

HUGIN*— a Shell for Building Bayesian Belief Universes for Expert Systems

Stig K. Andersen and Kristian G. Olesen
Finn V. Jensen and Frank Jensen
Institute of Electronic Systems
Aalborg University
Strandvejen 19, DK 9000 Aalborg, Denmark

Abstract

Causal probabilistic networks have proved to be a useful knowledge representation tool for modelling domains where causal relations in a broad sense are a natural way of relating domain objects and where uncertainty is inherited in these relations. This paper outlines an implementation the HUGIN shell- for handling a domain model expressed by a causal probabilistic network. The only topological restriction imposed on the network is that, it must not contain any directed loops. The approach is illustrated step by step by solving a genetic breeding problem. A graph representation of the domain model is interactively created by using instances of the basic network components—nodes and arcs—as building blocks. This structure, together with the quantitative relations between nodes and their immediate causes expressed as conditional probabilities, are automatically transformed into a tree structure, a junction tree. Here a computationally efficient and conceptually simple algebra of Bayesian belief universes supports incorporation of new evidence, propagation of information, and calculation of revised beliefs in the states of the nodes in the network. Finally, as an example of a real world application, MUNIN an expert system for electromyography is discussed.

1 Introduction

In recent years, locus has been put on the development of expert systems based on a probabilistic scheme where the structure of the domain knowledge is represented in a directed graph. Contrary to earlier probabilistic attempts, this approach makes the domain structure explicit and utilizes the topology of the graph in the control of the inference. The probabilistic scheme has as opposed to rule based and logic based systems—an intrinsic distinction between so-called 'expectation evoked' and 'explanation evoked' inference [Pearl,

* HUGIN stands for Handling Uncertainty In General Inference Network. Coincidentally, according to Norse mythology, Hugin and Munin are two ravens, whispering intelligence into the ears of the god Odin. MUNIN is also used as acronym for an expert system for electromyography.

1988]. This knowledge representation has been termed differently in the literature: Bayes belief net [Pearl, 1986a], causal probabilistic network [Andreassen *et al.*, 1987], causal networks [Lauritzen and Spiegelhalter, 1988], probabilistic causal network [Cooper, 1984], influence diagrams [Howard and Matheson, 1981], and so on.

A causal probabilistic network, which is the term we have chosen to use in this paper, is— at a qualitative level—a graph where the nodes represent domain objects and the links between nodes represent causal relations between these objects. At a quantitative level, the relations expressed by the links are represented by conditional probabilities. Together with methods derived from Bayesian decision theory, this formalism offers a consistent means of handling uncertainty inherent in expert systems [Horvitz *et al.*, 1988].

The key issue of this formalism is the ability to reduce the computation to a series of local calculations using only variables obtainable from one object and its neighbors in a graph structure. Hereby a calculation of the global joint, probability distribution is avoided. Furthermore, the graph representation showing explicit dependencies and independencies—is a powerful tool in the knowledge acquisition and verification process [Pearl and Paz, 1987].

When the network is singly connected (i.e., at most one path between pairs of nodes is allowed), Kim and Pearl [1983] gives a conceptually simple scheme for passing messages along links between nodes in the causal probabilistic network. Unfortunately, not all real world problems can be modelled as a singly connected network, and a multiply connected network (i.e., more than one path between pairs of nodes may exist) is required in order to catch the facets of the domain. Several schemes are proposed in the literature such as *reasoning by assumption* [Pearl, 1986], and *value preserving arc-reversing* [Shachter, 1988].

This paper outlines the expert system shell MUCIN. This shell provides (1) a set of tools for creation of a causal probabilistic network which simultaneously creates a graphical interface to the domain model, and (2) a set of tools for entering quantitative information about the state of the domain and for propagating this in the network. Behind the scenes, the *Algebra of Bayesian Relief Universes* is used. This scheme [Andersen *et al.*, 1987, Jensen, 1988a, Jensen *et al.*, 1988, Olesen and Andersen, 1988], which is a substantial simplification of [Lauritzen and Spiegelhalter, 1988], allows prop-

agation of evidence in acyclic multiply connected networks. The algebra is conceptually as simple as the method of Kim and Pearl [1983] for singly connected causal probabilistic networks. This simplicity is achieved by transforming the multiply

connected network into a set of *belief universes* organized as a tree, a *junction tree*, and by providing the operations for propagation of information between belief universes in the tree. The basic operation is *calibration* of one belief universe with another. Calibration is the basis for two operations: *Distribute Evidence* and *Collect Evidence*, which distributes evidence from one belief universe to all other belief universes, and collects evidence to one belief universe from all other belief universes, respectively. The idea of a separate representation for propagation purposes is also used in the approaches of Dechler *et al* [1988] and Shaler and Sheuoy [1988].

In order to illustrate the HUGIN shell, the next section deals with an example from genetics. The reader is guided through the major steps from specifying the model to making it operational, and finally inferring beliefs based on data entered into the model. In Section 3, the use of the algebra of Bayesian belief universes is outlined. Section 1 contains an overview of the HUGIN shell architecture, and finally Section 5 discusses problems in applying HUGIN to real world applications, exemplified by MUNIN, an expert system for electromyography.

2 An example: Breeding

We will demonstrate the HUGIN shell by going through an example taken from genetics (inspired by Spiegelhalter [1989]). The example concerns breeding and would typically be found within animal husbandry, but in order to make it more attentive, we state it in terms of human beings:

Florence and Gregory are about to reproduce. However, Gregory is Florence's nephew, and in the annals of Bartholomew, the father of Florence and grandfather of Gregory, a life-threatening disease has haunted. The disease is caused by a dominant allele *A*, and appears in a rather late stage of the individual's life-time. Bartholomew married twice, hence Florence and Gregory's mother are half-siblings. Neither Florence nor Gregory, their parents, nor Gregory's grandmother have shown any signs of the disease. What is the risk that their child will inherit the fatal characteristic?

2.1 Qualitative modelling

The first step in a HUGIN session is to formulate the problem in terms of a causal probabilistic network. The genealogical structure of our example is shown in Figure 1.

Each individual is represented by name, and parent-child relations are represented by arrows from parent to child. Note that the diagram contains two different paths from Bartholomew to H: via Eliza and Gregory, and via Florence.

In general, we are interested in determining the genotype of the individuals. To each individual, we associate a node labelled with his/her initial and as states the possible genotypes identified by the combination of alleles. Hence, each individual is of exactly one of the genotypes *AA*, *Aa*, or *aa*.

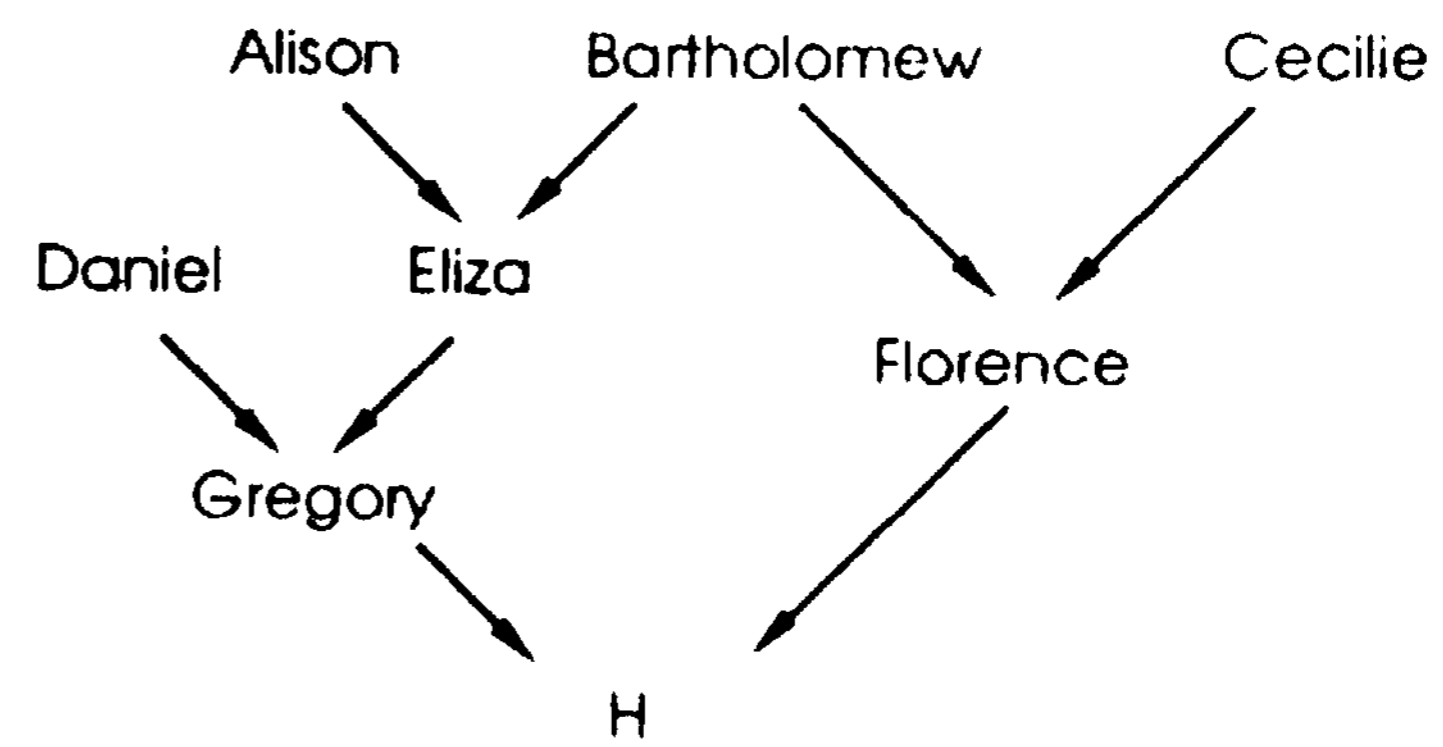


Figure 1: Genealogical structure for the example.

Figure J: Genealogical structure for the example.

In our case, where one allele is dominant, this means that the genotypes *AA* and *Aa* are carriers of the disease, whereas *aa* is not. What can be observed is whether the disease is present or not, but this can only be determined when the individual has reached a mature age due to the sneaky character of the disease. In order to be able to enter relevant information on observed occurrences and ask for expectations of the disease, we add disease nodes to the elder and the coming generation. These nodes, labelled by the individuals initial with index *d*, have two possible states, *yes* and *no*, corresponding to the presence and absence of the disease, respectively.

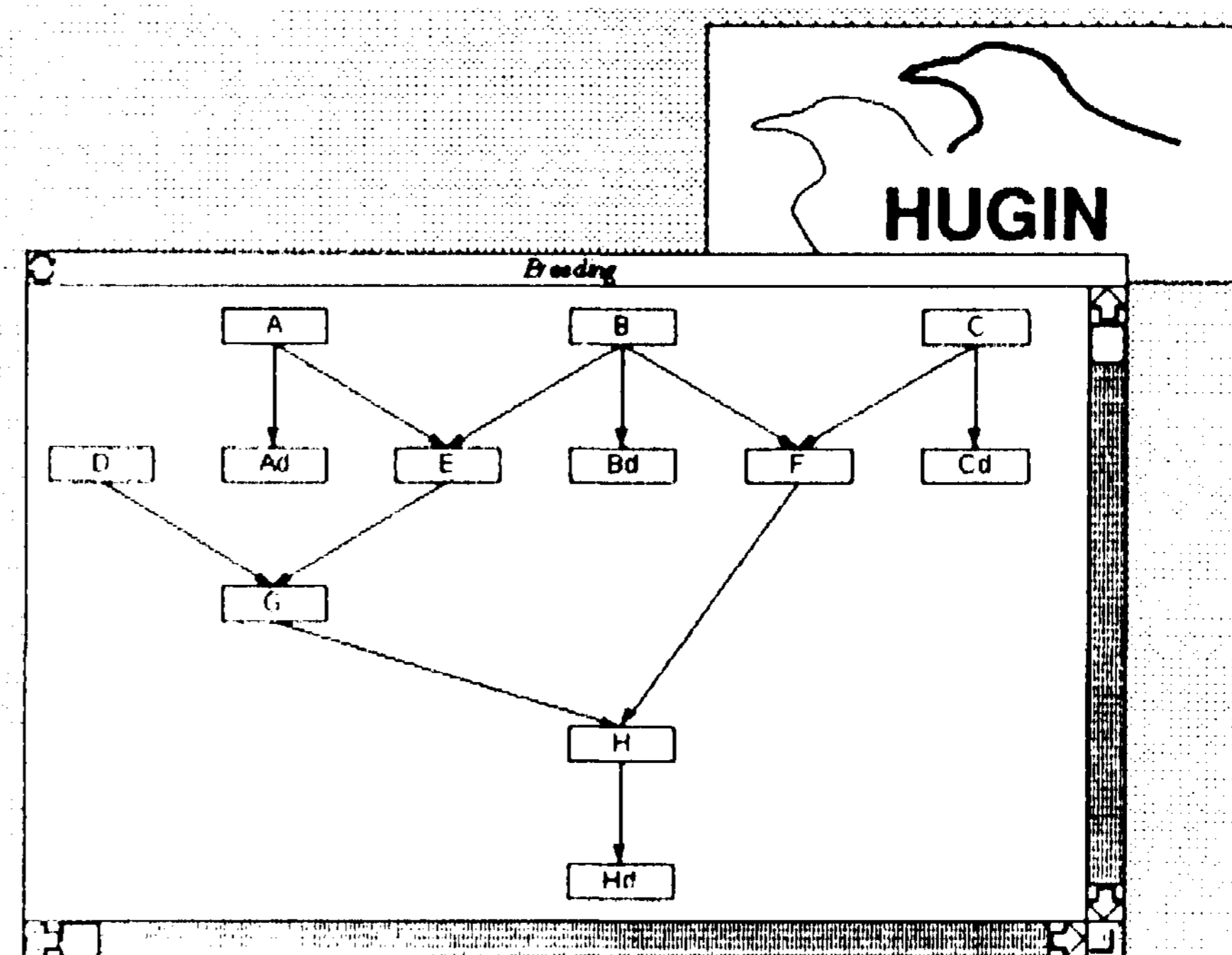


Figure 2: HUGIN screen dump.

At this point, the qualitative structure of our problem is determined, and we invoke the HUGIN shell. The shell is menu-driven, and most operations are done by mouse selection. Figure 2 shows the main screen with the model entered. Here a domain specific window labelled *Breeding* is created, and the model is constructed: this is done by selecting the *Create domain* entry on the main menu (attached to the HUGIN icon). All nodes and links are entered, and the corresponding graphical symbols are created. For each node, a display name and the states are typed in. Entering states for the disease nodes is not necessary though, as the states *yes* and *no* apply as default. If so desired, the domain model can be saved at any point, for later continuation by reloading.

¹The reader may feel a need for a discussion of temporal relationships appropriate, but this is outside the scope of this paper.

2.2 Quantitative modelling

Let us complete the model by specifying the numeric part of the links. This is done by means of conditional probability tables. For each node, we give a conditional probability for each state given the states of its parents. Thus for the disease nodes, we have tables as specified in Table 1.

Disease	Genotype		
	AA	Aa	aa
yes	1	1	0
no	0	0	1

Table J: Conditional probability table for genotype-disease relations. Each entry specifies the probability of the disease given the genotype.

P_1	P_2		
	AA	Aa	aa
AA	(1, 0, 0)	($\frac{1}{2}, \frac{1}{2}, 0$)	(0, 1, 0)
Aa	($\frac{1}{2}, \frac{1}{2}, 0$)	($\frac{1}{4}, \frac{1}{2}, \frac{1}{4}$)	(0, $\frac{1}{2}, \frac{1}{2}$)
aa	(0, 1, 0)	(0, $\frac{1}{2}, \frac{1}{2}$)	(0, 0, 1)

Table 2: Conditional probability table for parent-child relations. Each entry specifies the probability distribution for the genotype of the child (A A, Aa, aa) given the genotypes of parents P_1 and P_2 .

For the parent-child relations, we get tables as shown in Table 2, where parents P_1 and P_2 have offspring C. Since top-nodes do not have any parents (in our model that is), their tables degenerate to a priori values for their genotypes. Assume that the conditions for Hardy-Weinberg equilibrium are fulfilled, and let the frequency of A be p and the frequency of a be q ($p+q = 1$). Then the a priori probabilities for the genotype of any individual N is $P_0(N) = (p^2, 2pq, q^2)$. For $p = 0.01$, we get $P(A) = P(C) = (0.001, 0.0198, 0.9801)$. For Bartholomew, we estimate $P(B) = (0.0025, 0.2500, 0.7475)$ as his grandmother was known to suffer from the disease.

We will now enter these considerations into HUGIN. Select the *Specify table* entry from the menu attached to the node 11 in the *Breeding* window, and a new dialogue window pops up (Figure 3).

Table for H			
Parent	State		
G	AA	NEXT	
F	Aa		
State	AA	Aa	aa
Probability	0.5	0.5	0

Figure 3: Entering table for H.

A skeleton for the table is automatically generated, and the numbers are typed in. Now the domain model is complete and a selection of *Compile* will generate an operational run-time system. Although this last step appears simple, it actually holds the key to the success of the method. What happens is that the state space of the model is factorized in order to allow for efficient computations. This is discussed in more detail in Section 3.

2.3 Interactive problem solving

Asking for the belief in node H_d , we see a risk of 10% (Figure 1) for // being a victim of the disease. This seems reasonably low, and we can give Florence and Gregory our blessings. Unfortunately, while structuring the model, Bartholomew, our young lovers' common ancestor, has caught the disease. To see whether this affects our approval of the marriage, let us enter this new fact into our model. Select *Set state* on B_d and enter the fact; then update the beliefs by selecting *Propagate*.

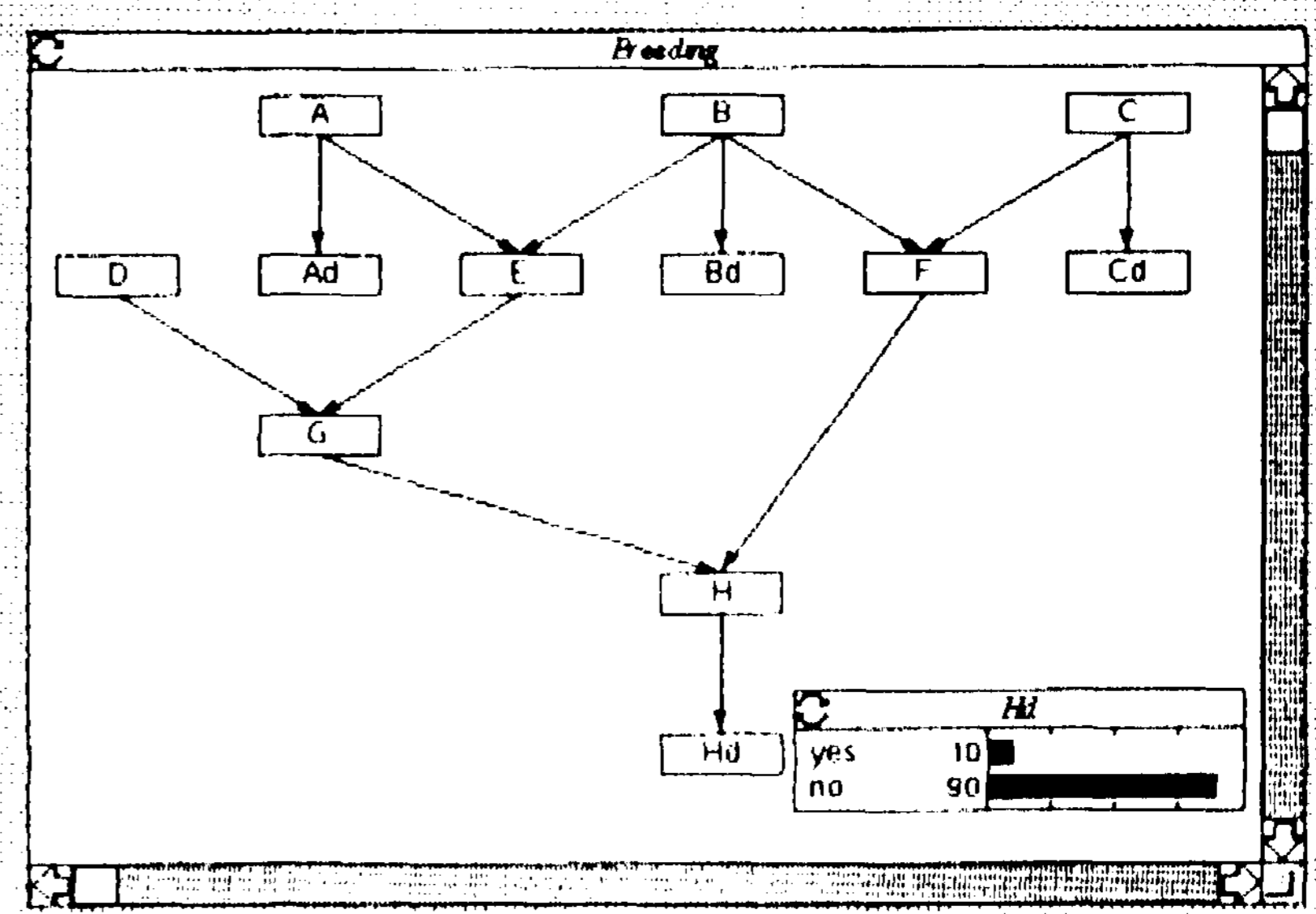


Figure 4: Display of belief in H_d .

A request for belief in H_d now yields a risk of 36%, forcing us to implore Florence and Gregory to reconsider their plans.

Suppose that the information about Bartholomew's alleged misfortune was wrong, but that instead Alison showed the disease. We would then *Retract* the finding on B_d (returning to the a priori belief for B) and follow the described procedure on A_d . Alternatively, we could choose *Initialize* and repeat as above. In both cases, we find IPs risk to be 22%.

At this point other possibilities could be investigated, but we will leave our friends with their worries here and close the section.

3 Behind the scenes

In the example of Section 2, several things happen behind the scenes, but the user does not need to worry about that unless the model has a complexity that requires manual intervention (see Section 5).

In this section, we outline the most important features from behind the scenes. In [Lauritzen and Spiegelhalter, 1988] and in the discussion by Jensen [1988b], and by Olesen and Andersen [1988] in the same article, an exposition of the methods can be found. A full description of the methods is given in [Jensen et al., 1988] and in [Jensen, 1988a].

3.1 Transformation of the causal network

The inference in a causal network is in HUGIN not done directly in the network. Instead, the domain represented by the network is partitioned into a set of subdomains called *belief universes*.

A belief universe U consists of a set of nodes, denoted $|U|$ and a *belief table*. For reasons to become clear soon, the node sets are called *cliques*. A belief table is a non-normalized table for the joint probabilities of the belief space.

The construction of a system of belief universes, equivalent to the original domain model, consists of the following steps:

- *Form the moral graph:* For each node in the network, add links between all of its parents. Drop the directions.
- *Triangulate the moral graph:* A graph is triangulated if all cycles of length > 3 have a chord. Add links to the moral graph until a triangulated graph is obtained.
- *Form the system of belief universes:* The node sets are the cliques of the triangulated graph (a clique is a maximal set of nodes, all of which are pairwise linked). The initial belief tables are calculated as appropriate products of the conditional probability tables (described by Lauritzen and Spiegelhalter [1988]).
- *Organize the system as a junction tree:* Links between belief universes are introduced. This is done such that a tree with the following property results: For all universes U, V , each belief universe on the unique path between U and V contains $|C| \cap |V|$. In Figures 5 and 6, a junction tree for the domain from Section 2 is given. As proved by Jensen [1988a], a junction tree can be constructed by a maximal spanning tree algorithm.

All steps except the second are deterministic: There is only one moral graph, and the set of cliques of a triangulated graph is unique. There may be several junction trees, though, but the differences are not (at this point) essential.

The second step—the triangulation—is not trivial. There is a wide range of literature on triangulation [Rose *et al.*, 1976, Yannakakis, 1981, Tarjan and Yannakakis, 1981, Kong, 1986], and it is an NP-complete problem to find a triangulation giving the minimal total table size [Gavril, 1977]. In HUGIN, the algorithm used for triangulation is maximum cardinality search [Tarjan and Yannakakis, 1984]. An algorithm—giving much better results on sparse graphs—has been constructed [Jensen and Kjierulff, 1989] and will be integrated into the next version of HUGIN.

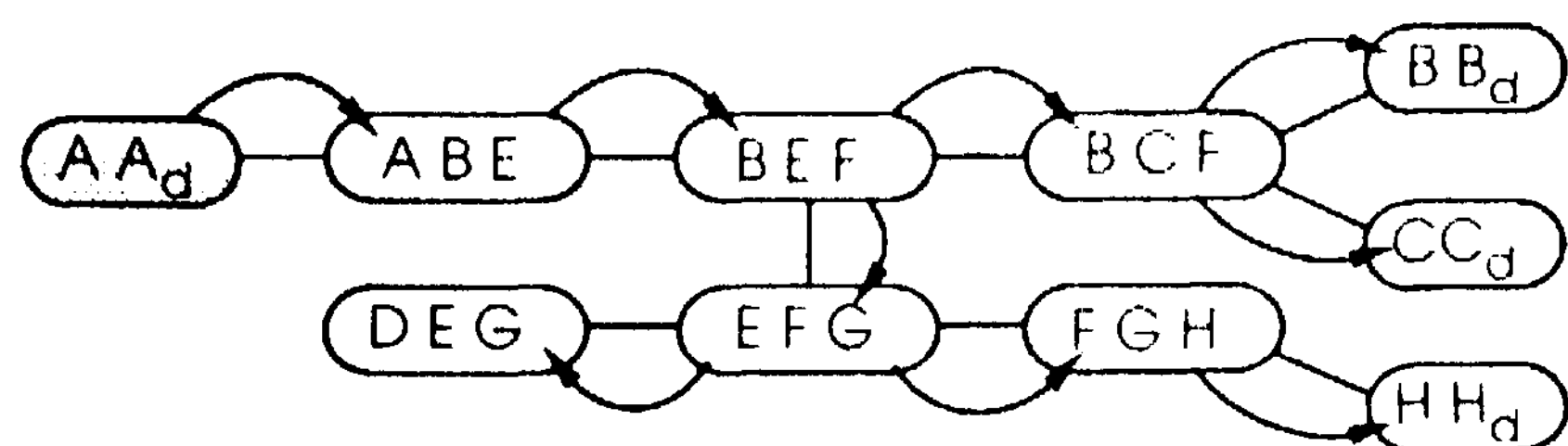


Figure 5: *Distribute Evidence* is called from $\{A, A_d\}$, and the evidence is propagated to all universes.

3.2 Propagation of evidence

A belief table is a non-normalized assessment of joint probabilities for a clique. Let V be a belief universe with belief

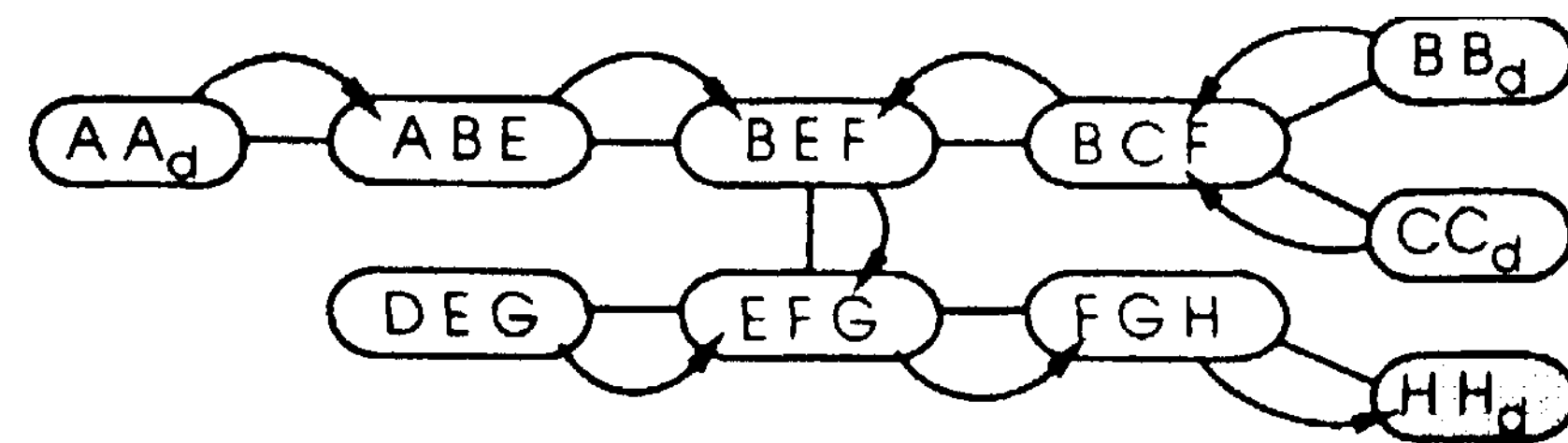


Figure (i): The flow of evidence when *Collect Evidence* is called in

table J , and let $S \subseteq |U|$. An assessment of joint probabilities for S can be obtained from B by summing up all beliefs in B for S . This operation is called *marginalization*. In particular, the belief in a single node can be obtained by marginalization in the normalized table of any belief universe containing it.

Evidence is transmitted between belief universes through the *calibration* operation. A belief universe U calibrates to another belief universe V as follows [Jensen *et al.* 1988]:

1. Calculate the belief table for $|C| \cap |V|$ by marginalizing in the belief table of C .
2. Calculate the belief table for the same intersection by marginalizing in V .
3. Multiply the belief table for V by the ratio of the tables achieved by (2) and (1) (i.e., the table achieved by (2) is divided by the table achieved by (1)).

In HUGIN, an object oriented style is used: Each belief universe is an object holding its own data and its own methods. They communicate only with neighbors in the junction tree.

Calibration is the local propagation method. For a global propagation, the evidence is propagated to all belief universes through the junction tree.

There are two ways of propagating evidence. A forward propagation performed by the operation *Distribute Evidence*, and a backward propagation performed by the operation *Collect Evidence*.

Distribute Evidence is used when evidence from a single belief universe must propagate to the entire system: Evidence is transmitted to neighbors, and their *Distribute Evidence* method is called (see Figure 5). *Collect Evidence* is used when evidence from the entire system must be propagated to a single belief universe: U asks neighbors to *Collect Evidence*, and then U makes a multiple calibration to them [Jensen *et al.*, 1988] (see Figure 6).

When findings have been entered into more than one belief universe, they can be propagated to the entire system by a call to *Collect Evidence*, followed by a call to *Distribute Evidence* from an arbitrary belief universe.

There are other methods for calculation of probabilities in networks with loops [Pearl, 1988, Shachter, 1988]. These methods also contain graph manipulations with a potential for a combinatorial explosion. These manipulations are performed every time evidence is propagated in the causal network. The methods in HUGIN have the advantage that the graph manipulations are performed once and for all when the graph is triangulated. When the junction tree has been constructed, an upper limit on time and space requirements for evidence propagation can easily be estimated.

In general, the task of probabilistic inference in causal networks is *NP-hard* [Cooper, 1987]. The methods in HUGIN makes it unnecessary to solve an NP-hard problem every time evidence is propagated. Instead the NP-hard problem is tackled in the construction phase, where much effort can be devoted to finding optimal solutions.

4 Schematic overview of HUGIN

A typical HUGIN session is divided into three subtasks: (1) Creation of a domain model, (2) generation of a run-time system, and (3) the actual problem solving. These tasks must be performed sequentially, but the session can be suspended at any point, and resumed later. The MUCIN shell is organized in a modular fashion as unhealed in Figure 7.

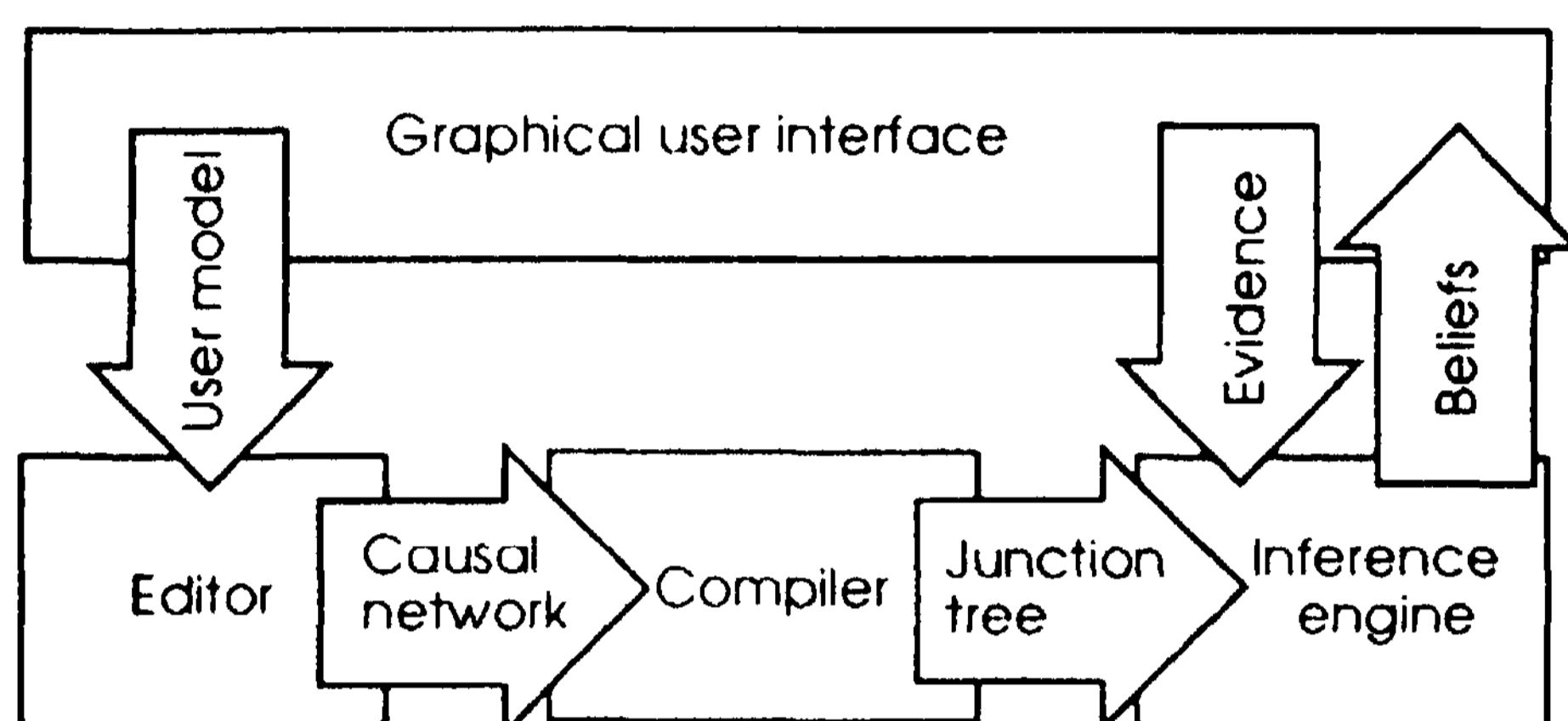


Figure 7: The structure of HUGIN, and the flow through the system from a domain model to a run-time system.

The graphical interface module does all the necessary book-keeping and provides access to facilities for creation, saving, loading, and deletion of domains. The editor module offers facilities for creating and editing a causal probabilistic network, i.e., creation and deletion of nodes and causal links, creation and editing of conditional probability tables etc. The compiler module creates a junction tree in which all calculations are done (cf. Section 3.1). The inference module maintains a set of current beliefs in all nodes in accordance with entered findings.

The HUGIN shell is implemented in C. and the graphical interface is clone in NeWS.²

5 The MUNIN networks

So far we have illustrated the use of HUGIN only by means of a simple example. In the ESPRIT project P599: "A Knowledge Based Assistant for Electromyography" [Andreassen *et al.*, 1987, Andreassen *et al.*, 1989, Olesen *et al.*, 1989], large networks are being constructed in order to diagnose muscle and nerve diseases. These nets constitute the MUNIN³ expert system.

In MUNIN, we are dealing with networks consisting of several hundred nodes. Handling networks on this scale of size raises new problems. The main problem concerns space and—as an implication hereof—time. The automatic triangulation in HUGIN is achieved by the maximum cardinality

²NeWS™ is a trademark of Sun Microsystems, Inc..

³MUScle and Nerve Inference Network.

search algorithm. This method is sufficient for small problems, but it is inadequate for large networks as it produces unnecessarily large belief universes.

As an example, one of the nets in MUNIN consists of 57 nodes, having from 2 to 21 states. In this network, at least one belief universe consisting of 14 nodes is created. This results in a huge belief table, and hence implies slow and space consuming propagation. The best known covering of this net consists of two 8-node belief universes and 13 other belief universes, giving a total size of about 2.2 million numbers in the belief tables.

6 Conclusion

We have presented a new expert system shell for modelling domains by causal probabilistic networks. The main achievement is the handling of multiply connected networks by transformation to a tree structure where the inference can be done efficiently. The shell, which employs the newest research results, has proved its capability to handle domain models of non-trivial size as exemplified by the MUNIN application.

HUGIN is now a well-suited tool for construction of research-like expert systems. However, much work still needs to be done before a full-size expert system shell is constructed. In particular, the following ideas and functionalities are under investigation and development:

- detection of conflicting evidence
- explanation facilities
- search for crucial findings
- planning
- learning
- extension to continuous variables

Acknowledgments

We thank David J. Spiegelhalter and in particular Steffen L. Lauritzen for the great inspiration they have given to this work.

We are indebted to our colleagues in the MUNIN project, Steen Andreassen, Uffe Kjaerulf, Annelise Rosenfalck, Anders R. Sprensen, and Marianne Woldbye, for many valuable discussions.

References

- [Andersen *et al.*, 1987] S. K. Andersen, F. V. Jensen, and K. C. Olesen. The HUGIN core preliminary considerations on systems for fast manipulations of probabilities. In *Proceedings of Workshop on Inductive Reasoning: Managing Empirical Information in AI-Systems*, Risø National Laboratory, Roskilde, Denmark, April 1987.
- [Andreassen *et al.*, 1987] S. Andreassen, M. Woldbye, B. Falck, and S. K. Andersen. MUNIN—A Causal Probabilistic Network for Interpretation of Electromyographic Findings. In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, pages 366-372, Milan, Italy, August 1987.

- [Andreassen *et al*, 1989] S. Andreassen, F. V. Jensen, S. K. Andersen, B. Falck, U. Kjaerulff, M. Woldbye, A. R. Sørensen, A. Rosenfalck, and F. Jensen. MUNIN -An Expert EMG Assistant. In *Computer-Aided Surface or Needle Electromyography and Expert Systems in Diagnosis*, J. E. Desmedt (editor), Elsevier Science Publishers, Amsterdam, The Netherlands, 1989.
- [Cooper, 1984] G. F. Cooper. NESTOR: A Computer Based Medical Diagnostic Aid that integrates Causal and Probabilistic Knowledge. Technical Report HHP-84-48, Medical Computer Science Group, Stanford University, Stanford, California, 1984.
- [Cooper, 1987] G. F. Cooper. Probabilistic Inference Using Belief Networks is NP-Hard. Research Report KLS-87-27, Medical Computer Science Group, Stanford University, Stanford, California, 1987.
- [Dechter *et al*, 1988] R. Dechter, A. Dechter, and J. Pearl. Optimization in Constraint Networks. In *Proceedings of Conference on Influence Diagrams for Decision Analysis, Inference, and Prediction*, UCB, Berkeley, California, 1988.
- [Gavril, 1977] F. Gavril. Some AfV -complete problems on graphs. In *Proceedings of the Eleventh Conference on Information Sciences and Systems*, Johns Hopkins University, Baltimore, Maryland, 1977.
- [Horvitz *et al*, 1988] E. J. Horvitz, J. S. Breese, and M. Henrion. Decision Theory in Expert Systems and Artificial Intelligence. To appear in *Journal of Approximate Reasoning, Special Issue on Uncertain Reasoning*, 1988.
- [Howard and Matheson, 1981] R. A. Howard and J. E. Matheson. Influence Diagrams. In *Readings in Decision Analysis*, R. A. Howard and J. E. Matheson (editors), Strategic Decisions Group, Menlo Park, California, chapter 38, 763-771, 1981.
- [Jensen, 1988a] F. V. Jensen. Junction Trees and Decomposable Hypergraphs. Judex Research Report, Aalborg, Denmark, 1988.
- [Jensen, 1988b] F. V. Jensen. Discussion published in [Lauritzen and Spiegelhalter, 1988], 1988.
- [Jensen *et al* 1987] F. V. Jensen, S. K. Andersen, U. Kjaerulff, and S. Andreassen. MUNIN: on the case for probabilities in medical expert systems—a practical exercise. In *Proceedings of the First Conference of European Society for Artificial Intelligence in Medicine*, J. Fox, M. Fieschi, and R. Engelbrecht (editors), pages 149-160, Springer-Verlag, Heidelberg, 1987.
- [Jensen and Kjaerulff, 1989] F. V. Jensen and U. Kjaerulff. Triangulation of graphs—Algorithms giving small total clique size, 1989. In preparation.
- [Jensen *et al*, 1988] F. V. Jensen, K. G. Olesen, and S. K. Andersen. An Algebra of Bayesian Belief Universes for Knowledge Based Systems. Research Report R-88-25, Institute of Electronic Systems, Aalborg University, Denmark, 1988. To appear in *Networks*, 1989.
- [Kim and Pearl, 1983] J. H. Kim and J. Pearl. A computational model for causal and diagnostic reasoning in inference systems. In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, pages 190-193, Karlsruhe, West Germany, 1983.
- [Kong, 1986] C. T. A. Kong. Multivariate Belief Functions and Graphical Models. Ph.D. Thesis, Harvard University, Research Report S-107, 1986.
- [Lauritzen and Spiegelhalter, 1988] S. L. Lauritzen and D. J. Spiegelhalter. Local Computations with Probabilities on Graphical structures and Their Application to Expert Systems. *J. Royal Statistical Society B*, 50(2): 157-224, 1988.
- [Olesen and Andersen, 1988] K. G. Olesen and S. K. Andersen. Discussion published in [Lauritzen and Spiegelhalter, 1988], 1988.
- [Olesen *et al*, 1989] K. G. Olesen, U. Kjaerulff, F. Jensen, F. V. Jensen, B. Falck, S. Andreassen, and S. K. Andersen. A MUNIN Network for the Median Nerve—A Case Study on Loops. *Applied Artificial Intelligence, Special Issue: Towards Causal AI Models in Practice*, 1989.
- [Pearl, 1986a] J. Pearl. Fusion, Propagation, and Structuring in Belief Networks. *Artificial Intelligence*, 29:241-288, 1986.
- [Pearl, 1986b] J. Pearl. A Constraint-Propagation Approach to Probabilistic Reasoning. In *Uncertainty in Artificial Intelligence*, L. N. Kanal and J. F. Lemmers (editors), pages 357-369, Elsevier Science Publishers, Amsterdam, The Netherlands, 1986.
- [Pearl, 1988] J. Pearl. Embracing Causality in Default Reasoning. *Artificial Intelligence*, 35:259-271, 1988.
- [Pearl and Paz, 1986] J. Pearl and A. Paz. Graphoids—A graph-based logic for reasoning about relevance relations. In *Proceedings of the Seventh European Conference on Artificial Intelligence*, pages 55-61, Brighton, UK, 1986.
- [Shachter, 1986] R. I. Shachter. Evaluating Influence Diagrams. *Operational Research*, 34:871-882, 1986.
- [Shachter and Heckerman, 1987] R. D. Shachter and D. Heckerman. A Backwards View for Assessment. In *AI magazine*, 8(3):55-62, 1987.
- [Shafer and Shenoy, 1988] G. Shafer and P. Shenoy. Bayesian and Belief-Function Propagation. School of Business, Working Paper 121, University of Kansas, Lawrence, 1988.
- [Spiegelhalter, 1989] D. J. Spiegelhalter. Fast algorithms for probabilistic reasoning in influence diagrams with application in genetics and expert systems. In *Influence Diagrams*, R. Barlow *et al* (editors), Wiley, Chichester, 1989.
- [Tarjan and Yannakakis, 1984] R. E. Tarjan and M. Yannakakis. Simple Linear-time Algorithms to Test Chordality of Graphs, Test Acyclicity of Hypergraphs, and Selectively Reduce Acyclic Hypergraphs. *SIAM J. Computing*, 13:566-579, 1984.
- [Yannakakis, 1981] M. Yannakakis. Computing the minimal fill-in is NP-complete. *SIAM J. Algebraic Discrete Methods*, 2:77-79, 1981.