes in Artificial Intelligence*, Vol. 770. Springer-Verlag, Berlin (1994).
16. K.R. Apt and M. Bezem, Acyclic programs. *New Generation Computing*, 335-363 (1991).
17. S. Srinivas and J. Breese, IDEAL: A software package for analysis of influence diagrams. In *Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence*, 212-219 (1990).
18. S.K. Andersen, K.G. Olesen, F.V. Jensen and F. Jensen, HUGIN a shell for building Bayesian belief universes for expert sytems. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, Detroit, Michigan, pp. 1080-1085 (1989).

19. R.D. Shachter, Probabilistic inference and influence diagrams, *Operations Res.*, **36**, 589–604 (1988).
20. R.P. Goldman and E. Charniak, Dynamic construction of belief networks. In *Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence*, Cambridge, Massachusetts, pp. 90-97 (1990).
21. R.P. Goldman and E. Charniak, A language for construction of belief networks, *IEEE Trans. Patt. Anal. Machine Intell.*, **15**, 196–208 (1993).
22. K. Kanazawa, A logic and time nets for probabilistic inference. In *Proceedings of the Ninth National Conference on Artificial Intelligence*, pp. 360-365 (1991).
23. P. Dagum, A. Galper and E. Horvitz, Dynamic network models for forecasting. In *Proceedings of the Eighth Conference on Uncertainty in Artificial Intelligence*, Stanford, pp. 41-48 (1992).

APPENDIX

A.1. *Formal Definitions of RAS and RPB*

We use completed logic programs proposed by Clark [9] for the semantics of the context base. Essentially, a completed logic program is formed from a logic program by adding to it: (1) some clauses so that the clauses in the original program become the only definitions of the predicates in the program; and (2) the equality theory to interpret the introduced equality predicate [9].

Let completed ($C \cup CB$) be the completed logic program with the associated equality theory [9] constructed from $C \cup CB$, then have the following definition of the set of relevant p-atoms.

Definition 23 *Given a set of evidence E, a set of context information C and a KB, the set of relevant p-atoms (RAS) is defined recursively by: (1) ground(E)$\subseteq$RAS; (2) if S is a ground instance of a probability sentence conforming to type constraints such that context(S) is a logical consequence of completed($C \cup CB$) and ante(S)$\subseteq$RAS then cons(S)$\in$RAS; (3) if a p-atom A is in RAS then Ext(A)$\subseteq$RAS; (4) RAS is the smallest set satisfying the above conditions.*

The RAS is constructed in a way similar to Herbrand least models for Horn programs. Context information is used to eliminate the portion of PB which is not related to the current problem.

Proposition 24 *Given a set of evidence E, a set of context information C and a KB, RAS always exists.*

Proof The proof is similar to that in [9] for the existence of the least Herbrand model for a Horn program. □

Definition 25 *Given a set of evidence E, a set of information C, and a KB; the set of relevant p-sentences (RPB) is constructed in the following way:*

(1) Let R be the set of all prob(S), such that S is a ground instance of a p-sentence in PB, context(S) is a logical consequence of completed($C \cup CB$), cons(S)$\in$RAS, and ante(S)$\subseteq$RAS.

(2) If $A \in$ground(E) and $\neg\exists S \in R$: cons(S)$\in$Ext(A) then add to R the following set $\{Pr(A)=1\} \cup \{Pr(A')=0 | A' \in Ext(A) \wedge A \neq A'\}$.

RPB is the final set R. Condition (2) augments the knowledge base KB with new knowledge about the evidence.

A.2. *Q-procedure*

Q-procedure uses a form of SLD-resolution (Linear resolution with Selection function for Definite clauses) to reason on PB and SLDNF-resolution (SLD with Negation-as-Failure) to answer context queries on $C \cup CB$ [9]. SLD and SLDNF are proof procedures which form the basis of the Prolog engine. The SLD-like portion of Q-procedure is more complex than SLD because it needs to collect all relevant sentences before combining rules can be used. For that purpose, Q-procedure needs to maintain a list of all ground p-sentences relevant to the current query atom. Q-procedure also calls a Bayesian network belief updating procedure. There are several available procedures for that purpose [10].

In checking for the validity of contexts, Q-procedure frequently calls the SLDNF proof procedure which works on C ∪ CB and queries provided by Q-procedure. SLDNF is sound and, for some classes of normal logic programs, complete under completed program semantics.

Figures 12 and 13 provide details of BUILD-NET and BUILD-C-NET called by Q-procedure.

A.3. *Proofs*

Proof of Theorem 13 *Let M be a possible (ι, τ)-model of CRPB. We arrange the atoms in M in the order of the influenced-by relationship: If A_i is influenced by A_j in (ι, τ)-CRPB then $j>i$. Let Pr^* be a probability distribution induced by (ι, τ)-CRPB. We have $Pr^*(M)=Pr^*(\wedge_{i=1}A_i)=Pr^*(A_1|\wedge_{i=2}A_i)\times Pr^*(\wedge_{i=2}A_i)=P(A_1|A11, ..., A_{1n})\times Pr^*(\wedge_{i=2}A_i)=\alpha''' \times Pr^*(\wedge_{i=2}A_i)$, where $P(A_1|A_{11}, ..., A_{1n})=\alpha$ is a sentence in (ι, τ)-CRPB with $\{A_{11}, ..., A_{1n}\}\subseteq M$, by the independence assumption. By using induction on the length of the sequence, we get a unique Pr^*.* □

Proof of Theorem 20 If SLDNF always stops then Q-procedure always stops for acyclic PBs with finite domains because in each pass of BUILD-NET the input atom is instantiated to ground and the function call to BUILD-C-NET transfers only atoms with smaller P_{ord}.

```
function BUILD-NET(A: an p-atom; var NET: a Bayesian net):set of substitutions;
var SUBSS, SUBSS1: a set of substitutions;
  RS, CRS: a set of p-sentences;
  S: a p-sentence;
  θ^S, θ^C, θ, θ': substitution;
  B: an atom;
BEGIN
   A:= A after replacing val(A) by a new variable name;
   SUBSS := {};
   RS := {};
   FOR each S ∈ P such that there exists
        a most general unifier θ^S of A and cons(S) DO
        { Assume that if context(S) is empty then there is exactly
             one θ^C, which is the empty substitution }
        FOR each θ^C which is a computed answer from
             (← context(S)θ^S, C ∪ CB) DO
             IF ante(S) = {}
             THEN RS := RS ∪{Sθ^Sθ^C}
             ELSE  BEGIN
                       SUBSS1 :=
                       BUILD-CONJUNCTION-NET(ante(S)θ^Sθ^C, NET);
                       RS:= RS ∪{Sθ^Sθ^Cθ'|θ' ∈ SUBSS1}
                  END;
   CRS := COMBINE(RS, CR);
   FOR each atom B in the head of some rule in CRS DO
        BEGIN
             SUBSS := SUBSS ∪ {θ|Aθ = B};
             IF obj(B) is not in NET THEN
                BEGIN construct node obj(B);
                        build the links from the parents
                END;
             assign the corresponding link matrix entries;
        END;
   RETURN SUBSS;
END.
```

Fig. 12. BUILD-NET procedure.

```
BUILD-C-NET
Function BUILD-C-NET( A_1 ∧ ... ∧ A_n: a conjunction of p-atoms; var NET:
        a Bayesian net; Extended_PB: a set of p-sentences): set of substitutions;
var SUBSS, SUBSS1: set of substitutions;
BEGIN
        SUBSS := BUILD-NET (A_1, NET, Extended_PB);
        IF n > 1 THEN BEGIN
           SUBSS1:= SUBSS;
           SUBSS:={};
           FOR each θ ∈ SUBSS1 DO
                SUBSS:= SUBSS∪{θθ'| θ' ∈ BUILD-C-NET(
                            (A_2∧... ∧ A_n)θ, NET, Extended_PB)};
        END;
        RETURN SUBSS;
END.
```

Fig. 13. BUILD-C-NET function.

The pruning step in the main procedure of Q-procedure does not affect the posterior probability of the queries [19]. By a proof similar to that of Theorem 13, we can show that the supporting network constructed by Q-procedure is sufficient for evaluating the posterior probability of the query given the evidence. The correctness of the theorem follows from the soundness of the bayesian network updating procedure [10]. □

Proof of Theorem 22 Under the conditions of the theorem, only time can have infinite domain. Hence, if we restrict to the finite interval $[\iota, \tau]$, all domains are finite. Because the domains are finite and CB ∪ C is acyclic, any ground SLDNF-refutation has a bounded length [16]. A simple version of SLDNF in which every goal (or subgoal) is instantiated to ground (like in BUILD-NET) before the unification will stop on any input goal (by the same reason that makes BUILD-NET stops). That version of SLDNF is sound and complete for acyclic logic program with finite domains [16]. The result follows directly from Theorem 20. □

About the Author—Liem Ngo received his BS degree in Computer Science from the University of HoChiMinh City, Vietnam, in 1985. He worked for 5 years (1985-1990) as a lecturer at the Department of Mathematics, University of HoChiMinh City. He received a MS degree in Computer Science from the Asian Institute of Technology, Bangkok, Thailand. His Master's thesis was on the semantics of logic programs. He received another MS degree in Management Information Systems from the School of Business, University of Wisconsin-Madison in 1994. He was a Fulbright fellow for 2 years (1992–1994). Currently, he is a PhD student at the Department of Electrical Engineering and Computer Science, University of Wisconsin-Milwaukee. His dissertation work is devoted to the foundation of Knowledge-Based Bayesian network model construction and its application to planning under uncertainty. His research interests include Bayesian networks, logic programming, knowledge representation and planning under uncertainty.

About the Author—Peter Haddawy received his BA degree in Mathematics from Pomona College in 1981. He received his MS in 1986 and his PhD in 1991, both in Computer Science from the University of Illinois at Urbana-Champaign. He is currently Associate Professor in the Department of Electrical Engineering and Computer Science at the University of Wisconsin-Milwaukee and director of the Decision Systems and Artificial Intelligence Lab. His research interests include artificial intelligence, decision theory, knowledge representation, planning, decision making, and medical applications. He is the author of Representing Plans Under Uncertainty: A Logic of Time, Chance, and Action. His current research deals with knowledge-based construction of belief networks, decision-theoretic planning, and uses of abstraction in decision-theoretic reasoning.

About the Author—Robert A. Krieger received his BS in Computer Science from University of Wisconsin-Milwaukee in 1990 and his MS in 1994. His Master's thesis, entitled Construction of Minimal Temporal Bayesian Networks from Probability Logic Knowledge Bases developed and proved the correctness of an implemented algorithm that dynamically constructs Bayesian networks from knowledge bases of probabilistic rules. He is currently Product Development Software Engineer for Merge Technologies, Milwaukee, WI, where he provides Ethernet networking solutions to hospitals and other users of medical scan data.

About the Author—JAMES W. HELWIG is finishing his MS in Computer Science at the University of Wisconsin in Milwaukee. His research at the university's Decision Systems and Artificial Intelligence Laboratory involves developing ASPIRE, a system for integrating abstraction-based planning and execution. As a member of the Medical Informatics and Decision Science Consortium of the University of Wisconsin and the Medical College of Wisconsin, his graduate work also includes applying decision-theoretic systems to problems in the medical domain. Before pursuing graduate work, he taught secondary mathematics and science after receiving his Bachelors of Science in Mathematics and Physics from the University of Wisconsin in River Falls. He currently lives with his wife in Madison, Wisconsin.